Monte Carlo Semantics

Robust Inference and Logical Pattern Processing with Natural Language Text

Richard Bergmair

University of Cambridge
Computer Laboratory
Churchill College

This dissertation is submitted for
the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

This dissertation does not exceed the extended word length of $68\,0000$, agreed with the Degree Committee (estimated word length $\approx 63,800$), including tables and footnotes, but excluding appendices and bibliography.

Monte Carlo Semantics
Robust Inference and Logical Pattern Processing with Natural Language Text

Richard Bergmair

Abstract

This thesis develops several pieces of theory and computational techniques which can be deployed for the purpose of allowing a computer to analyze short pieces of text (e.g. 'Socrates is a man and every man is mortal.') and, on the basis of such an analysis, to decide yes/no questions about the text ('Is Socrates mortal?'). More particularly, the problem is seen as a logical inferencing task. The computer must decide whether or not a logical consequence relation 'therefore' holds between the two pieces of text. ('Socrates is a man and every man is mortal, therefore Socrates is mortal.')

This problem is a pervasive theme in logic and semantics but has also been subject over the last five years to a wave of renewed attention in computational linguistics sparked by the Recognizing Textual Entailment (RTE) challenge. A critical reevaluation of this line of work is presented here which demonstrate several problems concerning the empirical methodology used at RTE and the results derived from it. This thesis is thus more theory-driven, but nevertheless inspired by RTE in that it addresses problems raised by RTE which have not previously received sufficient attention from a theoretical viewpoint, such as the problem of robustness.

With this goal in mind, two of the results on Natural Language Reasoning (NLR) established here become particularly important: (1) Assuming the syllogism as a benchmark fragment of NLR, the model theory which underlies NLR is not necessarily a two-valued logic, but it can be the many-valued Łukasiewicz logic. (2) Despite the fact that the syllogism is a logical language of less expressive power than natural language as a whole, a good approximation to NLR can still be obtained by using the method outlined here for rewriting natural language text into syllogistic premises.

These two properties of NLR enable the approach to robust inference and logical pattern processing called Monte Carlo semantics, which, in turn, demonstrates that a single logically based theory can account for the semantic informativity of deep techniques using theorem proving and for the robustness of bag-of-words shallow inference.

Acknowledgments

When Ann Copestake takes on a PhD project, strange things happen. To mention just two of them: An Irish Sanskritist discovers for himself the power of mathematics, and an Indian mathematician the semantics of discourse. And now this. – This project required of a supervisor an exceptionally open mind and a great deal of patience. It took someone to listen to unorthodox ideas and to see the merit in them at a stage where they could be described as half-baked at best. Ann has acquired among her students a reputation for bringing these rare qualities to the table, and deservedly so. It is therefore no exaggeration to say that this project would not have been possible without her.

I would also like to thank Ulrich Bodenhofer for opening up the world of multi-valued logic for someone like myself to explore. His support of this project has gone far beyond what one could expect outside the scope of institutional ties.

Just how I ended up in Cambridge, finding myself engaged in this piece of research, is still somewhat of a mystery to me, but a large part was certainly played by a line of teachers who appeared at different points in my student life to provide just that little bit of extra support that the classroom couldn't provide. In reverse-chronological order, these are: Stefan Katzenbeisser, Gerhard Höfer, and Peter Huemer. Ulrich Bodenhofer deserves to be thanked a second time here.

I'd like to thank Ted Briscoe and Simone Teufel for the influence they've had on me, primarily through their teaching of the 2005/06 M.Phil. course in Computer Speech, Text, and Internet Technology, and Ted, once more, for his support of my work in his role as second supervisor.

I have been financially supported during my time as an M.Phil. student and my first year as a Ph.D. student by my parents and an EPSRC studentship awarded by the Computer Lab. I have also received numerous travel grants from the Computer Lab and Churchill College. The bulk of the funding for this Ph.D. project has been provided by the Austrian Academy of Sciences through a DOC fellowship.

Finally, I would like to thank my parents and friends for their support on so many different levels I cannot even begin to recount here.

Contents

1. Introduction & Motivation

In this thesis, we will develop several pieces of theory and computational techniques which can be deployed for the purpose of allowing a computer to analyze short pieces of text (e.g. 'Socrates is a man and every man is mortal.') and, on the basis of such an analysis, to decide yes/no questions about the text ('Is Socrates mortal?'). More particularly, we approach the problem as a logical inferencing task. The computer must decide whether or not a logical consequence relation 'therefore' holds between the two pieces of text. ('Socrates is a man and every man is mortal, therefore Socrates is mortal.')

There are currently two prominent approaches to this problem: (1) In the shallow bag-of-words approach, a piece of text is represented as a vector, each dimension representing a word count. This translates into an inference technique which scores candidate inferences, for example based on the size of the bag-of-words overlap as a proportion of the length of the consequent. (2) In the traditional deep approach, text is represented as a formula, and logical theorem proving techniques are then applied to draw inferences.

The problem with these is that the shallow approach is robust but not semantically informed, while the deep approach is semantically informed but not robust. Our goal herein will be to contribute to the development of reasoning techniques which improve over the shallow approach in terms of semantic informativity and over the deep approach in terms of robustness. We will define what exactly we mean by semantic informativity and robustness in chapter 5. Although deep/shallow integration has come a long way in the domain of semantic representation (see e.g. Copestake 2007), the problem is relatively new on the agenda as far as logical inference is concerned. One other approach which attempts this, however, is the one by Bos & Markert (2005*a*,*b*, 2006*a*,*b*), which we will have more to say about in chapter 5.

1.1. Aims & Methodology

A New Paradigm: Relationalism

The notion which will remain centre-stage herein is that of logical consequence as a relation between pieces of natural language text. In adopting this view, we align ourselves with a recent paradigm shift in computational semantics which parallels an earlier development in the field of formal logic, when consequence relations (Tarski 1930) were recognized as a basic abstraction over logic alongside denotations (Frege 1879, 1892).

For computational semantics, the denotationalist paradigm has in practice meant that pieces of text are translated to formulae of FOPC, the first-order predicate calculus. The problem of drawing inferences on the basis of text can thereby be reduced to the more well-understood problem of drawing inferences on the basis of FOPC formulae. This strategy was adopted by Montague (1970*a*,*b*, 1973) and has also been at the heart of other influential treatments on semantics, such as the monograph by Kamp & Reyle (1993) and that of Blackburn & Bos (2005), nowadays often cited as a standard textbook on computational semantics. But this raises the question of just what representation language and what logic it is that is most appropriate for drawing inferences with natural language text, and whether FOPC is anything more than an arbitrary choice resulting from the lack of a practicable alternative. When considered in isolation, this problem seems intractable.

The paradigm shift towards relationalism, however, has meant that the emphasis is now on trying to infer the logic which is used in connection with a natural language from a set of acceptable candidate inferences in much the same way as a linguist would infer its grammar from a set of acceptable expressions. This parallels the approach of Tarski (1935), to whom a logic was simply whatever it needed to be in order to model the inferences properly, rather than starting with the rather abstract question "What is truth?" and then assigning truth-functional denotations to expressions.

A Synthetic Approach: Theory-Driven but Empirically Grounded

Within the relationalist paradigm, one can distinguish two kinds of methodology: (1) One can adopt the same kind of empiricism which underlies corpus linguistics. This idea motivated the methodology of the RTE recognizing textual entailment challenge (Dagan et al. 2005) and of related evaluation schemes like the AVE answer validation task at QA@CLEF (Peñas et al. 2007). (2) One can rely on introspection in much the same way as a linguist would, when using carefully chosen examples and counterexamples to substantiate a given set of working hypotheses. This kind of methodology was adopted for the FraCaS project (Cooper et al. 1996) in its testset-based evaluation methodology. – It is this latter methodology which we adopt for this thesis, and, given certain intuitions derived from such a method of introspection, we connect the dots by formal methods.

In addition to being theoretically-driven, our approach will also be synthetic in nature. Thus, our theory will inherit properties from the component theories from which it has been synthesized. This, in particular, includes certain properties concerning its power to describe empirically observable phenomena. For example: We will establish traditional deep inference as a special case of our theory which represents one limit case, and we also establish bag-of-words shallow inference as another special case which represents another limit case on the opposite end of the deep/shallow spectrum. To the extent that both of these special cases have previously been studied empirically, we have nothing to add concerning the validation of these results, but neither do we do anything to the theory which would invalidate the results. – Our own work will be concerned primarily with the generalization itself.

Finally, we arrive at certain claims. Here one must distinguish two different types: (a) A priori claims relate to definitions, conventions, and theory ('Water is wet'). (b) A posteriori claims relate to descriptions of empirical facts ('Sea water contains salt'). – In this thesis, we will arrive at claims of the former type. So our work is not a description of language by means of logic, but rather an extension of logic inspired by language.

The distinction is not of merely philosophical interest. Someone sufficiently indoctrinated into the ways of mainstream computational linguistics might feel compelled to ask: "Where is the empirical validation for all this?" But asking such a question of logic, as Łukasiewicz pointed out, would be like asking for empirical validation of the fact that $2+2=4$, making a survey, and concluding that, on average, $2+2$ equals 4.12. Conversely, if one has written a computer program to implement arithmetic, one might fallaciously conclude its correctness from the testcase $2+2=4$. Thus, it is important to note from the outset one thing about the claims which we will ultimately arrive at in this thesis: They all state the that we have successfully constructed a mathematical structure of some form or another which reflects certain intuitions which one may hold about language. They do not say anything about the empirical validity of these mathematical structures as models of observable phenomena.

The fact that we do not engage in an empirical study, however, is not to say that such empirical study is impossible. Quite to the contrary: we will have much to say about how one would go about it in practice, particularly in chapter 2. And it is the very fact that our theory is amenable to empirical study which makes it a scientific theory.

1.2. Related Work

Due to its abstract goals, this thesis draws on a large body of related work, with each particular piece of previous work contributing only a small piece to a large puzzle. For example, now that the main task of the RTE recognizing textual entailment challenge was run for the sixth and last time, the number of publications which have made it into that forum alone is over a hundred, not counting related evaluations like the RTE-3 pilot or AVE. In an attempt to account for this work, we will, in chapter 2, look at

RTE-4 systems in particular and reduce each system to a point in a geometrical space of results. This will make it possible to draw several interesting conclusions from patterns observable in that space, thus dealing with a large body of literature quantitatively rather than discussing each approach qualitatively.

As far as logic, natural logic in particular, is concerned, the relevant pieces of theory are so fundamental that we will frequently have to refer to them and that they are better dealt with in the particular contexts in which they become relevant in chapters 3, 4, and 5. The approach of syntactic pattern rewriting relates largely to the problem of representation, and will therefore be dealt with in chapter 4.

Then, of course, each chapter will also bring with it a particular body of relevant literature which we build on: Chapter 3 will draw heavily on the work of Jan Łukasiewicz on many-valued logic (Łukasiewicz & Tarski 1930) and on the syllogism (Łukasiewicz 1951). Chapter 4 on decomposition will build on the approach to semantic composition put forward by Copestake et al. (2001, 2005), Copestake (2007, 2009), as well as the grammar by Flickinger (2000) and related grammars. And, finally, in chapter 5, we will pay particular attention to the textual inference engine by Bos & Markert (2005*a*,*b*, 2006*a*,*b*), which represents a previous approach at logically-based textual inference with robustness properties.

Recognizing Textual Entailment

The Recognizing Textual Entailment Challenge (RTE)[1] has gained some prominence as an evaluation framework on textual inference, so it is worth devoting some attention to the methodology implied before delving into a new treatment of the subject. Another noteworthy piece of related work is the FraCaS project, which as previously introduced a testsuite (Cooper et al. 1996) similar in its intent to the RTE dataset. Figure 1.1 shows some candidate inferences which might be included in these. We will always write candidate inferences in a form where we first state an antecedent ('Socrates is a man and every man is mortal'), and then a consequent ('Socrates is mortal'), the candidate inference itself being the proposition which states that the latter follows from the former.

RTE datasets are constructed by sampling candidate inferences from large corpora using a model-free methodology which is meant to reflect the needs of several different applications such as information extraction (IE), information retrieval (IR), question answering (QA), and summarization (SUM). The candidate inferences would then be judged by naïve annotators, who attach their inference decisions on purely intuitive grounds (section 2.1.2). The problem of constructing an inference engine is then a model-fitting

[1] RTE-1: Dagan et al. (2005), RTE-2: Bar-Haim et al. (2006), RTE-3: Giampiccolo et al. (2007), RTE-3 pilot: Voorhees (2008), RTE-4: Giampiccolo et al. (2008), RTE-5: Bentivogli et al. (2009)

An Italian became the world's greatest tenor.
∴ Was there an Italian who became the world's greatest tenor? (F.1)

Every European has the right to live in Europe.
Every European is a person.
Every person who has the right to live in Europe can travel freely within Europe.
∴ Can every European travel freely within Europe? (F.18)

Some delegates finished the survey on time.
∴̸ Did any Irish delegates finish the survey on time? (F.70)

Just one accountant attended the meeting.
∴ ¬Did no accountants attend the meeting? (F.105)

(a) FraCaS examples (Cooper et al. 1996)

By the end of 2002, approximately 300,200 persons were reported as HIV-positive in the 15 countries of the former USSR, with the rate of HIV infection increasing rapidly. Throughout Eastern Europe, the period 2000-01 saw a sharp increase in infections, especially among intravenous drug users.
∴ AIDS victims increase in Europe. (R4.5)

AIDS could cut population numbers in some of the worst-hit African countries - the first falls attributable to disease since bubonic plague ravaged Europe.
∴̸ AIDS victims increase in Europe. (R4.7)

Reports from other developed nations were corroborating these findings. Europe, New Zealand and Australia were also beginning to report decreases in new HIV cases.
∴ ¬AIDS victims increase in Europe. (R4.8)

(b) RTE examples (RTE-4 data, task: IE)

Figure 1.1.: some example inferences from standard datasets

exercise with the implicit aim of maximizing a statistic which measures the agreement of system decisions with the gold standard reference decisions.

The FraCaS testsuite, on the other hand, can be understood as a relationalist rendering of logical-semantic phenomena in the form of manually constructed textbook example inferences. The creators of the FraCaS testsuite leave open the question of how to evaluate a given set of decisions as assigned by a model to the gold standard decisions. Yet, the testsuite clearly lends itself to the kind of methodology which would also be employed by a linguist or grammar engineer when constructing a syntactic grammar, while observing and describing the overgeneration and undergeneration behaviour of a specific grammar w.r.t. specific phenomena by means of well-chosen examples. – This implies a more theory-driven approach.

The ideological divide between the two approaches has previously given rise to some controversy: For example Zaenen et al. (2005) have taken a critical view on RTE methodology in its most empirical interpretation, eliciting a defense from Manning (2006), which, in turn, received a rebuttal (Crouch et al. 2006). In chapter 2 we will, among other things, undertake a theoretical investigation to shed some more light on this topic, while trying, as far as possible, to steer clear of dogma.

Recall, in this context, that, concerning the deep/shallow debate, we align ourselves with neither school of thought but are, rather, trying to hybridize both approaches and achieve a kind of deep/shallow integration which draws on the strengths of both. Our stance on evaluation is a similar one: The practice of RTE-style evaluation on one hand and FraCaS-style testing on the other, in terms of their methodological underpinnings, are taken herein as complementing, rather than contradicting, each other.

Especially concerning the RTE evaluations, we will highlight many fundamental problems which exist with the current implementation of the methodology underlying it. But such shortfalls in implementation are only to be expected of a young and evolving field and should not distract from the fact that the underlying idea is sound: In particular: RTE evaluations have brought relationalism into the spotlight within computational linguistics and natural language processing, and, for the first time, they have applied an empirical methodology to a problem that has been previously thought of only in theoretical terms. Furthermore, the fact that Harabagiu & Hickl (2006) have been able to successfully deploy an RTE engine to improve the performance of a question answering system seems promising.

Fuzzy Logic

Why is it necessary for us to devote an entire chapter to the rather foundational topic of logic in a thesis such as this, where we have quite specific aims that do not immediately refer to this area? After everything that has been said elsewhere about fuzzy logic, the so-called probability logic, Bayesianism, etc. is there anything substantially new which we need to say about this topic in order to justify our approach to textual inference?

The answer is yes, and it is precisely the fact that so much has been said about this topic which has made the body of related work so difficult to navigate. Especially the AI-motivated treatment of fuzzy logic which was popular in the 1970s has done a great disservice to the cause of fostering a more widespread understanding of many-valued logic on the interdisciplinary stage. – Refer to Elkan (1994), and the debate which ensued thereafter in *IEEE Expert* for a case in point. – For example, Hájek (1998), in one of the most groundbreaking contributions to the field, feels it necessary to refer to that legacy and make explicit the claim that "fuzzy logic is neither a poor man's logic, nor a poor man's probability".

I have found myself confronted with similar misconceptions when I first started presenting my ideas on a many-valued model theory for natural language to computational linguists. Negative opinions were widespread, citations to back up those opinions were generally rare, and even in the literature, confusion has often prevailed. For example, the misconception "I have heard that fuzzy logic is a bad idea, as it is provably incomplete" could be a reference to Morgan & Pelletier (2004), who construct a particular class of fuzzy logics, claiming the impossibility of a proof-theoretic account of it, while failing to discuss Hájek (1998) and other pertinent earlier work establishing classes of fuzzy logics which do come with proof theories and completeness results.

The only defence against prejudice of this sort is formal rigour, which is why, in chapter 3, we will establish from first principles a model theory, a proof theory, and an algebra, together with the relevant completeness proofs, for the particular logics which we use.

Aside from the question whether fuzzy logic has something to contribute to computational semantics to improve over classical logic, fuzzy logic has, in the past, also been contrasted with other more wildly non-classical logics which have been proposed for the purpose of modelling natural language semantics, one case in point being Pinkal (1985). None of these alternative proposals, however, have been nearly as well developed as fuzzy logic, and there now seems to be a more widespread recognition of the fact that fuzzy logic in computational semantics deserves another chance. For example, where van Deemter (1995, p. 82) previously stated that "systems resulting from" fuzzy logic "seem as badly disposed to solve the Sorites paradox as classical logic", he has now adopted an approach similar to my earlier proposal (Bergmair 2006*a*), and is actively advocating the use of fuzzy logic in computational semantics (van Deemter 2010*a*,*b*), particularly as a model for vagueness.

Here, we will not be interested in fuzzy logic as a model for any natural language phenomena at all, although such investigations might well benefit from the results we are about to establish. Rather, what motivates our use of a many-valued logic is the fact that this will turn out to be useful in chapter 5 computationally.

Finally, another possible pitfall is to misunderstand probability as logic: The formula $\mathbb{P}(\mathbf{ab}) = \mathbb{P}(\mathbf{a})\,\mathbb{P}(\mathbf{b})$ expresses the probability of the cooccurrence of two stochastically independent events $\mathbf{a}$ and $\mathbf{b}$. This is not the same as a logical conjunction, as it refers to the notion of stochastic independence, and is therefore not truth-functional. Conversely, the notion of stochastic independence would have no interpretation in a logic. Despite the fact that there are many deep relationships between probability and fuzzy logic, it is important not to confuse them from the outset.

In this thesis, we will take the viewpoint that logic and probability are models for different kinds of phenomena, and so they complement, rather than contradict, each other. Logic is primarily about language. It establishes the language of proof and it assigns, in a compositional manner, truth values to expressions in a formal language. Probability is primarily about knowledge and the lack thereof, i.e. uncertainty, and so is not a theory of formal language but a theory of formal epistemology. This is how we will put these theories to use in this thesis, despite the fact that advocates of fuzzy logic have not always drawn this distinction. Chapter 3 will be concerned with logic and its formal language, while chapter 5 will be concerned, among other things, with probability as a model of formal epistemology to approach the problem of uncertainty. So, in response to the question "Fuzzy logic or probability?", our answer is "both".

Natural Logic

Another body of literature which is important to us is what, for lack of a better term, often comes under the heading of "natural logic". This term, however, means many different things to many different people. Of particular relevance to us is that work within natural logic which studies fragments of the predicate calculus corresponding to fragments of natural language, given the usual style of predicate calculus representation for natural language. In particular this includes work by van Benthem (1986, 1991), McAllester & Givan (1992), Zamansky et al. (2006), van Eijk (2007), Pratt-Hartmann (2003, 2004), Pratt-Hartmann & Third (2006), Pratt-Hartmann & Moss (2009).

These fragments of the predicate calculus are usually studied in terms of their metatheoretic properties, i.e. proof-theoretic, model-theoretic and algebraic formalizations thereof and corresponding completeness theorems, computational properties etc.

Of these fragments of logic, we consider only two very basic fragments which are of special interest to us: (1) The syllogism, as we will outline shortly, plays a central role in providing the interface between our model theory (chapter 3) and our theory of semantic decomposition (chapter 4). (2) A fragment of natural logic which we will call substitution logic has an important role to play in competing approaches to textual inference.

Minimal Recursion Semantics and the English Resource Grammar

Some substitution-based approaches to textual inference attach substitution rules directly to surface form patterns, rather than patterns of syntactic or semantic deep structure. This means that they would have to extract, for example, active/passive alternations as logical rewrite patterns, rather than treating them as an element of grammatical analysis. – The main thrust of our argument here is as follows: When it comes to textual inference, one either deals explicitly with problems of grammar, i.e. syntax and compositional semantics, or there is a theory of grammar which is implied by one's approach to textual inference with surface patterns.

In this thesis, we will deploy the former approach and frequently refer to prior work surrounding the English Resource Grammar (Flickinger 2000) with its implementation of compositional semantics based on Minimal Recursion Semantics (Copestake et al. 2001, 2005, Copestake 2007, 2009), a metalanguage and algebra for semantic composition based on a predicate calculus-style object language with abstract predicate and quantifier symbols.

The advantages of using MRS, as opposed to using the language of the predicate calculus directly, are its ability to facilitate inference and composition in the face of quantifier scope ambiguity, as well as a higher degree of canonicity with regard to certain linguistic tranformations which are logico-semantically invariant. The ProtoForm language which we will introducte in section 4.1 is heavily inspired by the MRS language. In section 4.2, we discuss MRS-style semantic composition which is the same composition process that could, in principle, be applied for semantic composition with ProtoForms. – As far as our prototype is concerned, it uses the English Resource Grammar and obtains ProtoForms by translation from MRS.

The design of MRS has also inspired our particular approach to textual inference more abstractly: Themes such as deep/shallow integration, underspecification of scope ambiguities, dependency-style interpretation of semantic forms etc. are cornerstones of MRS, and they appear in many places throughout this thesis, particularly in chapters 4 and 5.

Bos & Markert's Nutcracker system

Our approach has much in common with that of Bos & Markert (2005*a*,*b*, 2006*a*,*b*) in that it is a logically-motivated treatment of textual inference based on translation of text into predicate calculus, and that it attempts deep/shallow integration.

The main point of difference is that NUTCRACKER uses standard FOPC reasoning tools, such as a theorem prover and model builder, and that its semantic composition uses a CCG-based grammar and DRT as a semantic representation formalism. We will discuss this approach in greater depth in chapter 5.

1.3. Overview

We approach two important questions about natural language reasoning (NLR). (1) Assuming the syllogism as a benchmark fragment of NLR, is the model theory which underlies NLR necessarily a two-valued logic, or can it be a many-valued logic? In chapter 3, we will develop the many-valued Łukasiewicz logic into such a model theory for

NLR. (2) Given the syllogism as a logical language of far less expressive power than natural language itself, can we still obtain a good approximation to NLR using the syllogism? In chapter 4, we will develop such a method of rewriting natural language text into syllogistic premises. Our approach to these These two results on NLR then enable the particular approach to robust inference and logical pattern processing which we call Monte Carlo semantics (chapter 5) and which aims at combining the semantic informativity of deep, logically based approaches to language processing with the robustness of shallow approaches such as bag-of-words representations.

Empirical Review & Methodology

Chapter 2[2] proposes a new empirical methodology for studying textual entailment which is inspired by the RTE recognizing textual entailment challenge[3]. However, it deviates from RTE methodology in a number of ways, and should thus be seen as an alternative to the RTE scheme, rather than an extension thereof.

The problems with RTE methodology which our treatment addresses start with the fact that it is not sufficiently clear what logical distinction between candidate inferences is actually being drawn by RTE judges: The distinction ENTAILED vs. UNKNOWN vs. CONTRADICTION could either correspond to a question of logical validity or a question of relevance or logical determinacy. We will see why the distinction is a crucial one to make.

We will also discuss in greater detail the evaluation criteria implicit to the methodology as part of the statistical scores used at RTE, such as accuracy and average precision, questioning their suitability and proposing mutual information as an alternative.

Having established, in section 2.1, a set of formal definitions which improve in various ways over the theory of evaluation which is implicit to the RTE scheme, we then go on, in section 2.2, to build on those theoretical foundations and use statistics on RTE inference data and submissions to conduct an empirical metaevaluation of the RTE evaluation scheme and reevaluation of the submitted systems.

From this investigation, we derive another result contradicting an idea which is constitutive of the task itself: Statistically, the task does not appear as a coherent abstraction over the different applications (question answering, information extraction, etc.)

Finally, our reevaluation of RTE systems shows that only a very small number of systems at the top of the ranking show any evidence of having successfully addressed the task. We derive this conclusion from the observation that the bag-of-words baseline performs relatively well on the task, and that the vast majority of systems exhibit error character-

[2] Some of the material in this chapter was previously published (Bergmair 2009).

[3] see Bentivogli et al. (2009), Giampiccolo et al. (2008), Voorhees (2008), Giampiccolo et al. (2007), Bar-Haim et al. (2006), Dagan et al. (2005)

istics which suggest that their labellings differ from the baseline randomly rather than trending towards the gold standard.

Łukasiewicz Logic & Syllogistic Semantics

Traditionally, one would think of natural language reasoning as being based on a bivalent model theory in which propositions are always either false or true, always either 0 or 1. Chapter 3[4] takes a different viewpoint.

Here, we establish that we can instead use the many-valued logic proposed by Łukasiewicz (Łukasiewicz & Tarski 1930), in which truth values are drawn from the entire unit interval $[0, 1]$, so that a proposition could be true, for example, to a degree 0.7.

In section 3.1, we will summarize the relevant literature to establish the propositional logic of Łukasiewicz and its associated proof theory and completeness result. The noteworthy property of this formal system is that an M-valued Łukasiewicz logic can be derived for any M, with the limit case of $M = 2$ reducing to standard propositional logic. Of particular interest to us will be the other limit case of $\aleph_0$-valued logic, which has infinitely many truth values and thus gives us a generalization of standard logic. This is a generalization in that it never proves a theorem which 2-valued logic does not prove. But there are also theorems which 2-valued logic proves which $\aleph_0$-valued logic does not.

In section 3.2, we will then move on to define a reduction of the language of the predicate calculus to the language of propositional logic. For example, if we have

$$\forall x : \mathsf{man}(x) \to \mathsf{mortal}(x),$$

we would rewrite the universal quantifier, traditionally taken to range over an infinite domain, as a conjunction over a domain of three individuals

$$\big(\mathsf{man}(\mathrm{c}_1) \to \mathsf{mortal}(\mathrm{c}_1)\big) \wedge \big(\mathsf{man}(\mathrm{c}_2) \to \mathsf{mortal}(\mathrm{c}_2)\big) \wedge \big(\mathsf{man}(\mathrm{c}_3) \to \mathsf{mortal}(\mathrm{c}_3)\big),$$

where $\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3$ are constants referring to those three individuals. So this is now simply a conjunction over three implications, each of which contains only predications over constants. But such predications over constants are simply propositional atoms, so we can rewrite this in propositional logic as

$$\big(\mathsf{man}_1 \to \mathsf{mortal}_1\big) \wedge \big(\mathsf{man}_2 \to \mathsf{mortal}_2\big) \wedge \big(\mathsf{man}_3 \to \mathsf{mortal}_3\big).$$

We are now deviating from standard logic in two regards: We move from 2-valued to $\aleph_0$-valued logic, and we change the interpretation of quantification. The question then arises whether this new logic is now any less adequate as a model for natural language

[4] Some of the material in this chapter was previously published (Bergmair 2008).

reasoning than standard logic. We will address this in section 3.3, where we take the syllogism as a benchmark for natural language reasoning. This approach was motivated by the central role which the syllogism plays in natural logic, and is further justified by our own results (chapter 4). Again, it is Łukasiewicz (1951) who pioneered the relevant foundations by establishing the syllogism as a formal logic.

The problem with his formalization, however, is that it relies on bivalent logic. This is why, in section 3.3, we will present a completeness proof which establishes that the fragment of our logic corresponding to the syllogism proves all and only those syllogisms traditionally considered valid. This completeness proof makes it necessary to develop some basic algebraic identities in the algebra of Łukasiewicz logic (section 3.1).

With this completeness result in place, it appears that, with our particular logic, despite the fact that it is slightly non-standard, we do not lose the ability to support the same kind of reasoning with natural language which standard logic would impose on the language.

This result turns out to be highly relevant in chapter 5. In particular, the fact that this model theory is $\aleph_0$-valued will be useful for compuational purposes. Furthermore, one of the two conjunction operators in Łukasiewicz logic, the strong conjunction, is commutative and associative but not idempotent. This amounts to a bag aggregation operator, and we will see that it can be used to reproduce the robustness effects of bag-of-words inference within a proper logical framework (chapter 5). But even aside from its importance to this thesis, this result on the many-valued semantics of natural language is interesting in and of itself. For example, many-valued logic might be useful for search, natural language interfaces to databases, and as a model for vagueness in natural language (see Bergmair 2006*a*,*b*, Bergmair & Bodenhofer 2006, van Deemter 2010*a*,*b*).

Semantic Decomposition

Having established a logical interpretation for the syllogism, the question now arises how to translate natural language to the language of the syllogism. This is the problem we address in chapter 4. We will arrive at a solution to this problem by means of a new kind of semantic representation scheme which supports semantic composition and semantic decomposition. Semantic composition is the well-understood problem of arriving at a semantic representation structure given a piece of text, and semantic decomposition is the problem of arriving at a logical formula suitable for purposes of inference given a semantic representation structure.

So, we begin the chapter in section 4.1 by establishing the semantic representation language of ProtoForms. Figure 1.2 shows an example of such a ProtoForm structure. The ProtoForm language is closely related to Hole Semantics (Bos 1996) and MRS (Copestake et al. 2005), but differs from these schemes in that it is recursive: A ProtoForm can be the subform of another ProtoForm. In particular, the recursive structure which we sentence:

Every representative of a company saw a sample.

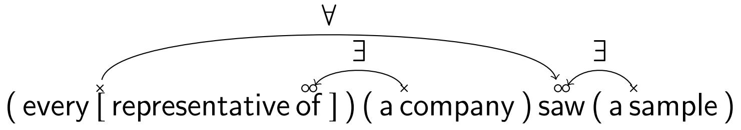
Underspecified ProtoForm:

$$\left[\begin{array}{l}
|\text{every}|_{(x_1)}\ \boxed{1}\ \text{--}, \\
\boxed{2}\left[\begin{array}{l} |\text{representative}|\,(\textsc{key}=x_1), \\ \text{-- \& --}, \\ |\text{of}|\,(\textsc{key}=/e_2/,\ \text{arg1}=x_1,\ \text{arg2}=x_2) \end{array}\right], \\
|\text{a}|_{(x_2)} \left[\,|\text{company}|\,(\textsc{key}=x_2)\right] \text{--}, \\
|\text{saw}|\,(\textsc{key}=e_1,\ \text{arg1}=x_1,\ \text{arg2}=x_3), \\
|\text{a}|_{(x_3)} \left[\,|\text{sample}|\,(\textsc{key}=x_3)\right] \text{--}, \\
\quad \boxed{1} \prec \boxed{2}
\end{array}\right].$$

SNF, logical form:

$$\left[\begin{array}{l}
\left[\begin{array}{l}
\left[\,|\text{every}|_{(x)} \left[\begin{array}{l} |\text{representative}|\,(\textsc{key}=x), \\ \text{-- \& --}, \\ |\text{of}|\,(\textsc{key}=/e_2/,\ \text{arg1}=x) \end{array}\right] \left[\,|\text{saw}|\,(\textsc{key}=/e_1/,\ \text{arg1}=x)\right]\right] \\
\quad \wedge \left[\,|\text{a}|_{(x)} \left[\,|\text{company}|\,(\textsc{key}=x)\right] \left[\begin{array}{l} |\text{representative}|\,(\ ), \\ \text{-- \& --}, \\ |\text{of}|\,(\textsc{key}=/e_2/,\ \text{arg2}=x) \end{array}\right]\right]
\end{array}\right] \\
\quad \wedge \left[\,|\text{a}|_{(x)} \left[\,|\text{sample}|\,(\textsc{key}=x)\right] \left[\,|\text{saw}|\,(\textsc{key}=/e_1/,\ \text{arg2}=x)\right]\right]
\end{array}\right]$$

SNF, dependency notation:


∀
∃
∃
(every [representative of]) (a company) saw (a sample)

SNF, McDonald's decomposition:

- F: Every representative saw \ something.
  Q: Who / saw? A: Every representative / saw.
- F: They were \ representatives of a company.
  Q: Who were they \ representatives of? A: Representatives of \ a company.
- F: Somebody / saw a sample.
  Q: What / was seen? A: A sample / was seen.

Figure 1.2.: example SNF structure

will impose on ProtoForms is the one used by Koller et al. (2009) for scoping purposes. As we will see, this is useful for semantic decomposition in a number of ways.

In section 4.2, we will summarize how ProtoForm composition is possible on the basis of a simplified MRS-based toy algebra. So, despite the fact that we will make some changes to the representation language, nothing fundamental changes about the process of composition. We will, however, have a number of things to say about decomposition.

In particular, section 4.4 discusses a decomposition process which produces from a ProtoForm what we call a syllogistic normal form (SNF). An SNF is a logical formula which is a conjunction of syllogistic premises (SPs). An example is shown in Figure 1.2.

We take the view that SNFs are not merely artefacts of our decomposition process, but that they have an interesting interpretation from a linguisic point of view. This can best be seen by considering the dependency notation used in Figure 1.2.

Section 4.3 considers substitution logic as an inference framework based on rewrite patterns. Here, we will find that, by using rewrite patterns over the bracketed structures of Figure 1.2, we can get a more accurate logic than by using rewrite patterns over syntax trees or syntactic dependency structures. However, this section will also show that rewrite patterns themselves run into limitations which can be overcome by using the full logic of the syllogism, rather than just substitution patterns.

Finally, in section 4.5, we will show that SNF dependency structures also adhere to the metatheoretical principles of grammar outlined by Harris (1982, 1991), which other kinds of dependency structures do not.

Monte Carlo Semantics

We thus have a way to translate natural language into the language of the syllogism and a reduction of the syllogism to propositional logic. So this immediately reduces the problem of natural language inference to that of inference with propositional logic. We could now use standard reasoning tools. But can we do better? This is the question we address in chapter 5[5].

Our requirements in an inference mechanism are as follows: (1) semantic informativity, the ability to take into account all available information; and (2) robustness, the ability to proceed on reasonable assumptions where such information is missing. – In section 5.1, we will substantiate these notions with some example inferences and delimit the scope of the inferences we can address.

As we will see, out-of-the-box reasoning tools and the traditional notions of satisfiability and validity are inadequate when it comes to the robustness properties we require. Note,

[5] Some of the material in this chapter was previously published (Bergmair 2008).

in this context, that a candidate inference $\varphi \to \psi$ is valid iff it is true for all valuations, i.e. iff the minimum truth value $\min_{w \in \mathcal{W}} \|\varphi \to \psi\|_w$ across all model-theoretic valuations w is ≥ 1. Similarly, it is considered classically satisfiable, iff the maximum truth value $\max_{w \in \mathcal{W}} \|\chi\|_w$ across all w is > 0.

But validity is too strong a criterion and satisfiability too weak for the purposes of open-domain NLP, where inferences will often hinge on common sense, real world or domain knowledge. In such a situation, it is to be expected that the vast majority of candidate inferences will be contingencies, i.e. formulae which are satisfiable but not valid. The classical validity and satisfiability notions do not allow one to draw distinctions between different grades of validity, so we cannot say which, of a given pair of candidate inferences, is closer to being valid, when none of them is strictly valid.

Our approach to the problem will be to use a statistic in between the minimum and the maximum: We use an arithmetic mean. In particular, we will denote by

$$\llbracket \varphi \to \psi \rrbracket_{\mathrm{W}} = \frac{1}{|\mathrm{W}|} \sum_{w \in \mathrm{W}} \|\varphi \to \psi\|_w$$

the degree of validity of the candidate inference $\varphi \to \psi$. In section 5.2.1, we discuss this definition and how, as a result of this definition, our degrees of validity relate to the validity and satisfiability notion of standard logic on one hand and to probability theory on the other.

In the case of deep linguistic analysis and a complete theory of background knowledge, our approach reduces to standard logic, which demonstrates semantic informativity (section 5.2.1). On the other hand, the definition also implies robustness properties, which we will show by establishing bag-of-words inference as another limit case (section 5.2.2).

In section 5.3 we will then show how one might approach the problem of estimating $\llbracket \varphi \to \psi \rrbracket_{\mathcal{W}}$ in the general case. Our algorithm is based on the naïve approach to inference in propositional logic where one generates model-theoretic valuations $w \in \mathcal{W}$ exhaustively and runs a model checker on each to check the truth value $\|\varphi \to \psi\|_w$ of the candidate inference. The problem with this approach in general is that $|\mathcal{W}| = 2^{\mathrm{N}}$, when N is the number of atomic propositions in the formula, and that, in order to estimate a minimum truth value, one must generate all of them in the worst case. An arithmetic mean, however, is better behaved when it comes to statistical estimation.

So, we will not attempt to determine its exact value and will instead take a random sample $\mathrm{W} \subseteq \mathcal{W}$ and use $\llbracket \varphi \to \psi \rrbracket_{\mathrm{W}}$ as an estimator for $\llbracket \varphi \to \psi \rrbracket_{\mathcal{W}}$. By statistical sampling theory, we know that the former will approach the latter as the sample size $|\mathrm{W}|$ approaches the population size $|\mathcal{W}|$. The sampling itself can be automated by means of a Monte Carlo method.

2. Empirical Review & Methodology

How well does a given theory of natural language inference describe empirically observable phenomena? Given two natural language inference engines which implement such theories, what are their characteristics when it comes to describing such phenomena, and how do they differ from each other?

We will formalize a theory of such evaluation and will discover several problems with the RTE evaluation scheme in the process of this formalization. We will then use the theory to conduct a new evaluation of systems previously submitted for the RTE challenge, as well as an empirical metaevaluation of previous RTE evaluations.

2.1. Theoretical Foundations & Review of Methodology

In this section, we will introduce some theoretical foundations which underly empirical work in textual inference. We will generalize over the definitions given in the past and go into greater analytic detail, so as to accommodate our criticism of the RTE evaluation scheme and our newly proposed methodological framework. The ideas also translate to a certain extent to the AVE answer validation exercise at QA@CLEF[1].

Here, we will only highlight as such those theoretical properties which arise from our definitions and which contradict incorrect intuitions and unjustified tacit assumptions underlying the RTE evaluation scheme. We will leave it for the next section (section 2.2) to discuss the limitations of the RTE evaluation scheme in more general terms and to show how our new methodology improves over it.

2.1.1. Fundamentals

In order to describe inference decisions, we need to impose some structure on the labels which can be assigned either by a model or by the gold standard to a candidate inference.

1. Inference decisions are drawn from the following set of *atomic decisions*:

$$\mathcal{D} = \{\boxplus, \boxminus, \diamondplus, \diamondminus\}.$$

[1] AVE-1: Peñas et al. (2007), AVE-2: Peñas et al. (2008), AVE-3: Rodrigo et al. (2009)

Note the two-dimensional structure of this notation: One fundamental dimension we will be interested in throughout the rest of this chapter will be the distinction + vs. −, the other dimension will be $\square$ vs. $\diamond$. The atomic decisions arise from writing the corresponding symbols on top of each other.

2. We will call the following subsets of $\mathcal{D}$ *structured labels*:

$$\begin{array}{lll}
\boxplus = \{\boxplus\}, & + = \{\boxplus, \diamondplus\}, & \triangle = \{\boxplus\}, \\
\boxminus = \{\boxminus\}, & - = \{\boxminus, \diamondminus\}, & \triangledown = \{\diamondplus, \diamondminus, \boxminus\}, \\
\diamondplus = \{\diamondplus\}, & \square = \{\boxplus, \boxminus\}, & \\
\diamondminus = \{\diamondminus\}, & \diamond = \{\diamondplus, \diamondminus\}. &
\end{array}$$

3. We define *labelsets*, which are sets of structured labels and partitions of $\mathcal{D}$:

$$\begin{array}{lll}
\mathcal{C}_4 = \{\boxplus, \boxminus, \diamondplus, \diamondminus\}, & \mathcal{C}_{+,-} = \{+, -\}, & \mathcal{C}_{\triangle,\triangledown} = \{\triangle, \triangledown\}, \\
\mathcal{C}_3 = \{\boxplus, \diamond, \boxminus\}, & \mathcal{C}_{\square,\diamond} = \{\square, \diamond\}. &
\end{array}$$

A few comments are in place about how these labelsets correspond to the labelsets used elsewhere in the literature. The labelset $\mathcal{C}_{\triangle,\triangledown}$ corresponds to the two-way distinction which has been used at the RTE since its inception and which was labelled ENTAILED vs. NOT ENTAILED at RTE-4 and RTE-5. The labelset $\mathcal{C}_3$ corresponds to the three-way distinction which was first introduced at the RTE-3 pilot, and subsequently used at RTE-4 and RTE-5 and which was labelled ENTAILED vs. UNKNOWN vs. CONTRADICTION. This is the same labelset which was also used for FraCaS, where the labels were YES vs. DON'T KNOW vs. NO.

Our own scheme is inspired by that of Wang & Zhang (2009), where the inference decision is reached by a two-stage process, the first stage deciding "relatedness" ($\square$ vs. $\diamond$), and the second stage deciding "entailment" (+ vs. −).[2]

4. For any labelset $\mathcal{C}$ and any atomic decision $d \in \mathcal{D}$, we define the equivalence class $[d]_{\mathcal{C}}$ as that $c \in \mathcal{C}$ for which $d \in c$. As an additional notational convenience, we may write $[d]_{\ldots}$ instead of $[d]_{\mathcal{C}_{\ldots}}$.

These equivalence classes express the idea that the NOT ENTAILED label at RTE is subdivided into UNKNOWN vs. CONTRADICTION. In our formalism, we can write out this relationship as follows: $[\boxminus]_{\triangle,\triangledown} = \triangledown$, and $[\diamond]_{\triangle,\triangledown} = \triangledown$, while $[\boxplus]_{\triangle,\triangledown} = \triangle$.

Now that we have defined the labels and the structure of labelsets, we need to attend to the candidate inferences themselves.

5. By $\mathcal{X}$, we denote the *inference language* of interest, the set of all candidate inferences that can be formed over some natural language, such as English.

[2] Note that the terminology we are about to adopt is different.

Note that each candidate inference $x \in \mathcal{X}$ is internally of the form $\varphi \to \psi$, a structure which we will make use of heavily throughout the rest of this thesis. For the purposes of this chapter, however, the internal structure of a candidate inference will not concern us.

Given this infinite language of candidate inferences, we can move towards a statistically more tangible notion:

6. An *inference sample* is a finite subset $\mathrm{X} \subseteq \mathcal{X}$ of $\mathcal{X}$, containing $|\mathrm{X}|$ candidate inferences

$$\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_{|\mathrm{X}|}\}.$$

So, we take a dataset, such as the RTE-4 dataset, to be a finite sample of candidate inferences drawn from an infinite language of possible candidate inferences. Here, our notion of sample includes not only statistically sampled datasets such as the ones used at the RTE challenge, but also analytical testsuites such as the FraCaS testsuite.

Having defined labelsets and inference samples, we can now move on to consider assignments of labels to candidate inferences.

7. The *model space* $\mathcal{H}$ is a set of mappings $\mathcal{X} \mapsto \mathcal{D}$ from the inference language $\mathcal{X}$ to the set of atomic decisions $\mathcal{D}$. Each such mapping $H \in \mathcal{H}$ is an *inference model.*

In the context of RTE-4, we can think of systems as inference models. In principle, the specifications of such systems as computer programs should allow them to assign an inference decision for any candidate inference in the language. This is why we have specified the inference model as applying to the infinite domain given by the entire inference language, despite the fact that we only observe inference decisions for finite samples.

For the sake of completeness, let us also mention the rankings which, besides labellings, are also used at the RTE evaluations as part of the assignment of inference decisions to inference samples:

8. Let X be an inference sample. We call a strict total order $\succ \subseteq \mathrm{X} \times \mathrm{X}$ a *ranking of* X.

We will have more to say about how these rankings enter into the evaluation when we define the evaluation measures which operate on them in section 2.1.4.

2.1.2. Decision Criteria & Decision Structure

Having defined a theoretical language for talking about samples of candidate inferences and their associated structured labels, we can now think about the meaning of such a relationship. We distinguish three criteria which may lead us to make a particular inference decision given a candidate inference: logical inference, intuitive inference, and application-oriented inference.

Logical Inference

We can obtain a criterion for making inference decisions on the basis of any logic which supports the inference language of interest and which is capable of assigning to any candidate inference $\varphi \to \psi$ a truth value $[\![\varphi \to \psi]\!]$ (see definition 90).

The following definitions give a criterion which is suitable to a large class of logics, including traditional bivalent FOPC and our particular logic which we define in the next chapter (chapter 3).

9. Let $x \in \mathcal{X}$ be a candidate inference of the form $\varphi \to \psi$. We say that x *is logically a □-instance* or that x *is logically determined* iff

$$[\![(\varphi \to \psi) \not\equiv (\varphi \to \neg\psi)]\!] > 0,$$

and that x *is logically a ◇-instance* or that x *is a logical contingency* otherwise.

10. Let $x \in \mathcal{X}$ be a candidate inference of the form $\varphi \to \psi$. We say that x *is logically a +-instance* or that x *is logically valid* iff

$$[\![\varphi \to \psi]\!] - [\![\varphi \to \neg\psi]\!] > 0,$$

and that x *is logically a −-instance* or that x *is logically unsatisfiable* otherwise.

Note that these definitions generalize over the traditional definitions of the notions of logical validity and satisfiability (definition 39). The generalization does not, however, have a role to play yet. We will estbalish it in due course.

Intuitive Inference

11. Let $x \in \mathcal{X}$ be a candidate inference. We say that x *is intuitively a □-instance* or that x *is intuitively relevant* iff, in response to question Q1 in the questionnaire in Figure 2.1, a naïve subject answers “YES”. Otherwise, we say that x *is intuitively a ◇-instance* or that x *is intuitively irrelevant*.

12. Let $x \in \mathcal{X}$ be a candidate inference. We say that x *is intuitively a +-instance* or that x *is intuitively valid* iff, in response to question Q2 in the questionnaire in Figure 2.1, a naïve subject answers “H is true”. Otherwise, we say that x *is intuitively a −-instance* or that x *is intuitively unsatisfiable*.

Although the RTE judges were never presented with a questionnaire like this, the methodology did depend on intuition. Just like any other decision criterion which relies on the subjective intuitions of naïve judges, it is, in this context, important to observe whether judges agree on their decisions, as this can be taken as evidence in support of the contention that they must have similar intuitions, and that therefore there is some universality to the intuitive criterion being captured.

T : AIDS could cut population numbers in some of the worst-hit African countries – the first falls attributable to disease since bubonic plague ravaged Europe.

H : AIDS victims increase in Europe.

(R4.7)

Q1. Given only common sense and the information provided in T, would you be willing to bet on whether or not the information provided in H is true? (YES / NO)

Q2. If you were forced to take such a bet, what would be your bet?
(H is true / H is false)

Figure 2.1.: intuitive inference questionnaire

For RTE-1, annotations were replicated independently of each other, and independently of the RTE organizers by Bos and Markert, Vanderwende et al., and Bayer et al. (Dagan et al. 2005). For RTE-2 and RTE-3, the organizers themselves carried out cross-annotation (Bar-Haim et al. 2006, Giampiccolo et al. 2007). For RTE-3, the annnotation was independently replicated by NIST for use in the RTE-3 3-way pilot evaluation (Voorhees 2008). In all cases, this led to agreement-levels of around 90% for the two-way distinction and 83% in the case of the 3-way distinction in the RTE-3 pilot.

As part of the sampling methodology, instances about which there was disagreement were discarded from the dataset, which is a controversial but not uncommon practice (Beigman-Klebanov & Beigman 2009). For RTE-4 and RTE-5, agreement levels were not officially reported.[3] Note, however, that the agreement statistics are subject to problems with bias and degradation in much the same way as accuracy scores for models, about which we have some reservations (section 2.1.6).

Application-Oriented Inference

The third and final criterion which we consider is the application-oriented criterion, motivated primarily from an engineering perspective. Here, we assume that the textual inference system is a component of a larger system, for example a question answering system which has an answer validation module based on textual inference. The inference

[3] Danilo Giampiccolo (in personal communication): "In RTE-4 and RTE-5, we decided not to report the data on the agreement, as it referred to the preliminary pair creation phase, and the actual agreement for all the pairs in the datasets was 100%. [...At RTE-5] out of 2538 pairs produced, 640 (25,22%) were discarded due to disagreement (all pairs being judged by three annotators)."

sample X of interest is then made up by the candidate inferences which are input to the inference subcomponent. We can then consider the inference model G assigned by the inference subcomponent as a variable, and try to optimize it in such a way as to maximize the measured end-to-end system performance.

13. Let $\mathrm{G_A} \in \mathcal{H}$ be an inference model. Let the function f map any given inference model $G \in \mathcal{H}$ to some ordered domain, so that $\mathrm{f}(G)$ is an application-oriented measure of end-to-end system performance. We call $\mathrm{G_A}$ an f*-induced application model*, iff

$$\mathrm{G_A} = \mathrm{argmax}_{G \in \mathcal{H}} \mathrm{f}(G).$$

Note that the function f represents the particular application and its evaluation criteria, and the particular system which is used as a model for how the performance of the inference subcomponent relates to the end-to-end system performance. So, while the criterion is attractive from an engineering perspective, its usefulness from a research perspective is limited by a lack of universality of any claim relating to an f-induced application model for a specific f. – From the induced application model, we can then derive the notions of validity and relevance.

14. Let $\mathrm{x} \in \mathcal{X}$ be a candidate inference and let $\mathrm{G_A}$ be the f-induced application model for some f. We say that x *is a □-instance in f* or that x *is relevant in f* iff, $[\mathrm{G_A}(\mathrm{x})]_{\square,\diamond} = \square$, and that x *is a ◇-instance in f* or that x *is irrelevant in f* otherwise.

15. Let $\mathrm{x} \in \mathcal{X}$ be a candidate inference and let $\mathrm{G_A}$ be the f-induced application model for some f. We say that x *is a +-instance in f* or that x *is valid in f* iff, $[\mathrm{G_A}(\mathrm{x})]_{+,-} = +$, and that x *is a −-instance in f* or that x *is unsatisfiable in f* otherwise.

Decision Criteria & Decision Structure at RTE

We have presented the inference decision as being composed of two dimensions representing independent aspects of the inference decision: relevance and validity.

At RTE-1 and RTE-2, some of the given example inferences hinge on negation to distinguish their labels, thus making it clear that their TRUE vs. FALSE distinction was not the pure relevance decision ($\mathcal{C}_{\square,\diamond}$). However, the definitions and overview papers remained ambiguous in that their entailment notion could have corresponded either to the validity decision ($\mathcal{C}_{+,-}$), or the logical conjunction of both validity and relevance ($\mathcal{C}_{\triangle,\triangledown}$). At the RTE-3 pilot, a new idea entered the scene: the UNKNOWN label as a subdivision of the negative class. Given that, we can assume that $\mathcal{C}_{\triangle,\triangledown}$ is the decision which is being used at RTE. What the annotators had in mind, of course, is hard to say in the absence of full annotation guidelines.

In section 2.2.1, we will show that, in the RTE-4 dataset, the relevance distinction ($\mathcal{C}_{\square,\diamond}$) receives much greater statistical weight than the validity distinction ($\mathcal{C}_{+,-}$). In section

2.1.3, we show that, on the other hand, some participants were explicitly addressing the problem of validity distinction ($\mathcal{C}_{+,-}$).

2.1.3. Negation Properties of Inference Decisions

The distinction between relevance and validity is nicely demonstrated by the impact of negation. We assume that any inference language $\mathcal{X}$ of interest will have a negation operator $\neg$, so that, whenever '$\varphi \to \psi$' $\in \mathcal{X}$, we also have '$\varphi \to \neg\psi$' $\in \mathcal{X}$. To abbreviate this notation, let us write $\neg x$ to denote '$\varphi \to \neg\psi$' when '$\varphi \to \psi$' denotes x.

The $(+, -)$-Symmetry & The $(\Box, \Diamond)$-Invariance

If x is valid, then $\neg x$ is unsatisfiable, and if x is unsatisfiable, then $\neg x$ is valid. For example: 'Socrates is a man and every man is mortal; Therefore Socrates is mortal.' This candidate inference is a +-instance. 'Socrates is a man and every man is mortal; Therefore Socrates is not mortal.' This must then be a −-instance. This relationship is quite obvious from a logical perspective, and it is, indeed, a corollary of definition 10. But it also seems likely that this hypothesis would be supported by the intuitive criterion of definition 12. More formally, we have for any atomic decision $d \in \mathcal{D}$:

$$[\neg d]_{+,-} = \begin{cases} - & \text{if } [d]_{+,-} = +, \\ + & \text{if } [d]_{+,-} = -. \end{cases}$$

It is this logic which participants had in mind when implementing, for example, a counter for negations, where an odd number of negations inverts the decision. Furthermore, this is the treatment afforded to negation by standard theorem provers as used, e.g. by Bos & Markert (2005*a*,*b*, 2006*a*,*b*).

On the other hand, if x is relevant, i.e. logically determined, then $\neg x$ will still be relevant. Conversely, if x is irrelevant, then $\neg x$ is equally irrelevant. Again, this is a corollary of the logical criterion of definition 9 and a hypothesis that seems likely to be supported by the intuitive criterion of definition 12. We define for any atomic decision $d \in \mathcal{D}$:

$$[\neg d]_{\Box,\Diamond} = \begin{cases} \Diamond & \text{if } [d]_{\Box,\Diamond} = \Diamond, \\ \Box & \text{if } [d]_{\Box,\Diamond} = \Box. \end{cases}$$

This notion of relevance can best be understood by considering the application of question-triggered information retrieval: Here, we might have a question such as 'Is Socrates mortal?' The sentence 'Socrates is Greek' does not provide an answer, and neither does 'Socrates is not Greek'. However, the sentence 'Socrates is mortal' does provide an answer, yet the same is true for its negation 'Socrates is not mortal.'

The difference is that the former answer is in the affirmative and the latter answer is in the negative, a distinction which may or may not play a role within any given application. If one is interested in retrieval applications, for example, one will often find that the validity distinction is drawn by the human user, not the computer, so that the retrieval model needs to handle only the relevance distinction.

The $(\triangle, \triangledown)$-Asymmetry

On the other hand, there are also applications where both relevance and validity plays a role, such as paraphrasing, semantic similarity, and text clustering of the kind performed in summarization. This problem has a straightforward formalization in the logical framework. Given two pieces of text φ and ψ, we need to determine $[\![\varphi \equiv \psi]\!]$.

In the logical framework, this is equivalent to saying that a candidate inference of the form $\varphi \to \psi$ is both relevant and valid, and that its converse $\psi \to \varphi$ is also both relevant and valid. This, again, is a corollary of definitions 9 and 10 of the logical criterion, and a hypothesis likely to be supported by the definitions 11 and 12 of the intuitive criterion. The relationship gives us a good idea of the notion of φ and ψ having the same meaning.

Let $x \in \mathcal{X}$ be some candidate inference which is both a $+$-instance and a $\square$-instance. Then x is a $\boxplus$-instance and thus a $\triangle$-instance. What if x is a $\triangledown$-instance? This may either be due to the fact that x is a $-$-instance, or due to the fact that x is a $\diamond$-instance. So, if x is a $\triangle$-instance, then $\neg x$ is a $\triangledown$-instance, but if x is a $\triangledown$-instance, then $\neg x$ may or may not be a $\triangle$-instance. We cannot express for arbitrary atomic decision $d \in \mathcal{D}$ the value of $[\neg d]_{\triangle,\triangledown}$ as a function of $[d]_{\triangle,\triangledown}$, but we could express it as a function of both $[d]_{+,-}$ and $[d]_{\square,\diamond}$.

2.1.4. Comparison Scores

In the previous sections, we have discussed candidate inferences, the structure of inference decisions, and the criteria for assigning such inference decisions to candidate inferences to arrive at empirical models. Within the empirical methodology, we also need to be able to compare models. For example one model G could be a gold standard reference model, and another model L could be that which a system has produced independently. What we need is a measure of how well L agrees with G.

16. Let $\mathcal{P}(\mathcal{X})$ be the powerset of the inference language $\mathcal{X}$. We say that α is a *comparison score* iff it is a mapping $\alpha : \mathcal{P}(\mathcal{X}) \times \mathcal{H}_{\mathrm{X}} \times \mathcal{H}_{\mathrm{X}} \mapsto \mathbb{R}$ mapping an inference sample and a pair of inference models to a number. For any $X \in \mathcal{P}(\mathcal{X})$, we will generally use the notation $\alpha^X(\mathrm{L}, \mathrm{G})$ instead of $\alpha(X, \mathrm{L}, \mathrm{G})$ to denote the numeric result obtained by comparing model L to model G on sample X.

17. Let $\mathcal{C}$ be a labelset, and let $\mathrm{G}, \mathrm{L} \in \mathcal{H}$ be inference models. For any inference sample $X \subseteq \mathcal{X}$, we define the usual contingency table and associated probabilities

$$\mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g, [\mathbf{L}]_{\mathcal{C}} = l\big) = \frac{1}{|X|} \sum_{x \in X} \mathbb{1}\Big([\mathrm{G}(x)]_C = g \;\wedge\; [\mathrm{L}(x)]_C = l\Big),$$

where $\mathbb{1}$ is a counter which takes on a numerical value of one when the logical expression in its argument is true and zero otherwise. We also define the associated marginals and conditionals as usual:

$$\begin{aligned}
\mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g\big) &= \sum_{l \in C} \mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g, [\mathbf{L}]_{\mathcal{C}} = l\big),\\
\mathbb{P}^X\big([\mathbf{L}]_{\mathcal{C}} = l\big) &= \sum_{g \in C} \mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g, [\mathbf{L}]_{\mathcal{C}} = l\big),\\
\mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g \,|\, [\mathbf{L}]_{\mathcal{C}} = l\big) &= \frac{\mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g, [\mathbf{L}]_{\mathcal{C}} = l\big)}{\mathbb{P}^X\big([\mathbf{L}]_{\mathcal{C}} = l\big)},\\
\mathbb{P}^X\big([\mathbf{L}]_{\mathcal{C}} = l \,|\, [\mathbf{G}]_{\mathcal{C}} = g\big) &= \frac{\mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g, [\mathbf{L}]_{\mathcal{C}} = l\big)}{\mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g\big)}.
\end{aligned}$$

Mutual Information

The comparison measure which we use in section 2.2 is mutual information. This has not been previously used as an evaluation measure at RTE.

18. Let $\mathcal{C}$ be a labelset, and let $\mathrm{G}, \mathrm{L} \in \mathcal{H}$ be empirical models. For any inference sample $X \subseteq \mathcal{X}$, we define the *prior entropy* $\mathbb{H}^X\big([\mathbf{G}]_{\mathcal{C}}\big)$ and the *mutual information* $\mathbb{I}^X\big([\mathbf{G}]_{\mathcal{C}}; [\mathbf{L}]_{\mathcal{C}}\big)$:

$$\begin{aligned}
\mathbb{H}^X\big([\mathbf{G}]_{\mathcal{C}}\big) &= -\sum_{g \in C} \mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g\big)\; \log\big(\mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g\big)\big);\\
\mathbb{H}^X\big([\mathbf{G}]_{\mathcal{C}} \,|\, [\mathbf{L}]_{\mathcal{C}} = l\big) &= -\sum_{g \in C} \mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g \,|\, [\mathbf{L}]_{\mathcal{C}} = l\big)\; \log\big(\mathbb{P}^X\big([\mathbf{G}]_{\mathcal{C}} = g \,|\, [\mathbf{L}]_{\mathcal{C}} = l\big)\big);\\
\mathbb{H}^X\big([\mathbf{G}]_{\mathcal{C}} \,|\, [\mathbf{L}]_{\mathcal{C}}\big) &= \sum_{l \in \mathrm{C}} \mathbb{P}^X\big([\mathbf{L}]_{\mathcal{C}} = l\big)\; \mathbb{H}^X\big([\mathbf{G}]_{\mathcal{C}} \,|\, [\mathbf{L}]_{\mathcal{C}} = l\big);\\
\mathbb{I}^X\big([\mathbf{G}]_{\mathcal{C}}; [\mathbf{L}]_{\mathcal{C}}\big) &= \mathbb{H}^X\big([\mathbf{G}]_{\mathcal{C}}\big) - \mathbb{H}^X\big([\mathbf{G}]_{\mathcal{C}} \,|\, [\mathbf{L}]_{\mathcal{C}}\big).
\end{aligned}$$

In what follows, we will consider evaluation measures which have been used as part of the RTE evaluation scheme.

Accuracy

Accuracy is simply the probability that, upon choosing a candidate inference at random from the sample, the model's decision will be equivalent in the chosen labelset with the gold standard decision. It was one of the measures used at RTE.

19. Let $\mathcal{C}$ be a labelset, and let $\mathrm{G}, \mathrm{L} \in \mathcal{H}$ be empirical models. For any inference sample $X \subseteq \mathcal{X}$, we define the *accuracy* $\mathrm{A}^{X}_{\mathcal{C}}\big(\mathrm{G};\mathrm{L}\big)$ as follows:

$$\mathrm{A}^{X}_{\mathcal{C}}\big(\mathrm{G};\mathrm{L}\big) = \frac{1}{|X|}\sum_{x\in X} \mathbb{1}\big([\mathrm{G}(x)]_{\mathcal{C}} = [\mathrm{L}(x)]_{\mathcal{C}}\big), = \sum_{c\in\mathcal{C}} \mathbb{P}^{X}\big([\mathbf{G}]_{\mathcal{C}} = c,\, [\mathbf{L}]_{\mathcal{C}} = c\big).$$

Average Precision & Confidence-Weighted Score

The previous measures compared, for a given sample, a given empirical model G to another model L. Average precision, however, compares an empirical model G to a retrieval-style ranking $\succ$. Confidence-weighted score compares the inference model G to another model L while taking into account $\succ$ as a confidence ranking.

20. Let $\mathrm{X} \subseteq \mathcal{X}$ be an inference sample. We define

$$\{\succeq x\} = \{x' \in \mathrm{X} \mid x' \succeq x\}.$$

21. Let $\mathrm{X} \subseteq \mathcal{X}$ be an inference sample, let $\mathcal{C}$ be a labelset, and $\mathrm{G}, \mathrm{L} \in \mathcal{H}$ be inference models. Furthermore, let $\succ$ be a ranking of X. We define the *confidence-weighted score* $\mathrm{CWS}^{\mathrm{X}}_{\mathcal{C}}(\mathrm{G};\mathrm{L},\succ)$ as follows:

$$\mathrm{CWS}^{\mathrm{X}}_{\mathcal{C}}(\mathrm{G};\mathrm{L},\succ) = \frac{1}{\mathrm{X}}\sum_{x\in\mathrm{X}} \mathrm{A}^{\{\succeq x\}}_{\mathcal{C}}\big(\mathrm{G};\mathrm{L}\big).$$

This confidence-weighted score was used at RTE-1 but then discontinued. As the name suggests, it is a weighted average of accuracy scores, where those weights come from a confidence-ranking $\succ$. Let L be the empirical model represented by the system. Then, if $\mathrm{G} = \mathrm{L}$, we will get a perfect score $\mathrm{CWS}_{\mathrm{C}}(\mathrm{G};\mathrm{L},\succ) = 1.0$, regardless of $\succ$. The weighting imposed by $\succ$ is thus a proper confidence weighting, in the sense that we rank x higher than x' iff we have higher confidence in hypothesizing that $\mathrm{G}(x) = \mathrm{L}(x)$ than we do in hypothesizing that $\mathrm{G}(x') = \mathrm{L}(x')$. Whenever a model labels an instance such that $\mathrm{G}(x) \neq \mathrm{L}(x)$, confidence-weighted score penalizes that model in a weighted fashion: The higher up in the ranking the incorrect decision occurs, the higher the penalty.

22. Let $\mathrm{X} \subseteq \mathcal{X}$ be an inference sample, let $\mathcal{C}$ be a labelset, and $\mathrm{G}, \mathrm{L} \in \mathcal{H}$ be inference models. Let $\succ$ be a ranking of X. We define the *average precision* $\mathrm{AP}_{\mathcal{C},c}(\mathrm{G},\succ)$ as follows:

$$\mathbb{P}\big([\mathrm{G}(\mathbf{x'})]_{\mathcal{C}} = c \mid \mathbf{x'} \succeq x\big) = \frac{\sum_{x'\in\mathrm{X}} \mathbb{1}\big([\mathrm{G}(x')]_{\mathcal{C}} = c\big) * \mathbb{1}\big(x' \succ x\big)}{\sum_{x'\in\mathrm{X}} \mathbb{1}\big(x' \succ x\big)};$$

$$\mathrm{AP}_{\mathcal{C},c}(\mathrm{G},\succ) = \frac{\sum_{x\in\mathrm{X}} \mathbb{P}\big([\mathrm{G}(\mathbf{x'})]_{\mathcal{C}} = c \mid \mathbf{x'} \succeq x\big) * \mathbb{1}\big([\mathrm{G}(x)]_{\mathcal{C}} = c\big)}{\sum_{x\in\mathrm{X}} \mathbb{1}\big([\mathrm{G}(x)]_{\mathcal{C}} = c\big)}.$$

This average precision measure was used from RTE-2 onwards. When using average precision as an interpretation for the ranking $\succ$, this ranking becomes a retrieval-style ranking, not a confidence ranking as used by confidence-weighted score. Average precision directly compares this ranking $\succ$ to the gold standard inference model G. Here, the model represented by the system L does not enter the picture.

For the average precision score $\mathrm{AP}_{\mathcal{C},c}$, we need to designate one label $c \in \mathcal{C}$ as denoting the positive class, all other labels then implicitly denoting the negative class. The retrieval-style ranking $\succ$ is considered to be in perfect agreement with the gold standard G, if for every pair of instances x and x' for which $\mathrm{G}(x) = c$ and $\mathrm{G}(x') \neq c$, we have $x \succ x'$. For example, the RTE evaluation scheme designates the label △ as positive, and a perfect model as one which successfully ranks all △-instances ahead of any ▽-instances. Whenever a model ranks a ▽-instance at a rank where there is supposed to be a △-instance, average precision penalizes that model in a weighted fashion: The higher up in the ranking the illegal insertion occurs, the higher the penalty. All a model needs to do here is position instances somewhere in a continuum between △-instances and ▽-instances, without ever having to apply a decision boundary to that continuum.

Average Precision vs. Confidence-Weighted Score at RTE

There was some confusion at past RTE challenges about the distinction between confidence rankings as imposed by confidence-weighted score at RTE-1 on one hand and retrieval-style rankings as imposed by the average precision score from RTE-2 onwards on the other.

For example, the average precision score of a submission by Padó et al. (2008) in RTE-4 goes up from .44 to .62 after reranking negatively-labelled instances to the bottom of the ranking and inverting their order so as to be consistent with the proper interpretation of average precision. – Six of the 26 groups who participated at RTE-4 submitted confidence-ranked three-way labellings, and four out of these six appear to have misunderstood the score in this way. Three of the four would have benefitted from reranking.

In the next section, we will have more to say about further unintuitive properties of the average precision score which may help explain this confusion. At this point it is, however, noteworthy that the problem applied only to ranked 3-way decisions, which invites the following speculation about the deeper causes: Participants who submitted ranked 2-way decisions may have been thinking about the task in terms of the retrieval analogy, thus producing retrieval-style rankings over the relevance criterion. Participants who submitted non-ranked 3-way decisions, on the other hand, may have been thinking about the task in logical terms. The clash of the two paradigms was apparent with those participants whose ranked 3-way decisions contradicted constraints implied by the very definition of the RTE evaluation. But it stands to reason that this clash of the retrieval analogy with the logical paradigm caused a more widespread misunderstanding about the underlying interpretation. – This thesis is, to the best of my knowledge, the first treatment of the RTE where this distinction between the logical paradigm and the retrieval analogy is being rigorously drawn.

2.1.5. Relabelling Isomorphicity

One common theme from the previous section is that a label in and of itself is meaningless. All comparison scores, except average precision, apply meaning to labels only to the extent that the labels serve to represent equivalence classes of candidate inferences. This is a commonplace property of logical semantics. In this section, it will serve to further substantiate the contrast between the logical paradigm of recognizing textual entailment on one hand, and the retrieval analogy on the other.

23. Let $\mathrm{X} \subseteq \mathcal{X}$ be an inference sample, $\mathcal{C}$ be a labelset, and $\mathrm{f} : \mathcal{D} \mapsto \mathcal{D}$ be some function on labels. We say that f is a *$\mathcal{C}$-relabelling morphism on* X iff for all $c, d \in \mathcal{D}$ we have $[f(c)]_{\mathcal{C}} = [f(d)]_{\mathcal{C}}$ only if $[c]_{\mathcal{C}} = [d]_{\mathcal{C}}$.

24. Let α be some comparison score. We say that α is *relabelling-isomorphic* iff, for every inference sample $X \subseteq \mathcal{X}$ and for every labelset $\mathcal{C}$, we have, for every $\mathcal{C}$-relabelling morphism f on X and for every pair of inference models $L, M \in \mathcal{H}$, that

$$\alpha^{\mathrm{X}}(G, L) = \alpha(f \circ G, f \circ L).$$

One can easily check from the above definitions that accuracy, mutual information, and confidence-weighted score are relabelling-isomorphic, but average precision is not. This is due to the fact that none of the relabelling-isomorphic comparison scores' definitions refer to labels directly, only to comparisons of labels. Thus, they cannot intrinsically make any distinctions between labels; labels are simply used as a symbolic representation for equivalence classes over $\mathcal{X}$.

This treatment of inference decision labels as equivalence classes on candidate inferences agrees well with the negation properties of inference decisions we discussed before (section 2.1.3). From this previous discussion, it follows that the function $\neg : \mathcal{D} \mapsto \mathcal{D}$ is a $\mathcal{C}_{+,-}$-relabelling morphism, and also a $\mathcal{C}_{\Box,\Diamond}$-relabelling morphism, but not a $\mathcal{C}_{\vartriangle,\triangledown}$-relabelling morphism.

Now, let G be a gold standard, L be a model, and let α be a comparison score. Let's say we derive from the original sample another sample by negating all the consequents, so that the new gold standard sample is $\neg\mathrm{G}$, where $(\neg\mathrm{G})(x) = \neg(\mathrm{G}(x))$ for all x. Furthermore, let's say the model is perfect at detecting such negation, so that the new model is $\neg\mathrm{L}$. Now, if α is relabelling-isomorphic, this will ensure that the model retains its original score, i.e. that $\alpha(\mathrm{G}, \mathrm{L}) = \alpha(\neg\mathrm{G}, \neg\mathrm{L})$.

In contrast to all the other comparison scores, average precision is not relabelling-isomorphic. Its definition refers to a particular label $c \in \mathcal{C}$ designated as the positive class. For example the $\mathrm{AP}_{\{\vartriangle,\triangledown\},\vartriangle}$ evaluation measure used at RTE makes an intrinsic distinction between the $\vartriangle$-label and the $\triangledown$-label by weighting errors that affect the rankings of $\vartriangle$-instances more heavily than those affecting the $\triangledown$-instances.

Since the set of $\vartriangle$-instances and the set of $\triangledown$-instances of X are disjoint, the two average precisions $\mathrm{AP}(\mathrm{G}; \succ)$ and $\mathrm{AP}(\neg\mathrm{G}; \succ')$, regardless of how $\succ$ relates to $\succ'$, are independent. – Note how this contradicts relabelling isomorphicity. Relabelling isomorphicity ensures that, when we apply some function to relabel G, then, by simply applying that same function to L also, a model can still satisfy the evaluation criterion. Average precision, on the other hand, rules out the existence of such a functional relationship between $\succ$ and $\succ'$ reflecting a given relabelling of G.

Average Precision & Relabelling Isomorphicity at RTE

So from the previous section it follows that average precision breaks up symmetries between its labels. This is unsurprising, given that in its traditional applications, this property is precisely what motivates its use. In information retrieval, for example, one does not wish irrelevant, non-retrieved documents to enter into evaluation scores, since they do not affect the user.

But this is a specific assumption which is made in information retrieval about the way in which users interact with ranked retrieval user interfaces. At the Text REtrieval Conference (TREC)[4], this property was particularly important, as scores were also subjected to macro-averaging across topics with different numbers of documents in the collection.

To the extent that an RTE system is to be evaluated for its usefulness in the particular context of a question-triggered retrieval system with a ranked-retrieval user interface, this property may apply to RTE evaluation, but it is hard to see how this would generalize, and how it would apply, for example, to information extraction or summarization.

In general, the retrieval analogy seems flawed. An RTE system must be evaluated not only on its ability to recognize $\vartriangle$-instances, but also on its ability to recognize $\triangledown$-instances. Nobody has, to the best of my knowledge, explicitly made an argument that errors pertaining to the former problem should carry more weight than errors pertaining to the latter, yet it has been adopted as an integral part of the evaluation methodology.

[4] For an overview on TREC, see Voorhees & Harman (2005).

2.1.6. Prior Distribution Effects

In this section, we will study how the prior distribution $\mathbb{P}(\mathbf{G} = g)$ affects the interpretation of scores α comparing some inference model G to other models L. In particular, we will consider bias and degradation, two properties of accuracy and related scores which contradict certain intuitions one may have about the RTE evaluation scheme. We will show that these counter-intuitive properties do not apply to mutual information.

We will use as a running example the following prior: $\mathbb{P}(\mathbf{G} = \boxplus) = 0.5$, $\mathbb{P}(\mathbf{G} = \Diamond) = 0.35$, $\mathbb{P}(\mathbf{G} = \boxminus) = 0.15$, which was the distribution of the 3-way gold standard labels both at RTE-4 and RTE-5. Note that this means that $\mathbb{P}(\mathbf{G} = \triangle) = 0.5$, $\mathbb{P}(\mathbf{G} = \triangledown) = 0.5$.

Bias

To demonstrate the effects of bias, it is instructive to consider the inference models $\mathrm{L}_{\boxplus}$, $\mathrm{L}_{\Diamond}$, and $\mathrm{L}_{\boxminus}$ produced by the constant choice baseline strategies which uniformly assign the same label $\boxplus$, $\Diamond$, and $\boxminus$, respectively. Let us also consider the average case scores of inference models $\mathrm{L}_{\ast}$ which assign labels randomly.

We observe that $\mathrm{A}(\mathrm{G};\mathrm{L}_{\boxplus}) = 0.5$, that $\mathrm{A}(\mathrm{G};\mathrm{L}_{\Diamond}) = 0.35$, and that $\mathrm{A}(\mathrm{G};\mathrm{L}_{\boxminus}) = 0.15$. For uniform random choice, we get $\mathrm{A}(\mathrm{G};\mathrm{L}_{\ast}) = 0.333$. And we get $\mathrm{A}(\mathrm{G};\mathrm{L}'_{\ast}) = 0.395$ for a random choice model which correctly reproduces the prior from the gold standard but assigns the labels to instances at random.

This already shows that it is potentially misleading to view accuracy as a one-dimensional projection of how good a model is in the context of an evaluation scheme. The three constant choice models and the random choice models are all equally uninformed about how to make inference decisions on the basis of candidate inferences, since, in all three cases, the candidate inferences themselves do not at all enter the picture. Yet, the accuracy score already favours certain inference models over others.

In fact, accuracy scores of up to 0.5 for such zero-information baselines seem like large numbers in the context of the RTE 3-way task. Both at RTE-4 and at RTE-5, one third of all submissions for the 3-way task scored lower. At the RTE-3 3-way pilot (Voorhees 2008), scores were even worse, with two thirds of all systems scoring lower than the constant choice baseline.

This seems counter-intuitive: Whenever one of these zero-information baselines happens to assign a label that turns out to be the correct label, this must be, by definition, a lucky guess. We would want a comparison score to quantify the extent to which a model can make distinctions between labels on the basis of candidate inferences beyond this level of lucky guessing.

20 (45)	25 (0)	5	$\mathbb{P}(\mathbf{G} = \boxplus)$ $= .5$
9 (27)	18 (0)	9	$\mathbb{P}(\mathbf{G} = \diamond)$ $= .36$
1 (8)	7 (0)	6	$\mathbb{P}(\mathbf{G} = \boxminus)$ $= .14$
$\mathbb{P}(\mathbf{L} = \boxplus)$ $= .3$ (.8)	$\mathbb{P}(\mathbf{L} = \diamond)$ $= .5$ (0)	$\mathbb{P}(\mathbf{L} = \boxminus)$ $= .2$ (.2)	N = 100

$$\begin{aligned} -\mathbb{H}(\mathbf{G}) &= .5 \log_2(.5) \\ &+ .36 \log_2(.36) \\ &+ .14 \log_2(.14) \\ &= -1.4277 \end{aligned}$$

$$\begin{aligned} -\mathbb{H}(\mathbf{G}|\mathbf{L} = \boxplus) &= \frac{20}{30} \log_2(\frac{20}{30}) \\ &+ \frac{9}{30} \log_2(\frac{9}{30}) \\ &+ \frac{1}{30} \log_2(\frac{1}{30}) \\ &= -1.0746 \end{aligned}$$

$$\begin{aligned} -\mathbb{H}(\mathbf{G}|\mathbf{L} = \diamond) &= \frac{25}{50} \log_2(\frac{25}{50}) \\ &+ \frac{18}{50} \log_2(\frac{18}{50}) \\ &+ \frac{7}{50} \log_2(\frac{7}{50}) \\ &= -1.4277 \end{aligned}$$

$$\begin{aligned} -\mathbb{H}(\mathbf{G}|\mathbf{L} = \boxminus) &= \frac{5}{20} \log_2(\frac{5}{20}) \\ &+ \frac{9}{20} \log_2(\frac{9}{20}) \\ &+ \frac{6}{20} \log_2(\frac{6}{20}) \\ &= -1.5395 \end{aligned}$$

$$\begin{aligned} -\mathbb{H}(\mathbf{G}|\mathbf{L}' = \boxplus) &= \frac{45}{80} \log_2(\frac{45}{80}) \\ &+ \frac{27}{80} \log_2(\frac{27}{80}) \\ &+ \frac{8}{80} \log_2(\frac{8}{80}) \\ &= -1.3280 \end{aligned}$$

$$\begin{aligned} \mathbb{H}(\mathbf{G}|\mathbf{L}) &= .3 * 1.0746 \\ &+ .5 * 1.4277 \\ &+ .2 * 1.5395 \\ &= 1.3441 \end{aligned}$$

$$\begin{aligned} \mathbb{H}(\mathbf{G}|\mathbf{L}') &= .8 * 1.3280 \\ &+ .2 * 1.5395 \\ &= 1.3703 \end{aligned}$$

Figure 2.2.: example contingency table and entropy calculations

By referring to our trivial models as zero-information baselines, we have, however, already anticipated the solution to the problem: mutual information. In order to demonstrate mutual information and how it deals with the problem of bias, let us consider the example contingency table of Figure 2.2.

Here, the unconditional entropy $\mathbb{H}(\mathbf{G})$ serves as a convenient measure of the hardness of the classification task itself, taking into account the number of labels and their prior distribution. This example has, again, been chosen so as to reflect our example prior, which is the one that was used for RTE-4 and RTE-5. It yields a value for $\mathbb{H}(\mathbf{G})$ of 1.4277 bits. This indicates that it is harder to guess the three-way gold standard label than it is to guess the two-way label or the outcome of a toss of a fair coin, which would both have an entropy of exactly 1 bit. On the other hand, it is easier to guess this outcome with the given prior than it would be if the prior were uniform. In that latter case, we would have an entropy of 1.5850 bits.

Similarly, we can calculate a conditional entropy $\mathbb{H}(\mathbf{G}|\mathbf{L} = l)$ over a conditional distribution of gold standard labels observed, given that the model has assigned label l to our randomly chosen candidate inference. In the example, we have calculated a value of 1.0746 bits for $\mathbb{H}(\mathbf{G}|\mathbf{L} = \boxplus)$. So, while the hardness of guessing the correct label without any additional knowledge is 1.4277, it will be easier to guess this label correctly once the model-assigned label is known to be $\boxplus$. Our best guess would be to always assign label $\boxplus$, which would be successful 50% of the time. But, among the cases where the model has assigned label $\boxplus$, this would be an even better guess. It would now be correct 66% of the time. We have gained information about the gold standard by taking into account the model-assigned label.

The conditional entropy $\mathbb{H}(\mathbf{G}|\mathbf{L})$ is the expected value of the conditional entropy $\mathbb{H}(\mathbf{G}|\mathbf{L} = l)$ across all possible labels l, when, as before, we draw a candidate inference at random.

One very noteworthy property of this measure is that all of the baseline models we considered, i.e. models assigning constant labels or models assigning labels at random, would have $\mathbb{H}(\mathbf{G}|\mathbf{L}) = \mathbb{H}(\mathbf{G})$, since the distribution of gold standard labels given the model labels, in all of these cases, is the same as the prior distribution. Furthermore, $\mathbb{H}(\mathbf{G}) = 1.4277$ is, in fact, an upper bound on $\mathbb{H}(\mathbf{G}|\mathbf{L})$. All the trivial baseline models would perform at this upper bound level, giving a mutual information $\mathbb{I}(\mathbf{G};\mathbf{L})$ of zero.

At the other extreme end of the spectrum, consider a perfect contingency table, where all the non-diagonal cells are zero. In this case all the conditional entropies $\mathbb{H}(\mathbf{G}|\mathbf{L} = l)$ would be entropies over delta distributions concentrating all probability mass on a single label. This would yield a value of $\mathbb{H}(\mathbf{G}|\mathbf{L}) = 0$, which is a lower bound for any entropy.

The model producing our contingency table performs worse than this ideal but better than the baselines, at $\mathbb{H}(\mathbf{G}|\mathbf{L}) = 1.3441$ which yields $\mathbb{I}(\mathbf{G};\mathbf{L}) = 1.4277 - 1.3441 = .0836$ bits. We have gained .0836 bits' worth of information by looking at the model-assigned label.

Degradation

Accuracy also suffers from the problem of degradation, by which we mean the property that accuracy may reward a model for collapsing distinctions between labels in its output. The numbers in the example of Figure 2.2 have been chosen so as to illustrate this.

In the example, the conditional distribution $\mathbb{P}(\mathbf{G} = g|\mathbf{L} = \Diamond)$ is the same as the unconditional distribution $\mathbb{P}(\mathbf{G} = g)$, so when it turns out that $\mathbf{L} = \Diamond$, no additional information has been revealed about $\mathbf{G}$. In information theoretic terms, one would argue that this is a good thing. We have separated a less informative category from the more informative ones, so we know what we know and what we don't know.

What happens if we now conflate the labels $\Diamond$ and $\boxplus$ in the model output? In Figure, 2.2, the numbers in brackets illustrate this. Previously, the model assigned label $\boxplus$ in 30% of

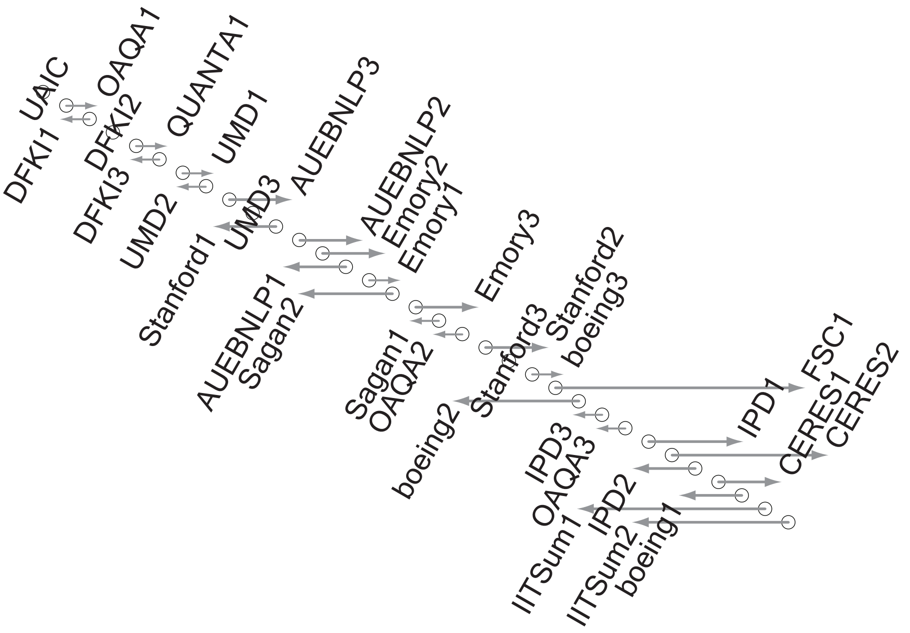


Figure 2.3.: RTE-4 submissions reranked by Mutual Information

all cases. In those cases, the model's choice was relatively well-informed, as ⊞ actually turned out to be the correct gold standard label 66% of the time. But now, with the labels conflated, the model chooses ⊞ in 80% of the cases; a choice which is now much less well-informed, as it is correct only 45% of the time.

Mutual information shows a drop from $.0836$ bits down to $.0262$ as accuracy increases from 44% to 51%. – Mutual information penalizes and accuracy rewards a model for destroying information in this example by conflating a well-informed label with a less well-informed label and thereby diluting the information content of the individual labels, and obscuring the output to less certainty and more guesswork.

Reranking RTE-4

Given these various advantages of mutual information over accuracy, the question arises how a move from accuracy to mutual information as an evaluation measure would affect rankings at RTE. Figure 2.3 shows a reranking of submissions for the 3-way task, the shaft of each arrow corresponding to the rank assigned to the submission by 3-way accuracy and the tip corresponding to the rank assigned by mutual information.

We observe that movements at the top of the ranking are small, compared to much larger movements towards the bottom of the ranking. This reflects the properties we have previously discussed, viz. that the move to mutual information is a recalibration of the baseline so that a zero score means an uninformed or random labelling, not zero agreement. For example, we have previously mentioned how models can end up performing at worse levels of accuracy than the zero-information models. On the other hand, the perfect model would have both a zero conditional entropy (maximal mutual information) and a 100% accuracy.

However, it is noteworthy that, even towards the top of the ranking, we often see the relative rankings of different runs from the same group inverting. For example, UMD runs 1 and 2, AUEBNLP runs 1 and 2, STANFORD runs 2 and 3, BOEING runs 2 and 3, and CERES runs 1 and 2 are cases where mutual information has different preferences than accuracy. – This means that, to the extent that participants' design choices and empirical conclusions have been informed by accuracy, those would have to be reviewed if the goal were to maximize mutual information rather than accuracy.

Weighting of Dimensions at RTE

Another immediate application of mutual information is in addressing the question we previously raised on how the two independent dimensions of the inference decision affect the structured labels. Recall, that the labelsets $\mathcal{C}_{\square,\diamond}$ and $\mathcal{C}_{+,-}$ represent relevance and validity, respectively, and that the labels $\mathcal{C}_{\triangle,\triangledown}$ used at RTE are a logical conjunction of those two indepenend criteria.

To that end, one can calculate that

$$\mathbb{I}\big([\mathrm{G}]_{\blacktriangle,\blacktriangledown};[\mathrm{G}]_{\square,\diamond}\big) = .6103,$$

and that

$$\mathbb{I}\big([\mathrm{G}]_{\blacktriangle,\blacktriangledown};[\mathrm{G}]_{+,-}\big) = .5330,$$

assuming the 3-way prior from RTE-4 and RTE-5 and stochastic independence of the two dimensions. So, we gain more information about the $\mathcal{C}_{\triangle,\triangledown}$-distinction by knowing the $\mathcal{C}_{\square,\diamond}$-distinction than we do by knowing the $\mathcal{C}_{+,-}$-distinction.

This is also reflected in accuracy. We get $\mathrm{A}_{\triangle,\triangledown}(\mathrm{G};\mathrm{L}) = .77$ for a model L which always correctly decides the $\mathcal{C}_{\square,\diamond}$-distinction, but which decides the $\mathcal{C}_{+,-}$-distinction by assigning labels randomly according to the correct prior. – This theoretical score was never beaten by any RTE submission throughout the history of the challenge.

For a model L′ which, conversely, decides the $\mathcal{C}_{+,-}$-distinction correctly and the $\mathcal{C}_{\square,\diamond}$-distinction randomly, we get a much lower theoretical accuracy of $\mathrm{A}_{\triangle,\triangledown}(\mathrm{G};\mathrm{L}') = .59$.

This supports the claim we made in section 2.1.2 that the relevance distinction has a greater bearing on the RTE datasets than the validity distinction. The effect is in fact so pronounced, that it is unclear whether the validity distinction task has ever been successfully addressed by any RTE submission at all.

2.2. Empirical Methodology & Review of Systems

Having established the necessary theoretical preliminaries in the previous section (section 2.1), we can now move on to put to use this theory to shed some more light on the results obtained for the RTE-4 evaluation.

As we will see, some of our hypothesis tests are less powerful than one might hope or not possible at all, due to limitations arising from the fact data collection practices employed at past RTE evaluations are imcompatible with the theory of evaluation presented here. Nevertheless, there are some interesting results which we can obtain from available data.

Note that, while referring to the RTE methodology as an evaluation scheme, I deliberately avoid this terminology in connection with the empirical methodology used here. The notion of evaluation employed in the context of recent competitive tasks in computational linguistics carries the connotation of collapsing a complex picture to a one-dimensional account. For a given shared task, the usual sort of one-dimensional evaluation scheme would test only a single hypothesis of the form "system X is better at the task than system Y" for some usually less than well-defined and less than agreed-upon interpretation of "better".

Our methodology is rather aimed at putting forward a battery of hypotheses which are collectively exhaustive and mutually independent in describing aspects of the data we have available to us: We will therefore be guided by the data structures which arise from the collection process itself: (1) The common matrix is the data structure obtained by making comparisons between a fixed gold standard and different empirical models. One dimension of the matrix compares different empirical models, the other dimension compares different samples and measurement criteria. Measurement criteria, in particular, include different ways of identifying and counting errors. (2) The concentration plot goes one step further. Rather than just counting errors, it visualizes error characteristics. Here, the gold standard and a designated baseline model are taken as reference points. Distances of points representing other empirical models then correspond to agreements and disagreements between labellings, making it easy to visually spot concentrations of systems with similar behaviour.

	QA	IR	IE	SUM	$[\cdot]_{\triangle,\triangledown}$	$[\cdot]_{+,-}$	$[\cdot]_{\square,\diamond}$	A	$\mathbb{I}$	...
CONST. $\boxplus$										
CONST. $\boxminus$										
CONST. ⟐										
CONST. ⬙										
RAND. UNIFORM										
RAND. PRIOR										
BOW										
SYS-1										
SYS-2										
...										

Figure 2.4.: schema of the common matrix

2.2.1. Data Collection & Analysis: The Common Matrix

Figure 2.4 visualizes the data structure which will be in the centre of attention for our empirical methodology. This figure lends a visual interpretation to our conceptualization of the RTE evaluation as one-dimensional. We propose a matrix of at least two dimensions to take the place of the one-dimensional tables published by RTE organizers in their overview papers. The traditional tables collapse the different data samples into one and thus report summary statistics corresponding to only a single column of the matrix. By reading off such a column from the matrix, one can compare the performance of different models. But, we believe it is equally important to be able to read off each row in the matrix the performance of a given model when measured under different conditions.

In particular, columns in the matrix represent different choices of comparison scores α, and, as part of that choice, different samples, different equivalence classes for labels, or different scoring methods. In principle these might be independent, to a certain extent, leading to a three- or four-dimensional matrix, but the choice of hypothesis to be tested, and the design of the associated hypothesis tests will dictate some combinations which are more useful than others.

Our main criticism with the RTE evaluations, in this context, is not so much with the one-dimensional physical layout of the results tables published as part of the overview papers, rather it is with the fact that not enough attention is being devoted to this second dimension of the matrix, and the fact that community participation is invited only for one dimension. Participants can submit systems, but they cannot submit data samples for which they want to observe the behaviour of other participants' systems.

In the rest of this section, we will consider some columns of the matrix for which the data has already been collected within past RTE evaluations, and we will then move on to suggest new columns that could be added for similar future work in the context of a community process.

Applications: QA vs. IR vs. IE vs. SUM

From an application-oriented point of view, textual entailment is widely understood as an abstraction over the central decision procedures in a number of different applications. In particular, the RTE organizers have advocated the viewpoint that textual entailment systems can be put to work in question answering (QA), information retrieval (IR), information extraction (IE), and summarization (SUM), and have employed four different sampling techniques in an attempt to represent the different needs of these four applications. The RTE dataset itself is the union of these four datasets. In this section, we will present some evidence to suggest that this may really be all there is to this dataset: the union of four unrelated datasets, not a statistically coherent sample of an abstract task.

Suppose, for the purpose of contradiction, that textual entailment were, in fact, a coherent task and that one given entailment engine were equally useful as a component of a larger system in any of the four applications, as implied by the tacit assumptions underlying RTE. Then, given two empirical models, $\mathrm{H}_1, \mathrm{H}_2 \in \mathcal{H}$, one would have to be preferable to the other. Let's take H_1 to be the one which is preferable. Also note that for this argument it does not matter what exactly it is that constitutes one's preference. Now H_1 would have to be preferable also on each of the four individual applications. If, however, it were possible for H_1 to be preferable to H_2 in, for example, QA, while H_2 is preferable to H_1 in IR, IE and SUM, then the notion of an RTE system, at least from this application-oriented point of view, loses not only its coherence but any kind of usefulness at all: averaging accuracy scores over the four applications would then only obscure, rather than illuminate the underlying issues.

H25. Let SUM, QA, and IE be the subsamples of the RTE-4 inference sample, and note that they form a partition. Let G be the gold standard model, and consider the different participating models L. We can use some appropriate comparison score α to evaluate $\alpha(\mathrm{G}, L)$ for each system L on each of the four subsamples separately.

> *(Null Hypothesis)* Systems generally exhibit the same behaviour on the four subsamples. When each system is scored against the four subsamples, these four scores do not differ significantly. When systems are ranked against each other according to their score on each of the four subsamples, then these four rankings do not differ significantly.
>
> *(Alternative Hypothesis)*

	$\hat{\mu}$	$\hat{\sigma}$	q1	med	q3	max
IR	.072	.0630	.022	.066	.103	.327
SUM	.052	.0531	.013	.042	.077	.282
QA	.022	.0463	.001	.003	.011	.205
IE	.020	.0451	.000	.001	.010	.229

(a) summary statistics on $\mathbb{I}([\mathrm{G}]_{\vartriangle,\triangledown};[\mathrm{L}]_{\vartriangle,\triangledown})$

IR	QA	SUM	IE	
	.15	.60	.41	IR
		.12	.21	QA
			.52	SUM
				IE

(b) Kendall's τ on $\mathbb{I}([\mathrm{G}]_{\vartriangle,\triangledown};[\mathrm{L}]_{\vartriangle,\triangledown})$

Figure 2.5.: scores for 81 systems on different applications

1. Scores differ significantly and relate to each other as follows:
   (a) IR > SUM, (b) SUM > QA, ~~(c) QA > IE~~.
2. Rankings of models differ when scored against different applications.

Test. This hypothesis is straightforwardly testable from the data collected for RTE-4. As per our arguments from the previous section (section 2.1), particularly those from section 2.1.6, mutual information seems the best choice of comparison score for the purposes of testing this hypothesis.[5]

As part of our choice of an appropriate comparison score α, we also need to choose a label equivalence. The obvious choice here is to use the $\mathcal{C}_{\vartriangle,\triangledown}$ labelset, since RTE-4 collected data based on that labelset for 81 different systems. We could have used $\mathcal{C}_3$, but this would have narrowed the data in the hypothesis test to the 45 systems which participated in the 3-way task.

The table in Figure 2.5a presents some summary statistics on mutual information, from which we can see that the sample means, seen as point estimates, agree with part 1 of the alternative hypothesis.

The tables in Figures A.2 and A.3 (appendix A) also list the relevant statistics for a sign test and for a Mann-Whitney U-test of our hypothesis.

As opposed to the t-test, which would make a normality assumption that seems unjustified on this data, these two tests are nonparametric. Both of them reject the null hypothesis in favour of parts 1a and 1b of the alternative hypothesis, but they do not reject it in favour of part 1c, even at the 20% level of significance for the two-tailed version of the test. So systems perform better on IR than they do on SUM, and better on SUM than they do on QA and IE. But there is not enough evidence to contradict the part of the null hypothesis which suggests that they perform the same on QA as they do on IE.

[5] Figures A.1, A.2, A.3, and A.4 in appendix chapter A also give the numbers to retrace this test based on accuracy to yield the same conclusions.

As for part 2 of the alternative hypothesis, Figure 2.5b presents Kendall's tau statistics to compare rankings. Recall that a Kendall's tau of zero means that the rankings are randomly reshuffled, and a Kendall's tau of one means that the rankings are the same.

As we can see, the ranking of systems by their performance on the QA subsample does not correlate well with rankings based on other subsamples. These are under the 20%-mark, roughly, which means that, picking 100 pairs of systems at random, we would expect that, at best, 60 pairs show the same relative ranking with 40 pairs showing opposite rankings on QA when compared to another subsample.

The other three subsamples show higher rank correlations, the highest being between SUM and IR at about 60% (80 concordant vs. 20 discordant pairs in every 100).

So we reject the null hypothesis also in favour of part 2 of the alternative hypothesis, concluding that the rankings do differ significantly. This statistical significance comes from the fact that, given our 81 systems, there are $\frac{80*81}{2} = 3240$ possible pairs of systems. So deviations in Kendall's tau are statistically significant beyond the level of precision reported in our table, and none of these rank correlations is anywhere near 1.0. □

So, in conclusion we note that this evidence casts doubt on the coherence and usefulness of the notion of textual entailment as an abstraction over the four applications considered at the RTE evaluations.

Decision Dimension: Relevance vs. Validity

One theme we keep coming back to is the two-dimensional nature of the inference decision, being composed of a relevance decision and a validity decision. We have previously seen, that (i) these decisions can be defined independently of each other (section 2.1.2), that (ii) they have different theoretical properties (section 2.1.3), and that (iii) due to the bias in the RTE-4/5 datasets, it is predominantly the relevance distinction which is being represented in the data (section 2.1.6). In this section, we will explore how systems perform on each of the dimensions individually.

H26. Let X be the RTE-4 inference sample, let G be the gold standard model, and consider the different participating inference models L. Let α be some comparison score such that $\alpha(\mathrm{G}, L)$ evaluates the performance of each system L on the validity distinction, and let α' be another comparison score such that $\alpha'(\mathrm{G}, L)$ evaluates the performance of L on the relevance distinction.

> *(Null Hypothesis)* There is no significant difference in models' performance on validity and their performance on relevance:
>
> $$\alpha(\mathrm{G}, L) \approx \alpha'(\mathrm{G}, L).$$

	$\hat{\mu}$	$\hat{\sigma}$	q1	med	q3	max
$\boxplus\boxminus/\Diamond$	.036	.0356	.010	.036	.062	.136
$\boxplus/\Diamond\boxminus$	.030	.0395	.006	.018	.040	.187
$\boxplus\Diamond/\boxminus$	.019	.0433	≈ 0	.004	.012	.229

Figure 2.6.: summary statistics on $\mathbb{I}'\big([G]; [L]\big)$: scores for 36 systems on different components of the 3-way inference decision

(Alternative Hypothesis) Systems are worse at validity than they are at relevance:

$$\alpha(G, L) < \alpha'(G, L).$$

Test. Ideally, what is needed to test this hypothesis is a comparison

$$\mathbb{I}\big([G]_{+,-}; [L]_{+,-}\big) \text{ vs. } \mathbb{I}\big([G]_{\Box,\Diamond}; [L]_{\Box,\Diamond}\big).$$

Since the RTE data uses only the three-way labelset $\boxplus$ vs. $\Diamond$ vs. $\boxminus$, and not the full four-way set of atomic decisions $\boxplus$ vs. $\diamondplus$ vs. $\diamondminus$ vs. $\boxminus$, we cannot directly observe performance on the validity distinction + vs. − separately from the relevance distinction $\Box$ vs. $\Diamond$.

However, we can decompose the three-way distinction into three two-way distinctions and observe performance on the different two-way distinctions which do and do not involve the validity distinction, viz. we can observe

$$\mathbb{I}\big([G]_{\boxplus,\Diamond\cup\boxminus}; [L]_{\boxplus,\Diamond\cup\boxminus}\big) \text{ vs. } \mathbb{I}\big([G]_{\boxplus\cup\Diamond,\boxminus}; [L]_{\boxplus\cup\Diamond,\boxminus}\big) \text{ vs. } \mathbb{I}\big([G]_{\boxplus\cup\boxminus,\Diamond}; [L]_{\boxplus\cup\boxminus,\Diamond}\big),$$

where the latter is a pure relevance distinction, but the former two do partially rely on a validity distinction.

This raises the problem that the different equivalence classes on labels are represented in the data with different levels of bias, and hence are not directly comparable. This is why we compute our mutual information scores over a rebalanced contingency table, where, for each gold standard label, the associated counts of system-assigned labels are multiplied by a constant multiplier chosen for each gold standard label in such a way as to yield a uniform distribution on the prior probabilities of gold standard labels, while not changing the relative frequencies of different system-assigned labels.

This can be interpreted as a stratified sampling technique, where the equivalence class of each gold standard label identifies a stratum and where we draw samples of equal sizes from each stratum separately. This means that the union of the subsamples is not a representative sample of the population across different strata. It also means that the prior distribution of system-assigned labels will be altered. The alteration would simulate what the system would do, if it could perfectly realign its decisions with the given alteration in the prior of the gold standard, which is not necessarily what a system would actually do if presented with the altered prior.

If we denote this rebalanced mutual information score as $\mathbb{I}'$, we arrive at the following scores to be observed.

$$\mathbb{I}'\big([\mathrm{G}]_{\boxplus,\Diamond\cup\boxminus};\,[\mathrm{L}]_{\boxplus,\Diamond\cup\boxminus}\big) \text{ vs. } \mathbb{I}'\big([\mathrm{G}]_{\boxplus\cup\Diamond,\boxminus};\,[\mathrm{L}]_{\boxplus\cup\Diamond,\boxminus}\big) \text{ vs. } \mathbb{I}'\big([\mathrm{G}]_{\boxplus\cup\boxminus,\Diamond};\,[\mathrm{L}]_{\boxplus\cup\boxminus,\Diamond}\big),$$

Figure 2.6 shows the summary statistics, and Figure A.5 (appendix A) shows the statistics for the nonparametric sign test and Mann-Whitney test. These numbers lead us to reject the null hypothesis in favour of the alternative hypothesis. □

So, we conclude that systems are comparatively good at the relevance distinction, but bad at the validity distinction. To the extent that these systems are based on model-fitting techniques, this might be attributed to the fact that the relevance distinction is being much more frequently represented in the gold standard data. Another possibility is that the validity decision is inherently harder than the relevance decision.

Another possible explanation is that the particular methods being employed in RTE-4 systems are inherently less suited to the validity distinction than they are to the relevance distinction. We will argue this in section 2.2.2.

Criteria: Logical vs. Intuitive vs. Application-Oriented

In section 2.1.2, we said that the application-oriented criterion which motives the RTE task might not coincide with the intuitive criterion by which the gold standard model was obtained from judges as part of the RTE data collection efforts and that these two criteria, in turn, do not necessarily correspond to any standard logic.

In this section, we will formalize the hypothesis more rigorously, though, unfortunately, we do not have the data to test it. This would require data on three different standard models to be derived from the three different criteria.

H27. Let $\mathrm{G_L}$ be the empirical model which assigns labels according to the logical criterion, let $\mathrm{G_I}$ be the empirical model which assigns labels according to the intuitive criterion, and let $\mathrm{G_A}$ be the f-induced standard model for some application f. We can then use some appropriate comparison score α, to evaluate $\alpha(\mathrm{G_L}, L)$, $\alpha(\mathrm{G_I}, L)$, and $\alpha(\mathrm{G_A}, L)$ for each participating model L.

> *(Null Hypothesis)* Systems exhibit the same behaviour on the three criteria as represented by the three different standard models. When each system is scored against each of the standard models, these three scores do not differ significantly.

When systems are ranked against each other according to their score on each of the three standard models, then these three rankings do not differ significantly.

(Alternative Hypothesis)

1. Scores do differ significantly.
2. Relative rankings do differ significantly.

In the absence of the proper data to test this hypothesis, one can only speculate about the relationship between the three different criteria. For example, the findings by Harabagiu & Hickl (2006) on QA and those by Romano et al. (2006) on IE have often been used to motivate the RTE task. They report on two specific instances of textual entailment components developed within the RTE framework which could be usefully redeployed as part of a QA and IE system. So these might be seen at least as two data points linking the intuitive criterion to the application-oriented criterion. As there is only very little overlap of systems entered both into the RTE evaluations and the Answer Validation Exercises at QA@CLEF, there is no ground for statistically significant comparisons between those two venues either, so ultimately very little is known about the relationship between systems' performance at RTE and their usefulness in applications.

2.2.2. Data Interpretation & Synthesis: The Concentration Plot

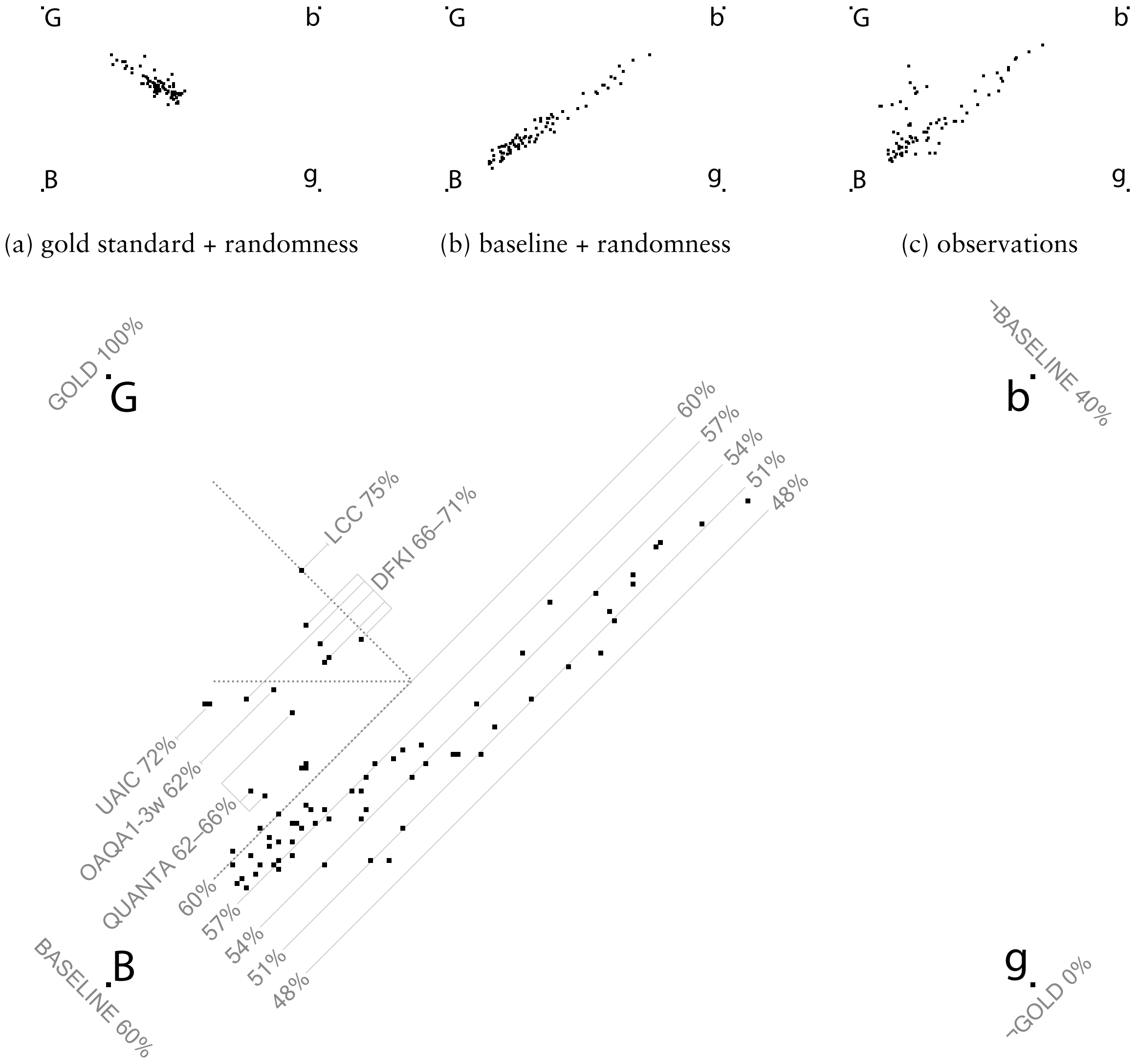
The Concentration Plot

Error Characteristics

Previously, we described the RTE evaluations as one-dimensional in the sense that too much attention is devoted to comparing a given empirical model M to another model N in terms of their performance as measured under some arbitrary set of conditions P; too little attention is devoted to comparing the performance of model M as measured under conditions P to its performance as measured under another set of conditions Q.

It is also one-dimensional in another sense: By using a comparison score to measure each of the empirical models M and N against only one gold standard model G, we get a sense of how M and N individually relate to G, but only a one-dimensional projection of how M relates to N, viz. we get error-counts, but no further qualification on how these errors relate to each other, no information on error characteristics, no error analysis.

But this kind of information is important: Consider as an example a dataset for textual inference containing 1000 candidate inferences. Each single candidate inference will contribute 0.1 percent to an accuracy score. If model M makes 400 errors and model N makes 450 errors, the fit of model M would be perceived as superior to that of model N,



(a) gold standard + randomness

(b) baseline + randomness

(c) observations

(d) observations

Figure 2.7.: scatterplot showing error characteristics at RTE-4

the methodological pitfall being to declare model N, at this point, as a research direction no longer worth pursuing but model M as warranting further work.

Subsequent error analysis may, however, show the following: The 400 errors of model M might be much the same as those made by a trivial baseline model B, say a bag-of-words model for textual inference, the other errors being the result of purely random deviations. The 450 errors of model N, on the other hand, might be predictable, e.g. they might apply only to inferences involving long sentences, and they might be stochastically independent of the errors made by the baseline B. By simply combining model N with a predictor of its own errors and the baseline technique, we can construct a model N′ which now makes only 40% of 450 = 180 errors, while such improvement would not be possible for model M.

Baseline

In this section, we will consider the error characteristics of the different models submitted at RTE-4 w.r.t. such a bag-of-words system as a trivial baseline: My particular implementation simply tokenizes words by whitespace, applies a Porter stemmer, counts the number of tokens in the antecedent and consequent, and finally evaluates the ratio of tokens which occur both in the antecedent and the consequent as a proportion of the number of tokens in the consequent. If this ratio is greater than $\frac{1}{2}$, the system decides ENTAILED, otherwise it decides UNKNOWN. The threshold of $\frac{1}{2}$ was what I arbitrarily determined. Later on, experimentation with the threshold showed that it happened to be optimal, but that variations of the threshold only yield relatively small variations in accuracy for the baseline system.

Construction of the Concentration Plot

The RTE-4 dataset is a collection of 1000 candidate inferences, so the labels assigned by a model G on a sample can be seen as a 1000-dimensional vector G, where each dimension represents a candidate inference x, and where the value of that component would be $[\mathrm{G}(x)]_{\vartriangle,\triangledown}$. So the space of all such labellings would be the 1000-dimensional Hamming space. The scatterplots in Figure 2.7 are a two-dimensional projection of the error characteristics observed at RTE-4 within this 1000-dimensional space.

Our concentration plot is constructed as follows: The point in the upper left corner, labelled G, represents the gold standard model, and the point in the lower right corner, labelled g, represents the negation of G, i.e. the model which makes the opposite decision for each of the candidate inferences. Similarly, point B is the bag-of-words baseline model, with point b representing its negation.

Since the 2-way accuracy of that system B is $\mathrm{A}_{\vartriangle,\triangledown}(\mathrm{G};\mathrm{B}) = .603$, we know that there are 603 components in the vector B for which the decisions assigned by B and G agree and 397 components for which they disagree. The position of any arbitrary 1000-dimensional vector L in our scatterplot is now defined as follows: Its horizontal component, i.e. the length of its projection in the Gb-direction, which is also the Bg-direction, as measured away from B, is the number of decisions among the 603 decisions about which G and B agree, where L disagrees with that decision. Its vertical component, i.e. the length of its projection in the GB-direction, which is also the bg-direction, as measured away from B, is the number of decisions among the 397 decisions for which G and B disagree, where L agrees with G, thus disagreeing with B.

Interpreting the Concentration Plot

This means that the Manhattan distance[6] of point L from g represents the accuracy of system L, and similarly the Manhattan distance of point L from b represents the percentage agreement between L and the baseline.

A few relationships can now be readily illustrated as geometric relationships within this space. To make the numbers simpler, suppose it measures 600 units by 400 units.

Consider a model which improves over the baseline B towards the gold standard G in such a way that it corrects 250 errors, but does not introduce any new errors. Such a model would have to fall on the line segment GB, a distance of 250 away from B towards G, so that the point is now only 150 units away from G.

One could also imagine a model which introduces 250 deviations into B at random. As $\frac{600}{400} = 1.5$, there are, in total, 1.5 instances where B and G agree for every instance where they disagree. So, if introducing deviations at random, we would also expect 1.5 of those deviations to affect an instance about which B and G agree for every deviation which affects an instance about which they disagree. So any system which introduces random deviations into B would fall on the line segment Bb. If introducing 250 such deviations, we simply need to note that $\frac{600}{400} = \frac{150}{100} = 1.5$ to find that, in terms of the statistical expected value, we would expect the point representing such a system at a position 150 units away from B and G in the Bg-direction (the horizontal), and 100 units away from B towards G in the GB-direction (the vertical).

Finally, consider a model which introduces 250 errors into G at random. We would expect the point to fall on segment Gg, 150 units away from B and G in the Bg-direction (the horizontal), and 100 units away from G towards B in the GB-direction (the vertical).

Statistical Patterns in the Concentration Plot

Figures 2.7a and 2.7b have been generated with a random number generator, in order to visually establish some initial intuitions about the interpretation of statistical data plotted in this diagram.

In Figure 2.7a, each point is generated as follows: Choose an error count c out of the set of RTE-4 submissions, i.e. a model's accuracy score. Then, instead of plotting the actual error characteristic of that model, use a random number generator to generate a random error characteristic by introducing c errors into the gold standard G at random. We can see that the resulting datapoints are centered around the line segment Gg, with the actual points falling within relatively short distances of that line segment. So this

[6] Recall that the Manhattan distance, or city-block distance, of a point (x_1, y_1) from another point (x_2, y_2) is $\text{abs}(x_1 - x_2) + \text{abs}(y_1 - y_2)$. So, in Figure 2.7d, the grey lines labelled with percentages are lines of equal Manhattan distance from G and g.

plot maintains the accuracies reported in the RTE-4 overview paper, but randomly regenerates the information which is in the data, in a way which is invariant to those reported accuracies.

In Figure 2.7b, on the other hand, each point is generated by choosing a deviation count d out of the set of RTE-4 submissions, where this deviation count represents the number of instances which deviate in the model decision from the baseline decision. Then, instead of plotting the actual deviation characteristics, we generate a random deviation characteristic by introducing d random deviations into B, and plot the resulting model. The datapoints are now centered around Bb.

Let me stress: These two plots do not visualize RTE-4 data as such. Rather, they provide a visual reference for the statistical properties of our plotting method.

Observed Concentration: Bag-of-Words Equivalence

This plotting method now enables us to ask a very basic question about the RTE systems and their error characteristics: Are the incorrectly labelled instances random deviations from the gold standard, or are the correctly labelled instances random deviations from the bag-of-words baseline? Figure 2.7c shows that the latter seems to be the case, possibly with the exception of a few outliers we will discuss shortly. Here, we have plotted the models we can actually observe in the RTE-4 data directly, observing a visual pattern which is much more similar to that of Figure 2.7b than it is to that of Figure 2.7a.

Figure 2.7d shows the same scatterplot larger with some additional labelling. The outliers are the LCC system (Hickl 2008, Bensley & Hickl 2008) and the DFKI system (Wang & Neumann 2008, Wang & Zhang 2009) which are, in fact, positioned closer to the gold standard than they are to the baseline. – Other outliers are the UAIC system (Iftene 2008, 2009), the OAQA system (Siblini & Kosseim 2008, 2009), and possibly the QUANTA system (Li et al. 2008), but these are still closer to the baseline than they are to the gold standard.

All participants other than our handful of exceptions submitted systems which exhibited a behaviour which is indistinguishable to our statistical test from a process behaving like a bag-of-words baseline but introducing random deviations into its output.

This also helps explain our previous observation that systems are better at deciding relevance than they are at deciding validity, as one would not expect to be able to draw validity distinctions on the basis of a bag of words. Relevance, however, may very well be decidable in such a way, as both deep and shallow systems suffer from the same lack of background knowledge: If the only reliable source of ontological knowledge is simply the assumption that two predicates refer to equivalent ontological categories when they relate to words which have the same spelling, then bag-of-words overlap is the only reliable indicator of relevance both for deep and shallow systems.

Note also, that the accuracies, which are easily observed in the diagram along the grey lines marked by percentages, do not measure linear progress from the baseline towards the gold standard. Even the UAIC system, which lies closest to the line segment BG and thus comes closest to this ideal of linear progress, introduces 34 errors for every 100 errors it corrects relative to the baseline. The LCC system, which comes closest to the gold standard G itself, introduces 46 errors for every 100 errors it corrects.

Phenomenology & Analytical Testsets: RTE vs. FraCaS

There is another interesting relationship that can be seen in the concentration plot in Figure 2.7d: Consider the grey non-dotted lines which are drawn at an angle of 45 degrees to the Bg-direction. Since all points falling on one of those lines have the same Manhattan distance from G, they would all map into the same accuracy. Similarly, we can think of lines of equal Manhattan distance from B, which would have to be at an angle of -45 degrees to the Bg-direction.

One can now imagine Figure 2.7d as rotated by -45 degrees, so as to view the grey lines as horizontal. Given this rotation, the vertical position of a model L will be its accuracy $\mathrm{A}(\mathrm{G}; \mathrm{L})$, with the horizontal position being its error rate when compared against the baseline as a reference model, i.e. $1 - \mathrm{A}(\mathrm{B}; \mathrm{L})$. If we leave aside the outliers mentioned earlier and concentrate only on the systems below 65% accuracy, we can then clearly see a correlation in the dataset: It is generally the points closest to B which are at the same time closest to G.

In more statistical terms, we can compute the Pearson correlation coefficient between $\mathrm{A}(\mathrm{G}; L)$ and $1 - \mathrm{A}(\mathrm{B}; L)$ over the different models L. This value, at RTE-4, is $\rho = .386$ including outliers, or $\rho = .682$ after removing systems with accuracy above 65%. These numbers are significantly different from zero, using a two-tailed test, and clearly positive.

In the rest of this section, we will argue that correlations like this pose a challenge to a particular phenomenology which is often tacitly assumed when applying statistical doctrine naïvely, viz. the assumption of stochastic independence for datapoints in a dataset like that used at RTE, or empirical models such as the systems submitted for RTE.

For example, a fact which the RTE organizers usually state in the abstract to the overview paper is the number of participants. By interpreting this number under the incorrect assumption of stochastic independence, one would grossly overestimate the size of the hypothesis space explored. For example the 25 participants at RTE-4 did not really produce 25 stochastically independent empirical models, rather there seem to be only a handful of clusters of models, each cluster exhibiting high mutual correlations between models. In Figure 2.7d, for example, we have identified bag-of-words equivalence as one very successful separation criterion. Our main criticism here is that there are easier ways of making this separation than the RTE evaluation methodology.

More generally, one must emphasize that different empirical models are only useful to the extent that they help us distinguish those datapoints where models succeed from those datapoints where models fail. – But the converse is also true. Different datapoints are only useful to the extent that they help us distinguish those empirical models which succeed on those datapoints from those models which fail on them, which brings us back to our running argument on the one-dimensionality of evaluation schemes like RTE.

To state this more formally: Let $\mathrm{H}_1, \mathrm{H}_2 \in \mathcal{H}$ be two models, let $\mathrm{X} \subseteq \mathcal{X}$ be a data sample, and let $\mathrm{G} \in \mathcal{H}_\mathrm{X}$ be the gold standard model. Furthermore, choose some comparison score α and a $\theta \in \mathbb{R}$. Now, we can define that $\mathrm{H}_1 \approx_\mathrm{X} \mathrm{H}_2$ iff both $\alpha^\mathrm{X}(\mathrm{H}_1; \mathrm{G}) \geq \theta$ and $\alpha^\mathrm{X}(\mathrm{H}_2; \mathrm{G}) \geq \theta$. – This way, we use the dataset X to partition the set of models into two subsets: The set of good models and the set of bad models.

Conversely, we can let $\mathrm{X}_1, \mathrm{X}_2 \in \mathcal{X}$ be two data samples, and $\mathrm{H} \in \mathcal{H}$ be some empirical model. Now we can define that $\mathrm{X}_1 \approx_\mathrm{H} \mathrm{X}_2$ iff both $\alpha^{\mathrm{X}_1}(\mathrm{H}; \mathrm{G}) \geq \theta$ and $\alpha^{\mathrm{X}_2}(\mathrm{H}; \mathrm{G}) \geq \theta$. So, we use the model H to partition the set of datasets into two subsets: The set of datasets for which the model is good, and the set of datasets for which the model is bad.

Now, denote the RTE-4 dataset as X_{1000}, and denote the following dataset, consisting of a single candidate inference as X_1:

$$\frac{\textsf{The cat chased the dog.}}{\textsf{The dog chased the cat.}}$$

We now arrive at a conclusion which seems unexpected from the point of view of statistical doctrine: The previously defined good/bad distinctions can be made almost as effectively, given the models which were to be evaluated at RTE-4, using the one datapoint in X_1 as they can be, using the 1000 datapoints in X_{1000}. The intuition behind this fact is that this particular good/bad distinction is dominated by the pattern of bag-of-words equivalence, and that this distinction can be as effectively made by looking at a single carefully chosen example as it can be by studying a relatively complex pattern within a a dataset of 1000 examples statistically.

More formally: At RTE-4, we would expect that, for the majority of participant models $H \in \mathcal{H}$, we would have $\mathrm{X}_{1000} \approx_\mathrm{H} \mathrm{X}_1$, and that, conversely, for every pair of models $H_1, H_2 \in \mathcal{H}$, we would have $\mathrm{H}_1 \approx_{\mathrm{X}_{1000}} \mathrm{H}_2$ iff $\mathrm{H}_1 \approx_{\mathrm{X}_1} \mathrm{H}_2$.

This is the methodological rationale behind the use of analytical testsuites like FraCaS (Cooper et al. 1996). If one subscribes naïvely to statistical doctrine, one may be mislead prematurely into discarding such analytical testsuites on the grounds of the dataset not being large enough, or the whole methodology somehow not being empirical. However, I believe that the extreme case of X_1 vs. X_{1000} for RTE-4 should make it clear that analytical testing is not simply the theory-driven antithesis to empiricism. In this particular case, it is rather clear that an analytical testsuite which has been well-designed with empirical considerations in mind is, on the whole, no less justifiable from an empirical viewpoint than the sort of data collection practice used at RTE. And while analytical testsuites are no less empirical in nature than traditional datasets, results become much more straightforward to interpret.[7]

This, incidentally, is what linguists traditionally do when observing the effect of a theory on a set of carefully chosen example sentences, or what logicists do when testing whether a given logic proves all and only those of a small number of carefully selected theorems which are intended to be provable in the logic.

[7] This is why, in appendix D, we provide an exhaustive list of FraCaS test instances, together with comments on how our inference mechanism deals with each. This provides some level of empirical grounding for our inference mechanism, although this empirical line of work is comparatively immaterial for the main thrust of the argument presented in this thesis.

3. Łukasiewicz Logic & Syllogistic Semantics

In the previous chapter, we reviewed the state of the art in empirically-driven textual inference, pointing out its main limitations, and concluding that an in-depth theoretical investigation is needed.

In this chapter, we establish a model theory to give an interpretation to expressions of a logical language. In the next chapter (chapter 4), we will show how to translate natural language into this logical language and, in the chapter after that (chapter 5), how we can computationally implement an appropriate inference procedure for this logic.

The defining feature which makes our logic particularly useful for the purposes of this thesis is the fact that it is a many-valued logic.

28. For any $M \in \mathbb{N}$ with $M \geq 2$, we define *truth value sets* as

$$\mathbb{V}_M \overset{\text{def}}{=} \left\{ \frac{0}{M-1}, \frac{1}{M-1}, \frac{2}{M-1}, \dots, \frac{M-1}{M-1} \right\};$$

$$\mathbb{V}_{\aleph_0} \overset{\text{def}}{=} \{v \mid v \in \mathbb{Q} \ \wedge \ 0 \leq v \leq 1\}.$$

If $\mathbb{V}$ is a truth value set, we call any $v \in \mathbb{V}$ a truth value.

Where, traditionally, truth values are drawn from the set $\mathbb{V}_2$, we will, in this chapter, establish how we can generalise this towards drawing truth values from $\mathbb{V}_M$ for $M > 2$, and we will be particularly interested in the limit case of $\mathbb{V}_{\aleph_0}$.

3.1. Many-Valued Propositional Logic

Due to the widespread misconceptions about fuzzy logic which we briefly outlined in the introduction (chapter 1), it has become very difficult to delimit concrete proposals such as ours against the often hazy and indistinct notions which one sometimes finds in connection with soft computing. Our approach will therefore be to build up our theory entirely from first principles. In doing so, we will, of course, make heavy use of prior work in the field, which is why this section, in particular, amounts simply to a summary of relevant aspects of the work of Łukasiewicz & Tarski (1930), Rose & Rosser (1958), Chang (1959) on Łukasiewicz logic. A more general and modern treatment of many-valued logic can be found, for example, in the monographs by Hájek (1998), Gottwald (2001), Metcalfe et al. (2008).

3.1.1. Language

In what follows, we define the expressions of the formal language to which we ultimately want to assign truth values.

29. We call Λ a *propositional signature*, iff Λ is a set of propositional symbols $\Lambda = \{\mathsf{p}_1, \mathsf{p}_2, \dots, \mathsf{p}_{|\Lambda|}\}$ of finite cardinality $|\Lambda|$.

30. Let $\mathbb{V}$ be a truth value set and let Λ be a propositional signature. The following recursive rules define by structured induction the notion of a *basic propositional formula over* $\mathbb{V}$ *and* Λ. For all v, p, φ, and ψ:

- if $v \in \mathbb{V}$, the *value constant* '$\overline{v}$' is a *formula*;
- if $p \in \Lambda$, the *proposition* 'p' is a *formula*;
- if φ and ψ are formulae, then the *implication* '$(\varphi \rightarrow \psi)$' is a *formula* as well;
- nothing else is a *formula*.

So, for example, if we take $\Lambda = \{\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3\}$, $\mathbb{V} = \mathbb{V}_3$, then '$\mathsf{p}_1$' will be a formula, but 'q' or 'p_4' will not be formulae. Similarly, '$\overline{0}$', '$\overline{.5}$' and '$\overline{1}$' will be formulae, but '$\overline{.7}$' will not be a formula. The latter will however be a formula for $\mathbb{V} = \mathbb{V}_{\aleph_0}$. Given these atomic formulae, we can use implication operators to construct formulae like '$\mathsf{p}_1 \rightarrow (\mathsf{p}_2 \rightarrow \mathsf{p}_3)$'. Obviously '$\mathsf{p}_1 \rightarrow\rightarrow \mathsf{p}_2$' or '$\mathsf{p}_1 \rightarrow \mathsf{p}_2\mathsf{p}_3$' are not formulae. For a formula to be a *basic* formula, implication is the only operator allowed.

A few more comments on notation: We enclose an expression in quotation marks, writing '$\cdots$', when there is a need to emphasize that it is taken as an expression of an object language under discussion, rather than an expression of the meta language used to describe the object language. To enhance readability, we are free to drop the quotation marks when no such confusion can arise.

Also, we will drop parentheses when no confusion can arise about the syntactic structure of the formula. However, we will not rely on any conventions regarding operator precedence. We treat this strictly as notational convenience. For purposes of mathematical induction on the syntactic structure of a formula, we will always treat it w.l.o.g. as if it were parenthesized in a strict fashion.

We can now go on to define more operators besides implication.

31. Let $\mathbb{V}$ be a truth value set and let Λ be a propositional signature. The following recursive rules define by structured induction the notion of an *extended propositional formula over* $\mathbb{V}$ *and* Λ. For all φ and ψ:

- if φ is a basic formula over $\mathbb{V}$ and Λ, then φ is an extended *formula* as well;
- if φ is a formula, then the *negation* "$\neg\varphi$" is a *formula* as well;
- if φ and ψ are formulae, then
  - the *strong conjunction* '$(\varphi \,\&\, \psi)$',
  - the *strong disjunction* '$(\varphi \veebar \psi)$',
  - the *weak conjunction* '$(\varphi \wedge \psi)$',
  - the *weak disjunction* '$(\varphi \vee \psi)$',
  - the *equivalence* '$(\varphi \equiv \psi)$', and
  - the *antivalence* '$(\varphi \not\equiv \psi)$'

  are *formulae* as well;
- nothing else is a *formula*.

32. For any extended formula χ over some $\mathbb{V}$ and Λ, we assign a *corresponding* basic formula $\mathcal{B}(\chi)$ over $\mathbb{V}$ and Λ. We define $\mathcal{B}(\cdot)$ recursively as follows. For any extended formulae φ and ψ:

$$\begin{aligned}
\mathcal{B}(\neg\varphi) &= \mathcal{B}(\varphi) \to \overline{0}, \\
\mathcal{B}(\varphi \to \psi) &= \mathcal{B}(\varphi) \to \mathcal{B}(\psi), \\
\mathcal{B}(\varphi \,\&\, \psi) &= \mathcal{B}\big(\neg(\varphi \to \neg\psi)\big), \\
\mathcal{B}(\varphi \veebar \psi) &= \mathcal{B}\big(\neg\varphi \to \psi\big), \\
\mathcal{B}(\varphi \wedge \psi) &= \mathcal{B}\big(\varphi \,\&\, (\varphi \to \psi)\big), \\
\mathcal{B}(\varphi \vee \psi) &= \mathcal{B}\big((\varphi \to \psi) \to \psi\big), \\
\mathcal{B}(\varphi \equiv \psi) &= \mathcal{B}\big((\varphi \to \psi) \,\&\, (\psi \to \varphi)\big) \\
\mathcal{B}(\varphi \not\equiv \psi) &= \mathcal{B}\big(\neg(\varphi \equiv \psi)\big).
\end{aligned}$$

33. Let $\mathbb{V}$ be a truth value set and let Λ be a propositional signature. We denote the set of all basic propositional formulae over $\mathbb{V}$ and Λ as $\mathcal{L}_{\mathbb{V},\Lambda}$.

This propositional language follows a traditional setup. The only noteworthy aspect is the use of two different conjunction operators, strong conjunction ('$\&$') and weak conjunction ('$\wedge$'), as well as two different disjunction operators, strong disjunction ('$\veebar$') and weak disjunction ('$\vee$') in definitions 31, 32.

One can verify some initial intuitions about definition 32 by checking that the given reductions are identities in Boolean algebra. In this case, weak and strong conjunction turn out to be identical. However, as we will see later on in corollary 38, this is not generally the case in a Łukasiewicz logic with more than two truth values.

We allude to the reduction of definition 32 whenever we talk simply about a *formula* without specifying whether it is a formula of the basic or the extended syntax. We always take such a formula to be of the extended syntax in the first instance. But for subsequent statements, we can then assume w.l.o.g. that it is in basic syntax, i.e. we then talk about the basic propositional formula which corresponds to the original extended propositional formula.

Our terminology, our notation for formulae, and our reduction of the extended fragment to the basic fragment have been chosen in accordance with Hájek (1998). The operators are used throughout the relevant literature, for example also by Rose & Rosser (1958). Different notations are in use for formulae of the kind we are considering, and it is quite common for different fragments of the logic to be taken as basic, together with different reductions of the extended fragment to the basic fragment. However, after some initial development of the logic, it is usually easy to see that the different ways of theoretically framing Łukasiewicz logic are equivalent.

3.1.2. Semantics & Model Theory

Having constructed formulae in our propositional logic, we can now go about assigning truth values to them. In order to do this, we first need to assign truth values to atomic formulae. We do this by means of a *valuation*. Truth values of composite formulae are then determined through their operators.

34. Let $\mathbb{V}$ be a truth value set and let Λ be a propositional signature. We call w a $(\mathbb{V}, \Lambda)$-*valuation* iff $\mathrm{w} : \Lambda \mapsto \mathbb{V}$ is a function from Λ to $\mathbb{V}$.

In traditional bivalent logic, we use truth tables to define operators model theoretically. For example implication might be defined as follows:

35. Let Λ be a propositional signature and w be a $(\mathbb{V}_2, \Lambda)$-valuation. Now, for any formula χ over $\mathbb{V}_2$ and Λ, the *truth value of* χ *in* w, denoted $\|\chi\|_{\mathrm{w}}$, is defined by structured induction as follows. For any φ and ψ:

$$\|\,\overline{0}\,\|_{\mathrm{w}} \overset{\text{def}}{=} 0;\;\; \|\,\overline{1}\,\|_{\mathrm{w}} \overset{\text{def}}{=} 1;$$

$$\|\,p\,\|_{\mathrm{w}} \overset{\text{def}}{=} \mathrm{w}(p),\;\text{for each } p \in \Lambda;$$

$$\|\varphi \to \psi\|_{\mathrm{w}} \overset{\text{def}}{=} \begin{cases} 1 & \text{if } \|\varphi\|_{\mathrm{w}} = 1 \text{ and } \|\psi\|_{\mathrm{w}} = 1, \\ 0 & \text{if } \|\varphi\|_{\mathrm{w}} = 1 \text{ and } \|\psi\|_{\mathrm{w}} = 0, \\ 1 & \text{if } \|\varphi\|_{\mathrm{w}} = 0 \text{ and } \|\psi\|_{\mathrm{w}} = 1, \\ 1 & \text{if } \|\varphi\|_{\mathrm{w}} = 0 \text{ and } \|\psi\|_{\mathrm{w}} = 0. \end{cases}$$

We can generalize this to the many-valued case as follows:

36. Let $\mathbb{V}$ be a truth value set, Λ be a propositional signature, and w be a $(\mathbb{V},\Lambda)$-valuation. Now, for any formula χ, over $\mathbb{V}$ and Λ, the *truth value of χ in* w, denoted $\|\chi\|_{\mathrm{w}}$, is defined by structured induction as follows. For any φ, ψ:

$$\|\,\overline{v}\,\|_{\mathrm{w}} \overset{\text{def}}{=} v,\;\text{for any } v \in \mathbb{V};$$

$$\|\,p\,\|_{\mathrm{w}} \overset{\text{def}}{=} \mathrm{w}(p),\;\text{for each } p \in \Lambda;$$

$$\|\varphi \to \psi\|_{\mathrm{w}} \overset{\text{def}}{=} \min(1, 1 - \|\varphi\|_{\mathrm{w}} + \|\psi\|_{\mathrm{w}}).$$

$*$**37.** *Classical semantic (def. 35) is a special case of Łukasiewicz semantic (def. 36).*

Proof. Simply substitute the truth values from the case distinction in definition 35 into the expression in definition 36 and validate that the assignments agree. □

From here on, by $\|\cdot\|$, we will always denote truth values as per Łukasiewicz semantic (definition 36). From the above corollary, we know that classical semantic (definition 35) can be treated as a special case.

$*$**38.** *Let $\mathbb{V}$ be a truth value set and let Λ be a propositional signature. Now for any $(\mathbb{V},\Lambda)$-valuation w, we have*

$$\begin{aligned} \|\varphi \,\&\, \psi\|_w &= \max(0, \|\varphi\|_w + \|\psi\|_w - 1), & \|\neg\varphi\|_w &= 1 - \|\varphi\|_w, \\ \|\varphi \veebar \psi\|_w &= \min(1, \|\varphi\|_w + \|\psi\|_w), & \|\varphi \to \psi\|_w &= \min(1, 1 - \|\varphi\|_w + \|\psi\|_w), \\ \|\varphi \wedge \psi\|_w &= \min(\|\varphi\|_w, \|\psi\|_w), & \|\varphi \not\equiv \psi\|_w &= \mathrm{abs}(\|\varphi\|_w - \|\psi\|_w), \\ \|\varphi \vee \psi\|_w &= \max(\|\varphi\|_w, \|\psi\|_w), & \|\varphi \equiv \psi\|_w &= 1 - \mathrm{abs}(\|\varphi\|_w - \|\psi\|_w), \end{aligned}$$

where abs *stands for absolute value.*

Proof. Use the reduction from definition 32 to substitute the left-hand formula for its corresponding basic formula, use definition 36 to obtain its truth value, and simplify. □

Now that we can assign to a formula its truth value as a function of the truth values of its atomic propositions, we can define validity. Again, we start with the traditional bivalent special case and then generalise to the many-valued case.

39. Let $\mathbb{V}$ be a truth value set and let Λ be a propositional signature. For any formula φ over $\mathbb{V}$ and Λ, we denote by $Ł(\mathbb{V},\Lambda) \vDash \varphi$ that φ is *valid in* $\mathbb{V}$ *and* Λ, which we define as:

$$Ł(\mathbb{V},\Lambda) \vDash \varphi \text{ iff } \|\varphi\|_w = 1 \text{ for any } (\mathbb{V},\Lambda)\text{-valuation.}$$

40. Let $\mathbb{V}$ be a truth value set, Λ be a propositional signature, and $\mathrm{t} \in \mathbb{V}$ be a *validity threshold*. For any formula φ over $\mathbb{V}$ and Λ, we denote by $Ł(\mathbb{V},\Lambda) \vDash_{\mathrm{t}} \varphi$ that φ is t*-valid in* $\mathbb{V}$ *and* Λ, which we define as:

$$Ł(\mathbb{V},\Lambda) \vDash_{\mathrm{t}} \varphi \text{ iff } \|\varphi\|_w \geq \mathrm{t} \text{ for any } (\mathbb{V},\Lambda)\text{-valuation.}$$

⁎41. *Classical validity (definition 39) is a special case of graded validity (definition 40). In particular, classical validity is* 1*-validity.*

Proof. This follows trivially from definitions 39 and 40. □

The fragment of the model theory considered here, consisting of implication and negation, as well as the attached notion of validity were first introduced in a publication by Łukasiewicz & Tarski (1930), but are attributed to Jan Łukasiewicz alone. The full set of operators considered here were used by Rose & Rosser (1958) and now appear throughout the relevant literature. The notion of graded validity we consider is chosen in accordance with the theoretic framework of Pavelka (1979). Our characterization of Pavelka logic, in turn, closely follows that of Hájek (1998).

3.1.3. Syntax & Proof Theory

In the previous section, we defined a model theory for Łukasiewicz logic, establishing traditional propositional logic as that special case of Łukasiewicz logic which arises when we restrict attention to model theoretic valuations which assign only the truth values 0 and 1. We will now switch to a proof theoretic viewpoint, showing how we can develop Łukasiewicz logic axiomatically. This will show that all theses provable in $\aleph_0$-valued Łukasiewicz logic are also provable in traditional propositional logic. Note however that it is not the case that, conversely, all theses provable in traditional logic are provable in Łukasiewicz logic.

We start this definition by establishing an axiom system, i.e. the set of formulae which we accept without proof:

42. Consider the following axiom schemata:

$$\varphi \to (\psi \to \varphi); \tag{$\star$Ł1}$$

$$(\varphi \to \psi) \to \big((\psi \to \chi) \to (\varphi \to \chi)\big); \tag{$\star$Ł2}$$

$$(\neg\varphi \to \neg\psi) \to (\psi \to \varphi); \tag{$\star$Ł3}$$

$$(\psi \vee \varphi) \to (\varphi \vee \psi); \tag{$\star$Ł4}$$

$$\overline{\mathrm{p}} \to \overline{\mathrm{q}} \equiv \overline{\mathrm{r}} \;\text{ for }\; \overrightarrow{\mathrm{T}}_{\text{Ł}}(\mathrm{p},\mathrm{q}) = \mathrm{r}; \tag{$\star$Ł5}$$

$$\bigvee_{r \in \mathbb{V}_{\mathrm{M}}} (\varphi \equiv \overline{r}). \tag{$\star$Ł6}$$

Let Λ be a propositional signature. In what follows, we define the conditions under which a formula ω over Λ and some $\mathbb{V}$ is considered an *instance* of one of these schemata.

- A formula ω is an *instance* of one of axiom schemata ($\star$Ł1), ($\star$Ł2), ($\star$Ł3), or ($\star$Ł4), iff it results from substituting particular formulae φ, ψ, and χ for the form variables φ, ψ and χ in the schema.
- A formula $\overline{r} \equiv \overline{p} \to \overline{q}$ is an *instance* of axiom schema ($\star$Ł5), iff $p, q, r \in \mathbb{V}$, so that '$\overline{p}$', '$\overline{q}$', and '$\overline{r}$' are formulae over $\mathbb{V}$, and $r = \min(1, 1 - p + q)$.
- A formula ω over $\mathbb{V}_M$ for some M is an *instance* of axiom schema ($\star$Ł6), iff it results from substituting a particular formula φ over $\mathbb{V}_M$ for the form variable φ in the schema, where the M in the axiom schema equals M. So, for example for $\mathbb{V}_3$, any formula of the form $\varphi \equiv \overline{0} \vee \varphi \equiv \overline{0.5} \vee \varphi \equiv \overline{1}$ is an instance of ($\star$Ł6).

We can then define axiom systems for Łukasiewicz logic as follows.

- The *axiom system* $\text{Ł}_{\mathbb{V}_{\aleph_0},\Lambda}$ is the set of all formulae ω over $\mathbb{V}_{\aleph_0}$ and Λ that are instances of any of the above axiom schemata except ($\star$Ł6).
- An *axiom system* $\text{Ł}_{\mathbb{V}_M,\Lambda}$, for any M, is the set of all formulae ω over $\mathbb{V}_M$ and Λ that are instances of any of the above axiom schemata, including ($\star$Ł6).

From this axiom system we can develop the entire logic simply by modus ponens. Again, we will first establish the traditional case, and then generalize to the graded case.

43. Let $\mathbb{V}$ be a truth value set and let Λ be a propositional signature. For any formula φ over $\mathbb{V}$ and Λ, we denote by $\text{Ł}(\mathbb{V},\Lambda) \vdash \varphi$ that φ is *provable in* $\mathbb{V}$ *and* Λ, which we define:

- if $\varphi \in \text{Ł}_{\mathbb{V},\Lambda}$, then $\text{Ł}(\mathbb{V},\Lambda) \vdash$ "φ";
- if $\text{Ł}(\mathbb{V},\Lambda) \vdash$ "φ" and $\text{Ł}(\mathbb{V},\Lambda) \vdash$ "$\varphi \to \psi$", then $\text{Ł}(\mathbb{V},\Lambda) \vdash$ "ψ";
- nothing else is *provable* in $\mathbb{V}$ and Λ.

44. Let $\mathbb{V}$ be a truth value set, Λ be a propositional signature, and $\mathrm{t} \in \mathbb{V}$ be a *standard of proof*. For any formula φ over $\mathbb{V}$ and Λ, we denote by $\text{Ł}(\mathbb{V},\Lambda) \vdash_{\mathrm{t}} \varphi$, that φ is t-*provable* *in* $\mathbb{V}$ *and* Λ, which we define as follows:

$$Ł(\mathbb{V},\Lambda) \vdash_{\mathrm{t}} \text{“}\varphi\text{” iff } Ł(\mathbb{V},\Lambda) \vdash \text{“}\bar{\mathrm{t}} \rightarrow \varphi\text{”}.$$

In section 3.1.5, we will outline the completeness result which establishes that the relation '$\vDash$', which we defined model theoretically is the same as the relation '$\vdash$', which we defined proof theoretically.

Some more comments on the origins of this proof theory: Axiom schemata ($\star$Ł1), ($\star$Ł2), ($\star$Ł3), and ($\star$Ł4) were given by Łukasiewicz himself (Łukasiewicz & Tarski 1930). He conjectured that these four axioms, together with a fifth axiom, would form an axiomatization for the semantic system he was considering, but did not present a completeness proof to that extent. In 1935, M. Wajsberg claimed to have proven this completeness result, but such a proof never appeared in print (Borkowski 1970). A completeness proof was later published by Rose & Rosser (1958). At the same time, it was also found that the fifth axiom considered by Łukasiewicz was in fact dependent (Meredith 1958, Chang 1958*b*), so that it does not need to be accepted axiomatically but rather can be deduced from the other four.

Our axiom schemata ($\star$Ł5) and ($\star$Ł6) serve bookkeping purposes. Schema ($\star$Ł5) establishes the construction of Pavelka logic due to Hájek (1998). Schema ($\star$Ł6) embeds the M-valued logic into the $\aleph_0$-valued logic. This construction is due to Rose & Rosser (1958). Finally, our notion of graded provability is set up in accordance with Pavelka (1979), following Hájek (1998).

3.1.4. Algebra

Besides the model theoretic and the proof theoretic account, one can also use algebra to work with this logic. In the 2-valued case, this process of algebraization would yield a Boolean algebra. In this section, we will consider the more general case of $\aleph_0$-valued Łukasiewicz logic, which, by the process of algebraization yields what we call Łukasiewicz algebra. The algebraic analysis of Łukasiewicz logic was pioneered and developed in great depth by Chang. The part of this algebra which focuses on conjunction and disjunction is referred to in the literature as MV algebra (Chang 1958*a*, 1959) and the part which focuses on implication is that of Wajsberg algebra (Font et al. 1984).

We will show how these algebraic systems interact to give equivalent characterizations of the algebra of Łukasiewicz logic. For the convenience of the reader, we have reproduced some useful algebraic identities and proven them directly from our definitions in appendix B. These proofs, however, proceed along the lines of similar proof theoretic proofs as they can be found in many places in the literature, including (Chang 1958*a*,

1959) and recent monographs such as that by Hájek (1998), that by Gottwald (2001) or that by Metcalfe et al. (2008). Furthermore, the equivalence of the algebraic systems which we show here also follows trivially from the obvious similarity between the axiomatic identities required for Wajsberg algebras and the axioms of the proof theory for Łukasiewicz logic, given the completeness result by Chang (1959).

It is this algebraic account which we will use in order to prove our main result in section 3.3. Note that all algebraic identities which are identities in Łukasiewicz algebra also hold in Boolean algebra, but the converse is not true. So, we first need to develop Łukasiewicz algebra to some extent, before we can use it to prove our main result. We start by defining what an algebra is, and work our way towards more concrete results.

45. We call $\mathbf{A} = (\mathrm{A}, \mathrm{op}_1, \mathrm{op}_2, \ldots, \mathrm{op}_\mathrm{n}, \overline{\mathrm{c}_1}, \overline{\mathrm{c}_2}, \ldots, \overline{\mathrm{c}_\mathrm{m}})$ an *algebra* iff it consists of

- the *carrier* A, a nonempty set of elements;
- a set of *operators* $\mathrm{op}_i : \mathrm{A}^{\mathrm{a}_i} \mapsto \mathrm{A}$ which are functions from A to A; and
- a set of *constants* $\overline{\mathrm{c}_j} \in \mathrm{A}$.

We can now define some reductions of operators to a basic set. Note how this parallels definition 32 which we have made for the language itself in section 3.1.1.

46. Let $\mathbf{A} = (\mathrm{A}, \rightarrow, \overline{0})$ be an algebra. We call $\mathbf{A}'$ the *Wajsberg-induced algebra* of $\mathbf{A}$, iff $\mathbf{A}' = (\mathrm{A}, \rightarrow, \neg, \&, \veebar, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ is an algebra and, for all $x, y, z \in \mathrm{A}$:

$$\overline{1} = \overline{0} \rightarrow \overline{0}, \qquad (*\mathrm{W}5) \qquad x \wedge y = x \,\&\, (x \rightarrow y), \qquad (*\mathrm{W}9)$$

$$\neg x = x \rightarrow \overline{0}, \qquad (*\mathrm{W}6) \qquad x \vee y = (x \rightarrow y) \rightarrow y, \qquad (*\mathrm{W}10)$$

$$x \,\&\, y = \neg(x \rightarrow \neg y), \qquad (*\mathrm{W}7) \qquad x \equiv y = (x \rightarrow y) \,\&\, (y \rightarrow x), \qquad (*\mathrm{W}11)$$

$$x \veebar y = \neg x \rightarrow y, \qquad (*\mathrm{W}8) \qquad x \not\equiv y = \neg(x \equiv y). \qquad (*\mathrm{W}12)$$

Now we define some identities over the basic fragment. Note how these identities parallel the axioms of definition 3.1.3.

47. Let $\mathbf{A} = (\mathrm{A}, \rightarrow, \overline{0})$ be an algebra, and let $\mathbf{A}' = (\mathrm{A}, \rightarrow, \neg, \&, \veebar, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ be the Wajsberg-induced algebra of $\mathbf{A}$. We call $\mathbf{A}$ a *Wajsberg algebra*, iff for all $x, y, z \in \mathrm{A}$:

$$\overline{1} \rightarrow y = y, \qquad (\star\mathrm{W}1)$$

$$(x \rightarrow y) \rightarrow \big((y \rightarrow z) \rightarrow (x \rightarrow z)\big) = \overline{1}, \qquad (\star\mathrm{W}2)$$

$$(\neg x \rightarrow \neg y) \ \rightarrow \ (y \rightarrow x) = \overline{1}, \qquad (\star\mathrm{W}3)$$

$$(x \vee y) = (y \vee x). \qquad (\star\mathrm{W}4)$$

We can arrive at what we will show to be the same algebra by taking conjunctions or disjunctions as basic, rather than taking implication as basic.

48. Let $\mathbf{A} = (\mathrm{A}, \underline{\vee}, \neg, \overline{0})$ be an algebra. We call $\mathbf{A}'$ the *MV-induced algebra* of $\mathbf{A}$, iff $\mathbf{A}' = (\mathrm{A}, \to, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ is an algebra and, for all $x, y, z \in \mathrm{A}$:[1]

$$
\begin{array}{rlrl}
\overline{1} = \neg\overline{0}, & (\star\mathrm{MV8}) & x \vee y = (x \,\&\, \neg y) \mathbin{\underline{\vee}} y, & (\star\mathrm{MV15}) \\
x \,\&\, y = \neg(\neg x \mathbin{\underline{\vee}} \neg y), & (\star\mathrm{MV12}) & x \wedge y = (x \mathbin{\underline{\vee}} \neg y) \,\&\, y, & (\star\mathrm{MV16}) \\
x \to y = \neg x \mathbin{\underline{\vee}} y, & (\star\mathrm{MV13}) & x \not\equiv y = \neg(x \equiv y). & (\star\mathrm{MV17}) \\
x \equiv y = (x \to y) \,\&\, (y \to x), & (\star\mathrm{MV14}) & &
\end{array}
$$

The basic identities are as follows.

49. Let $\mathbf{A} = (\mathrm{A}, \underline{\vee}, \neg, \overline{0})$ be an algebra and let $\mathbf{A}' = (\mathrm{A}, \to, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ be its MV-induced algebra. We call $\mathbf{A}$ an *MV algebra* iff for all $x, y, z \in \mathrm{A}$:

$$
\begin{array}{rlrl}
x \mathbin{\underline{\vee}} y = y \mathbin{\underline{\vee}} x, & (\star\mathrm{MV1}) & x \mathbin{\underline{\vee}} \overline{1} = \overline{1}, & (\star\mathrm{MV4}) \\
x \mathbin{\underline{\vee}} (y \mathbin{\underline{\vee}} z) = (x \mathbin{\underline{\vee}} y) \mathbin{\underline{\vee}} z, & (\star\mathrm{MV2}) & x \mathbin{\underline{\vee}} \overline{0} = x, & (\star\mathrm{MV5}) \\
x = \neg\neg x, & (\star\mathrm{MV7}) & x \vee y = y \vee x. & (\star\mathrm{MV9})
\end{array}
$$

We can now establish a useful duality result, which gives us identities on conjunctions from identities on disjunctions.

∗50. *Let* $\mathbf{A} = (\mathrm{A}, \underline{\vee}, \neg, \overline{0})$ *be an MV algebra and* $\mathbf{A}' = (\mathrm{A}, \to, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ *be the MV-induced algebra of* $\mathbf{A}$. *Then* $\mathbf{B} = (\mathrm{A}, \&, \neg, \overline{1})$ *is an MV algebra, and the MV-induced algebra of* $\mathbf{B}$ *is of the form* $\mathbf{B}' = (\mathrm{A}, \to', \neg, \underline{\vee}, \&, \vee, \wedge, \equiv', \not\equiv', \overline{1}, \overline{0})$.

Proof. See appendix B.1 (p. 159). □

Finally, we can make more rigorous the claim that Wajsberg algebras and MV algebras are equivalent ways of algebraizing Łukasiewicz logic.

∗51. *Let* $\mathbf{A} = (\mathrm{A}, \to, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ *be an algebra. Then* $\mathbf{A}$ *is the Wajsberg-induced algebra of a Wajsberg algebra* $(\mathrm{A}, \to, \overline{0})$, *iff* $\mathbf{A}$ *is the MV-induced algebra of an MV algebra* $(\mathrm{A}, \underline{\vee}, \neg, \overline{0})$.

Proof. See appendix B.1 (p. 164). □

This is why, from here on, we will not make the distinction between MV-algebras and Wajsberg algebras explicit. Rather we will just call them Łukasiewicz algebra.

52. We call $\mathbf{A}$ a Łukasiewicz algebra, iff $\mathbf{A}$ is the Wajsberg-induced algebra of a Wajsberg algebra, or, equivalently, if $\mathbf{A}$ is the MV-induced algebra of an MV algebra.

[1] The equation numbers marked here with the prefix 'MV' are coordinated with the numbers used by Chang (1958*a*, 1959).

We can now develop the algebra some more. First, let's establish the rest of the identities which had a central role to play for Chang (1958*a*, 1959).

$\ast$**53.** *Let* $\mathbf{A} = (\mathrm{A}, \rightarrow, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ *be a Łukasiewicz algebra. For all* $x, y, z \in \mathrm{A}$*:*

$$x \,\underline{\vee}\, \neg x = \overline{1}, \tag{†MV3}$$

$$\neg(x \,\underline{\vee}\, y) = \neg x \,\&\, \neg y, \tag{†MV6}$$

$$x \vee (y \vee z) = (x \vee y) \vee z, \tag{†MV10}$$

$$x \,\underline{\vee}\, (y \wedge z) = (x \,\underline{\vee}\, y) \wedge (x \,\underline{\vee}\, z). \tag{†MV11}$$

Proof. See appendix B.1 (p. 168). □

Then, we get a number of identities directly from duality.

$\ast$**54.** *Let* $\mathbf{A} = (\mathrm{A}, \rightarrow, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ *be a Łukasiewicz algebra. For all* $x, y, z \in \mathrm{A}$*:*

$$x \,\&\, y = y \,\&\, x, \quad (\dagger\mathrm{MV1}') \qquad x \,\&\, \overline{0} = \overline{0}, \quad (\dagger\mathrm{MV4}')$$

$$x \,\&\, (y \,\&\, z) = (x \,\&\, y) \,\&\, z, \quad (\dagger\mathrm{MV2}') \qquad x \,\&\, \overline{1} = x, \quad (\dagger\mathrm{MV5}')$$

$$x = \neg\neg x, \quad (\dagger\mathrm{MV7}') \qquad x \wedge y = y \wedge x. \quad (\dagger\mathrm{MV9}')$$

Furthermore, for all $x, y, z \in \mathrm{A}$*, we have*

$$x \,\&\, \neg x = \overline{0}, \tag{†MV3′}$$

$$\neg(x \,\&\, y) = \neg x \,\underline{\vee}\, \neg y, \tag{†MV6′}$$

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z, \tag{†MV10′}$$

$$x \,\&\, (y \vee z) = (x \,\&\, y) \vee (x \,\&\, z). \tag{†MV11′}$$

Proof. This follows from definition 49, theorem 53, and corollary 50. □

Finally, we can establish the rest of the algebraic identities which we will need for our main result (section 3.3).

$\ast$**55.** *Let* $\mathbf{A} = (\mathrm{A}, \rightarrow, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ *be a Łukasiewicz algebra. For all* $x, y, z \in \mathrm{A}$*:*

$$x = y \text{ \textit{iff} } x \rightarrow y = \overline{1} \text{ \textit{and} } y \rightarrow x = \overline{1}; \tag{†W13}$$

$$\text{\textit{if} } x \rightarrow y = \overline{1} \text{ \textit{and} } y \rightarrow z = \overline{1} \text{ \textit{then} } x \rightarrow z = \overline{1}. \tag{†W14}$$

Furthermore, the following identities hold for all $x, y, z \in \mathrm{A}$*:*

$$x \rightarrow x = \overline{1}, \quad (\dagger\mathrm{W15}) \qquad \neg x \rightarrow \neg y = y \rightarrow x, \quad (\dagger\mathrm{W18})$$

$$x \rightarrow \overline{1} = \overline{1}, \quad (\dagger\mathrm{W16}) \qquad x \rightarrow (y \rightarrow x) = \overline{1}, \quad (\dagger\mathrm{W19})$$

$$\overline{0} \rightarrow x = \overline{1}, \quad (\dagger\mathrm{W17}) \qquad x \rightarrow (y \rightarrow z) = y \rightarrow (x \rightarrow z), \quad (\dagger\mathrm{W20})$$

$$\neg(x \vee y) = \neg x \wedge \neg y, \tag{†Ł1}$$
$$\neg(x \wedge y) = \neg x \vee \neg y, \tag{†Ł1′}$$
$$y \to (x \vee y) = \overline{1}, \tag{†Ł2}$$
$$(x \wedge y) \to y = \overline{1}, \tag{†Ł4}$$
$$x \to (x \vee y) = \overline{1}, \tag{†Ł3}$$
$$(x \wedge y) \to x = \overline{1}, \tag{†Ł5}$$
$$x \vee x = x, \tag{†Ł6}$$
$$x \wedge x = x, \tag{†Ł6′}$$
$$x \vee (x \wedge y) = x, \tag{†Ł7}$$
$$x \wedge (x \vee y) = x, \tag{†Ł7′}$$
$$x \vee \overline{1} = \overline{1}, \tag{†Ł8}$$
$$x \wedge \overline{0} = \overline{0}, \tag{†Ł8′}$$
$$x \vee \overline{0} = x, \tag{†Ł9}$$
$$x \wedge \overline{1} = x, \tag{†Ł9′}$$

$$(x \to z) \to \Big((y \to z) \to \big((x \vee y) \to z\big) \Big) = \overline{1}, \tag{†Ł10}$$
$$(z \to x) \to \Big((z \to y) \to \big(z \to (x \wedge y)\big) \Big) = \overline{1}, \tag{†Ł11}$$
$$(x \,\&\, y) \to z = x \to (y \to z), \tag{†Ł12}$$
$$(x \,\&\, y) \to z = (\neg z \,\&\, y) \to \neg x, \tag{†Ł13}$$
$$(x \,\&\, y) \to z = (x \,\&\, \neg z) \to \neg y, \tag{†Ł14}$$
$$x \to \big(y \;\to\; (x \,\&\, y)\big) = \overline{1}, \tag{†Ł15}$$
$$(x \to y) \to \big((z \,\&\, x) \to (z \,\&\, y)\big) = \overline{1}, \tag{†Ł16}$$
$$\big((x_1 \to y_1) \,\&\, (x_2 \to y_2)\big) \;\to\; \big((x_1 \,\&\, x_2) \to (y_1 \,\&\, y_2)\big) = \overline{1}. \tag{†Ł17}$$

Proof. See appendix B.1 (p. 170). □

From these identities, we can now see a number of relationships between our Łukasiewicz algebra and other types of algebras:

∗56. *Let* $\mathbf{A} = (\mathrm{A}, \to, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ *be a Łukasiewicz algebra. Now*

- $(\mathrm{A}, \vee, \wedge, \overline{0}, \overline{1})$ *is a bounded lattice,*
- $(\mathrm{A}, \underline{\vee}, \overline{0})$ *is an abelian monoid with neutral element* $\overline{0}$*,*
- $(\mathrm{A}, \&, \overline{1})$ *is an abelian monoid with neutral element* $\overline{1}$*,*
- $(\mathrm{A}, \underline{\vee}, \&, \neg)$ *is a DeMorgan algebra, and*
- $(\mathrm{A}, \vee, \wedge, \neg)$ *is a DeMorgan algebra.*

Proof. This follows from definition 49, theorems 53, and 55. □

3.1.5. Metatheory

In the previous sections, we have presented Łukasiewicz logic from a model theoretic point of view and from a proof theoretic point of view, and we have developed it from its axiomatic seeds into a collection of algebraic identities which will turn out to be useful throughout the rest of this work, and in particular for the proof of our main result in section 3.3. In this section, we will conclude our presentation of this logic by giving some pointers to results on the metatheory which connects these three viewpoints.

First, note that our model theoretic definition 36 leads to the so-called standard algebra of Łukasiewicz logic. Here we take $\mathbb{V}_{\aleph_0}$ as the support of the algebra, we take $x \to y = \min(1, 1 - x + y)$ as the implication operator, and we take $\overline{0} = 0$ for $0 \in \mathbb{V}_{\aleph_0}$ as the falsity constant. The Wajsberg-induced algebra which uses this implication operator is then a Łukasiewicz algebra.

We can also approach Łukasiewicz algebra from a proof theoretic angle, by using the Lindenbaum-Tarski process to obtain an algebra over sets of formulae which can be derived from each other. This Lindenbaum-Tarski algebra, too, is a Łukasiewicz algebra.

These two results about the standard algebra and the Lindenbaum-Tarski algebra of Łukasiewicz logic can be easily checked from what we have established so far, and parallel claims are found in Hájek (1998), Gottwald (2001), Metcalfe et al. (2008). This line of work leads up to the algebraic completeness result of Chang (1959) which in some form or another can be found in many places in the literature, including Hájek (1998), Gottwald (2001), and which establishes that an identity which holds in the standard Łukasiewicz algebra holds in all Łukasiewicz algebras.

We can now state the completeness result for this logic more rigorously:

⁎57. *Let $\mathbb{V}$ be a truth value set and let Λ be a propositional signature. For any formula φ over $\mathbb{V}$ and Λ,*

$$Ł(\mathbb{V}, \Lambda) \vdash \varphi \textit{ iff } Ł(\mathbb{V}, \Lambda) \vDash \varphi.$$

⁎58. *Let $\mathbb{V}$ be a truth value set, Λ be a propositional signature, and* $\mathrm{t} \in \mathbb{V}$ *be a* degree of validity. *For any formula φ over $\mathbb{V}$ and Λ,*

$$Ł(\mathbb{V}, \Lambda) \vdash_{\mathrm{t}} \varphi \textit{ iff } Ł(\mathbb{V}, \Lambda) \vDash_{\mathrm{t}} \varphi.$$

This result has two additional points of interest: The fact that it holds for M-valued Łukasiewicz logic, and the fact that it holds for graded validity and graded provability. The former follows from the completeness proof of Rose & Rosser (1958), the latter has been established by Pavelka (1979), and later re-formulated by Hájek (1998).

For our purposes, however, it is sufficient to note that we will build our logic simply by extending the propositional language and by using the identities of Łukasiewicz algebra to derive further identitites. Since these identities will hold in all Łukasiewicz algebras, this will hold in particular for the standard algebra, implying a model-theoretic validity, and for the Lindenbaum-Tarski algebra of the logic, implying a provability. If we wish to establish the validity and provability of a formula, we will simply prove its identity with the element $\overline{1}$, if we wish to establish its non-validity and non-provability we will prove its identity with the element $\overline{0}$.

3.2. Predicate Logic

The main complication in moving from propositional logic to predicate logic is the fact that quantifications in first-order logic are usually taken as ranging over infinite domains. The predicate logic we establish here does not go down that road. Instead, we assume strictly finite domains, so that the predicate language can be established as merely syntactic sugar over propositional language. We will assign to each formula χ of the predicate language a purely propositional formula χ', which then implicitly gives us a model theory, a proof theory and an algebra for our logic by reduction to propositional logic.

3.2.1. Language

A predicate signature establishes symbols for variables and constants. It also assigns symbols to predicates and defines their arities.

59. We call $\Lambda = (\mathrm{P}, \mathrm{arp}, \mathrm{X}, \mathrm{C})$ a *predicate signature* iff it consists of:

- a finite set P of predicate symbols;
- a function $\mathrm{arp} : \mathrm{P} \mapsto \mathbb{N}$ mapping each predicate symbol $P \in \mathrm{P}$ to its arity;
- a finite set $\mathrm{X} = \{\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_{|\mathrm{X}|}\}$ of variable symbols; and
- a finite set $\mathrm{C} = \{\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_{|\mathrm{C}|}\}$ of constant symbols.

For example, if we wanted to express statements like 'Socrates is mortal' and 'Every man loves a woman', we would need

$$\begin{aligned}\Lambda = \big(\; & \{\mathsf{man}, \mathsf{woman}, \mathsf{mortal}, \mathsf{loves}\}, \\ & \{(\mathsf{man}, 1), (\mathsf{woman}, 1), (\mathsf{mortal}, 1), (\mathsf{loves}, 2)\}, \\ & \{\mathsf{x}\}, \\ & \{\mathsf{Socrates}\} \; \big)\end{aligned}$$

60. Let $\Lambda = (\mathrm{P}, \mathrm{arp}, \mathrm{X}, \mathrm{C})$ be a predicate signature. The following recursive rules define by structured induction the notion of a *predicate formula* over $\mathbb{V}$ and Λ:

- For any propositional formula χ over $\mathbb{V}$ and propositional signature $\varnothing$, we say that χ is a *formula* with no free variables, viz. a formula with free variables $\varnothing$, and that χ is *propositionally constant*.

- For any propositional formula χ over $\mathbb{V}$ and propositional signature $\{\varphi\}$, where φ is a formula with free variables X, we say that χ is a *formula* with free variables X, and that φ is a *subformula* of χ.
- For any propositional formula χ over $\mathbb{V}$ and propositional signature $\{\varphi, \psi\}$, where φ and ψ are formulae with free variables X and Y respectively, we say that χ is a *formula* with free variables $X \cup Y$, and that φ and ψ are *subformulae* of χ.
- For any χ of the form ‘$P(\mathrm{u}_1, \mathrm{u}_2, \dots, \mathrm{u}_{\mathrm{arp}(P)})$’, where $P \in \mathrm{P}$ and all $\mathrm{u}_i \in \mathrm{X} \cup \mathrm{C}$, we say that χ is a *formula* with free variables $\{\mathrm{u}_1, \mathrm{u}_2, \dots, \mathrm{u}_{\mathrm{arp}(P)}\} \setminus \mathrm{C}$, and that χ is a *predication*.
- For any χ of the form ‘$\forall_{(x)}\{\varphi\}\{\psi\}$’, ‘$\exists_{(x)}\{\varphi\}\{\psi\}$’, ‘$\nexists_{(x)}\{\varphi\}\{\psi\}$’, or ‘$\not\forall_{(x)}\{\varphi\}\{\psi\}$’, where φ and ψ are formulae with free variables X and Y respectively, we say that χ is a *formula* with free variables $(X \cup Y) \setminus \{x\}$, that χ is a *quantification*, and that φ and ψ are *subformulae* of χ.
- Nothing else is a *formula*.

So, for example, ‘$\forall_{(\mathsf{y})}\{\mathsf{man}(\mathsf{y})\}\{\mathsf{mortal}(\mathsf{y})\}$’ and ‘$\mathsf{man}(\mathsf{Socrates})$’ are formulae with no free variables, ‘$\mathsf{man}(\mathsf{y})$’ has free variable y, and ‘$\mathsf{man}(\mathsf{woman})$’ is not a formula.

Note that this is the standard language of FOL, with the exception that we always denote quantifiers as binary where it is more common in FOL to denote them as unary. For example, it would be more common to write ‘$\forall_{(\mathsf{y})}\{\mathsf{man}(\mathsf{y}) \rightarrow \mathsf{woman}(\mathsf{y})\}$’. Using the binary notation, however, we can more closely line up our predicate language with the linguistic structures envisioned by Barwise & Cooper (1981), with the object language used by grammars that produce MRS semantics (Copestake et al. 2005) such as the ERG (Flickinger 2000), and with the linguistic structures we will introduce in the next chapter (chapter 4). Also, we will find that this notation is more efficient when dealing with the syllogism in the next section (section 3.3).

Note, however, that this does not change the expressive capacity of the language, as we can always rewrite our binary quantifiers into a unary quantifier as suggested by the above example. In order to rewrite a unary quantifier into a binary quantifier, we simply need a predicate $\top$ which always has $\|\top(x)\| = 1$ for all x. We could then write ‘There is a woman’ as ‘$\exists_{(\mathsf{y})}\{\top(\mathsf{y})\}\{\mathsf{woman}(\mathsf{y})\}$’ and ‘Everything sucks’ as ‘$\forall_{(\mathsf{y})}\{\top(\mathsf{y})\}\{\mathsf{sucks}(\mathsf{y})\}$’.

We can now move on to show how we can rewrite expressions in our predicate language into a purely propositional language. First, we need to translate predicate signatures into propositional signatures.

61. Let $\Lambda = (\mathrm{P}, \mathrm{arp}, \mathrm{X}, \mathrm{C})$ be a predicate signature. We call Λ' the *induced propositional signature* of Λ over D with $|\mathrm{D}| \geq 3$ iff

$$\Lambda' = \{‘P_{i_1, i_2, \dots, i_{\mathrm{arp}(P)}}’ \mid P \in \mathrm{P} \text{ and all } i_j \in \mathrm{D} \cup \mathrm{C}\}.$$

Consider our example predicate signature, and the domain $\mathrm{D} = \{1, 2, 3\}$. The atomic propositions possible are then

$$\begin{aligned}\Lambda' = \{&\mathsf{man}_1, \mathsf{man}_2, \mathsf{man}_3, \mathsf{man}_{\mathsf{Socrates}},\\ &\mathsf{woman}_1, \mathsf{woman}_2, \mathsf{woman}_3, \mathsf{woman}_{\mathsf{Socrates}},\\ &\mathsf{mortal}_1, \mathsf{mortal}_2, \mathsf{mortal}_3, \mathsf{mortal}_{\mathsf{Socrates}},\\ &\mathsf{loves}_{1,1}, \mathsf{loves}_{1,2}, \mathsf{loves}_{1,3}, \mathsf{loves}_{1,\mathsf{Socrates}},\\ &\mathsf{loves}_{2,1}, \mathsf{loves}_{2,2}, \mathsf{loves}_{2,3}, \mathsf{loves}_{2,\mathsf{Socrates}},\\ &\mathsf{loves}_{3,1}, \mathsf{loves}_{3,2}, \mathsf{loves}_{3,3}, \mathsf{loves}_{3,\mathsf{Socrates}},\\ &\mathsf{loves}_{\mathsf{Socrates},1}, \mathsf{loves}_{\mathsf{Socrates},2}, \mathsf{loves}_{\mathsf{Socrates},3}, \mathsf{loves}_{\mathsf{Socrates},\mathsf{Socrates}}\}\end{aligned}$$

Finally, we can show how to rewrite entire predicate formulae into propositional formulae over such signatures.

62. Let $\mathbb{V}$ be a truth value set, let $\Lambda = (\mathrm{P}, \mathrm{arp}, \mathrm{X}, \mathrm{C})$ be a predicate signature and Λ' be the induced propositional signature of Λ over D. We call a mapping b a *binding* iff $b : \mathrm{X} \cup \mathrm{C} \mapsto \mathrm{D} \cup \mathrm{C}$, such that for all $c \in \mathrm{C}$, we have $\mathrm{b}(c) = c$. For any predicate formula χ over $\mathbb{V}$ and Λ with free variables X and any binding b, we assign a *corresponding* propositional formula $\mathcal{B}(b, \chi)$ over $\mathbb{V}$ and Λ' under binding b. We define $\mathcal{B}(b, \cdot)$ recursively as follows. For any predicate formulae φ and ψ with free variables X and Y respectively and any binding b:

$$\begin{aligned}\mathcal{B}\big(b, P(u_1, u_2, \ldots, u_{\mathrm{arp}(P)})\big) &= \mathsf{p}_{P, b(u_1), b(u_2), \ldots, b(u_{\mathrm{arp}(P)})},\\ \mathcal{B}\big(b, \forall_{(x)}\{\varphi\}\{\psi\}\big) &= \bigwedge_{i \in \mathrm{D} \cup \mathrm{C}} \mathcal{B}\big(b \cup (x \mapsto i), \varphi\big) \to \mathcal{B}\big(b \cup (x \mapsto i), \psi\big),\\ \mathcal{B}\big(b, \exists_{(x)}\{\varphi\}\{\psi\}\big) &= \bigvee_{i \in \mathrm{D} \cup \mathrm{C}} \mathcal{B}\big(b \cup (x \mapsto i), \varphi\big) \wedge \mathcal{B}\big(b \cup (x \mapsto i), \psi\big),\\ \mathcal{B}\big(b, \nexists_{(x)}\{\varphi\}\{\psi\}\big) &= \mathcal{B}\big(b, \neg\exists_{(x)}\{\varphi\}\{\psi\}\big),\\ \mathcal{B}\big(b, \not\forall_{(x)}\{\varphi\}\{\psi\}\big) &= \mathcal{B}\big(b, \neg\forall_{(x)}\{\varphi\}\{\psi\}\big).\end{aligned}$$

If χ is a formula with no free variables, its corresponding propositional formula is $\mathcal{B}(\varnothing, \chi)$, otherwise its corresponding propositional formula is left undefined if no reference is made to a binding.

So, for example, '$\mathsf{man}(\mathsf{Socrates})$' would be written as

$$\mathsf{man}_{\mathsf{Socrates}}.$$

The formula '$\forall_{(\mathsf{y})}\{\mathsf{man}(\mathsf{y})\}\{\mathsf{mortal}(\mathsf{y})\}$' would be written as

$$(\mathsf{man}_1 \to \mathsf{mortal}_1) \wedge (\mathsf{man}_2 \to \mathsf{mortal}_2) \wedge (\mathsf{man}_3 \to \mathsf{mortal}_3) \wedge (\mathsf{man}_{\mathsf{Socrates}} \to \mathsf{mortal}_{\mathsf{Socrates}}),$$

and the formula '$\exists_{(\mathsf{y})}\{\mathsf{man}(\mathsf{y})\}\{\mathsf{mortal}(\mathsf{y})\}$' would be written as

$$(\mathsf{man}_1 \wedge \mathsf{mortal}_1) \vee (\mathsf{man}_2 \wedge \mathsf{mortal}_2) \vee (\mathsf{man}_3 \wedge \mathsf{mortal}_3) \vee (\mathsf{man}_{\mathsf{Socrates}} \wedge \mathsf{mortal}_{\mathsf{Socrates}}).$$

3.3. Syllogistic Logic

At this point, we can employ the results summarized in this chapter to establish our main result on the syllogism.

Recall that the syllogism plays a central role in this thesis, which we previously summarized as follows: (1) Assuming the syllogism as a benchmark fragment of NLR, it turns out that the model theory which underlies NLR is not necessarily two-valued, but can be a many-valued Łukasiewicz logic. This is what we will establish in this section. (2) Given the syllogism as a logical language of far less expressive power than natural language itself, we can still obtain a good approximation to NLR using the syllogism. This is the result which we will establish in the next chapter (chapter 4).

The syllogism was originally established by Aristotle in his *Prior Analytics*. The first treatment of the syllogism from the standpoint of modern formal logic, and perhaps still the authoritative reference on the subject, is the monograph by Łukasiewicz (1951). More recent modern treatments of the syllogism have been established by Crabbé (2001) and by Moss (2007*a*,*b*).

3.3.1. Language

Our exposition of this logical fragment, again, begins by constructing its formulae.

63. We call Λ a *syllogistic signature* iff Λ is a set of term symbols $\Lambda = \{\mathsf{A}, \mathsf{B}, \ldots\}$.

64. Let Λ be a syllogistic signature. We call the predicate signature $\Lambda' = (\Lambda, \mathrm{arp}, \{\mathsf{y}\}, \varnothing)$ the *induced predicate signature* of Λ iff $\mathrm{arp}(P) = 1$ for all $P \in \Lambda$.

So a syllogistic signature is simply a predicate signature which consists exclusively of unary predicate symbols, one variable, and no constants. Next, we can impose some structure on formulae of the syllogism.

65. Let Λ be a syllogistic signature.

- We say that χ is an *atomic syllogistic proposition* about S and P for any term symbols $S, P \in \Lambda$, iff, χ is of one of the forms

  $$\forall_{(\mathsf{y})}\, S(\mathsf{y})\, P(\mathsf{y}) \text{ or } \exists_{(\mathsf{y})}\, S(\mathsf{y})\, P(\mathsf{y}),$$

  or is the negation

  $$\nexists_{(\mathsf{y})}\, S(\mathsf{y})\, P(\mathsf{y}) \text{ or } \not\forall_{(\mathsf{y})}\, S(\mathsf{y})\, P(\mathsf{y}),$$

  of one of those forms.

syllogism	pre-syllogism	equivalent syllogism
$(1)\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})}; \end{array}$	$(4')\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}; \end{array}$	$(4)\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}; \end{array}$
$(2)\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})}; \end{array}$	$(2')\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}; \end{array}$	$(2)\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}; \end{array}$
$(3)\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})}; \end{array}$	$(3')\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}; \end{array}$	$(3)\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}; \end{array}$
$(4)\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})}; \end{array}$	$(1')\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}; \end{array}$	$(1)\ \begin{array}{ll} (& \Psi_1 {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})} \\ \& & \Psi_2 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} M {\scriptstyle(\mathrm{y})}\ ) \\ \hline \to & \Psi_3 {\scriptstyle(\mathrm{y})} P {\scriptstyle(\mathrm{y})} S {\scriptstyle(\mathrm{y})}. \end{array}$

Figure 3.1.: syllogisms, pre-syllogisms and their figures

- We say that χ is a *pre-syllogism* about S, M, and P, for any term symbols $S, M, P \in \Lambda$, iff χ is of the form $(\varphi_1 \& \varphi_2) \to \psi$, where φ_1 is an atomic syllogistic proposition about M and P, φ_2 is an atomic syllogistic proposition about S and M, and ψ is an atomic syllogistic proposition about S and P.

- It can be seen that any pre-syllogism χ can be written in one of the eight forms listed in the first or second column of Figure 3.1, where $S, M, P \in \Lambda$, and where the form variables Ψ_1, Ψ_2, and Ψ_3 can stand for one of $\forall$, $\exists$, $\nexists$, and $\not\forall$. If a pre-syllogism χ is written in the form labelled i in the table,
  - We call i its *figure*.
  - We call the combination (Ψ_1, Ψ_2, Ψ_3) its *mood*.
  - We call the combination $(i, \Psi_1, \Psi_2, \Psi_3)$ of figure and mood its *schema*.

- We call a mood (Ψ_1, Ψ_2, Ψ_3)
  - *affirmative* iff $\Psi_1, \Psi_2, \Psi_3 \in \{\forall, \exists\}$,
  - *negative* iff $\Psi_1, \Psi_2, \Psi_3 \in \{\nexists, \not\forall\}$,
  - *mixed* otherwise.

- We say that χ is a *proper syllogism* over Λ, iff it is a pre-syllogism in one of the figures 1, 2, 3, or 4, i.e. excluding the figures $1'$, $2'$, $3'$, and $4'$.

So we can freely combine any of the eight figures with any of the 64 moods to obtain the 512 schemas for pre-syllogisms. A schema determines a pre-syllogism uniquely, up to substitution of its term symbols.

As pre-syllogisms are at the same time formulae of our predicate calculus, we can immediately establish their semantics:

66. Let Λ be some syllogistic signature, let Λ' be its induced predicate signature, and let Λ'' be the induced propositional signature of Λ'. Let φ be a pre-syllogism over Λ and let φ' be its corresponding propositional formula over Λ'' and some $\mathbb{V}$. Let $t \in \mathbb{V}$ be any validity threshold. Now we define

$$\mathrm{SYL}(\Lambda) \vDash \varphi \text{ iff } \text{Ł}(\mathbb{V}, \Lambda) \vDash_t \varphi''$$

Then, we can break up some symmetries within the space of pre-syllogisms:

67. Let Λ be some syllogistic signature. For any pre-syllogism χ over Λ, we say that χ' is the commutation of χ and that χ is the commutation of χ', iff χ is of the form listed in the center column of Figure 3.1 and χ' is of the form listed next to it in the right column.

***68.** *Let Λ be some syllogistic signature. For every pre-syllogism φ over Λ, if φ' is the commutation of φ, we have:*

$$\mathrm{SYL}(\Lambda) \vDash \varphi \textit{ iff } \mathrm{SYL}(\Lambda) \vDash \varphi'.$$

Thus, to every pre-syllogism there corresponds a proper syllogism which, by commutation, is equivalent to it.

Proof. It follows via the commutativity of strong conjunction (†MV1′) that a pre-syllogism χ is identical in any Łukasiewicz algebra to its commutation. Validity follows from the fact that the standard algebra of Łukasiewicz logic is a Łukasiewicz algebra. The third column in Figure 3.1 shows how, by means of term substitution and commutation, we obtain a proper syllogism from any pre-syllogism. □

69. Let Λ be some syllogistic signature. For any syllogism χ over Λ, we say that χ' is the *first contraposition* (1CP) of χ, iff χ is listed in the left column of figure 3.2 and χ' is of the form listed next to it in the center column. Similarly, we say χ'' is the *second contraposition* (2CP) of χ, iff χ'' of the form listed in the right column.

***70.** *Let Λ be some syllogistic signature. For every syllogism φ over Λ, if φ' is the 1CP of φ, and φ'' is the 2CP of φ, we have:*

$$\mathrm{SYL}(\Lambda) \vDash \varphi \textit{ and } \mathrm{SYL}(\Lambda) \vDash \varphi' \textit{ and } \mathrm{SYL}(\Lambda) \vDash \varphi'' \textit{ are mutually equivalent.}$$

Proof. This is similar to our previous proof. This time, we need the contraposition identities (†Ł13) and (†Ł14). □

$$\begin{array}{ccc}
\text{orig.} & \text{1CP} & \text{2CP} \\
(1)\;\begin{array}{l} (\;\;\Psi_{1\,(y)}\,M_{(y)}\,P_{(y)} \\ \;\&\;\;\Psi_{2\,(y)}\,S_{(y)}\,M_{(y)}\;) \\ \hline \rightarrow\;\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)}; \end{array}
& (2)\;\begin{array}{l} (\;\;\Psi_{1\,(y)}\,M_{(y)}\,P_{(y)} \\ \;\&\;\;\neg\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)}\;) \\ \hline \rightarrow\;\neg\Psi_{2\,(y)}\,S_{(y)}\,M_{(y)}; \end{array}
& (3)\;\begin{array}{l} (\;\;\neg\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)} \\ \;\&\;\;\Psi_{2\,(y)}\,S_{(y)}\,M_{(y)}\;) \\ \hline \rightarrow\;\neg\Psi_{1\,(y)}\,M_{(y)}\,P_{(y)}; \end{array} \\
(2)\;\begin{array}{l} (\;\;\Psi_{1\,(y)}\,P_{(y)}\,M_{(y)} \\ \;\&\;\;\Psi_{2\,(y)}\,S_{(y)}\,M_{(y)}\;) \\ \hline \rightarrow\;\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)}; \end{array}
& (1)\;\begin{array}{l} (\;\;\Psi_{1\,(y)}\,P_{(y)}\,M_{(y)} \\ \;\&\;\;\neg\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)}\;) \\ \hline \rightarrow\;\neg\Psi_{2\,(y)}\,S_{(y)}\,M_{(y)}; \end{array}
& (3)\;\begin{array}{l} (\;\;\Psi_{2\,(y)}\,S_{(y)}\,M_{(y)} \\ \;\&\;\;\neg\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)}\;) \\ \hline \rightarrow\;\neg\Psi_{1\,(y)}\,P_{(y)}\,M_{(y)}; \end{array} \\
(3)\;\begin{array}{l} (\;\;\Psi_{1\,(y)}\,M_{(y)}\,P_{(y)} \\ \;\&\;\;\Psi_{2\,(y)}\,M_{(y)}\,S_{(y)}\;) \\ \hline \rightarrow\;\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)}; \end{array}
& (2)\;\begin{array}{l} (\;\;\neg\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)} \\ \;\&\;\;\Psi_{1\,(y)}\,M_{(y)}\,P_{(y)}\;) \\ \hline \rightarrow\;\neg\Psi_{2\,(y)}\,M_{(y)}\,S_{(y)}; \end{array}
& (1)\;\begin{array}{l} (\;\;\neg\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)} \\ \;\&\;\;\Psi_{2\,(y)}\,M_{(y)}\,S_{(y)}\;) \\ \hline \rightarrow\;\neg\Psi_{1\,(y)}\,M_{(y)}\,P_{(y)}; \end{array} \\
(4)\;\begin{array}{l} (\;\;\Psi_{1\,(y)}\,P_{(y)}\,M_{(y)} \\ \;\&\;\;\Psi_{2\,(y)}\,M_{(y)}\,S_{(y)}\;) \\ \hline \rightarrow\;\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)}; \end{array}
& (4)\;\begin{array}{l} (\;\;\neg\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)} \\ \;\&\;\;\Psi_{1\,(y)}\,P_{(y)}\,M_{(y)}\;) \\ \hline \rightarrow\;\neg\Psi_{2\,(y)}\,M_{(y)}\,S_{(y)}; \end{array}
& (4)\;\begin{array}{l} (\;\;\Psi_{2\,(y)}\,M_{(y)}\,S_{(y)} \\ \;\&\;\;\neg\Psi_{3\,(y)}\,S_{(y)}\,P_{(y)}\;) \\ \hline \rightarrow\;\neg\Psi_{1\,(y)}\,P_{(y)}\,M_{(y)}; \end{array}
\end{array}$$

Figure 3.2.: contrapositions

A	1CP	2CP	A	1CP	2CP
$(1,\forall,\forall,\forall)$	$(2,\forall,\neg\forall,\neg\forall)$	$(3,\neg\forall,\forall,\neg\forall)$	Barbara	Baroco	Bocardo
$(1,\forall,\exists,\exists)$	$(2,\forall,\neg\exists,\neg\exists)$	$(3,\neg\exists,\exists,\neg\forall)$	Darii	Camestres	Ferison
$(3,\forall,\exists,\exists)$	$(2,\neg\exists,\forall,\neg\exists)$	$(1,\neg\exists,\exists,\neg\forall)$	Datisi	Cesare	Ferio
$(3,\exists,\forall,\exists)$	$(2,\neg\exists,\exists,\neg\forall)$	$(1,\neg\exists,\forall,\neg\exists)$	Disamis	Festino	Celarent
$(4,\exists,\forall,\exists)$	$(4,\neg\exists,\exists,\neg\forall)$	$(4,\forall,\neg\exists,\neg\exists)$	Dimaris	Fresison	Camenes

Figure 3.3.: left: valid syllogisms; right: corresponding scholastic names

3.3.2. Valid Syllogisms

71. Let Λ be some syllogistic signature. For any syllogism χ over Λ, we say that χ is *traditionally correct*, iff the schema of χ is listed in Figure 3.3.

Note that we consider what Crabbé (2001) calls the weakening-free fragment of the syllogism, i.e. we do not permit subalternation as a valid inference rule, so we do not include among our list of correct moods the four moods Darapti, Felapton, Bramantip, Fesapo, or their five weakened variants. These are considered correct in Aristotle's classical syllogism, and in the treatment by Łukasiewicz (1951). However, they commit what is now sometimes called the existential fallacy. Whether or not this is, in fact, a fallacy relies, of course, on the intended semantics of the logic. We will demonstrate the controversy by considering the mood Bramantip.

This mood takes as antecedents $\forall_{(y)} Z_{(y)} Y_{(y)}$ and $\forall_{(y)} Y_{(y)} X_{(y)}$, from which it would follow, by the mood Barbara, that $\forall_{(y)} Z_{(y)} X_{(y)}$. The mood Bramantip would conclude $\exists_{(y)} X_{(y)} Z_{(y)}$, from which it would also follow that $\exists_{(y)} Z_{(y)} X_{(y)}$.

Now, if we think of the intended semantics as set-theoretic in nature, there is a problematic case where Z is the empty set. Then $\forall_{(y)} Z_{(y)} Y_{(y)}$ and $\forall_{(y)} Z_{(y)} X_{(y)}$, as it might be concluded by the mood Barabara, would be true vacuously. However, $\exists_{(y)} Z_{(y)} X_{(y)}$ would assert the existence of elements in Z, contradicting the intended semantic.

We have decided to go with the modern treatment, on the basis that it would still be possible to introduce a universal quantifier with existential import, i.e. if needed we can define $\forall_{+\,(y)} X_{(y)} Y_{(y)} = \forall_{(y)} X_{(y)} Y_{(y)} \mathbin{\&} \exists_{(y)} X_{(y)} Y_{(y)}$. The existential quantifier $\exists$ would then interact with this universal quantifier $\forall_{+}$ in the classical way.

3.3.3. Metatheory

We can now go on to show our main result: The syllogistic fragment of the predicate calculus we have constructed here proves all and only those syllogisms which are traditionally considered correct.

Soundness

$\ast$72. *Let Λ be some syllogistic signature. For every syllogism φ over Λ, we have*

$$\mathrm{SYL}(\Lambda) \vDash \varphi \textit{ if } \varphi \textit{ is listed in Figure 3.3.}$$

Before we can prove this soundness result, we first need a lemma.

$\ast$73. *Let $(\mathcal{V}, \rightarrow, \neg, \mathbin{\&}, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ be a Łukasiewicz algebra. For all $x_1, x_2, y_1, y_2, z_1, z_2 \in \mathcal{V}$, and all N, we have*

$$(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \ldots \vee (x_N \wedge y_N) = (y_1 \wedge x_1) \vee (y_2 \wedge x_2) \vee \ldots \vee (y_N \wedge x_N), \tag{$\star$SYL$\cdot$comm}$$

$$\begin{aligned} \Big(\; & \big((y_1 \rightarrow z_1) \wedge (y_2 \rightarrow z_2) \wedge \ldots \wedge (y_N \rightarrow z_N)\big) \\ \mathbin{\&} \; & \big((x_1 \rightarrow y_1) \wedge (x_2 \rightarrow y_2) \wedge \ldots \wedge (x_N \rightarrow y_N)\big) \; \Big) \\ \rightarrow \; & \Big(\; (x_1 \rightarrow z_1) \wedge (x_2 \rightarrow z_2) \wedge \ldots \wedge (x_N \rightarrow z_N) \; \Big) = \overline{1}, \end{aligned} \tag{$\star$SYL$\cdot$1$\cdot$A}$$

$$\begin{aligned} \Big(\; & \big((y_1 \rightarrow z_1) \wedge (y_2 \rightarrow z_2) \wedge \ldots \wedge (y_N \rightarrow z_N)\big) \\ \mathbin{\&} \; & \big((x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \ldots \vee (x_N \wedge y_N)\big) \; \Big) \\ \rightarrow \; & \Big(\; (x_1 \wedge z_1) \vee (x_2 \wedge z_2) \vee \ldots \vee (x_N \wedge z_N) \; \Big) = \overline{1}. \end{aligned} \tag{$\star$SYL$\cdot$2$\cdot$A}$$

Proof. See appendix B (p. 171). □

Identity (⋆SYL·comm) states the commutativity of the existantial affirmative, i.e.

$$\exists\,(\mathsf{y})\,X(\mathsf{y})\,Y(\mathsf{y}) \;=\; \exists\,(\mathsf{y})\,Y(\mathsf{y})\,X(\mathsf{y}).$$

Identity (⋆SYL·1·A) is the schema Barbara,

$$\forall\,(\mathsf{y})\,Y(\mathsf{y})\,Z(\mathsf{y}) \;\&\; \forall\,(\mathsf{y})\,X(\mathsf{y})\,Y(\mathsf{y}) \;\to\; \forall\,(\mathsf{y})\,X(\mathsf{y})\,Z(\mathsf{y}),$$

and identity (⋆SYL·2·A) is the schema Darii,

$$\forall\,(\mathsf{y})\,Y(\mathsf{y})\,Z(\mathsf{y}) \;\&\; \exists\,(\mathsf{y})\,X(\mathsf{y})\,Y(\mathsf{y}) \;\to\; \exists\,(\mathsf{y})\,X(\mathsf{y})\,Z(\mathsf{y}).$$

Note that, in our naming scheme, the number stands for a line in Figure 3.3, and the suffixes ‘A’, ‘1CP’, or ‘2CP’ identifies the colum.

We can view these three algebraic identities as axiomatic for the theory of the syllogism. What we have shown here, however, is that, instead of accepting them as axiomatic, we can also construct them from a reduction of the language of the syllogism to the language of propositional logic. Now we can show the rest of the soundness proof.

Proof of theorem 72. Let $(\mathcal{V}, \to, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ be a Łukasiewicz algebra. Now, from the algebraic identities in lemma 73, we can also derive the following identities:

$$\begin{aligned} &\Big(\; \big((y_1 \to z_1) \wedge (y_2 \to z_2) \wedge \ldots \wedge (y_N \to z_N)\big) \\ &\;\&\; \big((y_1 \wedge x_1) \vee (y_2 \wedge x_2) \vee \ldots \vee (y_N \wedge x_N)\big) \;\Big) \\ &\to \Big(\; (x_1 \wedge z_1) \vee (x_2 \wedge z_2) \vee \ldots \vee (x_N \wedge z_N) \;\Big) \;=\; \overline{1}, \end{aligned} \tag{†SYL·3·A}$$

$$\begin{aligned} &\Big(\; \big((y_1 \wedge x_1) \vee (y_2 \wedge x_2) \vee \ldots \vee (y_N \wedge x_N)\big) \\ &\;\&\; \big((y_1 \to z_1) \wedge (y_2 \to z_2) \wedge \ldots \wedge (y_N \to z_N)\big) \;\Big) \\ &\to \Big(\; (z_1 \wedge x_1) \vee (z_2 \wedge x_2) \vee \ldots \vee (z_N \wedge x_N) \;\Big) \;=\; \overline{1}, \end{aligned} \tag{†SYL·4·A}$$

$$\begin{aligned} &\Big(\; \big((x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \ldots \vee (x_N \wedge y_N)\big) \;\Big) \\ &\;\&\; \big((y_1 \to z_1) \wedge (y_2 \to z_2) \wedge \ldots \wedge (y_N \to z_N)\big) \\ &\to \Big(\; (z_1 \wedge x_1) \vee (z_2 \wedge x_2) \vee \ldots \vee (z_N \wedge x_N) \;\Big) \;=\; \overline{1}. \end{aligned} \tag{†SYL·5·A}$$

This can easily be seen, because Datisi (†SYL·3·A) results from Darii (⋆SYL·2·A) simply by applying the commutativity of the existential affirmative (⋆SYL·comm) in the antecedent. Then Disamis (†SYL·4·A) results from Datisi (†SYL·3·A) by applying the commutatvity of strong conjunction (†MV1′), and the commutativity of the existential affirmative (⋆SYL·comm) in the consequent. Finally, Dimaris (†SYL·5·A) results from Disamis (†SYL·4·A) by applying the commutativity of the existential affirmative (⋆SYL·comm) in the antecedent again.

n	$\nexists, \not\forall$	M	$\curvearrowleft\!\!\curvearrowright$	P	$S = \{1\}$	32
	$\nexists, \not\forall$	S	$\curvearrowleft\!\!\curvearrowright$	M	$M = \{2\}$	
	$\nexists, \not\forall$	S		P	$P = \{1\}$	
b	$\nexists, \not\forall$	M	$\curvearrowleft\!\!\curvearrowright$	P	$S = \{1\}$	32
	$\nexists, \not\forall$	S	$\curvearrowleft\!\!\curvearrowright$	M	$M = \{2\}$	
	$\forall, \exists$	S		P	$P = \{3\}$	
c	$\forall, \exists$	M	$\curvearrowleft\!\!\curvearrowright$	P	$S = \{1\}$	32
	$\forall, \exists$	S	$\curvearrowleft\!\!\curvearrowright$	M	$M = \{1\}$	
	$\nexists, \not\forall$	S		P	$P = \{1\}$	
a.1.1	$\exists$	M		P	$S = \{1\}$	12
	$\forall, \exists$	S		M	$M = \{1, 2\}$	
	$\forall, \exists$	S		P	$P = \{2\}$	
a.1.2	$\exists$	M		P		2
	$\forall$	M		S		
	$\exists$	S		P		
a.1.3	$\exists$	M		P	$S = \{1, 2\}$	2
	$\forall$	M		S	$M = \{1\}$	
	$\forall$	S		P	$P = \{1\}$	

a.2.1.1	$\forall$	M		P		2
	$\exists$	S		M		
	$\exists$	S		P		
a.2.1.2	$\forall$	P		M	$S = \{1\}$	2
	$\exists$	S		M	$M = \{1, 2\}$	
	$\exists$	S		P	$P = \{2\}$	
a.2.2	$\forall$	M	$\curvearrowleft\!\!\curvearrowright$	P	$S = \{\}$	4
	$\forall$	S	$\curvearrowleft\!\!\curvearrowright$	M	$M = \{\}$	
	$\exists$	S		P	$P = \{\}$	
a.2.3	$\forall$	M	$\curvearrowleft\!\!\curvearrowright$	P	$S = \{1, 2\}$	4
	$\exists$	S		M	$M = \{1\}$	
	$\forall$	S		P	$P = \{1\}$	
a.2.4.1	$\forall$	M		P		1
	$\forall$	S		M		
	$\forall$	S		P		
a.2.4.2	$\forall$	P		M	$S = \{1\}$	1
	$\forall$	S		M	$M = \{1, 2\}$	
	$\forall$	S		P	$P = \{2\}$	
a.2.4.3	$\forall$	M	$\curvearrowleft\!\!\curvearrowright$	P	$S = \{1, 2\}$	2
	$\forall$	M		S	$M = \{1\}$	
	$\forall$	S		P	$P = \{1\}$	

Figure 3.4.: model-theoretic counterexamples for invalid syllogisms

From these algebraic identities, the corresponding validities of correct affirmative mood syllogisms listed in the left column of Figure 3.3 follow via the fact that the standard algebra of Łukasiewicz logic is itself a Łukasiewicz algebra.

Then, observe that each of these correct affirmative moods has an associated first contraposition (1CP) and second contraposition (2CP), listed in the center and right columns of table 3.3 respectively. The validities for these syllogisms follows from the validity of the associated affirmative mood syllogism via corollary 70. □

Completeness

$*$**74.** *Let Λ be some syllogistic signature. For every syllogism φ over Λ, we have*

$\mathrm{SYL}(\Lambda) \vDash \varphi$ *iff φ is listed in Figure 3.3.*

Proof. The 'if'-part was demonstrated above in theorem 72. We will show the 'only if'-

part by considering its contrapositive, i.e. we will show that whenever *it is not the case that* φ is listed in Figure 3.3, then it is also not the case that $\mathrm{SYL}(\Lambda) \vDash \varphi$.

As per the construction of our model theory, let Λ' be the induced predicate signature of Λ, and Λ'' be the induced propositional signature of Λ'. Let φ' be the propositional formula over Λ'' and some $\mathbb{V}$ corresponding to φ. Let $\mathrm{t} \in \mathbb{V}$ be any validity threshold.

Now, note that $\mathbb{V}_2 \subseteq \mathbb{V}$ for all $\mathbb{V}$. So, in order to establish this result, it is sufficient to show that, for each φ, there is a $(\mathbb{V}_2, \Lambda'')$-valuation w, for which $\|\varphi''\|_w = 0$, unless φ is listed in Figure 3.3. Such a valuation would act as a counter-example, showing that it cannot be the case that $\mathrm{SYL}(\Lambda) \vDash_t \varphi$ for any t. These counter-examples are listed in Figure 3.4. To make our list of counter-examples more comprehensible, we have imposed some structure on it.

The table lists in the third column of each line a valuation w in set notation, where $\|X_x\|_w = 1$ if $x \in X$ and $\|X_x\|_w = 0$ otherwise.

The second column shows which syllogisms the valuation w is a counter-example for. Syllogisms are written in three lines, where the first and second line represent the two conjuncts of the antecedent, and where the third line represents the consequent.

Every line in the abbreviated notation can have a number of configurations, each configuration corresponding to a syllogistic proposition.

- A line of the form $\forall\ X\ Y$ has one configuration, corresponding to the proposition $\forall_{\mathrm{(y)}} X_{\mathrm{(y)}} Y_{\mathrm{(y)}}$.
- Analogously, a line of the form $\not\forall\ X\ Y$ has one configuration, corresponding to the proposition $\not\forall_{\mathrm{(y)}} X_{\mathrm{(y)}} Y_{\mathrm{(y)}}$.
- A line of the form $\forall\ X \leftrightarrow Y$ has two configurations, corresponding to the propositions $\forall_{\mathrm{(y)}} X_{\mathrm{(y)}} Y_{\mathrm{(y)}}$ and $\forall_{\mathrm{(y)}} Y_{\mathrm{(y)}} X_{\mathrm{(y)}}$.
- Analogously, a line of the form $\not\forall\ X \leftrightarrow Y$ has two configurations, corresponding to the propositions $\not\forall_{\mathrm{(y)}} X_{\mathrm{(y)}} Y_{\mathrm{(y)}}$ and $\not\forall_{\mathrm{(y)}} Y_{\mathrm{(y)}} X_{\mathrm{(y)}}$.
- A line of the form $\exists\ X\ Y$ or $\exists\ X \leftrightarrow Y$ has two configurations, corresponding to the propositions $\exists_{\mathrm{(y)}} X_{\mathrm{(y)}} Y_{\mathrm{(y)}}$ and $\exists_{\mathrm{(y)}} Y_{\mathrm{(y)}} X_{\mathrm{(y)}}$, if it occurs in an antecedent. However, it has only one configuration, corresponding to the original proposition $\exists_{\mathrm{(y)}} X_{\mathrm{(y)}} Y_{\mathrm{(y)}}$, if it occurs in the consequent.
- Analogously, a line of the form $\nexists\ X\ Y$ or $\nexists\ X \leftrightarrow Y$ has two configurations, corresponding to the propositions $\nexists_{\mathrm{(y)}} X_{\mathrm{(y)}} Y_{\mathrm{(y)}}$ and $\nexists_{\mathrm{(y)}} Y_{\mathrm{(y)}} X_{\mathrm{(y)}}$, if it occurs in an antecedent. However, it has only one configuration, corresponding to the original proposition $\nexists_{\mathrm{(y)}} X_{\mathrm{(y)}} Y_{\mathrm{(y)}}$, if it occurs in the consequent.
- A line of the form $\Psi_1, \Psi_2\ X\ Y$ has as its configurations all configurations of $\Psi_1\ X\ Y$ plus all the configurations of $\Psi_2\ X\ Y$.

The above definition mentions explicitly the number of configurations for each line with one quantifier. If there are two quantifiers, we simply add up the number of configurations for each of the individual quantifiers to obtain the number of configurations for the line. In order to obtain a configuration for the entire three-line syllogism, we can independently choose a configuration for each of the three lines. So by multiplying the number of configurations for each of the lines, we get the number of configurations for the entire syllogism, which is listed in the third column of each row.

Since the first line only refers to the terms M and P, the second to S and M, and the third to S and P, we know that these are pre-syllogisms, and that they are syntactically distinct. Since S and P always occur in the same order in the consequent, they are, in fact, syntactically distinct proper syllogisms, not just syntactically distinct pre-syllogisms.

In order to verify that each valuation w is in fact a valid counter-example for the corresponding syllogisms, simply verify that $\|\varphi_1\|_w = 1$ for all configurations φ_1 of the first line, that $\|\varphi_2\|_w = 1$ for all configurations φ_2 of the second line, and that $\|\psi\|_w = 0$ for all configurations ψ of the third line.

Some rows in the table list one or more syllogism, but no corresponding valuation. By comparison with Figure 3.3, it can be verified that these are the valid syllogisms.

Now all that remains to be done is to verify that Figure 3.4 accounts for each of the 256 syntactically distinct syllogisms which can be formed by combining one of the 64 moods (Ψ_1, Ψ_2, Ψ_3) with one of the four figures. – To that end, consider the following eight cases, each of which accounts for eight moods with four figures, and verify that they are pairwise mutually exclusive, and comprehensively exhaustive.

i. $\Psi_1, \Psi_2, \Psi_3 \in \{\nexists, \not\forall\}$: The counter-example is listed under (n).

ii. $\Psi_1 \in \{\nexists, \not\forall\}$, and $\Psi_2, \Psi_3 \in \{\forall, \exists\}$: By inspecting Figure 3.2, it can be seen that, by corollary 70 and the algebraic double-negation identity ($\star$MV7), each of the four figures for case (ii) can be rewritten into some figure for case (i). So this case reduces to case (i), viz. the counter-example listed under (n) is also a counter-example for the 32 case-ii-syllogisms.

iii. $\Psi_2 \in \{\nexists, \not\forall\}$, and $\Psi_1, \Psi_3 \in \{\forall, \exists\}$: This case reduces to case (i), just as case (ii) reduces to case (i).

iv. $\Psi_3 \in \{\nexists, \not\forall\}$, and $\Psi_1, \Psi_2 \in \{\forall, \exists\}$: The counter-example is listed under (c).

v. $\Psi_1, \Psi_2, \Psi_3 \in \{\forall, \exists\}$: The counter-example are listed with the prefix "a.", and fall within exactly one of the following cases:

   - Ψ_1 = "$\exists$": This accounts for 16 syllogisms, listed with the prefix "a.1".
   - Ψ_1 = "$\forall$": This accounts for 16 syllogisms, listed with the prefix "a.2".

vi. $\Psi_1 \in \{\forall, \exists\}$, and $\Psi_2, \Psi_3 \in \{\nexists, \not\forall\}$: This case reduces to case (v.), just as case (ii) reduces to case (i).

vii. $\Psi_2 \in \{\forall, \exists\}$, and $\Psi_1, \Psi_3 \in \{\nexists, \not\forall\}$: This case reduces to case (v.), just as case (iii) reduces to case (i).

viii. $\Psi_3 \in \{\forall, \exists\}$, and $\Psi_1, \Psi_2 \in \{\nexists, \not\forall\}$: The counter-example is listed under (b). □

With this completeness result, we have arrived at a logic which I believe can be more usefully applied to textual inference problems than classical logic due to the advantageous computational properties of many-valued logic, which we will discuss in chapter 5.

It is, at this point, perhaps also worth reiterating that Łukasiewicz logic proves a subset of all theorems provable in classical logic, and that the present completeness result is nontrivial in that it shows that those theorems which classical logic proves and which Łukasiewicz logic does not prove are not required for the purposes of reasoning within a fragment of propositional logic which corresponds to the syllogism.

To the extent that one takes the syllogism as a benchmark fragment of natural language reasoning, this result is particularly noteworthy in that bivalence is often one of the very first axioms postulated about logic, even when one is talking about the logic of natural language. One of the wider implications of this thesis, however, is that one can get a surprisingly long way towards natural langauge reasoning, even without assuming bivalence. In particular: This chapter has shown that one gets at least as far as the syllogism. In the next chapter (chapter 4), we will then see why the syllogism, in turn, gets us surprisingly far towards the goal of full-scale natural langauge reasoning.

4. Semantic Decomposition

In this chapter, we will address the topic of semantic representation and of how we can arrive from pieces of natural language text at logical formulae that are suitable for inference purposes. Here, we distinguish semantic composition from semantic decomposition, *semantic composition* being the problem of arriving at a semantic representation structure, given a piece of text, and *semantic decomposition* being that of arriving at a logical formula suitable for inference purposes, given a semantic representation structure.

To that end, we first need to establish what we mean by representation and then what it is that makes a representation for text a semantic representation. We think of a *representation scheme* as either a formal language the formal semantics of which can stand in within an inference mechanism for the true semantics of the natural language of interest or of the domain of an algebra which supports relevant logical operations on text.

This notion of representation is commonplace throughout the literature on semantics, but it usually involves an a-priori commitment to a particular logic. Noteworthy wide-coverage implementations of semantic composition include that of Bos (2005) in a CCG grammar, based on DRT (Kamp & Reyle 1993), and that of Dalrymple et al. (1993) within LFG, based on Glue Semantics (see e.g. Lev 2007). Both DRT and Glue Semantics are meta-level formalisms which are grounded in FOPC as an object-level formalism so that the problem of decomposition is solved relatively trivially.

Herein, we will instead build on the MRS algebra for semantic composition (Copestake et al. 2001, 2005, Copestake 2007) and on the general principles underlying the implementations of MRS-based semantic composition in the ERG grammar (Flickinger 2000) and similar grammars. This line of work fits particularly well with our empiricist-relationalist viewpoint as outlined in the introduction (1), precisely because it avoids to a greater extent such a-priori commitments to a particular logic. This makes it possible to address the problem of formulating a logic which is suitable for drawing inferences with text, without committing the fallacy of begging the question. The only commitment we accept a-priori is the methodological need to fit our model of semantics to a particular set of inferences we wish to make. – Semantic decomposition thus becomes a problem which is nontrivial and distinct from that of semantic composition, and it will be the prime focus of the present chapter.

But, given this generalized viewpoint, the notion of *semantic representation* seems to become a pleonasm, as every form of representation of text has some formal semantics, and every theory of the semantics of text, either explicitly or implicitly, avails itself of a form of representation. For our purposes, the notion of semantics serves purely as a distinction from syntax. Syntactic representation arises specifically from the need to draw inferences about the grammaticality of a piece of text. Semantic representation, on the other hand, aims at drawing inferences about the meaning of text. So, when we talk about semantic composition and decomposition, we are really talking about how the syntactic and the semantic criteria for drawing inferences about text differ from and relate to each other.

One notion which is instructive concerning this distinction is that the logic for reasoning about grammaticality and the logic for reasoning about meaning may really be the same. We will take the recent thesis by MacCartney (2009) as prototypical for this idea, but it is, in fact, commonplace throughout the field of natural language processing. Although it often comes under the heading of natural logic, we prefer the more specific reference to what we will call syntactically driven substitution logic (SynSL), as we will contrast this syntactic approach with a semantically driven substitution logic (SemSL), showing that there are inference phenomena which SemSL can adequately handle which SynSL cannot. But, we will then, in turn, have to discard the working hypothesis of SemSL in favour of a more expressive fragment of predicate calculus: the syllogism.

This argument goes into the heart of the question why one would want to do semantic composition and decomposition in the first place: Predicate calculus leads down the road of what has traditionally fallen under the heading of computational semantics and semantic composition in particular. SynSL, however, describes approaches to textual inference which avoid any such explicit treatment of semantics, but which do lead to an implied theory of semantics which is, as we argue, inadequate. In particular, this includes inference engines working with rewrite-patterns over syntax trees or the Carroll-Briscoe dependency structures (Carroll et al. 1999) produced by many parsers.

Once we arrive at the syllogism, we will stop there to consider some of its properties which seem linguistically interesting. In particular, we will demonstrate how syllogistic premises impose dependency structures on text that fulfill the metatheoretical principles of grammar outlined by Harris (1982, 1991). These dependency structures differ from Briscoe-Carroll style dependencies, which sheds some more light on the nature of the syntax/semantics interface we have in mind.

4.1. The ProtoForm Language

In this section, we will introduce the new ProtoForm representation language for semantic structures. Given the proliferation of different semantic representation schemes over the past decade or two, the question arises whether it is really necessary to introduce yet another. In response to this question, we emphasize that the ProtoForm language is heavily inspired by, closely related to, and fully compatible with previous work on MRS. In 4.2, we will show that the MRS algebra for semantic composition can, in fact, be directly applied to the problem of semantic composition with ProtoForms. Besides enforcing this theoretical relationship, I have also implemented an algorithm which is linear in memory and runtime complexity to translate MRS structures as produced by the ERG into ProtoForms.[1] Thus, the existence of the ProtoForm language is not confined to the drawing board. It takes full advantage of the experience in broad-coverage grammar engineering which has gone into the ERG and related HPSG grammars, and there exists a mature software platform for future experimentation with ProtoForms.

Given the current state of the art in MRS-based semantics, the ProtoForm formalism makes two important contributions: (1) It facilitates the decomposition operation outlined in section 4.4. (2) It makes explicit a number of theoretical properties which are only implicit within current implementations of MRS-based composition in particular grammars such as the ERG, but which have not been explicitly mentioned in the literature previously.

The ProtoForm language has disadvantages relative to the MRS language when it comes to the applications which have traditionally been the domain of MRS: (1) In the MRS language, composition operators can be directly implemented in typed feature structures, which is useful for grammar engineering in certain frameworks. (2) MRS has been designed so as to maximize the level of canonicity it imposes on the representation of semantic structures, which is useful for generation and transfer-based machine translation. – These two features are not shared by ProtoForms but irrelevant for purposes of semantic decomposition.

One particular problem within semantic decomposition is that of solving the scope underspecification problem which would necessarily be involved when trying to translate an MRS-style structure into a formula in FOPC. This problem of scoping MRS structures has been addressed in a series of publications by Koller et al. (2009). This work, however, has further-reaching implications than those applying to the problem of scope enumeration. We will argue that the recursive structure they impose on an MRS-style representation is central to semantic decomposition in general. The language of Proto-

[1] It is available freely as part of PyPES, the Python Platform for Experimentation with Semantics at http://www.semantilog.org/pypes.html

Forms can be seen as an MRS-style representation language which allows for making this kind of recursive structure explicit.

The notions of semantic composition and decomposition now take on a more specific meaning: Semantic composition within the algebra of MRS is all about removing syntactic recursion from the semantic representation. Decomposition within the language of ProtoForms, on the other hand, is about making explicit a different kind of recursion: Here we are interested in the semantic recursion that supports inferences, rather than the syntactic recursion which arises during composition due to the idiosyncrasies of the particular language and the structure imposed on it by the grammar. Although the two are clearly related to each other, they are nevertheless distinct as we will see throughout the rest of this chapter.

4.1.1. ProtoForm Concepts

Different Kinds of Subforms

Predications

The simplest kind of subform which can appear in a ProtoForm is a predication. Our notation for predications differs from standard FOPC notation in that we use explicitly named labels to identify arguments, rather than identifying arguments by their position in a sequence. We would write, for example

$$|\mathsf{gave}|\,\big(\,\mathsf{KEY} = e_0,\ \mathsf{arg1} = x_1,\ \mathsf{arg2} = x_2,\ \mathsf{arg3} = x_3\,\big),$$

and this would mean the same as

$$|\mathsf{gave}|\,\big(\,\mathsf{arg3} = x_3,\ \mathsf{arg2} = x_2,\ \mathsf{arg1} = x_1,\ \mathsf{KEY} = e_0\,\big).$$

We use the notation $|\cdot|$ to remind ourselves of the fact that a word is uninterpreted and may need some kind of logical interpretation in order to be applicable within a logical inference mechanism.

Quantifications

This is another kind of subform:

$$|\mathsf{the}|\,(x_1)\Big[\,|\mathsf{representative}|\,\big(\,\mathsf{KEY} = x_1\,\big)\Big]\Big[\,|\mathsf{arrived}|\,\big(\,\mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1\,\big)\Big].$$

It might appear in a semantic representation for 'The representative arrived'. The quantification binds a single variable (in this case x_1), and it has two scopal arguments: a restrictor and a scope. It can be easily seen how this notation corresponds to the notation for quantifiers we used previously (section 3.2.1), with the restrictor appearing as the left-hand scopal argument, and the body appearing as the right-hand scopal argument. In this case, we have directly plugged a ProtoForm into the restrictor and another ProtoForm into the body of this quantfication, but we will later on see, that ProtoForms also support a number of placeholder structures which can go into such scopal argument slots. A ProtoForm is always written in brackets [·]. In this case, each of the subordinate ProtoForms itself contains one subform. In our example these are predications. The quantifier itself, in the above example, is a word. But for purposes of logical inferences it is useful to be able to substitute a logical operator for an uninterpreted word. For example, we might write:

$$\exists\,(x_1) \Big[\lvert\mathsf{representative}\rvert\,(\,\mathsf{KEY} = x_1\,) \Big] \Big[\lvert\mathsf{arrive}\rvert\,(\,\mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1\,) \Big],$$

where the logical operator '∃' has replaced the word |the|.

Connections

Now consider

$$\begin{array}{l} \lvert\mathsf{the}\rvert\,(x_1) \Big[\Big[\lvert\mathsf{new}\rvert\,(\,\mathsf{arg1} = x_1\,) \Big] \ \& \ \Big[\lvert\mathsf{representative}\rvert\,(\,\mathsf{KEY} = x_1\,) \Big] \Big] \\ \quad \Big[\lvert\mathsf{arrived}\rvert\,(\,\mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1\,) \Big] \end{array},$$

which might appear in a semantic representation for 'The new representative arrived'. The subform which appears in the restrictor of the quantifier is a connection. These connections are always denoted in infix notation. As before, we could have used a word instead of a logical operator here, and write |and|, for example when representing the phrase 'new and inexperienced' or 'arrived and checked in'. A connection has two scopal arguments, which we simply call the lefthand scope and the righthand scope.

Modalifications

Finally, words like |said| give rise to a type of subform which, for lack of a better term, we will call a modalification (modal modification). The subform

$$\begin{array}{l} \lvert\mathsf{the}\rvert\,(x_1) \Big[\lvert\mathsf{representative}\rvert\,(\,\mathsf{KEY} = x_1\,) \Big] \\ \quad \Big[\lvert\mathsf{said}\rvert\,(\,\mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1\,) \Big[\lvert\mathsf{rained}\rvert\,(\,\mathsf{KEY} = e_2,\ \,) \Big] \Big] \end{array}$$

might appear in a representation for 'The representative said it rained', but modalifications need not have predicate-style arguments, as in 'It possibly rained', or they might have more than one, as in 'Smith bet Jones a dime that it rained'. In any case, however, a modalification has exactly one scopal argument, which we call the scope of the modalification.

Representing Scope

The ProtoForm language offers a number of different ways for representing scope. Depending on which mechanisms are used, we speak of a ProtoForm as either a type-I, type-II, or type-III ProtoForm. In what follows, we will give examples of each type of ProtoForm and the associated mechanisms for representing scope.

Type-I ProtoForms

All examples we have previously given are type-I ProtoForms, as they represent scope only by plugging ProtoForms into scopal arguments directly. They use none of the mechanisms for representing scope ambiguity.

Type-II ProtoForms

The following ProtoForm imposes a less strict kind of recursion:

$$\left[\begin{array}{l} |\mathsf{the}|_{(x_1)} \left[|\mathsf{representative}| \left(\mathsf{KEY} = x_1 \right) \right] \_\_, \\ |\mathsf{arrived}| \left(\mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1 \right) \end{array}\right],$$

Here, the ProtoForm consists of two subforms, and the body of the quantifier is marked by what we call an anonymous hole, denoted ‘__’. This is what makes this example a type-II ProtoForm: type-I ProtoForms may not use holes of any kind. The idea behind our use of holes is the same as the holes in Hole Semantics (Bos 1996; also see Blackburn & Bos 2005, Koller et al. 2003). Anonymous holes are a particularly simple special case of this idea: A type-II ProtoForm simply has N anonymous holes and $N + 1$ subforms, where any subform can be plugged into any anonymous hole. When each hole has been filled in this manner, we get a type-I ProtoForm which we call a configuration of the type-II ProtoForm. One can see how this can be used to model scope ambiguity. For example

$$\left[\begin{array}{l} |\mathsf{every}|_{(x_1)} \left[|\mathsf{representative}| \left(\mathsf{KEY} = x_1 \right) \right] \_\_, \\ |\mathsf{a}|_{(x_1)} \left[|\mathsf{sample}| \left(\mathsf{KEY} = x_2 \right) \right] \_\_, \\ |\mathsf{saw}| \left(\mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1,\ \mathsf{arg2} = x_1 \right) \end{array}\right],$$

has two configurations:

$$\left[\begin{array}{l} |\mathsf{every}|_{(x_1)} \left[|\mathsf{representative}| \left(\mathsf{KEY} = x_1 \right) \right] \\ \quad \left[\begin{array}{l} |\mathsf{a}|_{(x_2)} \left[|\mathsf{sample}| \left(\mathsf{KEY} = x_2 \right) \right] \\ \quad \left[|\mathsf{saw}| \left(\mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1,\ \mathsf{arg2} = x_1 \right) \right] \end{array}\right] \end{array}\right],$$

$$\left[\begin{array}{l} |\mathsf{a}|_{(x_2)} \left[|\mathsf{sample}| \left(\mathsf{KEY} = x_2 \right) \right] \\ \quad \left[\begin{array}{l} |\mathsf{every}|_{(x_1)} \left[|\mathsf{representative}| \left(\mathsf{KEY} = x_1 \right) \right] \\ \quad \left[|\mathsf{saw}| \left(\mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1,\ \mathsf{arg2} = x_1 \right) \right] \end{array}\right] \end{array}\right].$$

We can use the same mechanism not only for genuine scope ambiguity, but also for defining a canonical representation for connections which involve commutative operators, such as conjunction:

$$\left[\begin{array}{l} |\text{the}|_{(x_1)} \left[\begin{array}{l} |\text{new}|\,(\,\text{arg1} = x_1\,), \\ \text{--}\ \&\ \text{--}, \\ |\text{representative}|\,(\,\text{KEY} = x_1\,) \end{array}\right] \text{--}, \\ |\text{arrived}|\,(\,\text{KEY} = e_1,\ \text{arg1} = x_1\,) \end{array}\right].$$

Note that a subform can only fill such holes which we call active holes within the ProtoForm of which it is a subform. We will introduce the notions of active and passive hole in greater detail later. For type-II ProtoForms, it simply means that subforms cannot float into subordinate ProtoForms, so the subform for 'arrived' above cannot fill a hole in the connection.

Type-III ProtoForms

The most complex and most expressive mechanism for representing scope consists in assigning explicit names to handles, in order to refer to handles in a constraint language which governs how subforms may or may not be plugged into each other. This is the idea behind underspecification in Hole Semantics, more generally. A ProtoForm which uses only this kind of mechanism, and never plugs a ProtoForm into a scopal argument directly, is what we call a type-III ProtoForm. The two examples we have previously encountered can be represented using type-III ProtoForms as follows:

$$\left[\begin{array}{l} \boxed{1}\,|\text{the}|_{(x_1)}\ \boxed{2}\ \text{--}, \\ \boxed{3}\,|\text{representative}|\,(\,\text{KEY} = x_1\,), \\ \boxed{4}\,|\text{arrived}|\,(\,\text{KEY} = e_1,\ \text{arg1} = x_1\,), \\ \qquad \boxed{1} < \boxed{4}, \\ \qquad \boxed{2} < \boxed{3} \end{array}\right], \left[\begin{array}{l} \quad\ |\text{every}|_{(x_2)}\ \boxed{1}\ \text{--}, \\ \boxed{2}\,|\text{representative}|\,(\,\text{KEY} = x_1\,), \\ \quad\ |\text{saw}|\,(\,\text{KEY} = e_1,\ \text{arg1} = x_1,\ \text{arg2} = x_1\,), \\ \quad\ |\text{a}|_{(x_1)}\ \boxed{3}\ \text{--}, \\ \boxed{4}\,|\text{sample}|\,(\,\text{KEY} = x_2\,), \\ \qquad \boxed{1} < \boxed{2}, \\ \qquad \boxed{3} < \boxed{4} \end{array}\right].$$

Such type-III ProtoForms now bear a resemblance to MRS structures (Copestake et al. 2005; also see Niehren & Thater 2003, Fuchss et al. 2004). In the first of the above examples, all three subforms are labelled using what we call named roots ($\boxed{1}$, $\boxed{3}$, $\boxed{4}$), and the restrictor of the quantifier carries what we call a named hole ($\boxed{2}$). The problem of determining a configuration, again, consists in plugging subforms into holes, but this time we also specify constraints that the resulting tree structure must fulfill.

For example, the constraint $\boxed{2} < \boxed{3}$ specifies that root $\boxed{3}$ must be either a direct or indirect descendant of hole $\boxed{2}$. This constraint would have to be explicitly inserted during semantic composition.

The constraint $\boxed{1} < \boxed{4}$ specifies that root $\boxed{4}$ must be a direct or indirect descendant of root $\boxed{1}$. This latter constraint is an example of an implicit binding constraint (Niehren & Thater 2003, Fuchss et al. 2004). It arises from the fact that the quantifier which binds a variable must occur on an outside scope of a predication which refers to that variable. This is a simple well-formedness condition in FOPC, so the constraint can be omitted during semantic composition and inserted later. We will generally not write out the implicit binding constraints, but simply assume that they are always implicit. The second of the above examples uses that convention.

The use of constraints further improves the expressive power of the formalism. For example, consider the following structure:

$$\begin{bmatrix} & |\mathsf{every}|\,(x_1)\ \boxed{1}\ \text{--}, \\ \boxed{2} & \begin{bmatrix} |\mathsf{representative}|\,(\,\textsf{KEY} = x_1\,), \\ \text{--}\ \&\ \text{--}, \\ |\mathsf{of}|\,(\,\mathsf{arg1} = x_1,\ \mathsf{arg2} = x_2\,) \end{bmatrix}, \\ & |\mathsf{a}|\,(x_2)\ \boxed{3}\ \text{--}, \\ \boxed{4} & |\mathsf{company}|\,(\,\textsf{KEY} = x_2\,), \\ & |\mathsf{saw}|\,(\,\textsf{KEY} = e_1,\ \mathsf{arg1} = x_1,\ \mathsf{arg2} = x_3\,), \\ & |\mathsf{a}|\,(x_3)\ \boxed{5}\ \text{--}, \\ \boxed{6} & |\mathsf{sample}|\,(\,\textsf{KEY} = x_3\,), \\ & \quad \boxed{1} < \boxed{2}, \\ & \quad \boxed{3} < \boxed{4}, \\ & \quad \boxed{5} < \boxed{6} \end{bmatrix}.$$

Here, the restrictor of |every| must be a hole due to the fact that either 'a company' or 'representative of' can fill this slot, and the body of |every| must be a hole too, due to the fact that either 'saw' or 'a sample' can fill this slot. But they cannot both be anonymous holes, since this would mean that everything that can fill the restrictor of |every| could also fill its body and vice-versa, which is clearly not the case. So the use of named holes with constraints is required.

Minimally, Maximally, & Fully Recursive ProtoForms

Previously, we have seen three types of semantic representations, as exemplified by (1) The representative arrived, (2) Every representative saw a sample, (3) Every representative of a company saw a sample. Example (1) has a direct representation as a fully recursive ProtoForm, previously called a type-I ProtoForm. This is a ProtoForm, which plugs ProtoForms directly into scopal arguments, but does not use holes or constraints. Examples (2) and (3) have no such representations, due to scope ambiguity. Examples (1) and (2) also have a direct representation as a type-II ProtoForm. This kind of ProtoForm plugs ProtoForms into scopal arguments wherever possible, and uses anonymous holes to represent scope ambiguities, but no named holes or roots, and no constraints. A type-II ProtoForm thus specifies a set of type-I ProtoForms, which we call its configurations. Due to the nature of its scope ambiguity, example (3) has no representation either as a type-I or a type-II ProtoForm. However, examples (1), (2), and (3) all have a minimally recursive, or type-III ProtoForm as a representation. This kind of ProtoForm never fills scopal arguments with ProtoForms directly, and instead uses named holes and explicit constraints to represent scope ambiguities. As before, a type-III ProtoForm specifies a set of type-I ProtoForms, which we call its configurations.

The notion of a type-II ProtoForm was used in this section only for instructive purposes. Throughout the rest of this work, it will be more useful to talk about fully recursive, maximally recursive, and minimally recursive ProtoForms. When writing a maximally recursive ProtoForm, we plug ProtoForms into scopal arguments directly wherever possible, and prefer the use of anonymous holes over the use of named holes with explicit constraints. Given these preferences we will, however, still need to satisfy the goal of adequately representing scope ambiguity by the use of named holes and constraints where this is necessary.

Consider the following example, which combines the use of the three types of techniques for representing scoping in a maximally recursive ProtoForm:

$$\left[\begin{array}{l}
|\textsf{every}|\,{}_{(x_1)}\,\boxed{1}\,\_\_,\\
\boxed{2}\left[\begin{array}{l} |\textsf{representative}|\,(\,\textsf{KEY}=x_1\,),\\ \_\_\;\&\;\_\_,\\ |\textsf{of}|\,(\,\textsf{arg1}=x_1,\ \textsf{arg2}=x_2\,)\end{array}\right],\\
|\textsf{a}|\,{}_{(x_2)}\left[\,|\textsf{company}|\,(\,\textsf{KEY}=x_2\,)\right]\_\_,\\
|\textsf{saw}|\,(\,\textsf{KEY}=e_1,\ \textsf{arg1}=x_1,\ \textsf{arg2}=x_3\,),\\
|\textsf{a}|\,{}_{(x_3)}\left[\,|\textsf{sample}|\,(\,\textsf{KEY}=x_3\,)\right]\_\_,\\
\quad\boxed{1}\prec\boxed{2}
\end{array}\right].$$

Maximally recursive ProtoForms are of central importance to us, as they define the notion of semantic head, which we will introduce in section 4.3.2. These semantic heads, in turn, form the operators in our characterization of operator grammar (section 4.5).

I have implemented an algorithm to bring minimally recursive ProtoForms into an equivalent maximally recursive form. This algorithm does not involve exhaustive scope enumeration. Rather, the invariant pluggings can be read directly off the packed representation produced by a scoping machinery such as that of Koller et al. (2009).

Active and Passive Holes

One final complication which arises with ProtoForms is illustrated by the example 'Every organizer who knew that a representative protested apologized'. If we were forced to represent this in a minimally recursive way, we would use the following ProtoForm:

$$\left[\begin{array}{l}
\quad |\text{every}|\,(x_1)\ \boxed{1}\ \text{--},\\
\boxed{2}\left[\begin{array}{l} |\text{organizer}|\,(\text{KEY} = x_1),\\ \text{--}\ \&\ \text{--},\\ |\text{know}|\,(\text{KEY} = e_1,\ \text{arg1} = x_1)\ {}^{\ulcorner}_{\llcorner}\boxed{3}{}^{\urcorner}_{\lrcorner}\end{array}\right],\\
\quad |\text{a}|\,(x_2)\ \boxed{5}\ \text{--},\\
\boxed{6}\,|\text{representative}|\,(\text{KEY} = x_2),\\
\boxed{4}\,|\text{protested}|\,(\text{KEY} = e_2,\ \text{arg1} = x_2),\\
\quad |\text{apologized}|\,(\text{KEY} = e_3,\ \text{arg1} = x_1),\\
\qquad \boxed{1} < \boxed{2},\\
\qquad \boxed{3} < \boxed{4},\\
\qquad \boxed{5} < \boxed{6}
\end{array}\right],$$

which has a number of configurations, among them the following two:

$$\left[\begin{array}{l}
|\text{every}|\,(x_1)\left[\begin{array}{l}
|\text{a}|\,(x_2)\left[\,|\text{representative}|\,(\text{KEY} = x_2)\right]\\
\quad\left[\begin{array}{l} |\text{organizer}|\,(\text{KEY} = x_1),\\ \text{--}\ \&\ \text{--},\\ |\text{knew}|\,(\text{KEY} = e_1,\ \text{arg1} = x_1)\left[\,|\text{protested}|\,(\text{KEY} = e_2,\ \text{arg1} = x_2)\right]\end{array}\right]
\end{array}\right]\\
\quad |\text{apologized}|\,(\text{KEY} = e_3,\ \text{arg1} = x_1)
\end{array}\right],$$

$$\left[\begin{array}{l}
|\text{a}|\,(x_2)\left[\,|\text{representative}|\,(\text{KEY} = x_2)\right]\\
\quad |\text{every}|\,(x_1)\left[\begin{array}{l} |\text{organizer}|\,(\text{KEY} = x_1),\\ \text{--}\ \&\ \text{--},\\ |\text{knew}|\,(\text{KEY} = e_1,\ \text{arg1} = x_1)\left[\,|\text{protested}|\,(\text{KEY} = e_2,\ \text{arg1} = x_2)\right]\end{array}\right]\\
\qquad |\text{apologized}|\,(\text{KEY} = e_3,\ \text{arg1} = x_1)
\end{array}\right].$$

Contrast this with our previous example 'The new representative arrived', where we said that an outside subform cannot float into a hole in an inside ProtoForm. What we have newly introduced here is the notation ${}^{\ulcorner}_{\llcorner}\boxed{3}{}^{\urcorner}_{\lrcorner}$ which can override this property. Here, hole $\boxed{3}$ is not an active handle in the ProtoForm in which it appears, but rather in the superordinate ProtoForm thereof, i.e. the ProtoForm in which the ProtoForm in which it appears appears as a subform. This notation may also be nested. For example ${}^{\ulcorner\ulcorner}_{\llcorner\llcorner}\boxed{3}{}^{\urcorner\urcorner}_{\lrcorner\lrcorner}$ would be active in the superordinate ProtoForm of the superordinate ProtoForm of the ProtoForm in which it appears.

4.1.2. ProtoForm Definitions

Having given some examples and established some initial intuitions about ProtoForms, we can now go on to define some more notational preliminaries, and then give a full formal definition of ProtoForms.

Preliminaries: Basic Data Structures

75. Let $\mathrm{D}_1, \mathrm{D}_2$ be disjoint sets and let f_1 be a function over domain D_1 and f_2 be a function over domain D_2. Then the *union of functions* f_1 *and* f_2, written $\mathrm{f} = \mathrm{f}_1 \cup \mathrm{f}_2$, is a function over domain $\mathrm{D}_1 \cup \mathrm{D}_2$, which is defined as follows: For all x,

$$\mathrm{f}(x) = \begin{cases} \mathrm{f}_1(x) & \text{if } x \in \mathrm{D}_1, \\ \mathrm{f}_2(x) & \text{if } x \in \mathrm{D}_2. \end{cases}$$

76. Let D, D' be sets with $\mathrm{D}' \subseteq \mathrm{D}$, and let f be a function over domain D. Then the *intersection of function* f *with subdomain* D', written $\mathrm{f}' = \mathrm{f} \cap \mathrm{D}'$, is a function over domain D' where, for all x, we have $\mathrm{f}'(x) = \mathrm{f}(x)$.

77. Let D be some set. We say that S is a *sequence over* D *of length* N, iff S is a function $\{1, 2, \dots, \mathrm{N}\} \mapsto \mathrm{D}$. Furthermore, we define the following.

- We write $\mathrm{S}^{[i]}$ to denote the value $\mathrm{S}(i)$ of this function at i.
- We have $s \in \mathrm{S}$ when there exists an i with $\mathrm{S}^{[i]} = s$.
- We write $\{\mathrm{S}\}$ to denote the set $\{s | s \in \mathrm{S}\}$.
- We write $\mathrm{S}^{[:i]}$ to denote the sequence over D of length i, for $i \leq \mathrm{N}$, which has $\mathrm{S}^{[:i][j]} = \mathrm{S}^{[j]}$ for all j.
- We write $\mathrm{S}^{[i:]}$, for $i \leq \mathrm{N}$, to denote the sequence over D of length $\mathrm{N} - i$ which has $\mathrm{S}^{[i:][j]} = \mathrm{S}^{[i+j]}$ for all j.
- Let S_1 and S_2 be two sequences over D of lengths N_1 and N_2. Let S_2' be the function over domain $\{\mathrm{N}_1 + 1, \mathrm{N}_1 + 2, \dots, \mathrm{N}_1 + \mathrm{N}_2\}$ which has $\mathrm{S}_2'(x) = \mathrm{S}_2(x - \mathrm{N}_1)$. Then, S is the *concatenation* of sequences S_1 and S_2, written $\mathrm{S} = \mathrm{S}_1 * \mathrm{S}_2$, iff $\mathrm{S} = \mathrm{S}_1 \cup \mathrm{S}_2'$ is the union of the two functions S_1 and S_2'.

Signatures

78. A *logical signature* $\mathrm{LSig} = (\mathrm{Funcs}, \mathrm{ArgLabels}, \mathrm{FuncArgs}, \mathrm{ArgValues})$ consists of:

- A set Funcs of *functors*. Each functor $f \in \mathrm{Funcs}$ can be one of the following:
  - a *word* such as |every|, |representative|, or |arrive|, which we identify by its lemma, and write using the notation $|\cdot|$; or

  - an *operator*, which we generally identify using a logical symbol such as ¬, →, ∧, &, ∨, ≡, ≢, ∀, ∄, ∃, ℩. This particular list of operator symbols is specific to the language of our logic (section 3.2.1).[2]

- A set ArgLabels of *argument labels* such as KEY, arg1, arg2, arg3, arg4, arg, carg, l-index, r-index. This particular list of argument labels arises specifically from semantic structures as produced by the ERG but has some normative bearing on related HPSG grammars.

- A relation FuncArgs ⊆ Funcs × ArgLabels, such that $(f, l) \in$ FuncArgs, iff the functor $f \in$ FuncArgs accepts and requires an argument with label $l \in$ L, e.g. $\{(|\text{arrive}|, \text{KEY}), (|\text{arrive}|, \text{arg1}), (|\text{bet}|, \text{KEY}), (|\text{bet}|, \text{arg1}), (|\text{bet}|, \text{arg2}), (|\text{bet}|, \text{arg3})\} \subseteq \text{FuncArgs}$.

- A set V of *argument values*. Each argument value $v \in$ V can be one of the following:
  - a *variable* such as $x_1, x_2, x_3, \ldots$ or $e_1, e_2, e_3, \ldots$
  - a *constant* which we write between slashes, e.g. /Jones/, /San Francisco/, /e_1/.

79. A *proto signature* PSig may either be the *empty signature*, denoted ⊥, or may be of the form PSig = (LSig, Holes, PSig′), consisting of

- a logical signature LSig;
- a set Holes ⊆ $\{1, 2, 3, \ldots\}$ of *holes*;
- a subordinate proto signature PSig′.

ProtoForms and Their Subforms

80. A *ProtoForm*

$$\text{PF} = (\text{Roots}, \text{ActHoles}, \text{Subfs}, \text{Conss})$$

of size N, for some $N \in \mathbb{N}$, *over a proto signature* PSig = (LSig, PassHoles, PSig′) contains:

- a sequence Roots of *roots* of length |{Roots}| over domain $\{1, 2, 3, \ldots\}$ (i.e. a sequence of numbers, where each number occurs only once).
- a set ActHoles ⊆ $\{1, 2, 3, \ldots\}$ of *active holes* with {Roots} ∩ ActHoles = ∅;
- a relation Conss ⊆ ({Roots} ∪ ActHoles) × ActHoles of *constraints*;

[2] In addition, it contains the symbol '℩'. This is an upside-down iota, a piece of notation introduced in the Principia Mathematica alongside the quantifier symbols '∀' and '∃'. It stands for Russell's definite description quantifier, which is outside the scope of our particular logic, but which we will nevertheless permit into our logical language. In the absence of a better model for definite descriptions one may think of it as an existential quantifier. We could have introduced the syllogistic quantifier '∀̸', standing for 'Some X is not Y' or 'It is not the case that all X are Y'. The fact that there does not seem to be a single word in the English language to express this quantifier has led us to omit it.

- a function Subfs, such that, for each $\mathrm{r}_i \in \{\mathrm{Roots}\}$, we have $\mathrm{Subfs}(\mathrm{r}_i) = \mathrm{sf}_i$, where sf_i is a subform over $\mathrm{PSig}_i = (\mathrm{LSig}, \mathrm{H}_i, \mathrm{PSig})$[3], where

$$\{\mathrm{Roots}\} = \{\mathrm{r}_1, \mathrm{r}_2, \ldots, \mathrm{r}_\mathrm{N}\},$$
$$\mathrm{ActHoles} = \mathrm{H}_1 \cup \mathrm{H}_2 \cup \ldots \cup \mathrm{H}_\mathrm{N}.$$

- the following conditions are fulfilled about the size N:
(a) $|\{\mathrm{Roots}\}| = \mathrm{N}$, (b) $|\mathrm{ActHoles}| = \mathrm{N} - 1$.

If PF is a ProtoForm over PSig, then PF is also a *subform over* PSig. Furthermore, we say about every subform sf_i that it is a *subform of* PF. Finally, let us define a notation for ProtoForms: Let the notation for each sf_i be ω_i and let

$$\mathrm{Conss} = \{(\mathrm{u}_1, \mathrm{l}_1), (\mathrm{u}_2, \mathrm{l}_2), \ldots (\mathrm{u}_{\mathrm{N}'}, \mathrm{l}_{\mathrm{N}'})\}.$$

Then the notation for the ProtoForm PF is as follows:

$$\left[\begin{array}{l} \boxed{\mathrm{r}_1}\,\omega_i, \\ \boxed{\mathrm{r}_2}\,\omega_2, \\ \quad \ldots, \\ \boxed{\mathrm{r}_\mathrm{N}}\,\omega_\mathrm{N}, \\ \qquad \boxed{\mathrm{u}_1} < \boxed{\mathrm{l}_1}, \\ \qquad \boxed{\mathrm{u}_2} < \boxed{\mathrm{l}_2}, \\ \qquad \ldots, \\ \qquad \boxed{\mathrm{u}_{\mathrm{N}'}} < \boxed{\mathrm{l}_{\mathrm{N}'}} \end{array}\right]$$

81. Let $\mathrm{PSig} = (\mathrm{LSig}, \mathrm{Holes}, \mathrm{PSig}')$ be a proto signature.

- If $h \in$ Holes, then we say that h is a *scope bearer over* PSig.
Its notation over PSig is $\boxed{h}$.
- Let PF″ be a ProtoForm over PSig″ with $\mathrm{PSig}'' = (\mathrm{LSig}, \mathrm{Holes}'', \mathrm{PSig})$.
Then, PF″ is a *scope bearer over* PSig.
- Consider a scope bearer over PSig′ with notation υ over PSig′. This scope bearer over PSig′ is then also a *scope bearer over* PSig. Its notation over PSig is $\lceil\!\lfloor\upsilon\rfloor\!\rceil$.

82. Let $\mathrm{PSig} = (\mathrm{LSig}, \mathrm{Holes}, \mathrm{PSig}')$ be a proto signature with logical signature $\mathrm{LSig} = (\mathrm{Funcs}, \mathrm{ArgLabels}, \mathrm{FuncArgs}, \mathrm{ArgValues})$. Then, let $\mathrm{f} \in$ Funcs be a functor, $\mathrm{l}_1, \mathrm{l}_2, \ldots \in$ ArgLabels be argument labels with $(\mathrm{f}, \mathrm{l}_i) \in$ FuncArgs for all l_i, and let $\mathrm{v}_1, \mathrm{v}_2, \ldots \in$ ArgValues be argument values. We say that

- ω is a *predication over* LSig, iff ω is of the form

$$\mathrm{f}\,(\,\mathrm{l}_1 = \mathrm{v}_1,\, \mathrm{l}_2 = \mathrm{v}_2,\, \ldots,\, \mathrm{l}_N = \mathrm{v}_N\,) \text{ for some } N;$$

[3] This notion of sf_i being a subform over PSig_i will be defined in due course.

- ω is a *quantification over* PSig, iff ω is of the form

  $$\mathrm{f}\,(v)\,\varphi\,\psi,$$

  where φ and ψ are scope bearers over PSig;
- ω is a *modalification over* PSig, iff ω is of the form

  $$\mathrm{f}\,(\,\mathrm{l}_1 = \mathrm{v}_1,\, \mathrm{l}_2 = \mathrm{v}_2,\, \ldots,\, \mathrm{l}_N = \mathrm{v}_N\,)\,\varphi, \text{ for some } N,$$

  where φ is a scope bearer over PSig;
- and that ω is of the form

  $$\varphi\,\mathrm{f}\,\psi,$$

  where φ and ψ are scope bearers over PSig.

If ω is a predication, quantification, modalification, connection or ProtoForm over PSig, then we also say that ω is also a *subform over* PSig.

ProtoForm Operations

83. Let PSig_1 and PSig_2 be two proto signatures. Their *union* $\mathrm{PSig}_1 \sqcup \mathrm{PSig}_2$ is defined as follows:

- if $\mathrm{PSig}_1 = \bot$, then $\mathrm{PSig}_1 \sqcup \mathrm{PSig}_2 = \mathrm{PSig}_2$;
- if $\mathrm{PSig}_2 = \bot$, then $\mathrm{PSig}_1 \sqcup \mathrm{PSig}_2 = \mathrm{PSig}_1$;
- if $\mathrm{PSig}_1 = (\mathrm{LSig}, \mathrm{Holes}_1, \mathrm{PSig}'_1)$ and $\mathrm{PSig}_2 = (\mathrm{LSig}, \mathrm{Holes}_1, \mathrm{PSig}'_2)$, then

  $$\mathrm{PSig}_1 \sqcup \mathrm{PSig}_2 = (\mathrm{LSig}, \mathrm{Holes}_1 \cup \mathrm{Holes}_2, \mathrm{PSig}'_1 \sqcup \mathrm{PSig}'_2).$$

84. Let $\mathrm{PF}_1 = (\mathrm{Roots}_1, \mathrm{ActHoles}_1, \mathrm{Subfs}_1, \mathrm{Conss}_1)$ be a ProtoForm over PSig_1 and let $\mathrm{PF}_2 = (\mathrm{Roots}_2, \mathrm{ActHoles}_2, \mathrm{Subfs}_2, \mathrm{Conss}_2)$ be a ProtoForm over PSig_2. Their *union* $\mathrm{PF}_1 \sqcup \mathrm{PF}_2$ is a ProtoForm over $\mathrm{PSig}_1 \sqcup \mathrm{PSig}_2$ and is defined as follows:

$$\mathrm{PSig}_1 \sqcup \mathrm{PSig}_2 = (\mathrm{Roots}_1 \cup \mathrm{Roots}_2, \mathrm{ActHoles}_1 \cup \mathrm{ActHoles}_2, \mathrm{Subfs}_1 * \mathrm{Subfs}_2, \mathrm{Conss}_1 \cup \mathrm{Conss}_2).$$

85. Let $\mathrm{PF} = (\mathrm{Roots}, \mathrm{Holes}, \mathrm{Subfs}, \mathrm{Conss})$ be a ProtoForm over PSig, let $\mathrm{r} \in \mathrm{Roots}$ be a root, and let $\mathrm{Subfs}(\mathrm{r})$ be a subform over $\mathrm{PSig}' = (\mathrm{LSig}, \mathrm{Holes}', \mathrm{PSig})$. The *selection* $\mathrm{PF} \sqcap \mathrm{r}$ is the ProtoForm $(\langle \mathrm{r} \rangle, \mathrm{Holes}', \mathrm{Subfs} \cap \{\mathrm{r}\}, \varnothing)$.

86. Let $\mathrm{PF} = (\mathrm{Roots}, \mathrm{Holes}, \mathrm{Subfs}, \mathrm{Conss})$ be a ProtoForm over PSig, let $\mathrm{r} \in \mathrm{Roots}$ be a root, and let i be its index, so that $\mathrm{Roots}^{[\mathrm{i}]} = \mathrm{r}$. Let $\mathrm{Roots}' = \mathrm{Roots}^{[:\mathrm{i}-1]} * \mathrm{Roots}^{[\mathrm{i}:]}$, and let $\mathrm{Subfs}(\mathrm{r})$ be a subform over $\mathrm{PSig}' = (\mathrm{LSig}, \mathrm{Holes}', \mathrm{PSig})$. Then the *deletion* $\mathrm{PF} \setminus \mathrm{r}$ is the ProtoForm $(\mathrm{Roots}', \mathrm{Holes}_1 \setminus \mathrm{Holes}', \mathrm{Subfs}' \cap \{\mathrm{Roots}'\}, \mathrm{Conss})$.

4.2. MRS-Style ProtoForm Composition

Having introduced ProtoForms, we can now address the question of how to derive a ProtoForm as a representation for a piece of text by grammatical composition. Thus, we show, to a proof-of-concept level, that the ProtoForm language is compositionally adequate, in the sense of being able to express the intermediate semantics of arbitrary syntactic constituents arising from a small toy grammar. Furthermore, our toy language will define the sentences we will use as linguistic examples later on, together with their syntax trees and semantic representations.

4.2.1. Grammar

In particular, our toy grammar is as follows:

$$\begin{array}{lll}
\texttt{S} \rightarrowtail \texttt{NP} \rhd \texttt{VP}, & \texttt{VP} \rightarrowtail \texttt{Adv} \lhd \texttt{VP}, & \texttt{PP} \rightarrowtail \texttt{P} \rhd \texttt{NP}, \\
\texttt{NP} \rightarrowtail \texttt{Det} \lhd \texttt{N'}, & \texttt{VP} \rightarrowtail \texttt{V'}, & \texttt{N'} \rightarrowtail \texttt{N'} \vartriangle \texttt{PP}, \\
\texttt{N'} \rightarrowtail \texttt{Adj} \vartriangle \texttt{N'}, & \texttt{V'} \rightarrowtail \texttt{V}, & \texttt{V'} \rightarrowtail \texttt{V'} \vartriangle \texttt{PP}, \\
 & \texttt{V'} \rightarrowtail \texttt{V} \rhd \texttt{NP}. &
\end{array}$$

The framework of this simplistic textbook context-free grammar has been chosen here for illustrative purposes. The implementation of my inference engine uses the HPSG-based ERG grammar instead, which is a typed feature structure grammar implementing MRS-based composition. The resulting MRS structures are converted to ProtoForms. For our purposes, however, it is not necessary to go into the additional complexities which come with the implementation of a broad-coverage grammar. The definitions of the operators $\vartriangle$, $\rhd$ and $\lhd$ as used above will be given shortly. For now, simply note that they give us an algebra which is a special case of the MRS algebra, but which is nevertheless sufficient to introduce the basic ideas about MRS-style semantic composition relevant to us. For a more detailed treatment of the general case, refer to Copestake et al. (2001, 2005), Flickinger (2000).

4.2.2. Composition Structures & Lexicon

Before we can go on to discuss the composition operators, let us first consider an example lexicon which might go with our above example grammar: Entries in the semantic lexicon are composition structures, which we will give a formal definition of shortly. For example, an entry might look like this:

$$\langle \boxed{X} \rangle \ \langle x \rangle \left[\boxed{X} |\mathsf{company}| \left(\ \textsc{key} = x \ \right) \right].$$

For now, we can think of such an entry as akin to a lambda expression of the form

$$\lambda\boxed{X}\lambda x\left[\,\boxed{X}|\mathsf{company}|\,(\,\textsf{KEY} = x\,)\right].$$

Here, the outside lambda expression $\lambda\boxed{X}$ belongs to the meta language, viz. the language of ProtoForms, with lambda variable $\boxed{X}$ providing a handle that ranges over roots and holes. The inside lambda expression λx belongs to the object language, with lambda variable x ranging over entities of the logical model theory. But n.b. that we mention this analogy with the lambda calculus only to establish a preliminary intuition about composition structures. A formal definition of the composition algebra, which does not use the lambda calculus at all, will follow shortly.

The lexicon might look like this:

$$\begin{aligned}
\texttt{every} &= \langle\boxed{X}\rangle\ \langle x\rangle\left[\,|\mathsf{every}|\,{\scriptstyle(x)}\ \boxed{X}\ \text{--}\right], && \texttt{Det} \rightarrowtail \texttt{every},\\
\texttt{a} &= \langle\boxed{X}\rangle\ \langle x\rangle\left[\,|\mathsf{a}|\,{\scriptstyle(x)}\ \boxed{X}\ \text{--}\right], && \texttt{Det} \rightarrowtail \texttt{a},\\
\texttt{representative} &= \langle\boxed{X}\rangle\ \langle x\rangle\left[\,\boxed{X}|\mathsf{representative}|\,(\,\textsf{KEY} = x\,)\right], && \texttt{N} \rightarrowtail \texttt{repr...},\\
\texttt{company} &= \langle\boxed{X}\rangle\ \langle x\rangle\left[\,\boxed{X}|\mathsf{company}|\,(\,\textsf{KEY} = x\,)\right], && \texttt{N} \rightarrowtail \texttt{company},\\
\texttt{sample} &= \langle\boxed{X}\rangle\ \langle x\rangle\left[\,\boxed{X}|\mathsf{sample}|\,(\,\textsf{KEY} = x\,)\right], && \texttt{N} \rightarrowtail \texttt{sample},\\
\texttt{saw} &= \langle\boxed{X}\rangle\ \langle x_2, x_1\rangle\left[\,\boxed{X}|\mathsf{see}|\,(\,\mathsf{arg1} = x_1,\ \mathsf{arg2} = x_2\,)\right], && \texttt{V} \rightarrowtail \texttt{see},\\
\texttt{large} &= \langle\boxed{X}\rangle\ \langle x\rangle\left[\,\boxed{X}|\mathsf{large}|\,(\,\mathsf{arg1} = x\,)\right], && \texttt{Adj} \rightarrowtail \texttt{large},\\
\texttt{of} &= \langle\boxed{X}\rangle\ \langle x_2, x_1\rangle\left[\,\boxed{X}|\mathsf{of}|\,(\,\mathsf{arg1} = x_1,\ \mathsf{arg2} = x_2\,)\right], && \texttt{P} \rightarrowtail \texttt{of},\\
\texttt{probably} &= \langle\boxed{X}\rangle\ \langle\rangle\left[\,|\mathsf{probably}|\,(\ )\ \boxed{X}\right], && \texttt{Adv} \rightarrowtail \texttt{probably}.
\end{aligned}$$

Here, we have, for each word, a syntactic production rule and an entry for a semantic lexicon, which is what we call a composition structure. One element of the composition structure is a ProtoForm which represents the lexical entry.

87. A *composition structure* CS = (Handles, Vars) over ProtoForm

$$\mathrm{PF} = (\mathrm{Roots}, \mathrm{ActHoles}, \mathrm{Subfs}, \mathrm{Conss})$$

over PSig = (LSig, PassHoles, PSig′) with LSig = (Funcs, ArgLabels, FuncArgs, ArgValues) consists of:

- a sequence

  $$\mathrm{Vals} = \langle v_1, v_2, \ldots, v_N\rangle$$

  for some N with all $v_i \in$ ArgValues;
- a sequence Handles which may either be empty, so that Handles = ⟨⟩, or which may contain one handle Handles = ⟨Handle⟩ which is either an active hole Handle ∈ ActiveHoles or a root Handle ∈ Roots in PF.

4.2.3. Composition Operators & Derivations

By parsing a piece of text using the above context-free grammar, we can impose on it a structure such as the following:

$$(\mathsf{Every} \lhd (\mathsf{representative} \vartriangle (\mathsf{of} \rhd (\mathsf{a} \lhd (\mathsf{large} \vartriangle \mathsf{company}))))) \vartriangle (\mathsf{saw} \rhd (\mathsf{a} \lhd \mathsf{sample})).$$

We have defined a composition structure which provides a semantic representation for each of these lexical items. Now, all that remains to be done is to define the composition operators $\lhd$, $\vartriangle$ and $\rhd$ as operators on composition structures.

88. Let $\mathrm{CS}_1 = (\mathrm{Handles}_1, \mathrm{Vars}_1)$, $\mathrm{CS}_2 = (\mathrm{Handles}_2, \mathrm{Vars}_2)$, and $\mathrm{CS} = (\mathrm{Handles}, \mathrm{Vars})$, be composition structures over $\mathrm{PF}_1 = (\mathrm{Roots}_1, \mathrm{ActHoles}_1, \mathrm{Subfs}_1, \mathrm{Conss}_1)$, $\mathrm{PF}_2 = (\mathrm{Roots}_2, \mathrm{ActHoles}_2, \mathrm{Subfs}_2, \mathrm{Conss}_2)$, and $\mathrm{PF} = (\mathrm{Roots}, \mathrm{ActHoles}, \mathrm{Subfs}, \mathrm{Conss})$ where PF_1, PF_2 are ProtoForms over proto signatures $\mathrm{PSig}_1 = (\mathrm{LSig}, \mathrm{PassHoles}_1, \mathrm{PSig}'_1)$, $\mathrm{PSig}_2 = (\mathrm{LSig}, \mathrm{PassHoles}_2, \mathrm{PSig}'_2)$, and PSig respectively. Then, if one of the following three sets of conditions is fulfilled, we say that CS is a *composition* of CS_1 and CS_2.

a. CS is the *intersectively coordinating composition* of CS_1 and CS_2, written $\mathrm{CS} = \mathrm{CS}_1 \vartriangle \mathrm{CS}_2$, iff the following holds:

- Let $\mathrm{R} \in \{\mathrm{Roots}\}$, and $\langle \mathrm{H}_1 \rangle = \mathrm{Handles}_1$, and $\langle \mathrm{H}_2 \rangle = \mathrm{Handles}_2$;
- let ConnPF be the ProtoForm which has the notation $\mathrm{ConnPF} = \Big[\, \_ \;\&\; \_ \,\Big]$;
- let $\mathrm{RootPF}' = (\mathrm{PF}_1 \smallsetminus \mathrm{H}_1) \sqcup \mathrm{ConnPF} \sqcup (\mathrm{PF}_2 \smallsetminus \mathrm{H}_2)$;
- let the notation of RootPF' be φ;
- let RootPF be the ProtoForm which has the notation $\mathrm{RootPF} = \Big[\boxed{\mathrm{R}}\,\varphi\Big]$;
- then $\mathrm{CS} = \mathrm{CS}_1 \vartriangle \mathrm{CS}_2$ iff:
  - $\mathrm{PF} = \mathrm{RootPF} \sqcup (\mathrm{PF}_1 \smallsetminus \mathrm{H}_1) \sqcup (\mathrm{PF}_2 \smallsetminus \mathrm{H}_2)$,
  - $\mathrm{Handles} = \langle \mathrm{R} \rangle$, and
  - $\mathrm{Vars}_1 = \mathrm{Vars}_2 = \mathrm{Vars}$.

b. CS is the *intersectively complementizing composition* of CS_1 and CS_2, written $\mathrm{CS} = \mathrm{CS}_1 \rhd \mathrm{CS}_2$, iff:

- $\mathrm{PF} = \mathrm{PF}_1 \sqcup \mathrm{PF}_2$,
- $\mathrm{Handles} = \mathrm{Handles}_1$,
- $\mathrm{Vars}_2 = \langle \mathrm{Vars}_1^{[1]} \rangle$, and $\mathrm{Vars} = \mathrm{Vars}_1^{[2:]}$.

c. CS is the *scopally subordinating composition* of CS_1 and CS_2, written $\mathrm{CS} = \mathrm{CS}_1 \lhd \mathrm{CS}_2$, iff the following holds:

- Let $\langle \mathrm{H}_1 \rangle = \mathrm{Handles}_1$, and $\langle \mathrm{H}_2 \rangle = \mathrm{Handles}_2$;
- let $\mathrm{ConsPF} = \Big[\, \boxed{\mathrm{H}_1} < \boxed{\mathrm{H}_2} \,\Big]$;
- then $\mathrm{CS} = \mathrm{CS}_1 \lhd \mathrm{CS}_2$ iff:

- $\mathrm{PF} = \mathrm{PF}_1 \sqcup \mathrm{PF}_2 \sqcup \mathrm{ConsPF}$,
- $\mathrm{Handles} = \langle\rangle$,
- $\mathrm{Vars}_1 = \mathrm{Vars}_2 = \mathrm{Vars}$.

$\ast$**89.** *Let* $\mathrm{CS}_1 = (\mathrm{Handles}_1, \mathrm{Vars}_1)$, $\mathrm{CS}_2 = (\mathrm{Handles}_2, \mathrm{Vars}_2)$, *and* $\mathrm{CS} = (\mathrm{Handles}, \mathrm{Vars})$, *be composition structures over* $\mathrm{PF}_1 = (\mathrm{Roots}_1, \mathrm{ActHoles}_1, \mathrm{Subfs}_1, \mathrm{Conss}_1)$, $\mathrm{PF}_2 = (\mathrm{Roots}_2, \mathrm{ActHoles}_2, \mathrm{Subfs}_2, \mathrm{Conss}_2)$, *and* $\mathrm{PF} = (\mathrm{Roots}, \mathrm{ActHoles}, \mathrm{Subfs}, \mathrm{Conss})$ *where* PF_1, PF_2 *are ProtoForms over proto signatures* PSig_1, PSig_2, *and* PSig *respectively. Then, if* CS *is a composition of* CS_1 *and* CS_2*, the following conditions are always fulfilled:*

- $\mathrm{Roots} = \mathrm{Roots}_1 \cup \mathrm{Roots}_2$,
- $\mathrm{ActHoles} = \mathrm{ActHoles}_1 \cup \mathrm{ActHoles}_2$,
- $\mathrm{Conss} = \mathrm{Conss}_1 \cup \mathrm{Conss}_2$,
- $\mathrm{PSig} = \mathrm{PSig}_1 \sqcup \mathrm{PSig}_2$.

Proof. Trivial. □

These definitions are demonstrated in Figure 4.1, which traces a semantic composition.

4.3. Substitution Logic

The previous two sections have been dedicated to ProtoForms and ProtoForm composition in an MRS algebra. As we will later see, this approach to semantic representation is sufficient in order to support the type of natural language reasoning we will discuss in the next chapter (chapter 5). – But before we go on to subscribe to this approach and all of the complexities it entails, let us first consider the question of whether it is in fact necessary, as well as being sufficient, or whether a simpler model might do.

This section will, in particular, consider the more simple approach of reasoning with substitutions over syntactic structures. We will see various inadequacies of this approach and move on to the idea of reasoning with substitutions over semantic structures, and then finally to reasoning with quantified structures in a predicate calculus. At each step of the way, we will show why the working hypothesis is inadequate and construct examples where it fails before we move on to the next.

4.3.1. Syntactically Driven Substitution Logic

Applying to our example grammar the idea of syntactic monotonicity composition, which one often finds in connection with treatments of natural logic, we would write the gram-

$$
\begin{aligned}
\mathsf{large} &= \langle \boxed{1} \rangle \; \langle x_1 \rangle \left[\boxed{1} |\mathsf{large}| \,(\, \mathsf{arg1} = x_1 \,) \right]; \\
\mathsf{company} &= \langle \boxed{2} \rangle \; \langle x_1 \rangle \left[\boxed{2} |\mathsf{company}| \,(\, \mathsf{KEY} = x_1 \,) \right]; \\
\mathsf{large} \vartriangle \mathsf{company} &= \langle \boxed{3} \rangle \; \langle x_1 \rangle \left[\boxed{3} \begin{bmatrix} |\mathsf{large}| \,(\, \mathsf{arg1} = x_1 \,), \\ \text{--} \,\&\, \text{--}, \\ |\mathsf{company}| \,(\, \mathsf{KEY} = x_1 \,) \end{bmatrix} \right]; \\
\mathsf{a} &= \langle \boxed{4} \rangle \; \langle x_1 \rangle \left[|\mathsf{a}|_{(x_1)} \, \boxed{4} \, \text{--} \right]; \\
\mathsf{a} \lhd (\mathsf{large} \vartriangle \mathsf{company}) &= \langle \rangle \; \langle x_1 \rangle \begin{bmatrix} |\mathsf{a}|_{(x_1)} \, \boxed{4} \, \text{--}, \\ \boxed{3} \begin{bmatrix} |\mathsf{large}| \,(\, \mathsf{arg1} = x_1 \,), \\ \text{--} \,\&\, \text{--}, \\ |\mathsf{company}| \,(\, \mathsf{KEY} = x_1 \,) \end{bmatrix}, \\ \boxed{4} \prec \boxed{3} \end{bmatrix}; \\
\mathsf{of} &= \langle \boxed{5} \rangle \; \langle x_1, x_2 \rangle \left[\boxed{5} |\mathsf{of}| \,(\, \mathsf{arg1} = x_2, \; \mathsf{arg2} = x_1 \,) \right]; \\
\mathsf{of} \rhd (\mathsf{a} \lhd (\mathsf{large} \vartriangle \mathsf{company})) &= \langle \boxed{5} \rangle \; \langle x_2 \rangle \begin{bmatrix} \boxed{5} |\mathsf{of}| \,(\, \mathsf{arg1} = x_2, \; \mathsf{arg2} = x_1 \,), \\ |\mathsf{a}|_{(x_1)} \, \boxed{4} \, \text{--}, \\ \boxed{3} \begin{bmatrix} |\mathsf{large}| \,(\, \mathsf{arg1} = x_1 \,), \\ \text{--} \,\&\, \text{--}, \\ |\mathsf{company}| \,(\, \mathsf{KEY} = x_1 \,) \end{bmatrix}, \\ \boxed{4} \prec \boxed{3} \end{bmatrix}; \\
\mathsf{representative} &= \langle \boxed{6} \rangle \; \langle x_2 \rangle \left[\boxed{6} |\mathsf{representative}| \,(\, \mathsf{KEY} = x_2 \,) \right]; \\
\mathsf{repr} \vartriangle (\mathsf{of} \rhd (\mathsf{a} \lhd (\mathsf{large} \vartriangle \mathsf{company}))) &= \langle \boxed{7} \rangle \; \langle x_2 \rangle \begin{bmatrix} \boxed{7} \begin{bmatrix} |\mathsf{representative}| \,(\, \mathsf{KEY} = x_2 \,), \\ \text{--} \,\&\, \text{--}, \\ |\mathsf{of}| \,(\, \mathsf{arg1} = x_2, \; \mathsf{arg2} = x_1 \,) \end{bmatrix}, \\ |\mathsf{a}|_{(x_1)} \, \boxed{4} \, \text{--}, \\ \boxed{3} \begin{bmatrix} |\mathsf{large}| \,(\, \mathsf{arg1} = x_1 \,), \\ \text{--} \,\&\, \text{--}, \\ |\mathsf{company}| \,(\, \mathsf{KEY} = x_1 \,) \end{bmatrix}, \\ \boxed{4} \prec \boxed{3} \end{bmatrix}.
\end{aligned}
$$

Figure 4.1.: Example composition

mar as follows:

$$\begin{array}{lll}
\mathtt{S} \rightarrowtail \mathtt{NP}^{\uparrow}\ \mathtt{VP}^{\uparrow}, & \mathtt{VP} \rightarrowtail \mathtt{Adv}^{\uparrow}_{\mathtt{really}}\ \mathtt{VP}^{\uparrow}, & \mathtt{PP} \rightarrowtail \mathtt{P}^{\uparrow}\ \mathtt{NP}^{\uparrow}, \\
\mathtt{NP} \rightarrowtail \mathtt{Det}^{\uparrow}_{\mathtt{every}}\ \mathtt{N'}^{\downarrow}, & \mathtt{VP} \rightarrowtail \mathtt{Adv}^{\uparrow}_{\mathtt{didn't}}\ \mathtt{VP}^{\downarrow}, & \mathtt{N'} \rightarrowtail \mathtt{N'}^{\uparrow}\ \mathtt{PP}^{\uparrow}, \\
\mathtt{NP} \rightarrowtail \mathtt{Det}^{\uparrow}_{\mathtt{some}}\ \mathtt{N'}^{\uparrow}, & \mathtt{VP} \rightarrowtail \mathtt{V'}^{\uparrow}, & \mathtt{V'} \rightarrowtail \mathtt{V'}^{\uparrow}\ \mathtt{PP}^{\uparrow}, \\
\mathtt{N'} \rightarrowtail \mathtt{Adj}^{\uparrow}\ \mathtt{N'}^{\uparrow}, & \mathtt{V'} \rightarrowtail \mathtt{V}^{\uparrow}, & \\
 & \mathtt{V'} \rightarrowtail \mathtt{V}^{\uparrow}\ \mathtt{NP}^{\uparrow}. &
\end{array}$$

Rather than augmenting a grammar like this with semantic composition rules, this grammar applies monotonicity markers directly to constituents. MacCartney (2009) has developed such a ruleset for use with the Stanford parser, but the above toy grammar will be sufficient for our purposes.

Given such a grammar, we can obtain syntax trees like these:

$$\begin{array}{l}
\textsf{tango}^{\uparrow}\ \langle \textsf{in}^{\uparrow}\ \textsf{Paris}^{\uparrow}\rangle^{\uparrow}, \\
\langle \textsf{Some}^{\uparrow}\ \textsf{dog}^{\uparrow}\rangle^{\uparrow}\ \textsf{barked}^{\uparrow}, \\
\langle \textsf{Every}^{\uparrow}\ \textsf{dog}^{\downarrow}\rangle^{\uparrow}\ \textsf{barked}^{\uparrow}, \\
\langle \textsf{Every}^{\uparrow}\ \textsf{dog}^{\downarrow}\rangle^{\uparrow}\ \langle \textsf{didn't}^{\uparrow}\ \textsf{bark}^{\downarrow}\rangle^{\uparrow},
\end{array}$$

which are decorated with monotonicity markers.

Given such markers, we can move on to the logical reasoning mechanism which we call syntactically driven substitution logic (SynSL) using the following idea: Let's assume that we have somehow established that, when ψ is substituted for φ, then φ also logically follows from ψ, perhaps on the basis of an ontological fact which we take as logically trivial or axiomatic. Later on (section 4.3.4), we will in fact make the point that this is not such a good assumption. But let us nevertheless develop this formal system as a working hypothesis. We write this relationship as $\varphi \Rightarrow \psi$.

We then define our substitution logic as follows: If $\varphi \Rightarrow \psi$, let the same substitution be permissible in a context f where φ occurs as a constituent in an upward-monotonic context. We write this relationship as $\mathrm{f}(\varphi^{\uparrow}) \Rightarrow \mathrm{f}(\psi^{\uparrow})$. Furthermore, if φ occurs in a downward-monotonic context f′, let the converse substitution be permissible: $\mathrm{f}'(\psi^{\downarrow}) \Rightarrow \mathrm{f}'(\varphi^{\uparrow})$. Then, we immediately have an impressively productive logic:

$$\begin{array}{l}
\quad \textsf{Paris} \Rightarrow \textsf{France} \\
\hline
\therefore\ \textsf{in Paris}^{\uparrow} \Rightarrow \textsf{in France}^{\uparrow}, \\
\quad \textsf{tango} \Rightarrow \textsf{dance} \\
\hline
\therefore\ \textsf{tango}^{\uparrow}\ \langle\textsf{in Paris}\rangle^{\uparrow} \Rightarrow \textsf{dance}^{\uparrow}\ \langle\textsf{in France}\rangle^{\uparrow}
\end{array}.$$

And, similarly:

$$\frac{\textsf{brown dog} \Rightarrow \textsf{dog}}{\therefore \textsf{Every dog}^{\downarrow}\ \textsf{barked} \Rightarrow \textsf{Every}\ \langle\textsf{brown dog}\rangle^{\downarrow}\ \textsf{barked}},$$

$$\frac{\textsf{brown dog} \Rightarrow \textsf{dog}}{\therefore \textsf{Some}\ \langle\textsf{brown dog}\rangle^{\uparrow}\ \textsf{barked} \Rightarrow \textsf{Some dog}^{\uparrow}\ \textsf{barked}}.$$

4.3.2. Semantically Driven Substitution Logic

We can now contrast syntactically driven substitution logic (SynSL) with semantically driven substitution logic (SemSL) as a working hypothesis to improve over SynSL. In the next section (section 4.3.3), we will then point out some limitations of SynSL which SemSL does not suffer from, and, in section 4.3.4, we will point out limitations with SemSL and finally arrive at the syllogism as a more adequate reasoning mechanism.

Semantic Heads

Semantic heads can easily be defined in terms of maximally recursive ProtoForms. A given set of subforms which appear in a sentence make up a semantic head, written in square brackets [⋯], iff they appear together in the same ProtoForm within a maximally recursive representation. Furthermore, we enclose a set of subforms in round parentheses (⋯) when they appear together in the same subform of a ProtoForm. Note that we used angle brackets ⟨⋯⟩ in the previous section to denote syntactic constituents.

The two configurations of 'Every representative saw a sample', for example, have the following semantic structure:

Every representative [(saw a sample)].

[(Every representative saw)] a sample.

Confer section 4.1.1 for the full ProtoForm corresponding to this abbreviated notation.

Here, we can still mark some of the semantic heads as such, even in a fully scope-underspecified, but maximally recursive, ProtoForm.

(Every representative) saw (a sample).

We mark those semantic heads as such which are semantic heads in all configurations, and we remove the offending brackets which cross each other in different configurations.

Note how the above structure, despite being underspecified for scope, still shows a useful parallelism with a structure like this:

(Every [new representative]) saw (an [impressive sample]).

Another one of our running examples is even more heavily underspecified for scope:

> Every [representative of] (a [large company]) saw (a sample).

Considering heads such as [representative of], it becomes clear that semantic heads do not always coincide with syntactic constituents.

A semantic head need not even consist of words which are consecutive in a sentence:

> (Every Scillonian) [pay in] taxes Britain.

This would be the semantic structure of 'Every Scillonian pays taxes in Britain'.[4]

From the point of view of this section, we will only be interested in these bracketed structures. So the purpose of semantic composition here is simply to produce semantic heads. But, let us emphasize that semantic heads are only one kind of information which we can extract from the ProtoForms resulting from the semantic composition process discussed in the previous section (section 4.2).

Semantic Monotonicity Markers

In the previous section, we applied monotonicity markers to syntactic constituents. We can do the same thing with semantic heads. For example, the scope of a negation, or the restrictor of a universal quantifier, are downward monotonic, while the body of a quantifier, or the restrictor of an existential quantifier are upward monotonic.

> not [(every dog$^{\downarrow}$ bark$^{\uparrow}$)]$^{\downarrow}$.
>
> (Every representative$^{\downarrow}$) saw$^{\uparrow}$ (a sample$^{\uparrow}$).

[4] One final comment may be in place concerning this abbreviated notation: The ordering of subforms within ProtoForms is never significant for inference purposes, so the above expression would be equivalent to, e.g.

> Britain [pay in] (Every Scillonian) taxes.

However, there are a number of purposes for which a canonical ordering of subforms is useful, one of them being the goal of making our abbreviated forms as readable as possible. This is why, in our previous definitions of ProtoForms, the roots of a ProtoForm were defined as members of an ordered sequence, not members of an unordered set. Also, the composition operators as previously defined, together with the union, selection, and deletion operations on ProtoForms maintain this sequence such as to keep the roots and associated subforms of a ProtoForm in the canonical order. The canonical order aims to make subforms occur in the same order as the words in the surface representation, except where a reordering is required due to the necessity of putting a given set of words between a given pair of brackets representing a semantic head. Where such a reordering is required, it is always the lefthand argument to the composition operator which determines the ordering.

Note that this assignment can be made directly to underspecified forms, and without the need to enumerate configurations. Instead, we can simply propagate markers through a chart representation. Such chart representations have previously been introduced by Koller & Thater (2005), Koller et al. (2009) for purposes of scoping.

Note how, in a substitution logic, the recursive nesting of monotonicity markers affects the property of substitution directionality. For example, if we make a substitution $\mathrm{f}(\varphi^{\uparrow}) \Rightarrow \mathrm{f}(\psi^{\uparrow})$ we note that $\mathrm{f}(\varphi)$ and $\mathrm{f}(\psi)$ are themselves formulae, which we might write φ' and ψ'. This would then, for example, allow a substitution $\mathrm{g}(\psi'^{\downarrow}) \Rightarrow \mathrm{g}(\varphi'^{\downarrow})$. Equivalently, we could simply note that the context $\mathrm{g} \circ \mathrm{f}$ where the upward monotonic context is nested inside a downward monotonic context is itself downward monotonic, so substitutions must be of the form $\mathrm{g} \circ \mathrm{f}(\varphi^{\downarrow}) \Rightarrow \mathrm{g} \circ \mathrm{f}(\psi^{\downarrow})$.

More generally, a downward monotonic context inverts its inside monotonicity markers, while an upward monotonic context leaves them as they are. Let's consider our earlier example 'Every dog didn't bark', which we might assign the following representation:

> not [(every [brown$^{\uparrow}$ dog$^{\uparrow}$]$^{\downarrow}$ bark$^{\uparrow}$)]$^{\downarrow}$.

If we resolve the outermost marker to the inner level, we get

> not [(every [brown$^{\uparrow}$ dog$^{\uparrow}$]$^{\uparrow}$ bark$^{\downarrow}$)],

and, if we resolve the next level,

> not [(every [brown$^{\uparrow}$ dog$^{\uparrow}$] bark$^{\downarrow}$)].

Substitution Logic & Ambiguous Monotonicity Markers

Now consider an example involving a more complex form of scope ambiguity:

> Every [representative of] (a company) arrived,

which has the following configurations:

> Every [([representative of]$^{\uparrow}$ a company$^{\uparrow}$)]$^{\downarrow}$ arrived$^{\uparrow}$,
> [(Every [representative of]$^{\downarrow}$ arrived$^{\uparrow}$)]$^{\uparrow}$ a company$^{\uparrow}$.

We can now see that 'arrived' is upward monotonic in both cases (either upward monotonic on the outermost context, or upward monotonic in an upward monotonic context). Similarly, [representative of] is downward monotonic in both cases (either upward monotonic in a downward monotonic context, or downward monotonic in an upward monotonic context). But 'company' can be either upward monotonic or downward monotonic, depending on the configuration used to resolve the scope ambiguity. In one configuration, it is upward monotonic in a downward monotonic context, in the other configuration it is upward monotonic in the outermost context.

So, we write the monotonicity markers of our underspecified ProtoForm as follows:

$$\text{Every [representative of]}^{\downarrow}\text{ (a company}^{?}\text{) arrived}^{\uparrow}.$$

Furthermore, we extend our substitution logic by blocking substitutions of the following forms: $\mathrm{f}(\varphi^{?}) \not\Rightarrow \mathrm{f}(\psi^{?})$ and $\mathrm{f}(\psi^{?}) \not\Rightarrow \mathrm{f}(\varphi^{?})$. But we will permit both $\mathrm{f}(\varphi^{?}) \Rightarrow \mathrm{f}(\psi^{?})$ and $\mathrm{f}(\psi^{?}) \Rightarrow \mathrm{f}(\varphi^{?})$ in case we also have both $\varphi \Rightarrow \psi$ and $\psi \Rightarrow \varphi$.

For example:

$$\begin{array}{l} [\text{representative of}] \Rightarrow [\text{salesman of}],\\ \text{registered} \Rightarrow \text{arrived},\\ \text{manufacturer} \Rightarrow \text{company},\\ \text{firm} \Rightarrow \text{company},\\ \text{company} \Rightarrow \text{firm},\\ \hline \therefore \dfrac{\text{Every [salesman of]}^{\downarrow}\text{ (a company}^{?}\text{) registered}^{\uparrow}}{\Rightarrow \text{ Every [representative of]}^{\downarrow}\text{ (a firm}^{?}\text{) arrived}^{\uparrow}},\\ \not\therefore \dfrac{\text{Every [representative of]}^{\downarrow}\text{ (a company}^{?}\text{) arrived}^{\uparrow}}{\Rightarrow \text{ Every [representative of]}^{\downarrow}\text{ (a manufacturer}^{?}\text{) arrived}^{\uparrow}},\\ \not\therefore \dfrac{\text{Every [representative of]}^{\downarrow}\text{ (a manufacturer}^{?}\text{) arrived}^{\uparrow}}{\Rightarrow \text{ Every [representative of]}^{\downarrow}\text{ (a company}^{?}\text{) arrived}^{\uparrow}}. \end{array}$$

Syntactic monotonicity composition cannot deal with such examples, as, by definition, it is incapable of detaching the notion of semantic scope from the recursive constituency structure of the syntax-tree or of detecting scope ambiguity. MacCartney (2009, p. 118) concedes that "classic scope ambiguities [. . .] may be one issue" for his approach.

As an example of a classic scope ambiguity MacCartney (2009) mentions 'Every man loves a woman'. This is, as he also points out, in fact no problem at all, due to the fact that, as we have seen, despite scope ambiguity, there is no ambiguity about the monotonicity markers involved. However, he fails to discuss examples such as the above, where genuine ambiguity does arise.

MacCartney (2009, p. 135) then goes on to state that "in practice this rarely causes problems", by which he probably means the absence of such phenomena in the particular inference datasets he works with, such as the FraCaS testsuite and RTE data. This can hardly be denied. But at the same time, one is reminded of how he motivates his study of syntactic monotonicity marking to begin with: The problem being addressed here is that the success of most RTE systems "depends on the prevalence of upward-monotone contexts, and thus can easily be derailed" by the infrequent, though existent, presence of downward-monotone contexts (MacCartney 2009, p. 92).

At its most basic level, the heart of the problem being addressed by MacCartney (2009) is that of applying monotonicity markers to text recursively in such a way as to allow for textual reasoning. So, regardless of the "practical" impact of his simplifying assumptions, it seems unsatisfactory to allow any admissions to the extent that the recursive structures he uses may, in fact, be linguistically inadequate for his purposes.

For example, his approach, to the best of my understanding, could not deal with cases such as 'Every brown dog does not bark', where there is no real ambiguity involved at all, but where the semantically preferred scoping

not [(every [brown dog] bark)]

simply does not line up with the syntactic structure of the sentence

⟨Every brown dog⟩ ⟨does not bark⟩.

4.3.3. Semantic Scoping Principles

Finally, we need to introduce some additional conventions about resolving scope. In particular, our distinction between quantifications, modalifications, and connections makes it possible to single out different semantically interesting groupings among the configurations of a given underspecified ProtoForm.

Scoping Connections vs. Quantifications & Modalifications

As a general rule, we will always have connectives take wide scope over quantifications and modalifications. So, for example, we might have a maximally recursive ProtoForm which, in its general scope-underspecified form, looks like this:

(The dog) barked and (the cat) meowed,

where, in general, quantifiers belonging to the different sentential clauses might float into each other's body scopes. But, due to the fact that the two sentential clauses share no variables in the ProtoForm, this kind of scope ambiguity would be entirely spurious and could be eliminated in a redundancy elimination framework such as that of Koller & Thater (2006), Koller et al. (2009). So, let's establish a convention to add brackets around clausal constituents:

[(The dog) barked] and [(the cat) meowed].

leading to the following maximally recursive ProtoForm:

[(The dog barked)] and [(the cat meowed)].

Scoping Quantifications vs. Modalifications

It is often useful to make a distinction, for a given modalification, between *de re* configurations in which one minimizes the number of quantifiers which take narrow scope under that modalification, and *de dicto* configurations in which one maximizes the number of such quantifiers.

Consider, for example, the following maximally recursive ProtoForm:

> (The organizers) believed (the representatives) arrived,

which has the following configurations:

a1. The organizers [([(believed arrived)] the representatives)],
a2. [(The organizers [(believed arrived)])] the representatives,
b. The organizers [(believed [(the representatives arrived)])].

We can now distinguish among configurations between group (a) and group (b), and represent the same distinction by adding the following brackets:

a. (The organizers) [believed arrived] (the representatives),
b. (The organizers) believed [(the representatives) arrived],

leading to the following maximally recursive ProtoForms:

a. (The organizers) [(believed arrived)] (the representatives),
b. (The organizers) [(believed [(the representatives arrived)])].

ProtoForm (a) represents the de re reading of 'believed' and ProtoForm (b) represents the de dicto reading of 'believed'.

The brackets suggest the significance of this distinction for logical inferences. Consider, for example, the following:

$$\frac{\text{The organizers believed (some representatives) arrived}}{\text{The organizers believed (some salesmen) arrived}}.$$

A logical relationship which might justify this inference is that every representative is a salesman. But we also need to know the modal force of such a claim, and the modal scope of (some representatives) and (some salesmen) in the sentence.

The inference should go through under the de re reading iff the relationship is valid on the topmost modal context, and under the de dicto reading iff it is valid as a belief held by the organizers about whom the sentence is reporting.

In this case, I would expect Davidson (1968) to argue that the complementizer 'that' in

> The organizers believed [that] the representatives arrived

should exhibit the same function as the demonstrative 'that' with the subordinate clause as a cataphoric referent:

> The organizers believed that. The representatives arrived.

Thus, scope ambiguity about quantifiers from the two sentential clauses floating into each other's scopes would become spurious in much the same way as for the sentential connectives discussed previously.

Following the principle of *semantic innocence* (Bach 1997), we would resolve this spurious ambiguity by resorting to the de re reading, thus generally choosing ProtoForm (a) and dropping ProtoForm (b). This maximizes the amount of reasoning which is pushed onto the outermost scope and thus becomes shared between contexts.

But Davidson's and Bach's account are not without their critics[5], and their arguments do not extend to modalities which have nothing to do with such propositional attribute reports. Examples like

> The organizers necessarily greeted the representatives

seem to exhibit a more genuine form of logical ambiguity in a modal logic between

> $\Box$[(The organizers) greeted (the representatives)], and
> (The organizers) [$\Box$greeted] (the representatives).

The question is what to infer when the claim that every representative is a salesman is true, but not necessarily so. In fact, even the past tense marker associated with 'greeted' acts as such a modality in a temporal logic. The question then arises what to do with this inference if it is now true that every representative is a salesman, this not having been the case until recently.

4.3.4. Ontological Limitations of Substitution Logic

Universal vs. Existential Substitutability: PP-Arguments & Modifiers

Verb Modifiers vs. Verb Complements

$$\frac{\text{Paul tangoed}}{\therefore\ \text{Paul danced}}, \quad \frac{\text{Paul tangoed in Paris}}{\therefore\ \text{Paul danced in Paris}},$$

$$\frac{\text{Jones works}}{\therefore\ \text{Jones lives}}, \quad \frac{\text{Jones works in London}}{\not\therefore\ \text{Jones lives in London}};$$

[5] for a good summary on the problems surrounding propositional attitude reports see e.g. McKay & Nelson (2005)

The linguistic examples offered by MacCartney (2009) and other advocates of substitution logic display a peculiar affinity for verbs which happen to be closely related to nouns that are in a hypernymy-hyponymy relation such as 'tango' and 'dance'. These are clean cases for inference purposes, since we accept all of the following:

- syntactically montonic substitution:
  'if you tango, you dance';
- WordNet "verb entailment":
  'tangoing cannot be done unless dancing is done';
- WordNet verb troponymy:
  'to tango is to dance';
- syllogistic premise:
  'every tango is a dance';
- noun hyponymy:
  'tango is a dance'; 'tango is a kind of dance'; 'tangoing is a kind of dancing'.

Let's look at a different case:

- syntactically montonic substitution:
  'if you work, you live';
- WordNet verb entailment:
  'working cannot be done unless living is done';
- WordNet verb troponymy:
  '*to work is to live' (wrong, or meaning shift).

For this case, there exists no straightforward way of turning the concept of living or working into a noun, while maintaining the entailment property: 'every work is a life', 'work is a kind of life', 'working is a kind of living', are all nonsensical, wrong, or suggest an unintended metaphor.

The phenomenon is easily accounted for within an ontology expressed in the language of FOPC. Consider one solution based on Davidsonian-style event variables:

$$\forall_{(e,x)} \{|\mathrm{tango}|(\mathrm{KEY} = e,\ \mathrm{ARG1} = x) \rightarrow |\mathrm{dance}|(\mathrm{KEY} = e,\ \mathrm{ARG1} = x)\},$$
$$\forall_{(x)} \left\{\exists_{(e)} \{|\mathrm{work}|(\mathrm{KEY} = e,\ \mathrm{ARG1} = x)\} \rightarrow \exists_{(e')} \{|\mathrm{live}|(\mathrm{KEY} = e',\ \mathrm{ARG1} = x)\}\right\}.$$

The former relationship is what we need in order to postulate tango ⇒ dance in a substitution logic. Here, variables of predicates are never explicitly represented. Instead it is assumed that all relationships which affect substitutability are based on implicit universal quantification. But the expressive power of such a logic does not permit an adequate representation of the kind of relationship which exists between 'work' and 'live' in the above example.

There is no straightforward way of addressing that problem within the confines of substitution logic: If the grammaticist goes with the usual natural logic treatment of 'tango in Paris', then the ontologist is left with a choice to either incorrectly infer '∴ Jones lives in London' by interpreting '⇒' as involving besides cases of troponymy also other cases of verb entailment and postulating work ⇒ live, or to miss out on '∴ Jones lives' by interpreting '⇒' as involving only clean cases of troponymy, thereby precluding the possibility to postulate work ⇒ live.

The grammaticist may attempt to come to the ontologist's rescue by treating '⇒' as involving besides cases of troponymy also cases of meronymy, and consequently blocking intersective modification of verb phrases by prepositional phrases altogether, thus allowing the ontologist to safely postulate both work ⇒ live and tango ⇒ dance, but this would mean we are now losing '∴ I dance in France.'

Finally, the ontologist and grammaticist may conspire to hypothesize that a sense distinction is the true culprit, the material existence sense of living behaving in a cleanly monotonic fashion with PP-modification, the habitation sense of living featuring an optional PP-complement which blocks monotonic substitutions. Of course, going down this road means possibly introducing into the grammar a syntactically spurious distinction, where only an ontological distinction might exist.

Noun Modifiers vs. Relational Noun Complements

Furthermore, it seems the phenomenon is not limited to verbs, but applies to any kind of predicate which is relational in the ontology, i.e. which takes more than one argument, including relational nouns.

$$\frac{\text{J. is an auditor.}}{\therefore\ \text{J. is an accountant.}},\quad \frac{\text{J. is an accountant.}}{\therefore\ \text{J. is an employee.}},\quad \frac{\text{J. is an auditor of IBM.}}{\not\therefore\ \text{J. is an employee of IBM.}}.$$

From the perspective of substitution logic, it is hard to block the chain of reasoning 'auditor ⇒ accountant ⇒ employee' which would be licensed by

$$\forall_{(x_1,x_2)}\ \{\mathsf{auditorOf}(\,\mathrm{ARG1}=x_1,\ \mathrm{ARG2}=x_2\,) \rightarrow \mathsf{accountantAt}(\,\mathrm{ARG1}=x_1,\ \mathrm{ARG2}=x_2\,)\},$$
$$\forall_{(x_1,x_2)}\ \{\mathsf{accountantAt}(\,\mathrm{ARG1}=x_1,\ \mathrm{ARG2}=x_2\,) \rightarrow \mathsf{employeeOf}(\,\mathrm{ARG1}=x_1,\ \mathrm{ARG2}=x_2\,)\},$$

but not by

$$\forall_{(x_1)} \left\{\begin{array}{l} \exists_{(x_2)}\ \{\mathsf{auditorOf}(\,\mathrm{ARG1}=x_1,\ \mathrm{ARG2}=x_2\,)\} \\ \quad \rightarrow \exists_{(x_2')}\ \{\mathsf{accountantAt}(\,\mathrm{ARG1}=x_1,\ \mathrm{ARG1}=x_2'\,)\} \end{array}\right\},$$
$$\forall_{(x_1)} \left\{\begin{array}{l} \exists_{(x_2)}\ \{\mathsf{accountantAt}(\,\mathrm{ARG1}=x_1,\ \mathrm{ARG2}=x_2\,)\} \\ \quad \rightarrow \exists_{(x_2')}\ \{\mathsf{employeeOf}(\,\mathrm{ARG1}=x_1,\ \mathrm{ARG1}=x_2'\,)\} \end{array}\right\}.$$

For nouns it seems even less tempting to speak of a grammatical distinction between non-monotonic optional PP-complements and monotonic PP-modifiers, since this distinction seems even more systematically spurious from the point of view of syntax.

However, within the ontology, it makes perfect sense to postulate a very close relationship between modifiers and optional complements when it comes to providing optional arguments for verbs and relational nouns. Incidentally, such a naming convention for relational nouns and verbs with, possibly optional, PP-complements is also used in the ERG (Flickinger 2000), where predicates such as employee+of or accountant+at induce sense-distinctions which are useful, for example, for machine translation.

Generally, the lack of expressive power of substitution logics when it comes to such knowledge stems from their inability to attach arguments to predicates and to allow the ontology to relate predicates to each other by quantifying over the individual arguments.

Conversely, if one were to write down the underlying ontology in natural language, one would have to resort to statements like 'If you work, you live' but 'If you work in a place, you do not necessarily live in that same place' vs. 'If you dance, you tango' implying 'If you dance in a place, you also tango in that same place' or 'Every auditor is an employee' but 'Every auditor of a company is not necessarily an employee of that company' vs. 'Every accountant is a person' implying 'Every happy accountant is a happy person'. I can think of no grammatically trivial way of expressing such relationships at all, without recourse to meta-level variable binding as in the case of anaphora or mathematical language.

Transitivity Dimensions: Space Needle & Mariana Trench

In the previous section, we argued that the mechanism provided by FOPC to state quantifications over explicit variables makes FOPC adequate in a way in which substitution logic is not, for purposes of expressing the ontological distinction between complement-taking and modification. In this section, we will consider another important metatheoretical principle of ontology: transitivity.

Consider the classical substitution logic example of the geographical location sense of 'in' and the following transitivity property, as expressed in FOPC:

$$\forall_{(x_1,x_2,x_3)} \left\{ \begin{array}{l} \mathrm{in}\,(\,\textsc{arg1} = x_2,\ \textsc{arg2} = x_1\,)\ \&\ \mathrm{in}\,(\,\textsc{arg1} = x_3,\ \textsc{arg2} = x_2\,) \\ \quad \rightarrow \mathrm{in}\,(\,\textsc{arg1} = x_3,\ \textsc{arg2} = x_1\,) \end{array} \right\}.$$

One can establish how the meaning of this preposition relates to that of 'outside', by postulating the following relationship:

$$\forall_{(x_1,x_2)}\ \{\mathrm{in}\,(\,\textsc{arg1} = x_1,\ \textsc{arg2} = x_2\,) \rightarrow \neg\mathrm{outside}\,(\,\textsc{arg1} = x_1,\ \textsc{arg2} = x_2\,)\}.$$

These two meaning postulates would logically entail the contrapositives

$$\forall_{(x_1,x_2)} \{\text{outside}\,(\,\text{ARG1} = x_1,\ \text{ARG2} = x_2\,) \rightarrow \neg\text{in}\,(\,\text{ARG1} = x_1,\ \text{ARG2} = x_2\,)\},$$

$$\forall_{(x_1,x_2,x_3)} \left\{ \begin{array}{l} \text{in}\,(\,\text{ARG1} = x_2,\ \text{ARG2} = x_1\,)\ \&\ \text{outside}\,(\,\text{ARG1} = x_3,\ \text{ARG2} = x_1\,) \\ \quad \rightarrow \text{in}\,(\,\text{ARG1} = x_3,\ \text{ARG2} = x_2\,) \end{array} \right\}.$$

Given this and 'The Space Needle is in Seattle', we get the following inferences:

$$\frac{\text{Jones was in the Space Needle}}{\therefore\ \text{Jones was in Seattle}}, \quad \frac{\text{Jones was outside the Space Needle}}{\not\therefore\ \text{Jones was outside Seattle}},$$

$$\frac{\text{Jones was in Seattle}}{\not\therefore\ \text{Jones was in the Space Needle}}, \quad \frac{\text{Jones was outside Seattle}}{\therefore\ \text{Jones was outside the Space Needle}}.$$

This example of 'in the Space Needle' ⇒ 'in Seattle' is parallel to the one given by MacCartney & Manning (2007), Chambers et al. (2007), whereby 'in Paris' ⇒ 'in France'.

Moreover, it is important to note that, due to the way in which MacCartney (2009) has set up his inference mechanism, he must arrive at this relationship in a compositional manner. In particular, he uses a two-stage strategy, where the first stage consists in identifying lexical substitutions, deletions, and insertions, and the second stage consists in determining compositionally the impact of such operations. So as soon as 'outside Paris' and 'outside France' enter the scene, it must be down to the relationship between the prepositions 'in' and 'outside' and the monotonicity properties among the complement NPs to derive monotonicity properties for the resulting PPs. These cannot be directly postulated in the ontology as relationships between PPs per se.

If this set of relationships is to be derived compositionally, it would have to be expressed in SynSL by the relationship 'Space Needle' ⇒ 'Seattle', together with the grammar rules $\texttt{PP} \rightarrowtail \texttt{in NP}^{\uparrow}$ and $\texttt{PP} \rightarrowtail \texttt{outside NP}^{\downarrow}$. The idea that 'outside' would have to be taken as imposing a downward-monotonic context on its complement is parallel to the one expressed by MacCartney (2009, p. 11) that 'without' should be downward-monotonic.

But this is problematic. The relationship 'Space Needle' ⇒ 'Seattle' or 'Paris' ⇒ 'France' is clearly not logically axiomatic, but rather a quite specific ontological thesis: geography relates the Space Needle and Seattle to areas in the geometric space of latitude and longitude, and geometry establishes inclusion properties for such areas which can be expressed by 'in' and 'outside'.

The fact that 'Space Needle' ⇒ 'Seattle' is, indeed, not logically axiomatic, is reflected in the fact that we could arrive at the same system of relationships, by doing the exact opposite throughout the grammar: we could have 'Seattle' ⇒ 'Space Needle', and the grammar rules $\texttt{PP} \rightarrowtail \texttt{in NP}^{\downarrow}$ and $\texttt{PP} \rightarrowtail \texttt{outside NP}^{\uparrow}$.

If we do grant 'Space Needle' ⇒ 'Seattle', we will find that it either significantly overgenerates inferences such as 'like the Space Needle' ⇒ 'like Seattle', or 'own the Space Needle' ⇒ 'own Seattle', where the inference does not refer to this geographical/geometric inclusion, or it would miss inferences.

The question here is how to express the transitivity property of a given ontological predicate. Since logical entailment is a transitive relation, one can arrive at such a transitivity property by reinterpreting logical entailment under the interpretation of the ontological predicate. But such an approach would be badly confused. The point is that a logic has only one entailment relation, but an ontology has many transitive predicates, each imposing an independent dimension along which inclusions and exclusions arise. Inclusions or exclusions along these ontological dimensions are not at all the same thing as logical entailment and logical disjointness.

This can easily be shown by extending our example about in/outside with further dimensions by introducing into our toy grammar vocabulary such as above/below. Our domain of Space Needle/Seattle could be extended by Mariana Trench/Pacific.

We see that, w.r.t. the above example inferences, Mariana Trench/Pacific behave exactly like Space Needle/Seattle, which might lead us to postulate 'Mariana Trench' ⇒ 'Pacific'.

Now, let's say we want the following inferences, given 'The Mariana Trench is below the Pacific' and 'The Space Needle is above Seattle':

$$\frac{\text{The balloon floats above the Space Needle}}{\therefore\ \text{The balloon floats above Seattle}},$$

$$\frac{\text{Oil was found below the Mariana Trench}}{\therefore\ \text{Oil was found below the Pacific}},$$

$$\frac{\text{Data were measured above the Mariana Trench}}{\not\therefore\ \text{Data were measured above the Pacific}},$$

$$\frac{\text{A ring was found below the Space Needle}}{\not\therefore\ \text{A ring was found below Seattle}}.$$

It can be seen, that, given the relationships postulated so far, the latter two incorrect inferences cannot be blocked. Regardless of what we do to entailment-directions for lexical items and projectivity markers in our example grammar, we will find that at least two out of our eight example inferences will fail.

The only recourse for SynSL would be to mark the complements of prepositions as not projecting any entailment information, and to characterize the relationships in an ontology of rewrite relationships between PPs, without attempting to build them compositionally. – FOPC, on the other hand, is perfectly capable of a compositional treatment of inferences arising from these transitivities, by simply adding meaning postulates such as the one expressed above for 'in'. Each of these transitivity postulates refers only to one lexical item at a time, not entire phrases.

This example, involving two spacial dimensions, is, of course, only the tip of the iceberg, as it has been specifically chosen so as to hinge on the transitivity of a single preposition. If we think of the vast number of transitive predicates which provide the ontological interpretations for other semantic heads such as [member of], [attaches to], supports, etc., it quickly becomes clear that an ontology should not only be able to distinguish two dimensions of transitive inclusions, but, in fact, a great many.

MacCartney's approach, however, lumps in with logical negations linguistic phenomena such as the preposition 'without', the verb 'avoid', the adverb 'rarely', the superlative adjective 'tallest', and the noun 'denial' (MacCartney 2009, p. 11). Based on our discussion in this section, it should be clear, however, that this is confusing metalinguistic predicates with linguistic predicates and logical axioms with ontological theses. It amounts to saying that the ontological dimension which counts how often an event occurs is the same as the dimension which measures how tall a person is.

4.4. Syllogistic Normal Form Decomposition

We began this chapter by giving an introduction to the ProtoForm semantic representation language and showing how to use grammatical composition to translate text into ProtoForms. We showed how ProtoForms impose a recursive structure on text, and, equating such recursive structures to the notion of semantic head, argued that semantically driven substitution logic is more suitable for purposes of logical inference than syntactically driven substitution logic.

In this section, we will go one step further and discuss how ProtoForms also impose relational dependencies on text. We call these structures syllogistic normal forms (SNFs), and the individual relational dependencies which make up such structures are what we call syllogistic premises (SPs). – One might think of SNFs in terms of three different interpretations as illustrated by the different forms of notation shown in Figure 4.2.

First, an SNF is a logical formula within a fragment of FOPC, which can be used within traditional theorem provers or other kinds of first order logical inference mechanisms. In particular, an SNF is a conjunction of SPs, each of which corresponds to a possible premise in the traditional logic of the syllogism. The semantics of the SNF within the logic serves as an approximation to the true semantics of the text for inference purposes.

Second, an SNF is a semantic dependency structure which is comparable, in terms of its metalanguage properties, to representations such as Briscoe-Carroll-style syntactic sentence:

Every representative of a company saw a sample.

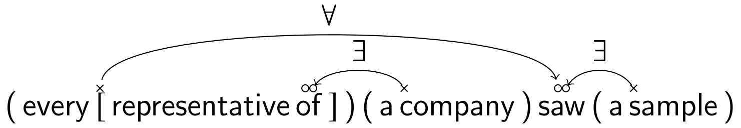
Underspecified ProtoForm:

$$\left[\begin{array}{l}
|\text{every}|\,{\scriptstyle (x_1)}\ \boxed{1}\ \text{--},\\
\boxed{2}\left[\begin{array}{l} |\text{representative}|\,(\,\text{KEY} = x_1\,),\\ \text{--}\ \&\ \text{--},\\ |\text{of}|\,(\,\text{KEY} = /e_2/,\ \text{arg1} = x_1,\ \text{arg2} = x_2\,)\end{array}\right],\\
|\text{a}|\,{\scriptstyle (x_2)}\left[\,|\text{company}|\,(\,\text{KEY} = x_2\,)\right]\ \text{--},\\
|\text{saw}|\,(\,\text{KEY} = e_1,\ \text{arg1} = x_1,\ \text{arg2} = x_3\,),\\
|\text{a}|\,{\scriptstyle (x_3)}\left[\,|\text{sample}|\,(\,\text{KEY} = x_3\,)\right]\ \text{--},\\
\quad \boxed{1} \prec \boxed{2}
\end{array}\right].$$

SNF, logical form:

$$\left[\begin{array}{l}
\left[\begin{array}{l}
\left[\begin{array}{l}|\text{every}|\,{\scriptstyle (x)}\left[\begin{array}{l}|\text{representative}|\,(\,\text{KEY} = x\,),\\ \text{--}\ \&\ \text{--},\\ |\text{of}|\,(\,\text{KEY} = /e_2/,\ \text{arg1} = x\,)\end{array}\right]\left[\,|\text{saw}|\,(\,\text{KEY} = /e_1/,\ \text{arg1} = x\,)\right]\end{array}\right]\\
\quad\wedge\left[\begin{array}{l}|\text{a}|\,{\scriptstyle (x)}\left[\,|\text{company}|\,(\,\text{KEY} = x\,)\right]\left[\begin{array}{l}|\text{representative}|\,(\ ),\\ \text{--}\ \&\ \text{--},\\ |\text{of}|\,(\,\text{KEY} = /e_2/,\ \text{arg2} = x\,)\end{array}\right]\end{array}\right]
\end{array}\right]\\
\quad\wedge\left[\,|\text{a}|\,{\scriptstyle (x)}\left[\,|\text{sample}|\,(\,\text{KEY} = x\,)\right]\left[\,|\text{saw}|\,(\,\text{KEY} = /e_1/,\ \text{arg2} = x\,)\right]\right]
\end{array}\right]$$

SNF, dependency notation:


∀
∃
∃
(every [representative of]) (a company) saw (a sample)

SNF, McDonald's decomposition:

- F: Every representative saw \ something.
  Q: Who / saw? A: Every representative / saw.
- F: They were \ representatives of a company.
  Q: Who were they \ representatives of? A: Representatives of \ a company.
- F: Somebody / saw a sample.
  Q: What / was seen? A: A sample / was seen.

Figure 4.2.: example SNF structure

How can I help?

I would like a large #3 Meal with Coke and with Sweet & Sour Sauce to eat in.

Would you like that large?

Yes.

What drink?

Coke.

What sauce?

Sweet & Sour.

To eat in or take out?

To eat in, please.

Figure 4.3.: example dialogue

dependency relations (Carroll et al. 1999), henceforth abbreviated GRs, and the representations used in semantic role labelling (see e.g. Màrquez et al. 2008). We will argue that SNFs are preferable for inference purposes to syntactic dependencies and that they are expressively more powerful than semantic role labels.

Third, we can think of SNF in terms of what, for lack of a better term, one might call a "McDonald's decomposition", where a possibly complex sentence is analyzed into a number of atomic factoids expressible using question/answer pairs of a particular format, in our case SPs. SPs are atomic in the sense that expressions are only allowed to use one quantification at a time and they can be represented either in a formal language, or in a controlled natural language of greatly reduced complexity. Factoid-based decomposition is, of course, nothing new (see e.g. Hickl 2008, Bensley & Hickl 2008), and text simplification techniques have been investigated before as well (see e.g. Siddharthan 2004). As a byproduct of our approach to semantic decomposition, however, SNFs present a new approach towards identifying in a logically-motivated way a fragment of natural language which serves the purposes of syntactic simplicity and canonicity of the language, expressive adequacy of the logic, and atomicity of the factoids.

4.4.1. From Composition-Derived Forms to SNF Logical Forms

Dialogues such as the one in Figure 4.3 take place hundreds of times each day in a typical fast food restaurant. Question/answer pairs in such dialogues might correspond to buttons on a computer register communicating orders to the kitchen and controlling the cashier's dialogue with the customer. This dramatically reduces the complexity of the language used in the dialogue by focusing on one variable at a time. – We can do the same with ProtoForms.

Consider our previous example:

$$\left[\begin{array}{l} \quad |\text{every}|\ {\scriptstyle (x_2)}\ \boxed{1}\ \text{--}, \\ \boxed{2}\,|\text{representative}|\ (\ \text{KEY} = x_1\ ), \\ \quad |\text{saw}|\ (\ \text{KEY} = e_1,\ \text{arg1} = x_1,\ \text{arg2} = x_2\ ), \\ \quad |\text{a}|\ {\scriptstyle (x_2)}\ \boxed{3}\ \text{--}, \\ \boxed{4}\,|\text{sample}|\ (\ \text{KEY} = x_2\ ), \\ \qquad \boxed{1} \prec \boxed{2}, \\ \qquad \boxed{3} \prec \boxed{4} \end{array}\right].$$

This ProtoForm has two variables: x_1 and x_2, and each has a quantification associated with it. We can go through the ProtoForm one quantification at a time, and, for each one, determine a partial scoping which has that quantifier on the outermost scope. For variable x_1 the ProtoForm which results from such partial scoping would look like this:

$$\left[\ |\text{every}|\ {\scriptstyle (x_1)}\ \Big[\ |\text{representative}|\ (\ \text{KEY} = x_1\ )\Big] \left[\begin{array}{l} \quad |\text{saw}|\ (\ \text{KEY} = e_1,\ \text{arg1} = x_1,\ \text{arg2} = x_2\ ), \\ \quad |\text{a}|\ {\scriptstyle (x_2)}\ \boxed{3}\ \text{--}, \\ \boxed{4}\,|\text{sample}|\ (\ \text{KEY} = x_2\ ), \\ \qquad \boxed{3} \prec \boxed{4} \end{array}\right]\right]$$

Now we have a ProtoForm in the restrictor of the quantification, and another ProtoForm in its body. The second step consists in applying what we might call a *variable filter* on both sides. Since we are currently focusing on x_1, we simply remove all subforms which do not refer to x_1. For those subforms which do, we remove all arguments referring to quantified variables other than x_1 but leave in place event variables such as e_1. In the above example, the restrictor ProtoForm is in the proper format already, so the filter does not alter anything. In the body, however, it will delete the quantification for x_2 and the predication referring only to x_2.

$$\Big[\ |\text{every}|\ {\scriptstyle (x_1)}\ \Big[\ |\text{representative}|\ (\ \text{KEY} = x_1\ )\Big]\ \Big[\ |\text{saw}|\ (\ \text{KEY} = e_1,\ \text{arg1} = x_1\ )\Big]\Big].$$

We now have a syllogistic premise (SP) and can move on to the next variable x_2:

$$\left[\ |\text{a}|\ {\scriptstyle (x_2)}\ \Big[\ |\text{sample}|\ (\ \text{KEY} = x_2\ )\Big] \left[\begin{array}{l} \quad |\text{every}|\ {\scriptstyle (x_2)}\ \boxed{1}\ \text{--}, \\ \boxed{2}\,|\text{representative}|\ (\ \text{KEY} = x_1\ ), \\ \quad |\text{saw}|\ (\ \text{KEY} = e_1,\ \text{arg1} = x_1,\ \text{arg2} = x_1\ ), \\ \qquad \boxed{1} \prec \boxed{2}, \end{array}\right]\right].$$

After filtering we get

$$\Big[\ |\text{a}|\ {\scriptstyle (x_2)}\ \Big[\ |\text{sample}|\ (\ \text{KEY} = x_2\ )\Big]\ \Big[\ |\text{saw}|\ (\ \text{KEY} = e_1,\ \text{arg2} = x_2\ )\Big]\Big].$$

Finally, we combine the two SPs by means of conjunction in a left-branching structure, leaving us with the SNF as displayed in the figure. Note that, since quantifier nesting is never allowed in SNFs, we can generally write each premise using the same variable, which we simply call x. In the figure, we have also turned the event variable e_1 into a constant $/e_1/$ for reasons we will discuss shortly (section 4.4.2).

4.4.2. SNFs vs. Scoped Logical Forms

By converting a ProtoForm with multiple quantifiers into an SNF, we remove the information pertaining to quantifier nesting, so it is instructive to revisit the problem of scope and scope ambiguity briefly. In standard examples such as

> Every man loves a woman,

common-sense knowledge associated with the predicate 'loves' may well have a role to play. If only one subject and only one object can participate in one given loving-event at a time, then nesting of quantifiers becomes important, and scope ambiguity arises. But in the example

> Every company gave to a politician,

one might equally well argue for a collective reading, i.e. that this sentence does not commit to either of the scoped readings, but rather just states the existence of a giving-event, where the set of all companies is a subset of the set of all individuals participating in subject-role, and the overlap between the set of all politicians and the set of all individuals participating in object-role is nonempty. The predication

$$|\mathsf{gave}|\,(\ \mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1,\ \mathsf{arg2} = x_2\ )$$

would break apart into two predications

$$|\mathsf{gave}|\,(\ \mathsf{KEY} = e_1,\ \mathsf{arg1} = x_1\ ), \qquad |\mathsf{gave}|\,(\ \mathsf{KEY} = e_1,\ \mathsf{arg2} = x_2\ ),$$

Note that, by saying that there exists at least one politician who is an object of this event, we are not ruling out the possibility that there may be many different politicians for which this is the case.

By appealing to common-sense knowledge, discourse-level variable binding, or formal or mathematical language, we might be able to enforce a particular scoped reading for a given sentence:

Every man loves a woman.

∴̸ There is this woman and every man loves her.’

A clean planet is important.
Every child needs a clean planet.

∴ A clean planet is important. Every child needs it.

In the first example, the every-outscopes-a reading is perhaps more likely, due to the fact that the competing scoping is less compatible with common-sense knowledge, but one might arrive at the same result by applying scoping heuristics such as scoping quantifiers in order of their appearance from left to right. In the second example, knowledge of the context in which such sentences are usually uttered and which implications are intended suggests that the a-outscopes-every reading is more likely to be the one intended. But these inference decisions are not quite as straightforward.

By using SNFs, an inference mechanism would generally allow both inferences to go through. This is due to the fact that it discards information about scope both in the antecedent and the consequent of an implication, and therefore ignores certain models which might make the consequent false while making the antecedent true and producing a counterexample to the implication. – So SNFs will tend to err on the side of proving too many positive propositions in the consequent, and too few propositions inside the scope of a negation in the consequent.

This differs from the alternative strategy of enforcing scoped forms, where an inference mechanism might block an inference based on a specific scoping, once such a scoping has been chosen. But this seems less natural than the treatment afforded by SNFs, which would tend to implicitly apply the scoping which is plausible in the context of whatever inference might refer to the sentence downstream in the discourse. For example, if the sentence ‘This season, every man loves a woman’ is uttered by a showmaster and followed by ‘And here she is; please welcome my next studio guest. . . ’, then a contradiction may not be perceived as such. Similarly, the sentence ‘Every child needs a clean planet’ might be uttered in a science fiction context, followed by ‘For £98, you can buy yours now’.

We can also use McDonald’s decompositions to further develop our intuitions surrounding SNFs. Under an SNF interpretation, the statement ‘Every company gave to a politician’ would fall apart into the conjunction of two factoids: ‘Every company gave to someone’, and ‘Someone gave to a politician’. So, the sentence implies ‘Every company gave’, ‘A politician received’, and, of course, it would imply itself, ‘Every company gave to a politician’.

In addition, we need to know which two factoids refer to the same event, so as to block the inference ‘Every company gave to a politician and every church gave to a charity’, ∴

'Every company gave to a charity and every church gave to a politician'. We achieve this by reifying event variables into logical constants, so that the two predicates

$$|\text{gave}|\,(\ \text{KEY} = /e_1/,\ \text{arg1} = x_1\ ), \qquad\qquad |\text{gave}|\,(\ \text{KEY} = /e_1/,\ \text{arg2} = x_2\ ),$$

would relate the subject and object to the same event, but not to a distinct event

$$|\text{gave}|\,(\ \text{KEY} = /e_2/,\ \text{arg1} = x_1\ ), \qquad\qquad |\text{gave}|\,(\ \text{KEY} = /e_2/,\ \text{arg2} = x_2\ ).$$

4.4.3. More on SNF Conversion

As we have seen, the main idea behind SNF conversion is to go through a ProtoForm one quantification at a time, and filtering its restrictor and body for the predications which depend on the quantified variable.

One further complication in this context is that, before this algorithm can be applied, the ProtoForm must be brought into a form which is maximally recursive except for the fact that quantifier holes do not get plugged. So all holes, other than quantifier holes, which have an invariant plugging in a scoping machinery get filled in by ProtoForms first.

Then, during SNF conversion, whenever the filtering mechanism hits a subordinate ProtoForm where at least one predication refers to the filter variable, all other predications must be kept as well. If this ProtoForm is nested inside other ProtoForms, all superordinate ProtoForms must be kept. In other words: removal of a subform during filtering can only occur, if this subform (a) does not reference the filter variable, (b) occurs either in the top-level ProtoForm, or inside a ProtoForm where no other subform refers to the filter variable, and where (c) this ProtoForm does not, itself, contain subordinate ProtoForms which are nonempty after filtering. – This essentially means that semantic heads never get broken up.

For ERG-derived ProtoForms, the additional condition (b) is important for adverbial modification. Consider for example the sentence 'No company gave grudgingly'. Here, the SP which establishes the subject would be

$$\left[\ |\text{no}|_{(x)} \left[\ |\text{company}|\,(\ \text{KEY} = x\ )\right] \begin{bmatrix} |\text{gave}|\,(\ \text{KEY} = /e_1/,\ \text{arg1} = x\ ), \\ \text{--}\ \&\ \text{--}, \\ |\text{grudgingly}|\,(\ \text{KEY} = /e_1/\ ) \end{bmatrix}\right]$$

In the body of this restrictor, we cannot remove the conjunction and the predication |grudgingly|, as this would incorrectly imply that 'No company gave'.

Condition (c) is important in cases where the quantifier binds the variable of a predicate which appears nested inside a modalification. So in the sentence 'Abrams said that Browne arrived', the SP which establishes the subject to |arrived| would be[6]

$$\Big[\,\text{ɿ}\,_{(x)}\,\Big[\,|\mathsf{Browne}|\,(\,\textsf{KEY} = x\,)\Big]\;\Big[\,|\mathsf{said}|\,(\,\textsf{KEY} = /e_1/\,)\;\Big[\,|\mathsf{arrived}|\,(\,\textsf{KEY} = /e_2/,\ \mathsf{arg1} = x\,)\Big]\Big]\Big]\,.$$

4.5. Operator Grammar & SNF Dependency Structures

In the previous section, we used the logical interpretation to show how SNFs can be obtained from ProtoForms coming out of the composition process. In this section, we will show how the logical interpretation relates to dependency structures.

In particular, we will discuss metatheoretic properties of various dependency-style representation schemes. In particular, we will use Briscoe-Carroll-style GRs (Carroll et al. 1999) as a point of reference, and briefly discuss dependency MRS (Copestake 2009).

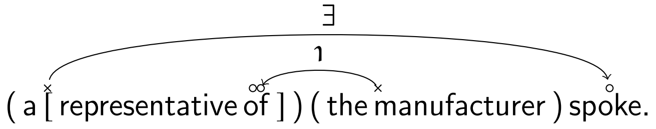
By the metatheoretic properties of a dependency structure, we mean theoretic properties which hold for the concrete symbols in the graph, not properties associated with the linguistic information they implicitly represent. As far as the latter is concerned, all three representations, SNFs, DMRSs, and GRs, are, to some extent, equivalent. For example, both SNFs and DMRSs can be obtained from the same MRS algebra, and, in the case of the ERG, the same grammar, and the same algebra might also be used to project GRs to such structures. So, as far as the most important linguistic properties are concerned, the three types of dependency structures are informationally equivalent.

For the purposes of this section, however, we will disregard such information which is implicit, and concentrate only on that information which is explicit in the associated graph structures. This distinction is particularly important, for example, for machine learning. If we feed dependency structures into a feature space by making each dependency a dimension in such a space, or if we use graph kernels, or, based on projectivity properties, tree kernels to represent the structures, we would expect a machine learner to be able to make use of the geometry of a feature space, or properties which arise directly from the graph-hood or tree-hood of a data structure. But it would be, in my opinion, naïve to suppose that a machine learner could learn, on the basis of GRs, a system of relationships among GR patterns which implicitly amounts to semantic composition, or that it could somehow pick up on the solution of a scope-underspecification problem represented by a DMRS. Even if it could, this would be unnecessary, if we can use a representation scheme where those kinds of information (semantic composition, scoping) are explicit, rather than implict.

[6] Recall that the symbol 'ɿ' is an upside-down iota and stands for Russell's definite description quantifier. We do not implement this quantifier in the model theory, but the symbol is nevertheless useful as a translation of the quantifier 'the', or for named entities.

4.5.1. From SNF Logical Forms to SNF Dependency Structures

Using the process described in the previous section (section 4.4), we can convert ProtoForms to SNFs in logical notation. But we can interpret such structures not only as logical formulae, but also as dependency structures like this:



This notation builds on the notation we have previously used for semantic structures, adding crosses to words and ProtoForms, circles to words, and arrows which always run from a cross to a circle. Each arrow corresponds to an SP. For example, the longer arrow in the above example would correspond to the following SP:

$$\begin{bmatrix} |\mathsf{a}|\ (x) & \begin{bmatrix} |\mathsf{representative}|\ (\textsc{key} = x), \\ \_ \ \&\ \_, \\ |\mathsf{of}|\ (\mathrm{arg1} = x) \end{bmatrix} & \begin{bmatrix} |\mathsf{spoke}|\ (\textsc{key} = /e_1/,\ \mathrm{arg1} = x) \end{bmatrix} \end{bmatrix}$$

We represent such a formula by drawing a cross above the ProtoForm in the restrictor of the quantification. We then identify the word in the body of the quantification which refers to the variable being quantified and represent its arguments by circles in the dependency notation. These are the arguments the predicate would have had in a ProtoForm before SNF decomposition, i.e. we are not speaking of the result of breaking up the predicate here, but rather of the predicate in its original lexical form. The circles, from left to right, correspond to arg1, arg2, and arg3. One of these arguments will be the one which refers to the variable being quantified in the SP, and this corresponds to the circle which defines the endpoint of the arrow. The label of the arrow is the quantifier itself, which is originally a word, or a quantifier induced by the grammar for purposes of representing a semantic construction.

In cases where this quantifier has a clear interpretation as a first-order quantifier, we can label the arrow with this quantifier to indicate the logical relationship being expressed by the semantic dependency. In many cases, we will omit this label, as it is a nontrivial problem and outside the scope of this work to define logical operators which adequately interpret these words and grammatical constructions.

4.5.2. Words & Syntactic Theory

Such SNF dependency structures are very closely related to dependency structures in operator grammar. In what follows, we will summarize the fundamental metatheoretic ideas behind operator grammar as envisioned by Harris (1991), and show, by the use of examples, that the relevant properties are fulfilled by SNF dependencies, but not, for example, by GRs and DMRSs.

Harris' entire "theory of sentences" (Harris 1991, p. 53) is built on little more than a few fundamental constraints on likelihood classes for word occurences and the nature of grammatical dependence. These constraints are, in fact, so basic, that one is quick to accept them as necessary for any theory of grammar and, in discarding them as trivial, to miss his main message, viz. that they are also very nearly sufficient:

> "At various points, the conclusions which are reached here turn out to be similar to well-known views of language [...]. The intent of this work, however, was not so much to arrive at such conclusions, as to arrive at them from first principles [...] The issue was not so much what was so, as whether and how the essential properties of language made it so." (Harris 1991, p. 6)

In particular:

> "The crucial property of language is that the presence of words in a sentence depends on how other words in the sentence depend on yet other words in it. This dependence is the essential constraint on equiprobability of words in sentences. Though more or less empirically come by, this dependence can be considered a construct of the syntactic theory." (Harris 1991, p. 54)

Note that he does not say that dependencies determine equiprobabilities, only that dependencies yield a constraint on equiprobabilities. In particular, this might refer to the viewpoint that they only serve to distinguish zero likelihoods from non-zero likelihoods.

Also note that he does not speak of dependencies on words, but rather of dependencies on dependencies on words. One way of fulfilling this dependence-on-dependence constraint is by expressing the algebra of dependence not at all as an algebra over words, but as an algebra over equivalence classes of words which enter into the same dependencies.

If, for example, our theory of syntax is such that the words in the classical example sentence 'Colorless green ideas sleep furiously' and the sentence 'Small young companies innovate tirelessly' enter into the same dependencies, then that would amount to the same constraint on equiprobabilty, viz. that the probability of the first is nonzero iff the probability of the second is nonzero. – By probability, we mean the expected occurence frequency of the expression in the language.

For the equivalence classes, this would mean that pairs of words entering into the same dependencies in the two sentences have to be assigned to the same equivalence class.

Conversely, if words such as 'black' and 'white' occur in the same context always either both with zero or both with nonzero probability, our theory of syntax must be such that they belong to the same equivalence classes and thus enter into the same dependencies.

So, concerning the sentence 'Colorless green ideas sleep furiously', we would not say that it has a zero likelihood of occurence, nor that it is nonsensical or that it has no meaning or semantic interpretation. Its likelihood of occurence may, of course, be so close to zero as to be, in practice, unobservable, but the same thing is true of sentences that are perfectly sensible if they have sufficiently nontrivial semantics.[7] This is notably different from an unlexicalized approach, which would go much further in saying not only that both example sentences must have nonzero likelihoods, but furthermore that they, indeed, have the same likelihood of occurence.

Topics surrounding polysemy, metonymy, and other productive phenomena which often come under the heading of lexical semantics are dealt with by Harris using further inequalities of likelihood besides the zero/non-zero distinction. This would be the sort of theory we would need to employ in order to interpret the sentence about green ideas.[8] This is outside the scope of our work, but it's nevertheless important to realize that, under this viewpoint, compositional semantics informs syntax, and lexical semantics informs compositional semantics, in the same way as syntax informs compositional semantics and compositional semantics informs lexical semantics. To the extent that one comes across citations of Harris in the context of bag-of-words semantics, one must keep in mind that this is almost certainly not what he had in mind.

All of this seems unsurprising to the linguistically informed reader. We have arrived simply at the idea of subsuming words under tags and writing grammars for tag sequences rather than individual words, yet have somehow managed to turn it into a two-page exegesis of three sentences written by Harris. – But what is important here is not what we have concluded, but how we have arrived at this conclusion. We have not yet had any need to define what a word is, how to assign tags to words, and how to assign grammatical dependencies to tag sequences: A word is simply the sort of thing which goes into equivalence classes that enter into dependencies. And dependencies are the sort of thing which establish relations between equivalence classes of words. This is taken by Harris as axiomatic and as a characteristic property which is inherent to the nature of linguistic distributionality. And it is this we must have in mind when addressing the much more complex problem of what constitutes a word, what defines a word class, and what makes a good grammatical dependency. This is different from starting out with a given theory of grammar and concluding what is merely a special case of this axiom.

[7] In that sense, word-classes in Harris' methodology might be seen as serving a similar sort of purpose as confidence intervals in statistical methodology when dealing with random variates over continuous ranges.

[8] Here is my submission to the contest: The green party is in a crisis. Their ideas appear to the electorate as colorless. Some also say the Greens are sleeping when it comes to addressing the relevant issues. Their lack of policial capital is leaving members furious. – a 39-word outline of a newspaper article with the headline "Colorless Green Ideas Sleep Furiously".

Having arrived at this point, we can now move on to put forward a working hypothesis which may, indeed, seem surprising, particularly to the linguistically informed reader: Under the metatheoretical constraints of operator grammar, it seems no less justifiable to define a word as what we have so far been calling a semantic head than it is to define a word as the sort of thing which occurs between white spaces.

Under this viewpoint, the semantic head [pay in] has all the properties in the metatheory surrounding SNF dependency structures which a word has in Harris' operator grammar. This is true, despite the fact that it is not even a subsequence of consecutive space-deliminted tokens in a string like 'Scillonians pay taxes in Britain' which might produce such a semantic head. And the dependencies which SNFs impose on semantic heads have all the properties which are imposed on words by operator grammar. This seems to be more than coincidental. In fact, Harris explicitly mentions in a footnote that the dependencies in his operator grammar "have similarity to the predicate structure in Aristotelian logic" (Harris 1991, p. 28). In what follows, we will argue that the similarity is perhaps much more concrete than he might have expected.

4.5.3. Dependence & Distributional Analysis

In operator grammar, dependence is defined as follows:

> "If A is a simple word and $b, \dots, e$ is an ordered set of classes of simple words, then A is said to depend on (or, require) $b, \dots, e$ if and only if for every sentence in the base, if A is in the sentence then there occurs in the sentence a sequence of simple words $B \dots E$ which are respectively members of $b, \dots, e$. Within the given sentence, A may then be said to depend on the word sequence $B \dots E$. If in the given sentence there is no other word G such that A depends on G and G depends on the given occurence of $B \dots E$, then A depends immediately on that occurence of $B \dots E$. A is then called the operator on that $B \dots E$, which in turn is called the argument of A in the sentence; B may be called the first argument, and so on." (Harris 1991, p. 55)

He then goes on to discuss dependence by using examples on how verbs depend on the presence of nouns. For example, an intransitive verb would act as an operator, denoted O, and depend on the presence of an argument, denoted N due to the fact that it will usually be a noun, as a subject. Such an intransitive verb would be of a word-class denoted O_N, as it is an operator which takes one argument. A transitive verb (O_{NN}) would, in addition, require a second argument as a direct object, and a ditransitive (O_{NNN}) would require a third argument as an indirect object. A coordinating sentential conjunction (O_{OO}) would have two operators as arguments, etc. – Let us discuss briefly how this idea of dependency differs from the one which is underlying GRs.

Harris does not put labels on dependencies as the Carroll-Briscoe scheme does (subj, dobj, obj2, iobj, xcomp, ...). In Harris' scheme, a dependency either exists or not. If it does exist then, by making a word depend on a sequence of word classes, rather than an unordered set, all we can say from the point of view of operator grammar is that subj and comp are distinct arguments. We would not try to assign a label with any kind of interpretation which is significant beyond distinguishing the arguments of a given word.

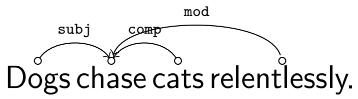
Harris also gives an empirical test which makes dependencies trivial to recognize for any given word class: A dependency of a word A on a word B of class b can be ruled out, as soon as we can show the existence of one sentence in which A occurs without any word of class b. This test is qualified only by requiring that the sentence providing such a counterexample be chosen from the basic fragment, i.e. a sentence where transformations or reductions have not occured.

But, conversely, this means that, if we know that A must depend on a b in any sentence, we know that a sequence of words in which A does occur without a b is not licensed syntactically. Now recall that, to Harris, syntax is about the distinction between zero and non-zero occurence likelihoods for word sequences, not about the magnitudes of such likelihoods. And what we have here is a test, based on syntactic theory, which distinguishes a zero likelihood of occurence from a likelihood of occurence which is not zero, but which can be arbitrarily close to zero. This is a distinction which would otherwise be very difficult or impossible to make on purely empirical grounds.

But a similar sort of empirical test could not reveal distinctions between GR labels. Consider, for example, the conj relation as in 'Jones arrived or Smith left' and the cmod relation as in 'Jones arrived because Smith left'. If one does insist on this distinction, one would have to admit that the empirical evidence which informs it is overt only in some but not all sentences, hence a single counterexample in which the distinction is not present would not rule out the possibility that the distinction could exist.

So, an important implication of Harris' approach to distributional analysis is that it precludes the possibility of using optionality as an empirical test identifying a type of dependency, as optionality is the very thing which rules out the existence of a dependency in Harris' methodology.

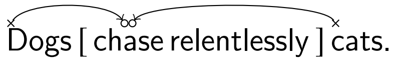
For example, when, in the sentences 'He gave an example' and 'He gave me an example', we feel tempted to postulate an optional complement to 'gave', Harris would admit as basic only the latter sentence, deriving the former by a reduction (i.e. ellipsis-like) process. This has important implications on the dependencies which, in the GR scheme, fall under the mod type, as optionality is an important empirical test for identifying them. Consider, for example, the sentence



Clearly, we must distinguish the sort of dependency which exists between 'chase' and 'relentlessly' from the sort of dependency which exists between 'chase' and 'Dogs' or 'cats'. In operator grammar, we do not allow ourselves to make the distinction simply by putting labels on dependencies.

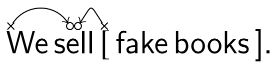
Harris' solution lies in using a transformation, whereby the above sentence results from transformation of 'Dogs chase cats, which is relentless', which, in turn would result from 'Dogs chase cats; Dogs' chasing of cats is relentless'. Here we can see that, in the second clause, 'chase' and 'relentless' provide non-optional arguments to the copula.

Our solution would be to simply remain agnostic to the exact mechanism by which the relation comes about and to allow the semantic head [chase relentlessly] to enter into the dependency structure as a single operator:



Although, rather than just remaining agnostic to the internal structure of [chase relentlessly], one might also say that it reflects precisely the meaning of Harris' second clause: 'The chasing is relentless'.

The same also goes for adjectives, where we would assign dependency structures like



As far as compositional semantics is concerned, there is no reason why we would need to work out the internal structure of the semantic head 'fake book'. In fact, there is good reason to believe that this should be left to the domain of lexical semantics and ontology. Just to repeat the usual argument: An illegal gun is a gun, a fake gun is not a gun. One might suspect that a fake book is similarly not a book, but to a Jazz musician it will be known that a fake book is a book containing simplified scores helping one "fake" a song, as opposed to playing the notes which exactly reproduce it.[9]

[9] Noun compounds in English can become rather complex. One example (proper attribution unfortunately unknown to me), is a sign at Gatwick Airport in the late 70s / early 80s, which read 'airport long term car park courtesy vehicle pick up point'. When faced with this phrase, the approach of putting it in brackets and accepting defeat may just be the honest thing to do for a grammar. One would almost certainly have to resort to the use of real world knowledge, as derived from lexical semantics or ontology, to work out the system of relationships implied.

4.5.4. Projectivity Properties

Harris goes on to say the following about dependencies and their relation to observable word sequences:

> "When the sentence consists only of A and its $B \ldots E$, the operator is necessarily contiguous to its arguments – before, after, or between them. When another word F depends on A as A depends on $B \ldots E$, the constraint would be simplest if the F is similarly contiguous (the resultant of) A together with its $B \ldots E$; this is indeed seen to be the case. The contiguity of operator to argument in the base makes it easy to check if no word G as above intervenes in the operator-argument relation of A to $B \ldots E$. In *I know sheep eat grass*, *know* is the operator on the pair *I, eat* as its argument, with *eat* in turn as the operator on the pair *sheep, grass* as argument." (Harris 1991, p. 55)

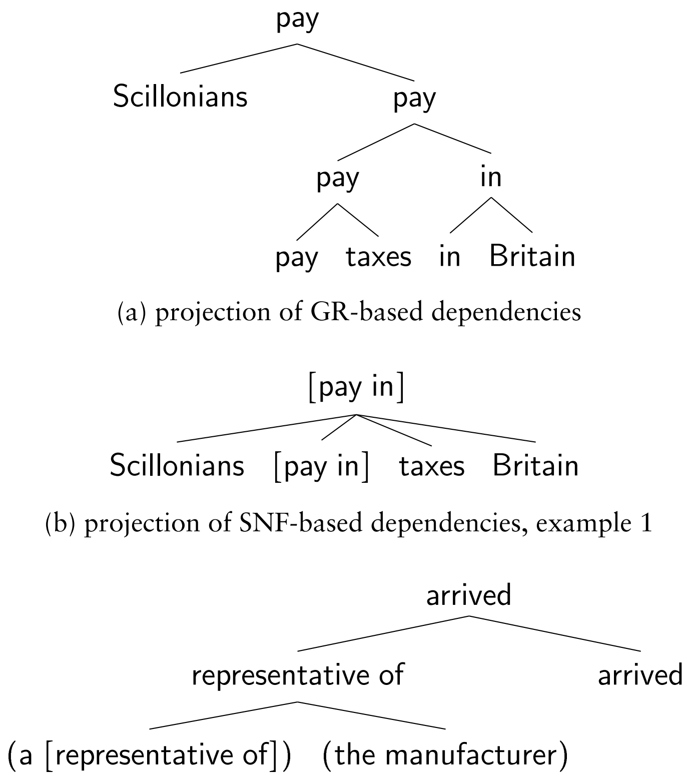
The notion of dependency which we arrive at by methodological observation of co-occurences, as described in the previous section, is transitively closed. By looking only at the set of words which co-occur in a sentence, we cannot distinguish the situation where one word directly depends on another from the situation where one word depends on a word which depends on the other. In order to make this distinction, we have to look not at a set of words, but at a sequence of words, and devise an empirical test which makes the distinction.

Harris gives such a test, which is based on the projectivity property of dependency structures. For GR-based dependency structures, the projectivity property means that the projection tree looks like a constituency tree. This is shown in Figure 4.4. But the projectivity property is also fulfilled for Harris' operator grammar, and for SNF-based dependency structures. This is true, despite the fact that a semantic head like [pay in] is not a set of words which occur consecutively, as Harris does not require this stronger form of projectivity. He merely requires that the projection under every operator be a set of words which are consecutive.

In the syntax tree, this means that the following sets of words must be consecutive:

- $\{\textsf{pay}, \textsf{taxes}\}$,
- $\{\textsf{in}, \textsf{Britain}\}$,
- $\{\textsf{pay}, \textsf{taxes}, \textsf{in}, \textsf{Britain}\}$,
- $\{\textsf{Scillonians}, \textsf{pay}, \textsf{taxes}, \textsf{in}, \textsf{Britain}\}$

In the SNF dependency tree for the same sentence, this reduces to the somewhat trivial observation that the words which must be consecutive are $\{\textsf{Scillonians}, \textsf{pay}, \textsf{in}, \textsf{taxes}, \textsf{Britain}\}$. Looking at a more nontrivial example, however, we might arrive at the following sets of words which must be consecutive:



(a) projection of GR-based dependencies

(b) projection of SNF-based dependencies, example 1

(c) projection of SNF-based dependencies, example 2

Figure 4.4.: projections of dependency structures

- {the, manufacturer},
- {a, representative, of, the, manufacturer},
- {a, representative, of, the, manufacturer, arrived}.

Another interesting piece of information we get from the projection structure is the locations of the splits which we can find along any line of heads which appear at the same vertical level in the projection tree. For example in a syntax tree for 'Most white cats are deaf', we would get, among others, the split

{Most, white, cats} / {are, deaf}.

The simple metatheoretic property that there is a split occuring here somewhere between 'white' and 'deaf' helps us distinguish the sentence from 'Most deaf cats are white'. Such a split also occurs in an SNF dependency structure, and in a logical formula in FOPC.

This is notably different from DMRS. Copestake (2009) explicitly mentions this example and encodes the distinction by putting labels on DMRS dependencies which indicate whether or not the codependants form what we would call a semantic head. Furthermore, Copestake (2009) mentions that EPs in DMRSs can have multiple heads, and that a head and its dependants are not necessarily consecutive in DMRS.

In essence, what we have shown in this section is that SNFs provide interpretations for the sort of information which is encoded in DMRSes in the form of labels on dependencies. By making this information explicit, which is only implicit in DMRS, we have arrived at a dependency structure which, in terms of its metatheoretic properties, looks more like what we would expect of a dependency structure. As we have pointed out earlier, this might be useful for machine learning and in lexical semantics, where one would not expect a machine learner to automatically pick up on what the equalities and inequalities in a DMRS actually mean, whereas one might reasonably expect, for example, a tree-kernel to make use of the information which comes with splits in a tree.

5. Monte Carlo Semantics

We now have all the relevant pieces of the puzzle in place. In this, final, chapter, we can put them together and describe the design of a rudimentary inference engine based on the theory we have developed. (Appendix C gives pointers concerning its implementation).

Recall from the statement of our main goals in the introduction (chapter 1), that we require two properties of our reasoning mechanism: (1) semantic informativity, the ability to take into account all available information; and (2) robustness, the ability to proceed on reasonable assumptions where such information is missing.

In section 5.1, we will give a description of what we mean by informativity and robustness and then, in section 5.2, we will discuss some ideas on how one might optimize the informativity/robustness tradeoff through deep/shallow integration. In particular, we will show how our theoretical framework allows us to formulate a theory of logic and epistemology which has bag-of-words logic as a limit case on the shallow end of the spectrum, and traditional theorem proving as a limit case on the deep end of the spectrum. Finally, section 5.3 describes a reasoning mechanism which addresses the general case.

5.1. The Informativity/Robustness Tradeoff

5.1.1. Semantic Informativity

Figure 5.1 gives some examples of candidate inferences one might consider in connection with a semantic informativity claim.

The non-starred examples 5.1–5.8, given all the necessary lexico-grammatical information, can be handled within our approach, while the starred examples *5.9–*5.13 all rely partially on knowledge sources which are outside the scope of our present treatment. Our system can therefore not reproduce these decisions in absolute terms. However, in the next section we will see that the robustness properties of our approach still apply, so the presence of these phenomena in inferences is not a problem as such.

Examples 5.1 and 5.2 demonstrate insertion and deletion of words in a sentence, where, depending on the semantic scope, such an insertion or deletion may or may not be licensed by the logical operators involved. Similarly, examples 5.3 and 5.4 demonstrate

$$\begin{array}{l}\text{Socrates is a Greek \& man.}\\ \hline \therefore\ \text{Socrates is a man.}\end{array} \qquad (5.1\text{a})$$

$$\begin{array}{l}\text{Socrates is a man.}\\ \hline \not\therefore\ \text{Socrates is a Greek \& man.}\end{array} \qquad (5.1\text{b})$$

$$\begin{array}{l}\text{Every Greek \& man is mortal.}\\ \hline \not\therefore\ \text{Every man is mortal.}\end{array} \qquad (5.2\text{a})$$

$$\begin{array}{l}\text{Every man is mortal.}\\ \hline \therefore\ \text{Every Greek \& man is mortal.}\end{array} \qquad (5.2\text{b})$$

(a) Insertions into and deletions from different semantic scopes

$$\begin{array}{l}\text{Some mortal \& man is Greek.}\\ \hline \therefore\ \text{Some Greek \& man is mortal.}\end{array} \qquad (5.3)$$

$$\begin{array}{l}\text{Every mortal \& man is Greek.}\\ \hline \not\therefore\ \text{Every Greek \& man is mortal.}\end{array} \qquad (5.4)$$

(b) Movement across semantic scopes

$$\begin{array}{l}\text{Socrates is a man.}\\ \hline \not\therefore\ \text{Socrates is not a man.}\end{array} \qquad (5.5)$$

$$\begin{array}{l}\text{Every man is mortal.}\\ \hline \not\therefore\ \text{Not every man is mortal.}\end{array} \qquad (5.6)$$

$$\begin{array}{l}\text{Socrates is a man and every man is mortal.}\\ \hline \therefore\ \text{Socrates is mortal.}\end{array} \qquad (5.7)$$

$$\begin{array}{l}\text{Socrates is a man and some men are mortal.}\\ \hline \not\therefore\ \text{Socrates is mortal.}\end{array} \qquad (5.8)$$

(c) Logical interpretations

$$\begin{array}{l}\text{Socrates addressed his accusers at the court.}\\ \hline \text{Socrates was at the court.}\end{array} \qquad (*5.9)$$

$$\begin{array}{l}\text{After Meletus accused Socrates, he apologized.}\\ \hline \text{Socrates apologized.}\end{array} \qquad (*5.10)$$

(d) PP-attachment, anaphoric reference

$$\begin{array}{l}\text{Socrates drank hemlock.}\\ \hline \therefore\ \text{Socrates drank poison.}\end{array} \qquad (*5.11)$$

$$\begin{array}{l}\text{Socrates drank the poison and died.}\\ \hline \therefore\ \text{Socrates is mortal.}\end{array} \qquad (*5.12)$$

$$\begin{array}{l}\text{Socrates is an Athenian \& man.}\\ \hline \therefore\ \text{Socrates is an Athenian \& citizen.}\end{array} \qquad (*5.13\text{a})$$

$$\begin{array}{l}\text{Xanthippe is an Athenian \& woman.}\\ \hline \not\therefore\ \text{Xanthippe is an Athenian \& citizen.}\end{array} \qquad (*5.13\text{b})$$

(e) Lexical knowledge, common sense knowledge, world knowledge

Figure 5.1.: Different kinds of information and background knowledge

the movement of words across semantic scopes. This is valid in the case of the restrictor and body of a some quantifier, but not in the case of an every quantifier. Our approach can correctly decide such inferences due to the fact that SNF decompositions would put the different semantic heads into different quantifier scopes. Our logic assigns a model-theoretic interpretation to the quantifiers involved and the conjunction operator, leading to correct decisions concerning inference phenomena of quantifier scope.

Examples 5.5–5.8 demonstrate inference patterns that require the inference engine to interpret logical function words such as determiners (e.g. 'every', 'some', 'a'), coordinators (e.g. 'and'), the copula ('is', 'are', . . .), or the negation not (e.g. 'is not', 'not every'). Logical interpretations also need to be applied to operators that do not directly correspond to words but are introduced by the grammar. For example, following our previous definitions of semantic composition (section 4.2), we denote the operator for intersective composition of an `Adj` and an `N'` as '&'. All of these are interpreted within our approach by model-theoretic implementations of the logical operators involved.

Example *5.9 demonstrates a PP-attachment ambiguity. Despite the fact that we have not had anything to say herein about problems such as this, a parse selection mechanism might be able to contribute this knowledge to our approach. The two different PP-attachments would lead to different MRS-based compositions, which, in turn would be reflected in different logical formulae, even after SNF decomposition. At this point, our inference engine would take the information into account model-theoretically. For the FraCaS-based experiment (appendix D), such problems are taken care of by using hand-selected syntactic analyses.

As we do not attempt to model discourse phenomena such as anaphora, candidate inferences of the type of example *5.10 are not addressed by our approach either.

Inferences such as examples *5.11–*5.13 require particular background knowledge. We need to know that 'Hemlock is poison' in example *5.11, that 'Socrates is a person and every person who dies is mortal' in example *5.12, and that 'Athenian women are not citizens' in example *5.13b. Note that we would be perfectly able to draw these inferences, if those sentences were part of the stated antecedents. As for example *5.11, we might be able to draw such an inference if the required knowledge is represented in WordNet as a hyponymy link. The kind of common sense knowledge needed for example *5.12, however, is not a straightforward case of hyponymy, as it involves a more complex relationship between entities of different syntactic types. Example *5.13, although it could be justified on the basis of straightforward hyponymy, as in example *5.11, presupposes ancient Greece as a discourse context and particular knowledge of a historic fact.

It should be stressed once again that all of these examples are merely desiderata. When the required lexico-grammatical information is present, our inference should fulfill them by guaranteeing that all candidate inferences of the types we just discussed will be decided correctly. This is equally true for the traditional approach which uses FOPC translation. As a matter of fact, in this special case where all lexico-grammatical information is present, our approach reduces to being functionally equivalent to an FOPC theorem prover. However, in addition to this, our approach takes into account the case of missing lexico-grammatical information.

5.1.2. Robustness

We call a reasoning mechanism robust, iff it makes use of heuristics which enable it to proceed on reasonable assumptions where hard information is missing. Figure 5.2 gives some examples of candidate inferences one might consider in connection with a robustness claim.

Our particular robustness heuristic does not decide upon the validity of a candidate inference in absolute terms. It only enables comparisons between candidate inferences, deciding, for a given pair of candidate inferences, which should be favoured.

We will see in section 5.2.2 how such comparisons are possible as a result of our many-valued logic and as a result of relaxing the usual completeness assumption concerning background knowledge in moving towards a probabilistic model of uncertainty about such information.

Throughout the rest of this section, we will discuss on purely intuitive grounds how the examples in Figure 5.2 can be decided even in the absence of background knowledge (Figures 5.2a and 5.2b) or lexico-grammatical knowledge (Figures 5.2c and 5.2d).

The examples in Figure 5.2a all rely on missing background knowledge about relations between certain words (e.g. philosopher and heretic). The inferences demonstrate the impact of the insertion or removal of a word (e.g. Greek) into a given semantic scope or its movement across semantic scopes.

Our approach here will be to let these preferences, intuitively, quantify the number of syllogistic premises which would have to be added to the background theory in order to make the inference valid, or it could be seen as quantifying the minimal number of illegal steps necessary in a proof of the candidate inference.

For example, if it were the case that every philosopher is an heretic, then the right-hand inference in example 5.14 would have to go through. But in order to justify the left-hand inference as well, we would, in addition to that, have to assume that everything Greek is mortal. Intuitively, the right-hand inference is easier to justify. One can also argue that the left-hand inference requires two invalid substitutions while the right-hand inference

$$\frac{\text{Some Greek \& heretic is mortal.}}{\text{./. Some old \& philosopher is mortal.}} < \frac{\text{Some Greek \& heretic is mortal.}}{\text{./. Some philosopher is mortal.}} \tag{5.14}$$

$$\frac{\text{Some heretic is mortal.}}{\text{./. Some philosopher is mortal.}} > \frac{\text{Some heretic is mortal.}}{\text{./. Some Greek \& philosopher is mortal.}} \tag{5.15}$$

$$\frac{\text{Some Greek \& heretic is mortal.}}{\text{./. Some Greek \& philosopher is mortal.}} = \frac{\text{Some Greek \& heretic is mortal.}}{\text{./. Some mortal \& philosopher is Greek.}} \tag{5.16}$$

$$\frac{\text{Every Greek \& heretic is mortal.}}{\text{./. Every old \& philosopher is mortal.}} > \frac{\text{Every Greek \& heretic is mortal.}}{\text{./. Every philosopher is mortal.}} \tag{5.17}$$

$$\frac{\text{Every heretic is mortal.}}{\text{./. Every philosopher is mortal.}} < \frac{\text{Every heretic is mortal.}}{\text{./. Every Greek \& philosopher is mortal.}} \tag{5.18}$$

$$\frac{\text{Every Greek \& heretic is mortal.}}{\text{./. Every Greek \& philosopher is mortal.}} \neq \frac{\text{Every Greek \& heretic is mortal.}}{\text{./. Every mortal \& philosopher is Greek.}} \tag{5.19}$$

(a) Quantifiers, world knowledge missing only

$$\frac{\text{Meletus accused Socrates.}}{\text{./. The people accused Socrates.}} > \frac{\text{Meletus accused Socrates.}}{\text{./. Socrates accused the people.}} \tag{5.20}$$

(b) Predicate arguments, world knowledge missing only

$$\frac{\text{Some\&Greek\&heretic\&is\&mortal.}}{\text{./. Some\&old\&philosopher\&is\&mortal.}} < \frac{\text{Some\&Greek\&heretic\&is\&mortal.}}{\text{./. Some\&philosopher\&is\&mortal.}} \tag{5.21}$$

$$\frac{\text{Some\&heretic\&is\&mortal.}}{\text{./. Some\&philosopher\&is\&mortal.}} > \frac{\text{Some\&heretic\&is\&mortal.}}{\text{./. Some\&Greek\&philosopher\&is\&mortal.}} \tag{5.22}$$

$$\frac{\text{Some\&Greek\&heretic\&is\&mortal.}}{\text{./. Some\&Greek\&philosopher\&is\&mortal.}} = \frac{\text{Some\&Greek\&heretic\&is\&mortal.}}{\text{./. Some\&mortal\&philosopher\&is\&Greek.}} \tag{5.23}$$

$$\frac{\text{Every\&Greek\&heretic\&is\&mortal.}}{\text{./. Every\&old\&philosopher\&is\&mortal.}} > \frac{\text{Every\&Greek\&heretic\&is\&mortal.}}{\text{./. Every\&philosopher\&is\&mortal.}} \tag{†5.24}$$

$$\frac{\text{Every\&heretic\&is\&mortal.}}{\text{./. Every\&philosopher\&is\&mortal.}} < \frac{\text{Every\&heretic\&is\&mortal.}}{\text{./. Every\&Greek\&philosopher\&is\&mortal.}} \tag{†5.25}$$

$$\frac{\text{Every\&Greek\&heretic\&is\&mortal.}}{\text{./. Every\&Greek\&philosopher\&is\&mortal.}} \neq \frac{\text{Every\&Greek\&heretic\&is\&mortal.}}{\text{./. Every\&mortal\&philosopher\&is\&Greek.}} \tag{†5.26}$$

(c) Quantifiers, lexico-grammatical information missing as well

$$\frac{\text{Meletus\&accused\&Socrates.}}{\text{./. The\&people\&accused\&Socrates.}} > \frac{\text{Meletus\&accused\&Socrates.}}{\text{./. Socrates\&accused\&the\&people.}} \tag{†5.27}$$

(d) Predicate arguments, lexico-grammatical information missing as well

Figure 5.2.: Robust inferences

requires only one invalid substitution, the deletion being perfectly valid in this particular semantic scope.

In example 5.15, we are dealing with insertion on the right-hand side, rather than deletion, so this is an invalid step and leads to the left-hand side being preferred. In example 5.16, we have only one illegal substitution, which is the same on the left and the right. Given the antecedent and one of the consequents, we could always infer the other. So both inferences are equally well justified. As we would expect from the previous section, the effect of insertion, deletion and movement of words is the opposite for the restrictor of 'every' compared to that of 'some'. This is why the preferences reverse for examples 5.17–5.19 compared to examples 5.14–5.16.

Example 5.20 relies on missing background knowledge about the relation between the referent of the people and that of Meletus. In order to justify the left-hand inference, we only need to assume that Meletus spoke for the people. But, in order to justify the right-hand inference, we would need to postulate further background knowledge to justify the reversal of the arguments of the predicate accuse. In terms of SNF decompositions, it can also be seen that, in the left-hand inference, one of the two syllogistic premises is the same for the antecedent and the consequent (Socrates was accused), whereas, in the right-hand inference, neither of the two syllogistic premises entails the other.

We have seen that lexico-grammatical information about predicates and their arguments as well as semantic scopes can be usefully applied for inferencing purposes, even when predicates are involved which cannot be related to each other due to missing background knowledge. But traditional logic fails in this respect. It loses the information concerning preferences among the candidate inferences we just discussed, as they would all fall within the same category, being considered satisfiable but not valid, a case that logicists usually rule out by requiring completeness for the theories providing the background knowledge. We will discuss this in greater detail in section 5.2.1.

What if we are missing lexico-grammatical information about the text, as well as background knowledge? As an extreme case, assume we only have information about tokenization. We can then take the individual words as atomic expressions of the logic, in the case of SNF-based reasoning perhaps syllogistic premises asserting the existence of an individual satisfying the predicate. We would combine these atoms using the bag aggregation operator, which we write as '&'. Since this operator must not require information about the structure of the text, it is necessary that this operator be commutative and associative but not idempotent. Incidentally, the strong conjunction of Łukasiewicz logic fits the bill perfectly (section 3.1.4).

Figures 5.2c and 5.2d now demonstrate what happens when we discard the lexico-grammatical information from the inferences we were considering in Figures 5.2a and

5.2b. The insertions, deletions, and movements of words in examples 5.21–5.23, which use the some quantifier, show the same behaviour under bag-of-words logic as in examples 5.14–5.16, where the same inferences do take into account lexico-grammatical information. This, however, is not true for examples †5.24–†5.27, which we mark with a dagger to indicate that the desired preferences are *not* being modeled correctly under bag-of-words logic. These candidate inferences, involving the every quantifier, exhibit the right behaviour when, as in examples 5.17–5.19, they can rely on lexico-grammatical information, but incorrect behaviour in bag-of-words logic, where such information is missing. Example †5.27 demonstrates that the same is true for predicate argument structure, which is not correctly accounted for by bag-of-words logic.

This theoretic framing of bag-of-words logic suggests a number of reasons which might explain why bag-of-words approaches have been so successful despite their linguistic naïveté: (1) Positives significantly outnumber negatives, and particulars and definites significantly outnumber universals in any corpus of naturally occuring text. So, cases in which bag-of-words logic responds correctly to insertions, deletions, and movements of words are more frequent than cases in which it responds incorrectly. (2) A given bag of words often has a strong tendency to occur in a particular semantic configuration. For example, occurences of 'The dog bit the man' or 'The man was bitten by the dog' would outnumber occurences of 'The man bit the dog' or 'The dog was bitten by the man' due to underlying ontological relationships. Compositional semantics, however, is not concerned with this prior but only with modelling semantic interpretation as a posterior to the information represented in the text. (3) Applications requiring the classification of text or the recognition of textual patterns predominantly apply target criteria which are invariant to semantic distinctions. For example, the aboutness and relevance criteria which feature in information retrieval exhibit this behaviour. Sentences like 'Some man is mortal.', 'Some man is not mortal.', and 'Every man is mortal.' might all be equally relevant to the same sort of information needs about the mortality of man.

This explains why it is counterproductive to force text into a form of representation making more fine-grained distinctions when these distinctions cannot be made reliably. Incorrect distinctions would negatively affect the applicability of the above three properties, leading to a comparatively small gain in semantic informativity being traded off for a comparatively large loss of robustness.

It would be interesting to test the above three hypotheses in a rigorous empirical study, but they hardly seem contentious. Furthermore, it is widely recognized that bag-of-words logic is highly robust and surprisingly hard to outperform, even with techniques that might promise a great deal more semantic informativity. The results of Bos & Markert (2005*a*,*b*, 2006*a*,*b*) and of MacCartney (2009) on the RTE task certainly agree with this. For these reasons, we will study bag-of-words logic in further depth, in an attempt to carry over its robustness properties into our generalized theoretical framework.

Finally, I must, once again, stress that our technique will only fall back on robust decisions of this kind when lexico-grammatical information is missing, thus preventing more semantically informative decisions. In the case where no lexico-grammatical information is available at all, our approach reduces to being functionally equivalent to bag-of-words overlap measurement. However, our approach can also take advantage of lexico-grammatical information whenever such information is provided.

5.2. Logic/Probability & Deep/Shallow Integration

Currently, systems which either explicitly or implicitly perform textual inference employ inference techniques which can be situated anywhere along a spectrum between deep and shallow techniques. A typical example for deep techniques is the RTE system of Bos & Markert (2005*a*,*b*, 2006*a*,*b*) which parses natural language texts using the CCG-based C&C tools (Curran et al. 2007) and translates them via Boxer (Bos 2005) to DRSes and ultimately to FOPC formulae. It then applies standard FOPC theorem provers and model builders against a theory of background knowledge derived from WordNet (Fellbaum 1998). On the other hand, many RTE systems have used variants of bag-of-words encoding as a frontend to a machine learning system. These systems mark the shallow end of the spectrum. Most systems are perhaps to be positioned between the two extreme ends, employing some symbolic, usually graph-based, representation mechanisms, and aligning the graphs in some way. A literature review making this distinction has been undertaken by MacCartney (2009). It comes as little surprise though, that shallow methods are robust but not semantically informative, while deep methods are semantically informative but not robust. Intermediate-level systems provide intermediate levels of both semantic informativity and robustness, but don't escape the tradeoff altogether.

I believe that successful deep/shallow integration in textual inference requires the formulation of a unified theory, such as the one we will consider in this section, where classical theorem proving, on one hand, and bag-of-words overlap measurement, on the other, can both be understood as special cases of the more general theoretic framework. Given that, the key to deep/shallow integration is to build a system that is robust on one hand, but which, on the other hand, also has a monotonicity property concerning lexico-grammatical and ontological information. Whenever such information is provided, this property must enable the inference mechanism to match the logical consequence relation of interest more closely. Conversely, whenever such information is required but missing, this property must ensure that the missing piece of the puzzle only has a local effect, with a robustness heuristic taking over to fill in the gap.

Such a monotonicity property is currently not available. Inference mechanisms on the deep end of the spectrum rely on correct information all of the time, even though it is often unavailable, while those on the shallow end of the spectrum remain ignorant to it all of the time, even though it is often available. Intermediate-level inference mechanisms restrict themselves to relying on certain kinds of information all of the time while remaining ignorant to others all of the time. In this section, we lay some groundwork for inference engines featuring truly monotonic deep/shallow integration.

5.2.1. Deep: From Completeness & Consistency to Uncertainty

The usual approach to knowledge representation in logic is to perform reasoning within a theory T, a set of formulae accepted without proof. We write $T \vDash \chi$ iff formula χ is valid within T.

The deduction theorem then defines when exactly a candidate entailment of the form "$\varphi \to \psi$" is valid. It states that $T \vDash \varphi \to \psi$ iff $T \cup \{\varphi\} \vDash \psi$. So, if we assume that the antecedent φ is valid, in addition to the formulae already in T and we can conclude that ψ is also valid then we also know that the candidate entailment $\varphi \to \psi$ is valid in T. – A slightly modified version of this deduction theorem also holds for our $\aleph_0$-valued Łukasiewicz logic. So, when evaluating a given candidate entailment, there are traditionally four cases to distinguish with regard to our knowledge about it:

(i) $T \cup \{\varphi\} \vDash \psi$ and $T \cup \{\varphi\} \nvDash \neg\psi$;
(ii) $T \cup \{\varphi\} \nvDash \psi$ and $T \cup \{\varphi\} \vDash \neg\psi$;
(iii) $T \cup \{\varphi\} \vDash \psi$ and $T \cup \{\varphi\} \vDash \neg\psi$;
(iv) $T \cup \{\varphi\} \nvDash \psi$ and $T \cup \{\varphi\} \nvDash \neg\psi$.

Consider, for example, the following formulae:

$$\begin{aligned} \varphi &: \text{Socrates is a man,} \\ \neg\varphi &: \text{Socrates is not a man,} \\ \psi &: \text{Socrates is mortal.} \end{aligned}$$

Assuming the empty theory $T = \varnothing$, the candidate inference $\varphi \to \varphi$ falls under the case of validity (case i). The candidate inference $\varphi \to \neg\varphi$ falls under the case of unsatisfiability (case ii). It is quite common for a logicist to require that a given theory $T \cup \{\varphi\}$ be *consistent*, i.e. that case (iii) be ruled out.

But what about $\varphi \to \psi$? In the absence of further knowledge, this would be a contingency (case iv), so we are dealing with an incomplete theory. In order to make the theory $T \cup \{\varphi\}$ *complete*, we could, for example, add χ : 'Every man is mortal' to T. We would then have $\{\chi, \varphi\} \vDash \psi$ and $\{\chi, \varphi\} \nvDash \neg\psi$, so the candidate inference would now fall under case (i) and be considered valid. Or, we could have added some χ', e.g. 'No man is mortal', to make it fall under case (ii) and be considered unsatisfiable.

For practical open-domain NLP applications, this background knowledge will be, to a large extent real-world and common-sense knowledge. How can we enter this into our logical theory? Bos & Markert, for example, use WordNet. Here, one might derive

$$\forall_{(x)} \{|\mathsf{cat}|\,(\,\textsc{key} = x\,) \rightarrow |\mathsf{animal}|\,(\,\textsc{key} = x\,)\}.$$

from a noun hyponymy hierarchy, or one might get

$$\forall_{(x,y,z)} \left\{ \begin{array}{l} \exists_{(e)} \{\mathsf{buyFrom}\,(\,\textsc{key} = e,\ \textsc{arg1} = x,\ \textsc{arg2} = y,\ \textsc{arg3} = z\,)\} \\ \quad \rightarrow \exists_{(e')} \{\mathsf{sellTo}\,(\,\textsc{key} = e,\ \textsc{arg1} = z,\ \textsc{arg2} = y,\ \textsc{arg3} = x\,)\} \end{array} \right\},$$

from a role-labelled verb lexicon. Knowledge of a more general type might be automatically acquired from text, and, given careful knowledge engineering, one might even be able to ensure that the resulting theories are consistent in a logical sense. But a completeness assumption still seems unrealistic given the present state of the art in natural language processing and real-world or common-sense ontology.

So this means that we cannot rule out what we have called case (iv) by the usual completeness assumption. Quite to the contrary. One would expect almost all candidate inferences to fall under case (iv), with cases (i) and (ii) occuring only as limit cases of theoretical interest. This is due to the fact that inferences will often hinge on real-world or common-sense knowledge, which is neither represented in the grammar nor the logic. This is perhaps the central problem when it comes to applying logical reasoning techniques for NLP applications.

> "Although in theory the method of finding proofs should work, in practice it does not work that well. This is mostly due to the lack of appropriate background knowledge without which many true entailments cannot be found."
> (Bos & Markert 2005*a*)

We will return to how Bos & Markert approached the problem in the next section (section 5.2.2). Our approach is as follows.

90. Let $\Lambda = \langle \mathsf{p}_1, \mathsf{p}_2, \ldots, \mathsf{p}_N \rangle$ be a propositional signature, and let W be a set of $(\mathbb{V}_{\aleph_0}, \Lambda)$-valuations. The *degree of validity of a formula* χ *over* W *and* Λ, denoted $[\![\chi]\!]^{\Lambda}_{\mathrm{W}}$, is defined as follows:

$$[\![\chi]\!]^{\Lambda}_{\mathrm{W}} = \frac{1}{|\mathrm{W}|} \sum_{w \in \mathrm{W}} \|\chi\|^{\Lambda}_{w}.$$

This generalizes the traditional logical notion of the validity of a formula within a theory towards a graded notion of validity, which we call *degree of validity*. For example, χ could be valid to a degree of 0.7, written $[\![\chi]\!] = 0.7$. Given this generalization, we can now get a new general case (iii) with limit cases (i), and (ii):

(i) $[\![\chi]\!] = 1.0$: Here, $T \cup \{\varphi\} \vDash \psi$ and $T \cup \{\varphi\} \nvDash \neg\psi$.

(ii) $[\![\chi]\!] = 0.0$: Here, $T \cup \{\varphi\} \nvDash \psi$ and $T \cup \{\varphi\} \vDash \neg\psi$.

(iii) $0.0 < [\![\chi]\!] < 1.0$: Here, $T \cup \{\varphi\} \vDash_t \psi$ and $T \cup \{\varphi\} \vDash_{t'} \neg\psi$, for some degree of validity $0 < t, t' < 1.0$.

In the new case (iii), we can now *compare* two given candidate entailments for their degree of validity; let us call them candidate 1, denoted $T \cup \{\varphi_1\} \vDash_{t_1} \psi_1$, and candidate 2, denoted $T \cup \{\varphi_2\} \vDash_{t_2} \psi_2$. It now may well be the case that we are missing knowledge, so that neither of them is strictly provable, in a proof-theoretic sense, that neither of them is a tautology, that neither of them is traditionally valid. But we can still determine, on the basis of the information we do have in T, which of them we would rather prove than the other, which of them is closer to being a tautology, which of them is valid to a higher degree. If $t_1 > t_2$, we prefer candidate 1, if $t_2 > t_1$, we prefer candidate 2. – So, this is why a graded notion of validity is useful. But why did we define it in this particular way, and what is its interpretation?

In a first step, consider the set of all $(\mathbb{V}_2, \Lambda)$-valuations, and denote this set as $\mathcal{W}$. So $|\mathcal{W}| = 2^{|\Lambda|}$. This definition can then easily be understood both logically and probabilistically. – For its logical interpretation, first recall the definitions of the classical notions of validity and satisfiability within such a model-theoretic framework.

Here, χ is considered classically valid, iff the truth value $\|\chi\|_w$ equals 1 in *all* valuations $w \in \mathcal{W}$. We could also say, χ is classically valid iff the *minimum* truth value $\min_{w \in \mathcal{W}} \|\chi\|_w$ across all w is ≥ 1. So the formula $\mathsf{p} \to \mathsf{q}$ would not be considered valid, as it has an assignment of truth values which make it false.

Similarly, χ is considered classically satisfiable, iff $\|\chi\|_w$ equals 1 in *some* valuation w, i.e. iff the *maximum* truth value $\max_{w \in \mathcal{W}} \|\chi\|_w$ across all w is > 0. So the formula $\mathsf{p} \to \mathsf{q}$ would not be considered valid, as it has an assignment (three, in fact) of truth values which make it true.

But, from a knowledge-engineering perspective, this traditional notion of validity is too strong, and this notion of satisfiability is too weak. This is why we use a statistic between the minimum and maximum. We use an arithmetic mean. So the formula $\mathsf{p} \to \mathsf{q}$ would now be considered valid to a degree of 0.75, as it has one assignment in which its truth value is 0.0, and three assignments in which its truth value is 1.0

This definition happens to coincide precisely with the definition of probability given by (De Finetti 1974) in his treatment of subjective probability. But let us first consider the more well-known formal epistemology of frequentist probability. Here, one would think of $\|\chi\|$ as a random variable indicating the truth value $\|\chi\|_w$, when a valuation w is chosen at random. The value of $[\![\chi]\!]$ is then the probability that the truth value of χ, for such a valuation w chosen at random, is 1, assuming for this choice a uniform distribution, which is easily motivated on the basis of an assumption of maximum entropy, i.e. maximum uncertainty and minimum knowledge about the missing postulates which would make our theory complete. For De Finetti, the question is not "Why assume a uniform distribution?" but rather "Why not?", as this assumption fulfills all of his coherence axioms from which he derives the notion of probability.[1]

In the special case of a complete theory of background knowledge, our approach reduces to being functionally equivalent to classical theorem proving. When background knowledge is missing, this raises uncertainty, which is dealt with probabilistically.

5.2.2. Shallow: From Overlap Measurement to Bag-of-Words Logic

In order to illustrate the definition of graded validity, let us consider an example which, at first sight, seems to have little to do with logical inference: bag-of-words inference.

When we have a bag-of-words level of analysis for two pieces of text T and H, we can think of them in a logical representation as conjunctions, in which the atomic conjuncts are simply words. Consider, for example, the Woody Allen mood of the syllogism:

$$\begin{array}{rl} \text{(T)} & \mathsf{socrates} \,\&\, \mathsf{is} \,\&\, \mathsf{a} \,\&\, \mathsf{man} \\ \hline \rightarrow \ \text{(H)} & \mathsf{so} \,\&\, \mathsf{every} \,\&\, \mathsf{man} \,\&\, \mathsf{is} \,\&\, \mathsf{socrates} \end{array}.$$

Note that those atoms could have additional internal structure. For example, they can be syllogistic premises.

Let's call the antecedent φ and the consequent ψ, and let's try to determine the degree of validity $[\![\varphi \rightarrow \psi]\!]$ for the bivalent case. This is possible using only basic combinatorics.

Let Λ_φ be the set of propositional symbols, in this case words, appearing only in the antecedent, not in the consequent, i.e. $\Lambda_\varphi = \{\mathrm{a}\}$. Similarly, let Λ_ψ be the set of propositional symbols appearing only in the consequent, not in the antecedent, i.e. $\Lambda_\psi = \{\mathrm{so}, \mathrm{every}\}$. Finally, let Λ_ω be the overlap, i.e. the set of propositional symbols appearing both in the antecedent and the consequent; $\Lambda_\omega = \{\mathrm{socrates}, \mathrm{is}, \mathrm{man}\}$.

There are $\mathrm{N} = |\Lambda_\varphi \cup \Lambda_\psi \cup \Lambda_\omega| = 6$ atomic propositions. We are dealing with the bivalent case, so there are $2^{\mathrm{N}} = 2^6 = 64$ possible valuations for this signature altogether. There are $2^{|\Lambda_\varphi|} = 2^1 = 2$ ways of assigning truth values to the antecedent, $2^{|\Lambda_\psi|} = 2^2 = 4$ ways of

[1] Another interesting property of De Finetti's theory of probability is that it readily deals with the generalization where we move from the bivalent case of using only $(\mathbb{V}_2, \Lambda)$-valuations to the $\aleph_0$-valued case of choosing random $(\mathbb{V}_{\aleph_0}, \Lambda)$-valuations. De Finetti calls this a *prevision* of the random quantity $\|\chi\|$ and establishes the mathematical properties of prevision in great detail.

assigning truth values to the consequent, and $2^{|\Lambda_\omega|} = 2^3 = 8$ ways of assigning truth values to the overlap.

In order to make the implication $\varphi \to \psi$ false, we must make the antecedent φ true, and the consequent ψ false. Clearly, only one out of the $2^{|\Lambda_\varphi \cup \Lambda_\omega|} = 2^1 * 2^3 = 16$ ways of assigning truth values to the antecedent makes the antecedent true. This is the case in which we assign the value 1 to all of the four conjuncts, thereby making the conjunction true. Out of the five conjuncts appearing in the consequent, this leaves only two unassigned, as we have already assigned truth values to the three conjuncts in the overlap set. There are 2^2 ways of assigning such truth values to the consequent, and only one of them makes the conjunction true, so the other $2^2 - 1 = 3$ all make the consequent false.

Therefore, out of the 2^6 possible valuations, only $1*3 = 3$ valuations make the implication false. If we count zero for each of these three valuations, count one for all of the others, and divide the result by 2^6, we arrive at the value $[\![\varphi \to \psi]\!] = \frac{64-3}{64} = .953125$.

More generally,

$$[\![\varphi \to \psi]\!] = 1 - \frac{2^{|\Lambda_\psi|} - 1}{2^{|\Lambda_\psi| + |\Lambda_\varphi| + |\Lambda_\omega|}}.$$

So, we can express the degree of validity for a given candidate entailment in a closed form depending only on the forms of the words and how they match up against each other, assuming we encode a given piece of text simply as a conjunction in bivalent logic.

Note that this closed form shares the same ordering properties with Dice's coefficient, the Jaccard index, or any other set overlap metric. These properties are as follows. (1) It acts as an overlap measure: Given φ or ψ, the ordering imposed by $[\![\varphi \to \psi]\!]$ on all ψ or φ, respectively, of the same length, is the same as that imposed by $|\Lambda_\omega|$. (2) It performs length normalization: Given φ or ψ, the ordering imposed by $[\![\varphi \to \psi]\!]$ on all ψ or φ, respectively, given a fixed overlap set Λ_ω, is inverse to the length of such ψ or φ.

So, we have arrived at a basic bag-overlap logic. Given an antecedent and a consequent, both of which are conjunctions of a number of propositions, the degree of validity $[\![\varphi \to \psi]\!]$ measures bag overlap.

Now note that SNF decompositions of possibly complex sentences are conjunctions of syllogistic premises, each syllogistic premise contributing one proposition. So, let's return to the previous example of predicate-argument structure:

$$\frac{\text{Meletus accused Socrates.}}{\text{./. The people accused Socrates.}} > \frac{\text{Meletus accused Socrates.}}{\text{./. Socrates accused the people.}} \tag{5.20}$$

Here we have one syllogistic premise in the overlap on the left-hand side, viz. the one which asserts that Socrates is being accused. On the right-hand side, the overlap set is empty. So the degree of validity would reflect exactly this preference. Similarly, the other preferences from Figure 5.2 are all fulfilled as indicated, given this approach.

This relationship between the form of a formula and its model-theoretic interpretation is precisely the reason why the bag-of-words approach works at all. The basic intuition is that the more conjuncts we have in a conjunction, the harder it will be to fulfill the constraint on the models which the conjunction represents. Consequently, the proportion of all models which fulfill the constraint will become smaller.

This also seems to be the intuition behind the approach taken by Bos & Markert, whose approach to missing background knowledge is to run a model builder and to use the size of the model which provides the counterexample as an entailment score. For our above example, a counterexample would be $\mathsf{socrates} \mathbin{\&} \mathsf{is} \mathbin{\&} \mathsf{a} \mathbin{\&} \mathsf{man} \mathbin{\&} \neg\mathsf{every}$. If we now increase the size of the overlap set, for example

$$\frac{(\mathrm{T})\quad \mathsf{socrates} \mathbin{\&} \mathsf{is} \mathbin{\&} \mathsf{an} \mathbin{\&} \mathsf{old} \mathbin{\&} \mathsf{man}}{\to\ (\mathrm{H})\quad \mathsf{so} \mathbin{\&} \mathsf{every} \mathbin{\&} \mathsf{old} \mathbin{\&} \mathsf{man} \mathbin{\&} \mathsf{is} \mathbin{\&} \mathsf{socrates}},$$

the size of the counterexample grows with the overlap set.

But this approach reacts differently to a conjunct being added to the non-overlap set, as the same counterexample would still apply to a longer sentence, e.g.

$$\frac{(\mathrm{T})\quad \mathsf{socrates} \mathbin{\&} \mathsf{is} \mathbin{\&} \mathsf{a} \mathbin{\&} \mathsf{old} \mathbin{\&} \mathsf{man}}{\to\ (\mathrm{H})\quad \mathsf{so} \mathbin{\&} \mathsf{every} \mathbin{\&} \mathsf{old} \mathbin{\&} \mathsf{man} \mathbin{\&} \mathsf{is} \mathbin{\&} \mathsf{socrates}}.$$

Also consider our previous example:

$$\frac{\mathsf{Some\ Greek} \mathbin{\&} \mathsf{heretic\ is\ mortal.}}{\not\to\ \mathsf{Some\ old} \mathbin{\&} \mathsf{philosopher\ is\ mortal.}} < \frac{\mathsf{Some\ Greek} \mathbin{\&} \mathsf{heretic\ is\ mortal.}}{\not\to\ \mathsf{Some\ philosopher\ is\ mortal.}} \tag{5.14}$$

This preference must clearly be fulfilled, as the right-hand consequent can be inferred from the left-hand consequent, but not the other way around. However, the size of the smallest counterexample is the same in both cases (e.g. $\mathsf{Greek} \mathbin{\&} \mathsf{heretic} \mathbin{\&} \neg\mathsf{philosopher}$).

5.3. Monte Carlo Semantics

In the previous section, we have seen that, if we could work out the value of $[\![\varphi \to \psi]\!]$, then that score would provide us with enough information to reproduce all of the semantic informativity properties of classical theorem proving, and all of the robustness properties of bag-overlap measurements.

But how do we go about this computationally? The general idea will be to consider various valuations $w \in \mathcal{W}$ and run a model checker on each valuation. Recall that, if $\varphi \to \psi$ is a formula over Λ, then, there are $2^{|\Lambda|}$ such valuations.

Now, in order to work out traditional validity or satisfiability, we need to find the minimum or maximum truth value $\|\varphi \to \psi\|_w$ we encounter for any w, so this means we would have to run a model checker $2^{|\Lambda|}$ times in the worst case.

But in our case of graded validity, we can exploit the fact that the arithmetic mean, in contrast to a maximum or a minimum, is very well behaved, when it comes to statistically estimating it. We will not attempt to logically determine its exact value. Instead, we will take a random sample $\mathrm{W} \subseteq \mathcal{W}$ and use $[\![\varphi \to \psi]\!]_{\mathrm{W}}$ as an estimator for $[\![\varphi \to \psi]\!]_{\mathcal{W}}$. By statistical sampling theory, we know that the former will approach the latter as the sample size $|\mathrm{W}|$ approaches the population size $|\mathcal{W}|$. This sampling can be automated using a Monte Carlo method.

5.3.1. 2-Valued vs. $\aleph_0$-Valued Logic and Simulation Error

The central question that arises then is how much information we obtain about $[\![\varphi \to \psi]\!]$ by simply assigning truth values to atomic propositions at random using a random number generator.

Let's consider a simple implication involving only atomic propositions: $[\![\mathrm{p} \to \mathrm{q}]\!] = 0.75$. We know that the truth table for this formula assigns the value 0 to only one valuation ($\|\mathrm{p}\| = 1$, $\|\mathrm{q}\| = 0$), and the value 1 to three valuations. Thus we have a $\frac{3}{4}$ chance of hitting the value 1.0 (error 0.25), and a $\frac{1}{4}$ chance of hitting the value 0 (error 0.75), which makes for a mean error of $\frac{3}{4} * 0.25 + \frac{1}{4} * 0.75 = 0.375$.

If we do this twice, we still have a $\frac{3}{4} * \frac{3}{4} = \frac{9}{16}$ chance of hitting an average value of 1.0 (error 0.25), a $\frac{3}{4} * \frac{1}{4} + \frac{1}{4} * \frac{3}{4} = \frac{6}{16}$ chance of hitting an average value of 0.5 (error 0.25) and finally a $\frac{1}{4} * \frac{1}{4} = \frac{1}{16}$ chance of hitting an average of 0.0 (error 0.75). We have a mean error of $\frac{9}{16} * 0.25 + \frac{6}{16} * 0.25 + \frac{1}{16} * 0.75 = 0.28125$.

As we increase the number of trials, the mean error will decrease. But can we speed up the process? We can increase the number of truth classes. This is what a truth table for 3-valued Łukasiewicz logic would look like:

p	**1.0**	1.0	**1.0**	0.5	0.5	0.5	**0.0**	0.0	**0.0**	
q	**1.0**	0.5	**0.0**	1.0	0.5	0.0	**1.0**	0.5	**0.0**	
p → q	**1.0**	0.5	**0.0**	1.0	1.0	0.5	**1.0**	1.0	**1.0**	$\mu = 0.77$

Four of these nine assignments coincide with bivalent logic, but we also insert five new values. We now have a mean truth value $[\![\mathrm{p} \to \mathrm{q}]\!] = 0.77$. We have a $\frac{6}{9}$ chance of hitting the value 1.0 (error 0.23), a $\frac{2}{9}$ chance of hitting the value 0.5 (error 0.27), and a $\frac{1}{9}$ chance of hitting the value 0.0 (error 0.77). The mean error is $\frac{6}{9} * 0.23 + \frac{2}{9} * 0.27 + \frac{1}{9} * 0.77 = 0.296$. If we run the model checker twice, with three truth values, we get a mean error of 0.19753.

We could also add a fourth truth value, which would give us a mean error of 0.26042 after running the model checker once, compared to a mean error of 0.28125 for using two truth values and running the model checker twice. So we get a better reduction in mean error by using more truth values than we do by using fewer truth values and running the model checker more often.

We can increase the number of truth values in the logic to $\aleph_0$, where $[\![p \to q]\!]$, which is 1.0 iff $p \leq q$, takes on the value 1.0 only at a 0.5 chance. – Of course we can do this only in theory. Computationally, there will have to be a limit. On a 64-bit machine, for example, choosing $M = 2^{63}$ and using an unsigned long integer to represent a truth value might make sense. The point is that we do not want to restrict the number of truth values any more than necessary, certainly not to anything as low as two, as this would be a waste of computation time, and a waste of entropy in estimating $[\![p \to q]\!]$.

5.3.2. Summary & A Worked Example

We now have all ideas in place needed to describe the inference engine in its entirety, and we will then conclude this section with a worked example.

- Step 1: Use the ERG (Flickinger 2000) to convert sentences in the antecedent and the consequent to MRS representations. In the case of our FraCaS experiment (appendix D) parse selection was performed manually.[2]
- Step 2: Convert the MRSes to ProtoForms and perform SNF decomposition. If antecedent and consequent consist of multiple sentences, conjoin them using strong conjunction. The candidate inference itself is represented as an implication (section 4.4) consisting of antecedent and consequent.
- Step 3: SNF formulae belong to the language of the predicate calculus. By interpreting quantifications as ranging over a finite domain of three individuals, we can embed this predicate calculus into a purely propositional calculus (section 3.2).
- Step 4: Construct a valuation by assigning truth values to the atomic propositions at random.
- Step 5: Work out the truth value of the candidate inference, as represented by the implication, in the given valuation. Add the truth value to a running sum.
- GOTO Step 4. REPEAT ≈ 1000 times.

[2] I would like to thank, at this point, Dan Flickinger for having undertaken this treebanking effort.

- Step 6: Divide the running sum by 1000, giving the average truth value. This is our entailment score.

The number of iterations will have to be chosen based on the required error bounds and could be dynamically assigned on the basis of the length and complexity of the formula. The number 1024 is what was chosen for the FraCaS experiment to work out the distinction between degrees of validity = 1.0 and < 1.0. This way we could correctly distinguish degrees of validity differing by an ϵ-magnitude in about 95% of all cases. The results reported in appendix D were obtained by increasing the number to 16384, which could distinguish the ϵ-difference in 100% of all cases. For many practical applications, it will be possible to choose a much smaller number, if we are content, for example, to be able to recognize differences of magnitude over 0.01 in 95% of all cases. The exact choice of the parameter will thus have to be tuned to particular application needs.

Steps 1 & 2

Let's consider a simple sentence:

φ : Some elephants are intelligent

Upon semantic composition in the ERG, SNF decomposition, and assignment of first-order quantifiers interpreting the generalized quantifiers, we would be left with a form like this:

$$\Big[\exists_{(x)} \Big[|\text{elephant}| (\textsc{key} = x) \Big] \Big[|\text{intelligent}| (\textsc{key} = x) \Big] \Big]$$

Step 3

By using a finite domain for its range, we can rewrite the existential quantification binding the variable as a disjunction of predications over constants:

$$\begin{aligned} &\big(|\text{elephant}| (\textsc{key} = /1/) \wedge |\text{intelligent}| (\textsc{key} = /1/)\big) \\ \vee\ &\big(|\text{elephant}| (\textsc{key} = /2/) \wedge |\text{intelligent}| (\textsc{key} = /2/)\big) \\ \vee\ &\big(|\text{elephant}| (\textsc{key} = /3/) \wedge |\text{intelligent}| (\textsc{key} = /3/)\big). \end{aligned}$$

These can be seen as atomic propositions in a propositional logic:

$$(e_1 \wedge i_1) \vee (e_2 \wedge i_2) \vee (e_3 \wedge e_3).$$

For brevity, we will work with only two individuals, rather than three in what follows. Let's consider as an example the following three sentences

φ : Some elephants are intelligent,

ψ : Some grey elephants are intelligent,

χ : Some clean grey elephants are intelligent.

which can then be represented by the formulae

$$\begin{aligned} \varphi &: \ (e_1 \wedge i_1) \vee (e_2 \wedge i_2) \\ \psi &: \ (e_1 \wedge g_1 \wedge i_1) \vee (e_2 \wedge g_2 \wedge i_2) \\ \chi &: \ (e_1 \wedge c_1 \wedge g_1 \wedge i_1) \vee (e_2 \wedge c_1 \wedge g_2 \wedge i_2) \end{aligned}$$

Step 4

	e_1	i_1	e_2	i_2	$(e_1 \wedge i_1) \vee$	$(e_2 \wedge i_2) =$	φ	$\rightarrow \psi$	$=$	$(\psi \rightarrow \varphi)$	χ	$(\varphi \rightarrow \chi)$	g_1	g_2	c_1	c_2
w_1	.99	.55	.47	.38	.55	.38	.55	.39	.84	1	.19	.64	.39	.19	.12	.97
w_2	.10	.58	.29	.00	.10	.00	.10	.10	1	1	.10	1	.98	.85	.62	.44
w_3	.13	.93	.59	.96	.13	.59	.59	.32	.73	1	.25	.66	.16	.32	.08	.25
w_4	.26	.64	.68	.74	.26	.68	.68	.68	1	1	.13	.45	.80	.99	.02	.13
w_5	.47	.10	.03	.76	.10	.03	.10	.10	1	1	.10	1	.65	.54	.10	.74
									.91	**1**		**.75**				

Now we can assign truth values to these propositions at random. Above, we have listed five different assignments of truth values. The truth values listed in the first and last four columns have been randomly generated, in this case using a standard spreadsheet tool. The values of the other columns are computed from these.

Recall from definition 36 that

$$\|p \rightarrow q\|_w = \max(1, 1 - \|p\|_w + \|q\|_w),$$

and from corollary 38 that

$$\begin{aligned} \|\varphi \wedge \psi\| &= \min(\|\varphi\|, \|\psi\|), \\ \|\varphi \vee \psi\| &= \max(\|\varphi\|, \|\psi\|). \end{aligned}$$

In the first valuation of the example, we have $\|e_1\|_{w_1} = 0.99$, $\|i_1\|_{i_1} = 0.55$, so $\|e_1 \wedge i_1\|_{w_1} = .55$, so 'individual 1 is an intelligent elephant' is true to degree $.55$. Similarly, $\|e_2 \wedge i_2\|_{w_1} = .38$, so 'individual 2 is an intelligent elephant' is true to degree $.38$. Finally $\|\varphi\|_{w_1} = \|(e_1 \wedge i_1) \vee (e_2 \wedge i_2)\|_{w_1} = \max(.55, .38) = .55$, so 'some individual is an intelligent elephant' is true to a degree $.55$.

If we use the values $\|g_1\|$ and $\|g_2\|$, we can analogously determine $\|\psi\|_{w_1} = .39$. Since $1 - .55 + .39 = .84$, the implication stating "if some elephants are intelligent, then some grey elephants are intelligent" is true to a degree $.84$ in valuation w_1. The converse implication is true to a degree 1.0. Similarly, we can determine $\|\varphi \rightarrow \chi\|_{w_1} = .64$. Note that $\|\psi \rightarrow \varphi\| \geq \|\top\| = 1.0$, in accordance with (2.a), that $1.0 = \|\top\| < \|\varphi \rightarrow \psi\|$, in accordance with (2.b), and that $\|\varphi \rightarrow \psi\| < \|\varphi \rightarrow \chi\|$, in accordance with (2.c). It should be obvious, at this point, that this is not a coincidence.

Steps 5 & 6

While the fundamental logical properties are already fulfilled for the truth values in the above example, the exact truth values are still a function of the random valuation we started out with. This is why we now repeat the process for different valuations w_2, w_3, etc., to obtain mean truth values, i.e. degrees of validity, of $[\![\psi \rightarrow \varphi]\!] = 1.0$, and $[\![\varphi \rightarrow \psi]\!] = .91$, and $[\![\varphi \rightarrow \chi]\!] = .75$. Observe that we have not entered any information at all giving ontological or lexical defintions of elephants, intelligence, etc. But these degrees of validity do reflect the robustness properties of Figure 5.2.

6. Conclusions

In this chapter, we will reiterate the particular claims we have made throughout this thesis and will then offer a viewpoint as to how these claims relate to each other and how they reflect on the broader fields to which they relate.

6.1. Empirical Review & Methodology

In chapter 2, we conducted a review of the methodology employed at the RTE recognizing textual entailment challenge.

Results & Claims

Formal Logic vs. Intuition vs. Application There are three different criteria which can lead to decisions on candidate inferences: The logical criterion, the intuitive criterion and the application-oriented criterion (section 2.1.2). Much of the attractiveness of RTE stems from the contention that the intuitive criterion by which the datasets are derived coincides with the application-oriented criterion which motivates the task, which is a hypothesis that has remained untested.

Textual Entailment as an Abstraction over Applications Another basic assumption underlying RTE is that the task can be seen as an abstraction over the application-oriented tasks of question answering, information extraction, information retrieval, summarization, etc. Two of our findings about RTE-4 submissions (section 2.2.1) cast doubts on this: (1) Systems generally performed better on information retrieval than they did on summarization, and better on summarization than they did on question answering and information retrieval. (2) Rank correlation was low when comparing rankings of submissions based on the different applications.

Relevance vs. Validity The notion of textual entailment employed at RTE fails to draw a clear distinction between relevance and validity (section 2.1.1). While much of what has been written about RTE treats the task as if it were primarily about logical validity, and while some participants specifically addressed the problem of validity (section 2.1.3), we found that the RTE-4 dataset put much more statistical weight on relevance than on validity (section 2.1.6) and also that systems were generally better at deciding relevance than they were at deciding validity (section 2.2.1). As it is primarily the problem of validity which requires logical inference, this means that the usefulness of RTE for evaluating logically-motivated approaches to inference is limited.

Evaluation Measures We found three problems with the various evaluation measures used at RTE (section 2.1.4): (1) The distribution of three-way gold standard labels is neither balanced nor representative of an application scenario. Yet, systems are rewarded by higher accuracy scores for learning this artificial bias from training data, while there is no indication of whether they could learn a different bias (section 2.1.6). (2) Average precision fails to properly reflect the symmetry imposed on textual entailment decisions by the possibility of negation (sections 2.1.3 and 2.1.5). (3) The notion of confidence ranking is misleading in the context of evaluating a ranking by average precision. This has lead to some confusion among participants who submitted confidence-ranked 3-way labellings at RTE-4 (section 2.1.5).

Bag-of-Words Equivalence At RTE-4, a bag-of-words baseline achieves an accuracy score of around 60%, which was outperformed by only about a quarter of all submissions. This raises the following question about the submissions which did: Are their incorrectly labelled instances random deviations from the gold standard, or are their correctly labelled instances random deviations from the baseline? In section 2.2.2, we show that, with the exception of two or three systems, the latter seems to be the case.

Concluding Remarks

Our review of the RTE evaluation scheme was relevant to us primarily in connection with the question of what sort of methodology to employ for our own studies into the subject of textual inference. In the introduction we mentioned two kinds of methodology: (1) The RTE scheme reflects a kind of empiricism where data is collected purely "out in the wild", as a corpus linguist would. (2) One can rely on introspection in much the same way as a linguist would, when using carefully chosen examples and counterexamples to substantiate a given working hypothesis, and connect the dots by logical deduction.

Our findings on the RTE scheme are by no means sufficient to contradict the entire approach of empiricsm. But, upon critical reevaluation, we do find that the state of the art in empirically driven computational semantics, promising as it seems on the basis of a superficial reading of the literature, has so far fallen short of delivering on those promises. So, despite the prominence which this approach has gained in the scientific discourse at this point in time, one can ill afford to neglect evidence obtained by introspection and deduction of the kind we put forward in this thesis.

6.2. Łukasiewicz Logic & Syllogistic Semantics

Chapter 3 established basic logical results. Ultimately, the chapter led up to a model-theoretic completeness proof of the syllogism. The particular properties of this model theory have had an important role to play throughout the rest of this thesis, notably the fact that it is based on an $\aleph_0$-valued logic and the fact that it has a strong conjunction operator which is commutative and associative but not idempotent.

Results & Claims

Choice of Propositional Logic The results outlined in section 3.1 are well-known and were summarized herein for the convenience of readers without a background in many-valued logic and for purposes of establishing notation and terminology. Despite the fact that, in and of itself, Łukasiewicz logic is well understood, it should however be pointed out that our choice to use Łukasiewicz logic, rather than any other many-valued logic, is a nontrivial result. For example, the completeness proof of the syllogism (section 3.3) would not have been possible with Gödel's many-valued logic or the product logic which results from a naive reinterpretation of probability theory as logic.

Finite Domain Size and Interpretations of Quantifiers Section 3.2 established an interpretation of quantifiers over finite domain sizes. We established that domains must contain a minimum of three distinct individuals and that universal quantification must be interpreted as weak rather than strong conjunction, and, similarly, that existential quantification must be interpreted as weak disjunction. Note that with a different choice of domain size and interpretation for quantifiers, the completeness proof of the syllogism (section 3.3) would not have been possible.

An $\aleph_0$-valued Model Theory for the Syllogism The result which the entire chapter led up to is the completeness proof of the syllogism (section 3.3) based on the model theory of our non-standard logic.

Concluding Remarks

The paradigm of bivalent logic is so deeply ingrained in our understanding of natural language semantics that Boolean algebra is often seen as synonymous with the very idea of propositional logic. The results of this chapter suggest that such an assumption of bivalence, however, may not be an inherent property of natural language. These results seem relevant not only to the present work, but also for search applications, natural language interfaces to databases, and as a model for vagueness in natural language (see Bergmair 2006*a*,*b*, Bergmair & Bodenhofer 2006, van Deemter 2010*a*,*b*).

The highly influential paradigm pursued by Montague and others of translating natural language to the language of FOPC may mislead one to think that the logic of FOPC is not only sufficient but necessary in its entirety for the purposes of interpreting those predicate calculus expressions. Herein, however, we established a result which lends a great deal of gravity to the notion that, as far as natural language is concerned, the move from propositional logic to predicate calculus may be nothing but syntactic sugar. The limit case of infinite domain sizes may be useful for certain theoretical purposes, but the property does not seem to be inherent to natural language reasoning. This leaves us with a logic which is different from the standard FOPC as implemented by out-of-the-box reasoning tools.

6.3. Semantic Decomposition

Chapter 4 showed how to translate natural language expressions into the language of the syllogism.

Results & Claims

ProtoForm Representation Language In section 4.1, we established ProtoForms as a new semantic representation language which is inspired by the MRS language, but which differs from previous designs of semantic underspecification languages in one important respect: It is recursive in the sense that a ProtoForm may be a subform of another ProtoForm and can therefore represent not only minimally recursive semantic structures but also partially scoped semantic structures and fully recursive logical formulae. In particular, the concept of a maximally recursive ProtoForm is constitutive of the notion of a semantic head, which played a central role throughout the rest of the chapter.

Using the MRS Algebra for ProtoForm Composition In section 4.2, we showed, on the example of a toy grammar, that ProtoForms are not only useful for scoping and decomposition, but are also an adequate form of representation for composition purposes. Otherwise, section 4.2 largely served to summarize relevant ideas for the convenience of readers without a background in compositional semantics.

Syntactically vs. Semantically-Driven Substitution Logic In section 4.3.3, we showed that syntactically driven substitution logic of the kind used by MacCartney (2009) does not adequately deal with semantic scope.

Limitations of Substitution Logic In section 4.3.4, we showed that substitution logic, more generally, cannot adequately represent the distinction between intersective modification and optional argument-taking. We showed that substitution logic is hopelessly inadequate when it comes to representing the kind of common sense ontology which would be needed to justify even very basic natural language inferences.

SNF Decomposition In section 4.4, we showed how the ProtoForms derived from MRS-style composition can be decomposed into conjunctions of syllogistic premises. We call these structures syllogistic normal forms (SNFs).

Operator Grammar & SNF Dependency Structures Section 4.5 reinterpreted SNFs as grammatical dependency structures, and showed that they fulfill all the metatheoretical properties set out by Harris (1982, 1991) for dependency structures. Thus, SNFs are not only artefacts which arise as a byproduct if one wants to define an inference engine on the basis of syllogistic logic, but rather they have an interpretation which is linguistically interesting in and of itself.

Concluding Remarks

When it comes to the interpretation of semantic representation structures, there are traditionally two different kinds of approaches. (1) On one hand, one can translate them to a logical language, such as FOPC. In practice, this translation has usually assumed a Montague-style relationship between natural langauge and the language of the predicate calculus, where quantifier nesting corresponds to quantifier nesting in natural language. (2) On the other hand, one can rely on the metatheoretical properties of semantic structures and treat them simply as graphs for purposes of graph-alignment, pattern rewriting, machine learning etc.

Our approach to the decomposition of semantic structures leads to a different kind of representation which is at the same time a logical formula in a fragment of FOPC and which has metatheoretical properties which make it a linguistically plausible dependency structure. This kind of dual interpretation has been highly useful herein as a theoretical framework for understanding the relationships between logically-based approaches to inference on one hand, and shallow and intermediate-level approaches on the other.

6.4. Monte Carlo Semantics

In chapter 5, we introduced some ideas on epistemology and, on the basis of a unified theory of deep and shallow inference, put forward a novel kind of inference mechanism which aims to be at the same time robust and semantically informed.

Results & Claims

Semantic Informativity & The Deep Limit Case We call a reasoning mechanism semantically informed, iff it is able to take into account all available information. We defined this notion in greater detail in section 5.1.1. In section 5.2.1, we established traditional theorem proving as the limit case of our approach which arises as a result of a complete theory of background knowledge and deep lexico-grammatical analysis.

Robustness & The Shallow Limit Case We call a reasoning mechanism robust, iff it makes use of heuristics which enable it to proceed on reasonable assumptions where hard information is missing. Section 5.1.2 defined this in detail, and, in section 5.2.2, we established bag-of-words pseudo-inference as the limit case of our approach which arises for empty theories of background knowledge and no lexico-grammatical analysis beyond tokenization.

Monte Carlo Semantics In section 5.3, we finally moved on to define the algorithm which scores candidate entailments for their grade of validity in the general case. The algorithm is based on the idea of assigning truth values to predications at random and running a model checker to determine for each randomization the truth value of a candidate entailment as represented by a material implication. The average truth value thus obtained is an estimate of our degreee of validity.

Implementation The ideas discussed in this dissertation were generated and tested to a proof-of-concept level in the course of an explorative software prototyping effort. In appendix C, we give some pointers concerning this software which is now freely available for use, modification, and redistribution, inviting future work to build on the various ideas surrounding ProtoForms, SNF decompositions and Monte Carlo semantics.

Concluding Remarks

Consider the following keywords used in different kinds of NLP publications:

(a) shallow processing, robustness, probability, statistics, machine learning;

(b) deep processing, semantics, logic, ontology, linguistics.

Furthermore, consider the following two expressions of opinions:

(a) "... you must be very naive to believe you can reason about language in logic. Even if you could, you're missing the knowledge to prove things. Even if you had that, logic would still be too computationally complex."

(b) "... you must be rather ignorant to believe a machine learner will magically acquire language competence if you don't build into it everything we know, and, in fact, some things we don't, about logic, epistemology, and linguistics, as well as common sense and real world knowledge."

Here, we are not saying that anyone in particular holds one of these viewpoints in its extreme form. But the stereotypes themselves are certainly recognizable for the deeply entrenched ideological divide which they represent.

To anyone subscribing to viewpoint (a), overly restrictive consistency and completeness assumptions, as well as the theoretical limitations of the traditional notion of validity seem like a bad idea. This is why we took the viewpoint that probability theory can do better than that.

To anyone subscribing to viewpoint (b), the formula 'every & man & is & mortal' will seem like a particularly bad idea, indeed. So, we took the viewpoint that existing methods of semantic composition can do better.

But, in response to viewpoint (a), we can now also say that knowledge engineering issues and problems of computational complexity are completely separate from the question of whether or not logic itself is a useful theoretical framework for approaching textual inference. It is all a question of how one represents text in logic. With SNF decompositions, we have identified a fragment of logic which is semantically informed almost[1] to the same extent as the full predicate calculus, but which is much more manageable in terms of knowledge engineering and computational complexity and can even reduce to bag-of-words reasoning where this is necessary.

[1] The exception is quantifier nesting.

In response to viewpoint (b), we took some steps towards accounting for the practical success of naïve approaches. We have shown that, in particular, the robustness properties associated with a gradual notion of validity are a key element.

So, we can emphasize, once again, that our approach subscribes neither to viewpoint (a) nor to viewpoint (b) exclusively. Rather it is an attempt at a unified theory which covers both and which tries to learn lessons from the failures and successes of each.

A. Statistical Tables

	$\hat{\mu}$	$\hat{\sigma}$	min	q1	med	q3	max
IR	.072	.0630	.000	.022	.066	.103	.327
SUM	.052	.0531	.000	.013	.042	.077	.282
QA	.022	.0463	.000	.001	.003	.011	.205
IE	.020	.0451	.000	.000	.001	.010	.229

(a) summary statistics on $\mathbb{I}\big([\mathbf{G}]_{\vartriangle,\triangledown};[\mathbf{L}]_{\vartriangle,\triangledown}\big)$

	$\hat{\mu}$	$\hat{\sigma}$	min	q1	med	q3	max
IR	.631	.0742	.477	.567	.643	.683	.820
SUM	.610	.0647	.495	.560	.620	.655	.780
QA	.537	.0711	.435	.495	.515	.545	.760
IE	.534	.0587	.443	.500	.513	.543	.773

(b) summary statistics on $\mathrm{A}_{\vartriangle,\triangledown}(\mathrm{G};\mathrm{L})$

Figure A.1.: scores for 81 systems on different applications at RTE-4 (i) (section 2.2.1)

	r	N	$2\,\mathcal{BN}\big(\mathrm{r};\mathrm{N},\frac{1}{2}\big)$
IR vs. SUM	15	79	≈ 0
SUM vs. QA	13	79	≈ 0
QA vs. IE	33	78	0.213

(a) sign statistics on $\mathbb{I}\big([\mathbf{G}]_{\vartriangle,\triangledown};[\mathbf{L}]_{\vartriangle,\triangledown}\big)$

	r	N	$2\,\mathcal{BN}\big(\mathrm{r};\mathrm{N},\frac{1}{2}\big)$
IR vs. SUM	15	78	≈ 0
SUM vs. QA	12	78	≈ 0
QA vs. IE	39	78	≈ 1

(b) sign statistics on $\mathrm{A}_{\vartriangle,\triangledown}(\mathrm{G};\mathrm{L})$

Figure A.2.: scores for 81 systems on different applications at RTE-4 (ii) (section 2.2.1)

	U_1	U_2	$2\mathcal{N}\big(\min(U_1, U_2); \hat{\mu}, \hat{\sigma}^2\big)$
IR vs. SUM	2544.5	4016.5	.014
SUM vs. QA	1477.5	5083.5	≈ 0
QA vs. IE	2902.0	3659.0	.205

$$\hat{\mu} = \frac{81^2}{2} = 3280.5$$

$$\hat{\sigma}^2 = \frac{81^2 * 82}{12} = 44833.5$$

(a) Mann-Whitney U-statistics on $\mathbb{I}\big([\mathbf{G}]_{\vartriangle,\triangledown}; [\mathbf{L}]_{\vartriangle,\triangledown}\big)$

	U_1	U_2	$2\mathcal{N}\big(\min(U_1, U_2); \hat{\mu}, \hat{\sigma}^2\big)$
IR vs. SUM	2589	3972	.0205
SUM vs. QA	1277	5284	≈ 0
QA vs. IE	3258.5	3302.5	.9413

$$\hat{\mu} = \frac{81^2}{2} = 3280.5$$

$$\hat{\sigma}^2 = \frac{81^2 * 82}{12} = 44833.5$$

(b) Mann-Whitney U-statistics on $A_{\vartriangle,\triangledown}\big(G; L\big)$

Figure A.3.: scores for 81 systems on different applications at RTE-4 (iii) (section 2.2.1)

IR	QA	SUM	IE	
	.15	.60	.41	IR
		.12	.21	QA
			.52	SUM
				IE

(a) Kendall's τ on $\mathbb{I}\big([\mathbf{G}]_{\vartriangle,\triangledown}; [\mathbf{L}]_{\vartriangle,\triangledown}\big)$

IR	QA	SUM	IE	
	.25	.63	.39	IR
		.26	.24	QA
			.40	SUM
				IE

(b) Kendall's τ on $A_{\vartriangle,\triangledown}\big(G; L\big)$

Figure A.4.: ranking 81 systems on different applications at RTE-4 (section 2.2.1)

	$\hat{\mu}$	$\hat{\sigma}$	min	q1	med	q3	max
$\boxplus\boxminus/\diamond$	.036	.0356	.010	.036	.062	.136	
$\boxplus/\diamond\boxminus$	.030	.0395	.006	.018	.040	.187	
$\boxplus\diamond/\boxminus$	.019	.0433	≈ 0	.004	.012	.229	

(a) summary statistics on $\mathbb{I}\big([\mathbf{G}];[\mathbf{L}]\big)$

	r	N	$2\mathcal{BN}\big(\mathrm{r};\mathrm{N},\frac{1}{2}\big)$
$\boxplus\boxminus/\diamond$ vs. $\boxplus/\diamond\boxminus$	4	34	≈ 0
$\boxplus\boxminus/\diamond$ vs. $\boxplus\diamond/\boxminus$	8	34	0.003

(b) sign statistics on $\mathbb{I}\big([\mathbf{G}];[\mathbf{L}]\big)$

	U_1	U_2	$2\mathcal{N}\big(\min(\mathrm{U}_1,\mathrm{U}_2);\hat{\mu},\hat{\sigma}^2\big)$
$\boxplus\boxminus/\diamond$ vs. $\boxplus/\diamond\boxminus$	500	796	0.096
$\boxplus\boxminus/\diamond$ vs. $\boxplus\diamond/\boxminus$	300	996	≈ 0

$$\hat{\mu} = \frac{36^2}{2} = 648$$

$$\hat{\sigma}^2 = \frac{36^2 * 37}{12} = 3996$$

(c) Mann-Whitney U-statistics on $\mathbb{I}\big([\mathbf{G}];[\mathbf{L}]\big)$

Figure A.5.: scores of 36 systems on components of the 3-way decision at RTE-4 (section 2.2.1)

B. Proofs

B.1. Proofs for section 3.1

Proof of corollary 50. It can be seen that $(\mathrm{V}, \underline{\vee}, \&, \neg)$ has a DeMorgan identity as follows. By $(*\mathrm{MV12})$, we have $\neg x \,\&\, \neg y = \neg(\neg\neg x \;\underline{\vee}\; \neg\neg y)$, hence, by $(\star\mathrm{MV7})$, $\neg x \,\&\, \neg y = \neg(x \;\underline{\vee}\; y)$. Now consider the algebra $\mathbf{B} = (\mathrm{V}', \underline{\vee}', \neg', \overline{0}')$, where $\mathbf{V}' = \{x' | x \in \mathrm{V}, x' = \neg x\}$, where $a \;\underline{\vee}\; 'b = \neg(a \;\underline{\vee}\; b)$, where $\neg' a = \neg\neg a$, and where $\overline{0}' = \neg\overline{0}$. It is clear that, whenever $a \;\underline{\vee}\; b = c$, in $\mathbf{A}$, $a' \;\underline{\vee}\; 'b' = c'$, in $\mathbf{B}$. Similarly, whenever $a \;\underline{\vee}\; \overline{0} = b$ or $\overline{0} \;\underline{\vee}\; a = b$ in $\mathbf{A}$, we must have $a' \;\underline{\vee}\; '\overline{0}' = b'$ or $\overline{0}' \;\underline{\vee}\; 'a' = b'$ respectively, in $\mathbf{B}$. Also, whenever $\neg a = b$ in $\mathbf{A}$, $\neg' a' = b'$ in $\mathbf{B}$. So it follows from the fact that $\mathbf{A}$ is an MV algebra, that $\mathbf{B}$ is an MV algebra. But we know, by $(\star\mathrm{MV7})$ and the fact that V is closed under $\neg$, that $\mathrm{V}' = \mathrm{V}$. Similarly, we know from DeMorgan's identity that $\underline{\vee}' = \&$, and from $(\star\mathrm{MV7})$ that $\neg' = \neg$. Finally, we know from $(*\mathrm{MV8})$ that $\overline{0}' = \overline{1}$. Also observe that definitions $(*\mathrm{MV15})$ and $(*\mathrm{MV16})$ are duals of each other. □

$*$91. *Let* $\mathbf{A} = (\mathrm{V}, \rightarrow, \overline{0})$ *be a Wajsberg algebra and let* $\mathbf{A}' = (\mathrm{V}, \rightarrow, \neg, \&, \underline{\vee}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ *be its Wajsberg-induced algebra. For all* $x, y \in \mathbf{V}$, *we have*

$$x = y \textit{ iff } x \rightarrow y = \overline{1} \textit{ and } y \rightarrow x = \overline{1}; \qquad (\dagger\mathrm{W13})$$

$$\textit{if } x \rightarrow y = \overline{1} \textit{ and } y \rightarrow z = \overline{1} \textit{ then } x \rightarrow z = \overline{1}. \qquad (\dagger\mathrm{W14})$$

Furthermore, the following identities hold for all $x, y, z \in \mathrm{V}$*:*

$$\neg\overline{0} = \overline{1}, \quad (\dagger\mathrm{W21}) \qquad \neg x \rightarrow \neg y = y \rightarrow x, \quad (\dagger\mathrm{W18})$$

$$\neg\overline{1} = \overline{0}, \quad (\dagger\mathrm{W22}) \qquad x \rightarrow (y \rightarrow x) = \overline{1}, \quad (\dagger\mathrm{W19})$$

$$x \rightarrow x = \overline{1}, \quad (\dagger\mathrm{W15}) \qquad x \rightarrow (y \rightarrow z) = y \rightarrow (x \rightarrow z), \quad (\dagger\mathrm{W20})$$

$$x \rightarrow \overline{1} = \overline{1}, \quad (\dagger\mathrm{W16}) \qquad \neg\neg x = x. \quad (\dagger\mathrm{W23})$$

$$\overline{0} \rightarrow x = \overline{1}, \quad (\dagger\mathrm{W17})$$

Furthermore, the identities from definitions 48 and 49 hold.

Proof.

$(\dagger\mathrm{W15})$ By $(\star\mathrm{W2})$, we have $\big(\overline{1} \rightarrow \overline{1}\big) \rightarrow \big((\overline{1} \rightarrow x) \rightarrow (\overline{1} \rightarrow x)\big) = \overline{1}$, hence, by $(\star\mathrm{W1})$, $\overline{1} \rightarrow (x \rightarrow x) = \overline{1}$, hence, by $(\star\mathrm{W1})$, $x \rightarrow x = \overline{1}$.

(†W21) By (∗W6) we have $\neg\overline{0} = \overline{0} \to \overline{0}$, hence, by (†W15), $\neg\overline{0} = \overline{1}$.

(†W22) By (∗W6) we have $\neg\overline{1} = \overline{1} \to \overline{0}$, hence, by (⋆W1) $\neg\overline{1} = \overline{0}$.

(†W16) By (†W15), we have $x \to \overline{1} = x \to (x \to x)$, hence, by (⋆W1), $x \to \overline{1} = (\overline{1} \to x) \to \big((\overline{1} \to x) \to x \big)$, hence, by (⋆W4) and (∗W10), $x \to \overline{1} = (\overline{1} \to x) \to \big((x \to \overline{1}) \to \overline{1} \big)$, hence, by (†W15), $x \to \overline{1} = (\overline{1} \to x) \to \big((x \to \overline{1}) \to (\overline{1} \to \overline{1}) \big)$, hence, by (⋆W2), $x \to \overline{1} = \overline{1}$.

(†W17) By (⋆W3), we have $(\neg x \to \neg\overline{0}) \to (\overline{0} \to x) = \overline{1}$, hence, by (†W21), $(\neg x \to \overline{1}) \to (\overline{0} \to x) = \overline{1}$, hence, by (†W16), $\overline{1} \to (\overline{0} \to x) = \overline{1}$, hence, by (⋆W1), $(\overline{0} \to x) = \overline{1}$.

(†W13) The "only if"-part of (†W13) follows directly from (†W16). To see that the "if"-part holds, note that by (⋆W1), we have $x = \overline{1} \to x$, hence, given that $y \to x = \overline{1}$, we have $x = (y \to x) \to x$, hence, by (⋆W4) and (∗W10), $x = (x \to y) \to y$, hence, given that $x \to y = \overline{1}$, we have $x = \overline{1} \to y$, hence, by (⋆W1), $x = y$.

(†W18) By (⋆W2), $(y \to x) \to \big((x \to \overline{0}) \to (y \to \overline{0}) \big) = \overline{1}$, hence, by (∗W6), $(y \to x) \to \big(\neg x \to \neg y \big) = \overline{1}$. By (⋆W3), $(\neg x \to \neg y) \to (y \to x) = \overline{1}$. So, by (†W13), $(\neg x \to \neg y) = (y \to x)$.

(†W19) By (⋆W1), $x \to (y \to x) = \overline{1} \to \big(x \to (y \to x) \big)$, hence, again by (⋆W1), $x \to (y \to x) = \overline{1} \to \big((\overline{1} \to x) \to (y \to x) \big)$, hence, by (†W16), $x \to (y \to x) = (y \to \overline{1}) \to \big((\overline{1} \to x) \to (y \to x) \big)$, hence, by (⋆W1), $x \to (y \to x) = \overline{1}$.

(†W14) By (⋆W2), $(x \to y) \to \big((y \to z) \to (x \to z) \big) = \overline{1}$, hence, given that $x \to y = \overline{1}$, $\overline{1} \to \big((y \to z) \to (x \to z) \big) = \overline{1}$, hence, by (⋆W1), $(y \to z) \to (x \to z) = \overline{1}$, hence, given that $y \to z = \overline{1}$, $\overline{1} \to (x \to z) = \overline{1}$, hence, by (⋆W1), $x \to z = \overline{1}$.

(†W20) By (⋆W2)

$$\begin{aligned} &\big(y \to ((y \to z) \to z) \big) \\ &\quad \to \Big(\big(((y \to z) \to z) \to (x \to z) \big) \to \big(y \to (x \to z) \big) \Big) = \overline{1}, \end{aligned}$$

hence, (⋆W4) and (∗W10),

$$\begin{aligned} &\big(y \to ((z \to y) \to y) \big) \\ &\quad \to \Big(\big(((y \to z) \to z) \to (x \to z) \big) \to \big(y \to (x \to z) \big) \Big) = \overline{1}, \end{aligned}$$

hence, by (†W19),

$$\overline{1} \to \Big(\big(((y \to z) \to z) \to (x \to z) \big) \to \big(y \to (x \to z) \big) \Big) = \overline{1},$$

hence, by (⋆W1),

$$\big(((y \to z) \to z) \to (x \to z) \big) \to \big(y \to (x \to z) \big) = \overline{1}.$$

By (⋆W2) $\big(x \to (y \to z) \big) \to \Big(\big((y \to z) \to z \big) \to (x \to z) \Big) = \overline{1}$. By (†W14), we therefore have $\big(x \to (y \to z) \big) \to \big(y \to (x \to z) \big) = \overline{1}$, hence, by (†W13), $\big(x \to (y \to z) \big) = \big(y \to (x \to z) \big)$.

(†W23) By (⋆W4) and (∗W10), we have $(x \to \overline{0}) \to \overline{0} = (\overline{0} \to x) \to x$, hence, by (†W17), $(x \to \overline{0}) \to \overline{0} = \overline{1} \to x$, hence, by (⋆W1), $(x \to \overline{0}) \to \overline{0} = x$, hence, by (∗W6), $\neg\neg x = x$.

(∗MV8) By (∗W5) and (∗W6).

(∗MV12) By (∗W7), $x \,\&\, y = \neg(x \to \neg y)$, hence, by (†W23), $x \,\&\, y = \neg(\neg\neg x \to \neg y)$, hence, by (∗W8), $x \,\&\, y = \neg(\neg x \veebar \neg y)$.

(∗MV13) By (∗W8) and (†W23).

(∗MV14) By (∗W11).

(∗MV15) By (∗W10), $x \vee y = (x \to y) \to y$, hence, by (†W23), $x \vee y = \neg\neg(x \to y) \to y$, hence, by (∗W8), $x \vee y = \neg(x \to y) \veebar y$, hence, by (†W23), $x \vee y = \neg(x \to \neg\neg y) \veebar y$, hence, by (∗W7), $x \vee y = \neg(x \,\&\, \neg y) \veebar y$.

(∗MV16) By (∗W9), $x \wedge y = x \,\&\, (x \to y)$, hence, by (∗W7), $x \wedge y = \neg(x \to \neg(x \to y))$, hence, by (†W18) and (†W23), $x \wedge y = \neg((x \to y) \to \neg x)$, hence, by (†W18), $x \wedge y = \neg((\neg y \to \neg x) \to \neg x)$, hence, by (⋆W4) and (∗W10), $x \wedge y = \neg((\neg x \to \neg y) \to \neg y)$, hence, by (∗W7), $x \wedge y = (\neg x \to \neg y) \,\&\, y$, hence, by (∗W8), $x \wedge y = (x \veebar \neg y) \,\&\, y$.

(∗MV17) By (∗W12).

(⋆MV1) By (†W18), $\neg x \to y = \neg y \to \neg\neg x$, hence, by (†W23), $\neg x \to y = \neg y \to x$, hence, by (∗W8), $x \veebar y = y \veebar x$.

(⋆MV2) By (†W18), $\neg x \to (\neg y \to z) = \neg x \to (\neg z \to \neg\neg y)$, hence, by (†W20) and (†W23), $\neg x \to (\neg y \to z) = \neg z \to (\neg x \to y)$, hence, by (†W18) and (†W23), $\neg x \to (\neg y \to z) = \neg(\neg x \to y) \to z$, hence, by (∗W8), $x \veebar (y \veebar z) = (x \veebar y) \veebar z$.

(⋆MV4) By (†W16), $\neg x \to \overline{1} = \overline{1}$, hence, by (∗W8), $x \veebar \overline{1} = \overline{1}$.

(⋆MV5) By (∗W6), $\neg x \to \overline{0} = \neg\neg x$, hence, by (†W23), $\neg x \to \overline{0} = x$, hence, by (∗W8), $x \veebar \overline{0} = x$.

(⋆MV7) By (†W23).

(⋆MV9) By (⋆W4). □

∗92. *Let* $\mathbf{A} = (\mathrm{V}, \veebar, \neg, \overline{0})$ *be an MV algebra and let* $\mathbf{A}' = (\mathrm{V}, \to, \neg, \&, \veebar, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ *be its MV-induced algebra. For all* $x, y, z \in \mathrm{V}$*, we now have*

$$x \veebar \neg x = \overline{1}, \qquad (\dagger\mathrm{MV3})$$

$$\neg(x \veebar y) = \neg x \,\&\, \neg y, \qquad (\dagger\mathrm{MV6})$$

Furthermore, the identities from definitions 46 and 47 hold.

Proof.

(†MV3) By (⋆MV9) and (∗MV15), $(x \mathbin{\&} \neg y) \mathbin{\underline{\vee}} y = (y \mathbin{\&} \neg x) \mathbin{\underline{\vee}} x$, hence, by (∗MV12), $\neg(\neg x \mathbin{\underline{\vee}} \neg\neg y) \mathbin{\underline{\vee}} y = \neg(\neg y \mathbin{\underline{\vee}} \neg\neg x) \mathbin{\underline{\vee}} x$, hence, by (⋆MV7) twice, $\neg(\neg x \mathbin{\underline{\vee}} y) \mathbin{\underline{\vee}} y = \neg(\neg y \mathbin{\underline{\vee}} x) \mathbin{\underline{\vee}} x$, hence, by substituting $\overline{1}$ for y, $\neg(\neg x \mathbin{\underline{\vee}} \overline{1}) \mathbin{\underline{\vee}} \overline{1} = \neg(\neg\overline{1} \mathbin{\underline{\vee}} x) \mathbin{\underline{\vee}} x$, hence, by (⋆MV4), $\overline{1} = \neg(\neg\overline{1} \mathbin{\underline{\vee}} x) \mathbin{\underline{\vee}} x$, hence, by (∗MV8), $\overline{1} = \neg(\neg\neg\overline{0} \mathbin{\underline{\vee}} x) \mathbin{\underline{\vee}} x$, hence, by (⋆MV7), $\overline{1} = \neg(\overline{0} \mathbin{\underline{\vee}} x) \mathbin{\underline{\vee}} x$, hence, by (⋆MV5), $\overline{1} = \neg x \mathbin{\underline{\vee}} x$.

(†MV6) By (∗MV12), $\neg x \mathbin{\&} \neg y = \neg(\neg\neg x \mathbin{\underline{\vee}} \neg\neg y)$, hence, by (⋆MV7), $\neg x \mathbin{\&} \neg y = \neg(x \mathbin{\underline{\vee}} y)$.

(∗W5) By (∗MV13), $\overline{0} \to \overline{0} = \neg\overline{0} \mathbin{\underline{\vee}} \overline{0}$, hence, by (†MV3), $\overline{0} \to \overline{0} = \overline{1}$.

(∗W6) By (∗MV13), $x \to \overline{0} = \neg x \mathbin{\underline{\vee}} \overline{0}$, hence, by (⋆MV5), $x \to \overline{0} = \neg x$.

(∗W7) By (∗MV12), $x \mathbin{\&} y = \neg(\neg x \mathbin{\underline{\vee}} \neg y)$, hence, by (∗MV13), $x \mathbin{\&} y = \neg(x \to \neg y)$.

(∗W8) By (∗MV13), $\neg x \to y = \neg\neg x \mathbin{\underline{\vee}} y$, hence, by (⋆MV7), $\neg x \to y = x \mathbin{\underline{\vee}} y$.

(∗W10) By (∗MV15), $x \vee y = (x \mathbin{\&} \neg y) \mathbin{\underline{\vee}} y$, hence, by (∗MV12), $x \vee y = \neg(\neg x \mathbin{\underline{\vee}} \neg\neg y) \mathbin{\underline{\vee}} y$, hence, by (⋆MV7), $x \vee y = \neg(\neg x \mathbin{\underline{\vee}} y) \mathbin{\underline{\vee}} y$, hence, by (∗MV13) twice, $x \vee y = (x \to y) \to y$.

(∗W9) By (∗MV16), $x \wedge y = (x \mathbin{\underline{\vee}} \neg y) \mathbin{\&} y$, hence, by (⋆MV7), $x \wedge y = (\neg\neg x \mathbin{\underline{\vee}} \neg y) \mathbin{\&} y$, hence, by (∗MV13), $x \wedge y = (\neg x \to \neg y) \mathbin{\&} y$, hence, by (∗W7), $x \wedge y = \neg((\neg x \to \neg y) \to \neg y)$, hence, by (⋆MV9) and (∗W10), $x \wedge y = \neg((\neg y \to \neg x) \to \neg x)$, hence, by (∗MV13), $x \wedge y = \neg(\neg(\neg y \to \neg x) \mathbin{\underline{\vee}} \neg x)$, hence, by (⋆MV7), $x \wedge y = \neg(\neg x \mathbin{\underline{\vee}} \neg(\neg y \to \neg x))$, hence, by (∗MV12), $x \wedge y = x \mathbin{\&} (\neg y \to \neg x)$, hence, by (∗MV13), $x \wedge y = x \mathbin{\&} (\neg\neg y \mathbin{\underline{\vee}} \neg x)$, hence, by (⋆MV7), $x \wedge y = x \mathbin{\&} (y \mathbin{\underline{\vee}} \neg x)$, hence, by (⋆MV1), $x \wedge y = x \mathbin{\&} (\neg x \mathbin{\underline{\vee}} y)$, hence, by (∗MV13), $x \wedge y = x \mathbin{\&} (x \to y)$.

(∗W11) By (∗MV14).

(∗W12) By (∗MV17).

(⋆W1) By (∗MV13), $\overline{1} \to y = \neg\overline{1} \mathbin{\underline{\vee}} y$, hence, by (∗MV8), $\overline{1} \to y = \neg\neg\overline{0} \mathbin{\underline{\vee}} y$, hence, by (⋆MV7), $\overline{1} \to y = \overline{0} \mathbin{\underline{\vee}} y$, hence, by (⋆MV5), $\overline{1} \to y = y$.

(⋆W2) Let a be as follows:

$$a \overset{\text{def}}{=} \big(x \to y\big) \to \big((y \to z) \to (x \to z)\big).$$

By (∗MV13),

$$a = \neg\big(\neg x \mathbin{\underline{\vee}} y\big) \mathbin{\underline{\vee}} \big(\neg(\neg y \mathbin{\underline{\vee}} z) \mathbin{\underline{\vee}} (\neg x \mathbin{\underline{\vee}} z)\big),$$

hence, by (⋆MV1) and (⋆MV2),

$$a = \big(\neg(\neg x \mathbin{\underline{\vee}} y) \mathbin{\underline{\vee}} \neg x\big) \mathbin{\underline{\vee}} \big(\neg(\neg y \mathbin{\underline{\vee}} z) \mathbin{\underline{\vee}} z\big),$$

hence, by (⋆MV1) and (⋆MV7),

$$a = \big(\neg(\neg\neg y \mathbin{\underline{\vee}} \neg x) \mathbin{\underline{\vee}} \neg x\big) \mathbin{\underline{\vee}} \big(\neg(\neg y \mathbin{\underline{\vee}} \neg\neg z) \mathbin{\underline{\vee}} z\big),$$

hence, by (∗MV12) and (⋆MV7),

$$a \;=\; \big((\neg y \,\&\, \neg\neg x) \veebar \neg x\big) \;\veebar\; \big((y \,\&\, \neg z) \veebar z\big),$$

hence, by (∗MV15),

$$a \;=\; (\neg y \vee \neg x) \;\veebar\; (y \vee z),$$

hence, by (⋆W4),

$$a \;=\; (\neg x \vee \neg y) \;\veebar\; (z \vee y),$$

hence, by (∗W10),

$$a \;=\; \big((\neg x \to \neg y) \to \neg y\big) \;\veebar\; \big((z \to y) \to y\big),$$

hence, by (∗MV13),

$$a \;=\; \big(\neg(\neg x \to \neg y) \veebar \neg y\big) \;\veebar\; \big(\neg(z \to y) \veebar y\big),$$

hence, by (⋆MV1) and (⋆MV2),

$$a \;=\; \big(\neg(\neg x \to \neg y) \veebar \neg(z \to y)\big) \;\veebar\; \big(\neg y \veebar y\big),$$

hence, by (†MV3) and (⋆MV4), $a = \overline{1}$, so, substituting for the definition of a,

$$\big(x \to y\big) \;\to\; \big((y \to z) \to (x \to z)\big) \;=\; \overline{1}.$$

(⋆W3) Let a be as follows:

$$a \overset{\text{def}}{=} (\neg x \to \neg y) \;\to\; (y \to x).$$

By (∗MV13),

$$a \;=\; \neg(\neg\neg x \veebar \neg y) \veebar (\neg y \veebar x),$$

hence, by (∗MV12) and (⋆MV7),,

$$a \;=\; (\neg x \,\&\, \neg\neg y) \veebar (\neg y \veebar x),$$

hence, by (⋆MV2),

$$a \;=\; \big((\neg x \,\&\, \neg\neg y) \veebar \neg y\big) \;\veebar\; x,$$

hence, by (∗MV15),

$$a \;=\; (\neg x \vee \neg y) \;\veebar\; x,$$

hence, by (⋆MV9),

$$a \;=\; (\neg y \vee \neg x) \;\veebar\; x,$$

hence, by (∗W10),

$$a \;=\; \big((\neg y \to \neg x) \to \neg x\big) \;\veebar\; x,$$

hence, by (∗MV13),

$$a \;=\; \big(\neg(\neg y \to \neg x) \mathbin{\underline{\vee}} \neg x\big) \;\mathbin{\underline{\vee}}\; x,$$

hence, by (⋆MV2),

$$a \;=\; \neg(\neg y \to \neg x) \;\mathbin{\underline{\vee}}\; (\neg x \mathbin{\underline{\vee}} x),$$

hence, by (†MV3) and (⋆MV4), $a = \overline{1}$, so, substituting for the definition of a,

$$(\neg x \to \neg y) \;\to\; (y \to x) \;=\; \overline{1}.$$

(⋆W4) By (⋆MV9). □

Proof of theorem 51. This follows immediately from lemmata 91 and 92. □

∗**93.** *Let* $(\mathrm{V}, \to, \neg, \mathbin{\&}, \mathbin{\underline{\vee}}, \wedge, \vee, \equiv, \not\equiv, \overline{0}, \overline{1})$ *be a Łukasiewicz algebra. For all* $x, y, z \in \mathrm{V}$, *we then have*

$$\neg(x \vee y) = \neg x \wedge \neg y, \tag{†Ł1}$$

$$y \to (x \vee y) = \overline{1}, \tag{†Ł2}$$

$$x \to (x \vee y) = \overline{1}, \tag{†Ł3}$$

$$(x \wedge y) \to y = \overline{1}, \tag{†Ł4}$$

$$(x \wedge y) \to x = \overline{1}, \tag{†Ł5}$$

$$x \vee x = x, \tag{†Ł6}$$

$$x \vee (x \wedge y) = x, \tag{†Ł7}$$

$$x \vee \overline{1} = \overline{1}, \tag{†Ł8}$$

$$x \vee \overline{0} = x, \tag{†Ł9}$$

$$(x \to z) \to ((y \to z) \to ((x \vee y) \to z)) = \overline{1}, \tag{†Ł10}$$

$$(z \to x) \to ((z \to y) \to (z \to (x \wedge y))) = \overline{1}, \tag{†Ł11}$$

$$(x \mathbin{\&} y) \to z = x \to (y \to z), \tag{†Ł12}$$

$$(x \mathbin{\&} y) \to z = (\neg z \mathbin{\&} y) \to \neg x, \tag{†Ł13}$$

$$(x \mathbin{\&} y) \to z = (x \mathbin{\&} \neg z) \to \neg y, \tag{†Ł14}$$

$$x \to \big(\; y \;\to\; (x \mathbin{\&} y) \;\big) = \overline{1}, \tag{†Ł15}$$

$$(x \to y) \to ((z \mathbin{\&} x) \to (z \mathbin{\&} y)) = \overline{1}, \tag{†Ł16}$$

$$\big((x_1 \to y_1) \mathbin{\&} (x_2 \to y_2)\big) \;\to\; \big((x_1 \mathbin{\&} x_2) \to (y_1 \mathbin{\&} y_2)\big) = \overline{1}. \tag{†Ł17}$$

Proof.

(†Ł1) By (∗MV15), $\neg(x \vee y) = \neg((x \mathbin{\&} \neg y) \mathbin{\underline{\vee}} y)$. hence, by (∗MV12), $\neg(x \vee y) = \neg(\neg(\neg x \mathbin{\underline{\vee}} \neg\neg y) \mathbin{\underline{\vee}} y)$, hence, by (⋆MV7), $\neg(x \vee y) = \neg(\neg(\neg x \mathbin{\underline{\vee}} \neg\neg y) \mathbin{\underline{\vee}} \neg\neg y)$, hence, by (∗MV12), $\neg(x \vee y) = (\neg x \mathbin{\underline{\vee}} \neg\neg y) \mathbin{\&} \neg y$, hence, by (∗MV16), $\neg(x \vee y) = \neg x \wedge \neg y$.

(†Ł2) By (†W19), $y \to \big((x \to y) \to y \big) = \overline{1}$, so, by (∗W10), $y \to (x \vee y) = \overline{1}$.

(†Ł3) By (†Ł2), $x \to (y \vee x) = 1$, so, by (⋆W4), $x \to (x \vee y) = 1$.

(†Ł4) By (†Ł2), $\neg y \to (\neg x \vee \neg y) = \overline{1}$, hence, by (†Ł1) and (⋆MV7), $\neg y \to \neg(x \wedge y) = \overline{1}$, hence, by (†W18), $(x \wedge y) \to y = \overline{1}$.

(†Ł5) By (†Ł3), $\neg x \to (\neg x \vee \neg y) = \overline{1}$, hence, by (†Ł1) and (⋆MV7), $\neg x \to \neg(x \wedge y) = \overline{1}$, hence, by (†W18), $(x \wedge y) \to x = \overline{1}$.

(†Ł6) By (∗W10), $(x \vee x) = \big((x \to x) \to x \big)$, hence, $(x \vee x) \to x = \big((x \to x) \to x \big) \to x$. hence, by (⋆W4) and (∗W10), $(x \vee x) \to x = \big(x \to (x \to x) \big) \to (x \to x)$, hence, by (†W15) twice, $(x \vee x) \to x = \big(x \to \overline{1} \big) \to \overline{1}$, hence, by (†W16) twice, $(x \vee x) \to x = \overline{1}$. By (†Ł2), $x \to (x \vee x) = 1$, so, by (†W13), $x = (x \vee x)$.

(†Ł8) By (∗MV15), $x \vee \overline{1} = (x \mathbin{\&} \neg\overline{1}) \mathbin{\underline{\vee}} \overline{1}$, hence, by (⋆MV4) $x \vee \overline{1} = \overline{1}$.

(†Ł9) By (∗MV15), $x \vee \overline{0} = (x \mathbin{\&} \neg\overline{0}) \mathbin{\underline{\vee}} \overline{0}$, hence, by (∗MV8) $x \vee \overline{0} = (x \mathbin{\&} \overline{1}) \mathbin{\underline{\vee}} \overline{0}$, hence, by (†MV5′) $x \vee \overline{0} = x \mathbin{\underline{\vee}} \overline{0}$, hence, by (⋆MV5) $x \vee \overline{0} = x$.

(†Ł10) By (⋆W2),

$$(x \to y) \to \big((y \to z) \to (x \to z) \big),$$

so, by (†W20),

$$(y \to z) \to \big((x \to y) \to (x \to z) \big).$$

So we have

$$(x \to z) \to \Big(\big((y \to x) \to x\big) \to \big((y \to z) \to z\big) \Big),$$

so, by (⋆W4) and (∗W10),

$$\underbrace{(x \to z)}_{a} \to \underbrace{\big((x \vee y) \to (y \vee z) \big)}_{b} = \overline{1}.$$

As a special case of this, we get

$$(y \to z) \to \big((y \vee z) \to (z \vee z) \big) = \overline{1},$$

and hence, by (†Ł6),

$$\underbrace{\big(y \to z \big)}_{c} \to \underbrace{\big((y \vee z) \to z \big)}_{d} = \overline{1}.$$

Then, note that, by (†Ł3), we have

$$\underbrace{\big((x \vee y) \to (y \vee z) \big)}_{b} \to \Big(\underbrace{\big((y \vee z) \to z\big)}_{d} \to \underbrace{\big((x \vee y) \to z\big)}_{e} \Big) = \overline{1}.$$

In what follows, we will continue the proof using the abbreviations a, b, c, d, e. Now, by (†Ł3), we have

$$\big(\ a\ \rightarrow\ b\ \big) \rightarrow \Big(\ \big(b \rightarrow (d \rightarrow e)\big)\ \rightarrow\ \big(a \rightarrow (d \rightarrow e)\big)\ \Big)\ =\ \overline{1}, \tag{$*$}$$

hence, by (†Ł2) twice, $a \rightarrow (d \rightarrow e) = \overline{1}$, hence, by (†W20),

$$d \rightarrow (a \rightarrow e)\ =\ \overline{1}.$$

As a variant of ($*$), we also get

$$\big(\ c\ \rightarrow\ d\ \big) \rightarrow \Big(\ \big(d \rightarrow (a \rightarrow e)\big)\ \rightarrow\ \big(c \rightarrow (a \rightarrow e)\big)\ \Big)\ =\ \overline{1},$$

so, again by (†Ł2) twice, $c \rightarrow (a \rightarrow e) = \overline{1}$, and, by (†W20),

$$\underbrace{(x \rightarrow z)}_{a} \rightarrow \Big(\ \underbrace{(y \rightarrow z)}_{c}\ \rightarrow\ \underbrace{\big((x \vee y) \rightarrow z\big)}_{e}\ \Big)\ =\ \overline{1}.$$

(†Ł11) By (†Ł10),

$$(\neg x \rightarrow \neg z) \rightarrow \Big(\ \big(\neg y \rightarrow \neg z\big)\ \rightarrow\ \big((\neg x \vee \neg y) \rightarrow \neg z\big)\ \Big)\ =\ \overline{1},$$

hence, by (†Ł1) and (⋆MV7),

$$(\neg x \rightarrow \neg z) \rightarrow \Big(\ \big(\neg y \rightarrow \neg z\big)\ \rightarrow\ \big(\neg(x \wedge y) \rightarrow \neg z\big)\ \Big)\ =\ \overline{1},$$

hence, by (†W18) three times,

$$(z \rightarrow x) \rightarrow \Big(\ \big(z \rightarrow y\big)\ \rightarrow\ \big(z \rightarrow (x \wedge y)\big)\ \Big)\ =\ \overline{1}.$$

(†Ł7) By (†Ł10),

$$((a \wedge b) \rightarrow a)\,\&\,(a \rightarrow a)) \rightarrow (((a \wedge b) \vee a) \rightarrow a)\ =\ \overline{1}.$$

But, by (†Ł4), $((a \wedge b) \rightarrow a) = \overline{1}$, and, by (†W15), $a \rightarrow a = \overline{1}$. So, by (†MV5′),

$$((a \wedge b) \vee a) \rightarrow a = \overline{1}.$$

By (†Ł2),

$$a \rightarrow ((a \wedge b) \vee a) = \overline{1}.$$

So, by (†W13),

$$((a \wedge b) \vee a) = a.$$

(†Ł12) By ($*$W7), $(x\,\&\,y) \rightarrow z = \neg(x \rightarrow \neg y) \rightarrow z$, hence, by (†W18) and (†W23), $(x\,\&\,y) \rightarrow z = \neg z \rightarrow (x \rightarrow \neg y)$, hence, by (†W20), $(x\,\&\,y) \rightarrow z = x \rightarrow (\neg z \rightarrow \neg y)$, hence, by (†W18) and (†W23), $(x\,\&\,y) \rightarrow z = x \rightarrow (y \rightarrow z)$.

(†Ł13) By (†Ł12), $(x \& y) \to z = x \to (y \to z)$, hence, by (†W18), $(x \& y) \to z = x \to (\neg z \to \neg y)$, hence, by (†W20), $(x \& y) \to z = \neg z \to (x \to \neg y)$, hence, by (†W18) and (⋆MV7), $(x \& y) \to z = \neg z \to (y \to \neg x)$, hence, by (†Ł12), $(x \& y) \to z = (\neg z \& y) \to \neg x$.

(†Ł14) By (†Ł13) and (⋆MV1), $(y \& x) \to z = (y \& \neg z) \to \neg x$.

(†Ł15) By (†W18), (†W23), and (†W13) $(z \to \neg x) \to (x \to \neg y) = \overline{1}$, hence, by (†W20), $x \to \big((y \to \neg x) \to \neg y \big) = \overline{1}$, hence, by (†W18) and (†W23), $x \to \big(y \to \neg(y \to \neg x) \big) = \overline{1}$, hence, by (∗W7), $x \to \big(y \to (x \& y) \big) = \overline{1}$.

(†Ł16) By (⋆W2), $(z \to \neg y) \to ((\neg y \to \neg x) \to (z \to \neg x))$, hence, by (†W20), $(\neg y \to \neg x) \to ((z \to \neg y) \to (z \to \neg x))$, hence, by (†W18), $(x \to y) \to ((z \to \neg y) \to (z \to \neg x))$, hence, by (∗W7), $(x \to y) \to (\neg(z \& y) \to \neg(z \& x))$, hence, by (†W18), $(x \to y) \to ((z \& x) \to (z \& y))$.

(†Ł17) First note that

$$x \& y = y \& x. \tag{†MV1′}$$

We get this from (⋆MV1) via duality (corollary 50). Now, by (†MV1′) and (†Ł16),

$$\underbrace{\big(x_1 \to y_1\big)}_{a} \to \underbrace{\big((x_1 \& x_2) \to (y_1 \& x_2)\big)}_{b} = \overline{1}.$$

Now, using the abbreviations a and b, again by (†MV1′) and (†Ł16), we get

$$\big(a \to b\big) \to \big((a \& c) \to (b \& c)\big) = \overline{1},$$

and hence, by (⋆W1),

$$(a \& c) \to (b \& c) = \overline{1}. \tag{∗}$$

Similarly, by (†Ł16),

$$\underbrace{\big(x_2 \to y_2\big)}_{c} \to \underbrace{\big((y_1 \& x_2) \to (y_1 \& y_2)\big)}_{d} = \overline{1},$$

and, using the abbreviations c and d, and again by (†Ł16),

$$\big(c \to d\big) \to \big((b \& c) \to (b \& d)\big) = \overline{1},$$

and hence, by (⋆W1),

$$(b \& c) \to (b \& d) = \overline{1}. \tag{∗∗}$$

Now recall that

$$b = \underbrace{(x_1 \,\&\, x_2)}_{e} \to \underbrace{(y_1 \,\&\, x_2)}_{f}$$

and

$$d = \underbrace{(y_1 \,\&\, x_2)}_{f} \to \underbrace{(y_1 \,\&\, y_2)}_{g}.$$

So, using the abbreviations e, f, g, by ($\star$W2),

$$\big(e \to f\big) \to \big((f \to g) \to (e \to g)\big),$$

hence, by (†Ł12),

$$\Big(\underbrace{(e \to f)}_{b} \,\&\, \underbrace{(f \to g)}_{d}\Big) \;\to\; \big(e \to g\big) \;=\; \overline{1}. \qquad (*\,*\,*)$$

Now, applying (†W14) twice to ($*$), ($**$), and ($*\,*\,*$), we get

$$(a \,\&\, c) \to (e \to g) = \overline{1},$$

and, by substituting for the abbreviations,

$$\Big(\underbrace{(x_1 \to y_1)}_{a} \,\&\, \underbrace{(x_2 \to y_2)}_{c}\Big) \;\to\; \Big(\underbrace{(x_1 \,\&\, x_2)}_{e} \to \underbrace{(y_1 \,\&\, y_2)}_{g}\Big) \;=\; \overline{1}.$$

□

Proof of theorem 53.

(†MV3) See lemma 92.

(†MV6) See lemma 92.

(†MV10) First note that, by (†Ł2) and by (†Ł3), we have

$$\big(x \vee y\big) \to \big((x \vee y) \vee z\big) \;=\; \overline{1},$$
$$z \;\to \big((x \vee y) \vee z\big) \;=\; \overline{1},$$
$$x \;\to (x \vee y) \;=\; \overline{1},$$
$$y \;\to (x \vee y) \;=\; \overline{1},$$

And by ($\star$W2), we have

$$\big(\,x \;\to\; (x \vee y)\,\big) \;\to\; \Big(\,\big((x \vee y) \to ((x \vee y) \vee z)\big) \to \big(x \to ((x \vee y) \vee z)\big)\,\Big) \;=\; \overline{1},$$

$$\big(\,y \;\to\; (x \vee y)\,\big) \;\to\; \Big(\,\big((x \vee y) \to ((x \vee y) \vee z)\big) \to \big(y \to ((x \vee y) \vee z)\big)\,\Big) \;=\; \overline{1},$$

hence, by (†Ł2) twice,

$$x \;\to ((x \vee y) \vee z) \;=\; \overline{1},$$

$$y \to ((x \vee y) \vee z) = \overline{1}.$$

From here on, let the abbreviation a stand for $((x \vee y) \vee z)$. Now, by (†Ł10),

$$(y \to a) \to \Big((z \to a) \to \big((y \vee z) \to a\big)\Big) = \overline{1}.$$

So, by (†Ł2) twice,

$$(y \vee z) \to a.$$

Similarly, by (†Ł10),

$$(x \to a) \to \Big(((y \vee z) \to a) \to \big((x \vee (y \vee z)) \to a\big)\Big) = \overline{1},$$

so, by (†Ł2) twice,

$$(x \vee (y \vee z)) \to a = \overline{1}.$$

Substituting for the abbreviation, we get

$$(x \vee (y \vee z)) \to ((x \vee y) \vee z) = \overline{1}.$$

By (⋆W4), this yields,

$$((z \vee y) \vee x) \to (z \vee (y \vee x)) = \overline{1},$$

which, by exchanging the roles of x and z through substitution, yields,

$$((x \vee y) \vee z) \to (x \vee (y \vee z)) = \overline{1}.$$

So, by (†W13) we have

$$(x \vee y) \vee z = x \vee (y \vee z).$$

(†MV11) By (†Ł16),

$$\Big(y \to (y \vee z) \Big) \to \Big((x \& y) \to (x \& (y \vee z)) \Big) = \overline{1},$$

hence, by (†Ł3) and (⋆W1),

$$\underbrace{\big(x \& y \big)}_{a} \to \underbrace{\big(x \& (y \vee z) \big)}_{c} = \overline{1}.$$

Similarly, by (†Ł16),

$$\Big(z \to (y \vee z) \Big) \to \Big((x \& z) \to (x \& (y \vee z)) \Big) = \overline{1},$$

hence, by (†Ł2) and (⋆W1),

$$\underbrace{\big(x \& z \big)}_{b} \to \underbrace{\big(x \& (y \vee z) \big)}_{c} = \overline{1}.$$

Then, using the abbreviations a, b, c, we have by (†Ł10),

$$\big(a \to c\big) \to \big((b \to c) \to ((a \vee b) \to c)\big) = \overline{1},$$

hence, by (†Ł2) twice, $(a \vee b) \to c = \overline{1}$, hence, substituting for the abbreviations,

$$\big((x \,\&\, y) \vee (x \,\&\, z)\big) \to \big(x \,\&\, (y \vee z)\big) = \overline{1}. \tag{$*$}$$

Next, note that, by (†Ł15),

$$x \to \big(z \to (x \,\&\, z)\big) = \overline{1},$$

hence, by (†W20),

$$z \to \underbrace{\big(x \to (x \,\&\, z)\big)}_{d} = \overline{1}.$$

Then, using the abbreviation d, we have, by (†Ł10),

$$\big(y \to d\big) \to \big((z \to d) \to ((y \vee z) \to d)\big) = \overline{1},$$

hence, by (†Ł2), $(y \to d) \to ((y \vee z) \to d) = \overline{1}$, hence, substituting for the abbreviation,

$$\big(y \to (x \to (x \,\&\, z))\big) \to \big((y \vee z) \to (x \to (x \,\&\, z))\big) = \overline{1},$$

hence, by (†W20) twice,

$$\big(x \to (y \to (x \,\&\, z))\big) \to \big(x \to \big((y \vee z) \to (x \,\&\, z)\big)\big) = \overline{1},$$

hence, by (†Ł12),

$$\big((x \,\&\, y) \to (x \,\&\, z)\big) \to \big(\big(x \,\&\, (y \vee z)\big) \to \big(x \,\&\, z\big)\big) = \overline{1},$$

hence, by (†W20),

$$\big(x \,\&\, (y \vee z)\big) \to \big(\big((x \,\&\, y) \to (x \,\&\, z)\big) \to \big(x \,\&\, z\big)\big) = \overline{1},$$

hence, by (*W10),

$$\big(x \,\&\, (y \vee z)\big) \to \big((x \,\&\, y) \vee (x \,\&\, z)\big) = \overline{1}. \tag{$**$}$$

Applying (†W13) to (*) and (**), we get

$$x \,\&\, (y \vee z) = (x \,\&\, y) \vee (x \,\&\, z).$$

Finally, (†MV11) follows from this by duality (corollary 50). □

Proof of theorem 55. This follows immediately from lemmata 91 and 93 and from corollary 50. □

B.2. Proofs for section 3.3

Proof of lemma 73.

($\star$SYL·comm) By ($\dagger$MV9′)

$$\begin{aligned} x_1 \wedge y_1 &= y_1 \wedge x_1, \\ x_2 \wedge y_2 &= y_2 \wedge x_2, \\ &\dots \\ x_N \wedge y_N &= y_N \wedge x_N, \end{aligned}$$

hence,

$$\begin{aligned} &(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \ldots \vee (x_N \wedge y_N) \\ &\quad = \ (y_1 \wedge x_1) \vee (y_2 \wedge x_2) \vee \ldots \vee (y_N \wedge x_N), \end{aligned}$$

(1) By ($\dagger$Ł4) and ($\dagger$Ł5), we have

$$\begin{aligned} x_1 \wedge x_2 \rightarrow x_1 &= \overline{1}, \\ x_1 \wedge x_2 \rightarrow x_2 &= \overline{1}, \\ y_1 \wedge y_2 \rightarrow y_1 &= \overline{1}, \\ y_1 \wedge y_2 \rightarrow y_2 &= \overline{1}. \end{aligned}$$

By ($\dagger$Ł17),

$$\begin{aligned} &\big((x_1 \wedge x_2) \rightarrow x_1\big) \rightarrow \Big(\big((y_1 \wedge y_2) \rightarrow y_1\big) \\ &\quad \rightarrow \Big(\big((x_1 \wedge x_2) \,\&\, (y_1 \wedge y_2)\big) \rightarrow (x_1 \,\&\, y_1)\Big)\Big) = \overline{1}, \\ &\big((x_1 \wedge x_2) \rightarrow x_2\big) \rightarrow \Big(\big((y_1 \wedge y_2) \rightarrow y_2\big) \\ &\quad \rightarrow \Big(\big((x_1 \wedge x_2) \,\&\, (y_1 \wedge y_2)\big) \rightarrow (x_2 \,\&\, y_2)\Big)\Big) = \overline{1}. \end{aligned}$$

So, by ($\star$W1) twice,

$$\begin{aligned} \underbrace{\big((x_1 \wedge x_2) \,\&\, (y_1 \wedge y_2)\big)}_{a} \rightarrow \underbrace{(x_1 \,\&\, y_1)}_{b_1} &= \overline{1}, \\ \underbrace{\big((x_1 \wedge x_2) \,\&\, (y_1 \wedge y_2)\big)}_{a} \rightarrow \underbrace{(x_2 \,\&\, y_2)}_{b_2} &= \overline{1}. \end{aligned}$$

By ($\star$W2),

$$\begin{aligned} \big(a \rightarrow b_1\big) \rightarrow \big((b_1 \rightarrow z_1) \rightarrow (a \rightarrow z_1)\big) &= \overline{1}, \\ \big(a \rightarrow b_2\big) \rightarrow \big((b_2 \rightarrow z_2) \rightarrow (a \rightarrow z_2)\big) &= \overline{1}, \end{aligned}$$

hence, by (⋆W1),

$$\underbrace{(b_1 \to z_1)}_{c_1} \to \underbrace{(a \to z_1)}_{d_1} \;=\; \overline{1},$$

$$\underbrace{(b_2 \to z_2)}_{c_2} \to \underbrace{(a \to z_2)}_{d_2} \;=\; \overline{1}.$$

By (†Ł17),

$$((c_1 \to d_1) \mathbin{\&} (c_2 \to d_2)) \to ((c_1 \mathbin{\&} c_2) \to (d_1 \mathbin{\&} d_2)) \;=\; \overline{1},$$

hence,

$$(\overline{1} \mathbin{\&} \overline{1}) \to ((c_1 \mathbin{\&} c_2) \to (d_1 \mathbin{\&} d_2)) \;=\; \overline{1},$$

hence, by (†MV5′) and (⋆W1),

$$(c_1 \mathbin{\&} c_2) \to (d_1 \mathbin{\&} d_2) \;=\; \overline{1}, \tag{$*$}$$

By (†Ł17),

$$\big(a \to z_1\big) \;\to\; \big((a \to z_2) \to (a \to (z_1 \wedge z_2))\big) = \overline{1},$$

hence, by (†Ł12),

$$\big(\underbrace{(a \to z_1)}_{d_1} \mathbin{\&} \underbrace{(a \to z_2)}_{d_2}\big) \;\to\; \big(a \to (z_1 \wedge z_2)\big) = \overline{1}, \tag{$**$}$$

hence, by applying (†W14) to (∗) and (∗∗),

$$\big(c_1 \mathbin{\&} c_2\big) \to \big(a \to (z_1 \wedge z_2)\big) = \overline{1}.$$

This is the same as

$$\begin{aligned} &\Big(\big(\underbrace{(x_1 \mathbin{\&} y_1)}_{b_1} \to z_1\big) \;\&\; \big(\underbrace{(x_2 \mathbin{\&} y_2)}_{b_2} \to z_2\big)\Big) \\ &\quad \to \Big(\underbrace{((x_1 \wedge x_2) \mathbin{\&} (y_1 \wedge y_2))}_{a} \;\to\; (z_1 \wedge z_2)\Big) \;=\; \overline{1}. \end{aligned} \tag{1}$$

(⋆SYL·1·A) Suppose that

$$\Big(\underbrace{(y_n \to z_n)}_{a_n} \mathbin{\&} \underbrace{(x_n \to y_n)}_{b_n}\Big) \to \underbrace{(x_n \to z_n)}_{c_n} \;=\; \overline{1}, \tag{$*$}$$

$$\begin{aligned} &\Big(\;\big(\underbrace{(y_1 \to z_1) \wedge (y_2 \to z_2) \wedge \ldots \wedge (y_{n-1} \to z_{n-1})}_{a'_{n-1}}\big) \\ &\mathbin{\&}\; \big(\underbrace{(x_1 \to y_1) \wedge (x_2 \to y_2) \wedge \ldots \wedge (x_{n-1} \to y_{n-1})}_{b'_{n-1}}\big)\;\Big) \\ &\to \Big(\;\underbrace{(x_1 \to z_1) \wedge (x_2 \to z_2) \wedge \ldots \wedge (x_{n-1} \to z_{n-1})}_{c'_{n-1}}\;\Big) \;=\; \overline{1}. \end{aligned} \tag{$**$}$$

Then, by (1),

$$\Big(\big((a'_{n-1} \,\&\, b'_{n-1}) \to c'_{n-1}\big) \;\&\; \big((a_n \,\&\, b_n) \to c_n\big) \Big)$$
$$\to \Big(\big((a'_{n-1} \wedge a_n) \,\&\, (b'_{n-1} \wedge b_n)\big) \;\to\; (c'_{n-1} \wedge c_n) \Big) \;=\; \overline{1},$$

hence, by (†MV5′) and (⋆W1),

$$\Big(\; \underbrace{\big(\underbrace{(y_1 \to z_1) \wedge (y_2 \to z_2) \wedge \ldots \wedge (y_{n-1} \to z_{n-1})}_{a'_{n-1}} \wedge \underbrace{(y_n \to z_n)}_{a_n} \big)}_{a'_n}$$
$$\& \; \underbrace{\big(\underbrace{(x_1 \to y_1) \wedge (x_2 \to y_2) \wedge \ldots \wedge (x_{n-1} \to y_{n-1})}_{b'_{n-1}} \wedge \underbrace{(x_n \to y_n)}_{b_n} \big)}_{b'_n} \Big)$$
$$\to \Big(\; \underbrace{\underbrace{(x_1 \to z_1) \wedge (x_2 \to z_2) \wedge \ldots \wedge (x_{n-1} \to z_{n-1})}_{c'_{n-1}} \wedge \underbrace{(x_n \to z_n)}_{c_n}}_{c'_n} \; \Big) \;=\; \overline{1},$$

We can now go on to show (⋆SYL·1·A): By (⋆W2), (†Ł12), and (†MV1′), we know that (⋆SYL·1·A) is fulfilled for $N = 1$, and, equivalently, that (∗) is fulfilled for any n and that (∗∗) is fulfilled for $n = 2$.

Our proof of (⋆SYL·1·A) for the case $N \geq 2$ proceeds by induction. First note that, for the base case $N = 2$, both (∗) and (∗∗) are fulfilled, and, by the above argument, so is (⋆SYL·1·A). This also shows (∗∗) for the case $n = 3$. Then, by the same argument we also have (⋆SYL·1·A) for $N = 3$, etc.

(⋆SYL·2·A) By (†Ł4), $(x \wedge y) \to y = \overline{1}$. By (†MV1′) and (†Ł16),

$$\big(\, (x \wedge y) \;\to\; y \, \big) \;\to\; \Big(\big((x \wedge y) \,\&\, (y \to z)\big) \;\to\; \big(y \,\&\, (y \to z)\big) \Big) \;=\; \overline{1},$$

hence, by (⋆W1),

$$\big((x \wedge y) \,\&\, (y \to z)\big) \;\to\; \big(y \,\&\, (y \to z)\big) \;=\; \overline{1},$$

hence, by by (∗W9),

$$\big((x \wedge y) \,\&\, (y \to z)\big) \;\to\; \big(y \wedge z\big) \;=\; \overline{1}.$$

By (†Ł4), $(y \wedge z) \to z = \overline{1}$. Hence, by (†W14),

$$\underbrace{\big((x \wedge y) \,\&\, (y \to z)\big)}_{d} \;\to\; z \;=\; \overline{1}.$$

Also note that, by (†Ł4), $(x \wedge y) \to x = \overline{1}$, so, by (†MV1′) and (†Ł16),

$$\big((x \wedge y) \to x\big) \;\to\; \big(((x \wedge y) \,\&\, (y \to z)) \to (x \,\&\, (y \to z))\big) = \overline{1},$$

hence, by ($\star$W1),

$$((x \wedge y) \,\&\, (y \to z)) \to (x \,\&\, (y \to z)) = \overline{1}. \tag{$*$}$$

Then note that, by (†W19),

$$x \to ((y \to z) \to x) = \overline{1},$$

so, by (†Ł12),

$$(x \,\&\, (y \to z)) \to x = \overline{1}.$$

So, by (†W14) on ($*$),

$$\underbrace{((x \wedge y) \,\&\, (y \to z))}_{d} \to x.$$

By (†Ł11),

$$(d \to z) \to ((d \to x) \to (d \to (z \wedge x))),$$

hence, by ($\star$W1) twice,

$$\underbrace{\big((x \wedge y) \,\&\, (y \to z)\big)}_{d} \to \big(x \wedge z\big) \;=\; \overline{1}.$$

So, by (†MV1′),

$$\Big(\underbrace{(y_n \to z_n)}_{a_n} \;\&\; \underbrace{(x_n \wedge y_n)}_{b_n}\Big) \to \underbrace{\big(x_n \wedge z_n\big)}_{c_n} \;=\; \overline{1}. \tag{$*$}$$

Also suppose

$$\begin{aligned}
&\Big(\;\Big(\underbrace{(y_1 \to z_1) \wedge (y_2 \to z_2) \wedge \ldots \wedge (y_{n-1} \to z_{n-1})}_{a'_{n-1}}\Big) \\
&\;\&\; \Big(\underbrace{(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \ldots \vee (x_{n-1} \wedge y_{n-1})}_{b'_{n-1}}\Big)\Big) \\
&\to \Big(\; \underbrace{(x_1 \wedge z_1) \vee (x_2 \wedge z_2) \vee \ldots \vee (x_{n-1} \wedge z_{n-1})}_{c'_{n-1}} \;\Big) \;=\; \overline{1}.
\end{aligned} \tag{$**$}$$

hence, by (†Ł14),

$$\begin{aligned}
(a'_{n-1} \,\&\, \neg c'_{n-1}) \to \neg b'_{n-1} &= \overline{1},\\
(a_n \,\&\, \neg c_n) \to \neg b_n &= \overline{1}.
\end{aligned}$$

By (1),

$$\Big(\big((a'_{n-1} \,\&\, \neg c'_{n-1}) \to \neg b'_{n-1}\big) \;\&\; \big((a_n \,\&\, \neg c_n) \to \neg b_n\big) \Big) \to \Big(\big((a'_{n-1} \wedge a_n) \,\&\, (\neg c'_{n-1} \wedge \neg c_n)\big) \;\to\; (\neg b'_{n-1} \wedge \neg b_n) \Big) \;=\; \overline{1},$$

hence, by (†MV5′) and (⋆W1),

$$\big((a'_{n-1} \wedge a_n) \,\&\, (\neg c'_{n-1} \wedge \neg c_n)\big) \;\to\; (\neg b'_{n-1} \wedge \neg b_n) \;=\; \overline{1},$$

hence, by (†Ł1),

$$\big((a'_{n-1} \wedge a_n) \,\&\, \neg(c'_{n-1} \vee c_n)\big) \;\to\; \neg(b'_{n-1} \vee b_n) \;=\; \overline{1},$$

hence, by (†Ł13),

$$\begin{aligned} \Big(\;& \Big(\underbrace{\underbrace{(y_1 \to z_1) \wedge (y_2 \to z_2) \wedge \ldots \wedge (y_{n-1} \to z_{n-1})}_{a'_{n-1}} \wedge \underbrace{(y_n \to z_n)}_{a_n}}_{a'_n} \Big) \\ \& \;& \Big(\underbrace{\underbrace{(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \ldots \vee (x_{n-1} \wedge y_{n-1})}_{b'_{n-1}} \vee \underbrace{(x_n \wedge y_n)}_{b_n}}_{b'_n} \Big) \Big) \\ \to \;& \Big(\underbrace{\underbrace{(x_1 \wedge z_1) \vee (x_2 \wedge z_2) \vee \ldots \vee (x_{n-1} \wedge z_{n-1})}_{c'_{n-1}} \vee \underbrace{(x_n \wedge z_n)}_{c_n}}_{c'_n} \Big) \;=\; \overline{1}. \end{aligned}$$

The rest of this proof of (⋆SYL·2·A) proceeds in analogy to the above proof of (⋆SYL·1·A). We have (⋆SYL·2·A) for $N = 1$, and, equivalently, $(*)$ for any n and that $(**)$ for $n = 2$.

For the induction, note that in the base case $N = 2$, both $(*)$ and $(**)$ are fulfilled, and, by the above argument, so is (⋆SYL·2·A). This also shows $(**)$ for the case $n = 3$. Then, by the same argument we also have (⋆SYL·2·A) for $N = 3$, etc.

□

C. Implementation Notes

Available Now! PyPES/McPIET

I generated the various ideas discussed in this dissertation in the course of a software prototyping effort, which has taken various approaches to a proof-of-concept level. The next section will describe briefly how these ideas evolved.

The implementation of the various procedures and algorithms described herein is called PyPES/McPIET, and it represents the final version of the software prototype. It is now freely available for use, modification, and redistribution[1]. PyPES relies on various software components distributed as part of the DELPH-IN family of tools[2], such as the ERG grammar, the Linguistic Knowledge Builder (LKB), and the `[incr tsdb()]` grammar profiling and treebanking tool. PyPES is a collection of Python Procedures for Experimentation with Semantics and provides ProtoForm handling functionality, as well as various interface and framework components. One particular component of PyPES is McPIET, the Monte Carlo Pseudo Inference Engine for Text.

The software comprises over 19k lines of code, written in Python3, 5k of which provide unit testing functionality. The library was developed over 5 years and underwent many revisions, some of the functionality undergoing two or three complete reimplementations. Documentation is available, as well as a website and a mailing list.

PyPES input/output and database storage:

- input of plaintext MRS or XML-MRS structures as produced by the LKB's MRS code either from the LKB or PET parser;
- through the use of `[incr tsdb()]` treebanks, hand-selected grammatical analyses can be entered;
- as part of the MRS input process, data are checked against the grammar's SEM-I; currently only the ERG grammar is supported;
- input/output of ProtoForms via a text format either directly or via the PyPES database.

[1] http://www.semantilog.org/pypes.html
[2] http://wiki.delph-in.net/moin

PyPES ProtoForm processing:

- scoping and scope enumeration based on the Koller-Thater algorithm;
- Koller-Thater-style redundancy elimination;
- the improved implementation allows not only the usual exhaustive scoping and scope enumeration, but also selective scoping and unpacking;
- conversion of minimally recursive ProtoForms to maximally recursive ProtoForms;
- conversion of minimally recursive ProtoForms to SNFs;
- ProtoForm subsumption and isomorphicity checking (i.e. reordering of subforms, renaming of variables etc.)
- a copula resolution mechanism which resolves certain occurences of the copula not handled by the grammar;
- approximation of ERG-derived word-prime semantics to operators in FOPC; if used together with the scoping mechanism, this can be used to translate ERG semantics to FOPC formulae that could be entered into FOPC theorem provers or model builders.

PyPES inference testing framework:

- conversion of inference datasets in the RTE/AVE format or Bill MacCartney's format into an internal format; plaintext sentences will be automatically cross-referenced with treebank items;
- conversion of RTE system submission files into the internal format;
- analyses for the 640 sentences in the FraCaS testsuite have been hand-selected for the ERG in `[incr tsdb()]`;
- various kinds of "cosmetic" preprocessing such as fixing punctuation in RTE datasets;
- bag-of-words baseline inference engine;
- inference datasets are hierarchically structured, supporting distinction between tasks in RTE and between sections in FraCaS;
- statistical evaluation for comparisons of results obtained by an inference engine; this includes all evaluation measures and visualization techniques discussed in chapter 2.

McPIET:

- McPIET is an implementation of the logic and the Monte Carlo inference algorithm described in this chapter.

McPIET@RTE4 vs. The New McPIET2

McPIET participated in the fourth RTE recognizing textual entailment challenge (RTE4) at the text analysis conference (TAC) in September 2008 (teamID: cambridge) and was submitted with a system report concentrating largely on some of the underlying theoretical ideas (Bergmair 2008), which were also presented at the workshop and in this dissertation.

Many of the ideas described in this dissertation were developed so as to address the shortcomings of this earlier version (McPIET@RTE4). The version of McPIET which is available for download now (McPIET2) was completely reimplemented from scratch so as to implement these new ideas.

The performance of McPIET@RTE4 was not significantly different in the evaluation from random, which could be attributed to a number of reasons. These include, obviously, the lack of lexical knowledge, and the lack of knowledge about time and space, and the lack of knowledge about names of people, places, and organizations. Also, McPIET does not address discourse phenomena, leading to huge gaps in reasoning patterns which involve semantic coreferences induced by anaphora etc.

All of these shortcomings are more or less to do with the scope and limited resources available for this particular PhD project. But there remains one piece of criticism against McPIET@RTE4 which seems more fundamental, and this is the particular setup of the linguistic processing infrastructure in McPIET@RTE4: It forced the parse selection mechanism to venture a wild guess as to a top-1 parse for each sentence. The inference mechanism then worked with this parse without further qualification of confidence etc. This means that, although McPIET@RTE4 does offer robustness when it comes to inference in the presence of uncertainty, it does not practice what we preach in section 5.1.2 when it comes to linguistic processing: By forcing fine-grained distinctions in the semantic representation, it trades off natural robustness effects for a promise of semantic informativity on which the grammar cannot be expected to deliver.

My current thinking on the subject is as follows: In order to set up the system robustly so as to deal with RTE-style data, what would be required is an inference mechanism which works directly off packed semantic representations for parse forests, rather than the semantic representation for a top-1 parse. This might work within our framework by running a model checker to determine the maximum and minimum truth value of a sentence, given the ambiguity represented in the packed representation. Such an algorithm could use dynamic programming to implement a lattice scoring mechanism.

Another approach could be to delete variables in SNFs which represent relationships about which there is ambiguity in the lattice. For example, in the sentence 'We saw a man with a telescope', one might simply delete the arg1 of the |with|-predicate entirely. Most

SPs would remain unaffected by this (We saw a man, a telescope was the object of a with-relationship). This way, SNFs give us the option of putting less information into a representation, rather than risking putting incorrect information into the representation. Our robustness heuristics could take over from there to fill the informational gap.

These ideas are, of course, nothing more than speculation at this point, as we have not implemented and tested them. It is therefore hard to anticipate the problems that would be involved. But it should be noted that this idea of using SNFs for packed inference was one of the major motivators for me to develop the theory surrounding SNFs. My hope is that future work will be able to develop these ideas more fully and eventually produce a version of McPIET2 which can be applied back to RTE-style data more successfully than its predecessor. In the meantime, my work on FraCaS-style data (appendix D) serves as an initial proof of concept.

D. FraCaS Testsuite

D.1. Generalized Quantifiers

D.1.1. Conservativity

An Italian became the world's greatest tenor.
∴ Was there an Italian who became the world's greatest tenor? (F.1)

Every Italian man wants to be a great tenor.
Some Italian men are great tenors.
∴ Are there Italian men who want to be a great tenor? (F.2)

All Italian men want to be a great tenor.
Some Italian men are great tenors.
∴ Are there Italian men who want to be a great tenor? (F.3)

Each Italian tenor wants to be great.
Some Italian tenors are great.
∴ Are there Italian tenors who want to be great? (F.4)

The really ambitious tenors are Italian.
∴ Are there really ambitious tenors who are Italian? 12 (*F.5)

No really great tenors are modest.
∴ ¬Are there really great tenors who are modest? (F.6)

Some great tenors are Swedish.
∴ Are there great tenors who are Swedish? (F.7)

Many great tenors are German.
∴ Are there great tenors who are German? 2 (F.8)

Several great tenors are British.
∴ Are there great tenors who are British? (F.9)

Most great tenors are Italian.
∴ Are there great tenors who are Italian? 1 (*F.10)

A few great tenors sing popular music.
Some great tenors like popular music. (F.11)
∴ Are there great tenors who sing popular music?

Few great tenors are poor. 1 (*F.12)
∴̸ Are there great tenors who are poor?

Both leading tenors are excellent.
Leading tenors who are excellent are indispensable. 7 (*F.13)
∴ Are both leading tenors indispensable?

Neither leading tenor comes cheap.
One of the leading tenors is Pavarotti. 10 (*F.14)
∴ ¬Is Pavarotti a leading tenor who comes cheap?

At least three tenors will take part in the concert. 3 (F.15)
∴ Are there tenors who will take part in the concert?

At most two tenors will contribute their fees to charity. 3 (F.16)
∴̸ Are there tenors who will contribute their fees to charity?

D.1.2. Upwards Monotonicity on 2nd Argument

An Irishman won the Nobel prize for literature. (F.17)
∴ Did an Irishman win a Nobel prize?

Every European has the right to live in Europe.
Every European is a person. (F.18)
Every person who has the right to live in Europe can travel freely within Europe.
∴ Can every European travel freely within Europe?

All Europeans have the right to live in Europe.
Every European is a person. (F.19)
Every person who has the right to live in Europe can travel freely within Europe.
∴ Can all Europeans travel freely within Europe?

Each European has the right to live in Europe.
Every European is a person. (F.20)
Every person who has the right to live in Europe can travel freely within Europe.
∴ Can each European travel freely within Europe?

The residents of member states have the right to live in Europe.
All residents of member states are individuals. (F.21)
Every individual who has the right to live in Europe can travel freely within Europe.
∴ Can the residents of member states travel freely within Europe?

No delegate finished the report on time.
·/· Did no delegate finish the report? (F.22)

Some delegates finished the survey on time.
∴ Did some delegates finish the survey? (F.23)

Many delegates obtained interesting results from the survey.
∴ Did many delegates obtain results from the survey? 2 (F.24)

Several delegates got the results published in major national newspapers.
∴ Did several delegates get the results published? (F.25)

Most Europeans are resident in Europe.
All Europeans are people.
All people who are resident in Europe can travel freely within Europe.
∴ Can most Europeans travel freely within Europe? 1 (F.26)

A few committee members are from Sweden.
All committee members are people.
All people who are from Sweden are from Scandinavia.
∴ Are at least a few committee members from Scandinavia? (F.27)

Few committee members are from Portugal.
All committee members are people.
All people who are from Portugal are from southern Europe.
·/· Are there few committee members from southern Europe? 1 (*F.28)

Both commissioners used to be leading businessmen.
∴ Did both commissioners used to be businessmen? (F.29)

Neither commissioner spends a lot of time at home.
·/· Does neither commissioner spend time at home? 5 (*F.30)

At least three commissioners spend a lot of time at home.
∴ Do at least three commissioners spend time at home? 5, 3 (F.31)

At most ten commissioners spend a lot of time at home.
·/· Do at most ten commissioners spend time at home? 5, 3 (*F.32)

D.1.3. Downwards Monotonicity on 2nd Argument

An Irishman won a Nobel prize.
·/· Did an Irishman win the Nobel prize for literature? (F.33)

Every European can travel freely within Europe.
Every European is a person.
Every person who has the right to live in Europe can travel freely within Europe.

./. Does every European have the right to live in Europe? (F.34)

All Europeans can travel freely within Europe.
Every European is a person.
Every person who has the right to live in Europe can travel freely within Europe.

./. Do all Europeans have the right to live in Europe? (F.35)

Each European can travel freely within Europe.
Every European is a person.
Every person who has the right to live in Europe can travel freely within Europe.

./. Does each European have the right to live in Europe? (F.36)

The residents of member states can travel freely within Europe.
All residents of member states are individuals.
Every individual who has the right to live anywhere in Europe
can travel freely within Europe.

./. Do the residents of member states have the right to live anywhere in Europe? (F.37)

No delegate finished the report.

∴ ¬Did any delegate finish the report on time? (F.38)

Some delegates finished the survey.

./. Did some delegates finish the survey on time? (F.39)

Many delegates obtained results from the survey.

./. Did many delegates obtain interesting results from the survey? 2 (F.40)

Several delegates got the results published.

./. Did several delegates get the results published in major national newspapers? (F.41)

Most Europeans can travel freely within Europe.
All Europeans are people.
All people who are resident in Europe can travel freely within Europe.

./. Are most Europeans resident in Europe? 1 (F.42)

A few committee members are from Scandinavia.
All committee members are people.
All people who are from Sweden are from Scandinavia.

./. Are at least a few committee members from Sweden? (F.43)

Few committee members are from southern Europe.
All committee members are people.
All people who are from Portugal are from southern Europe.

∴ Are there few committee members from Portugal? 1 (*F.44)

Both commissioners used to be businessmen.
$\not\therefore$ Did both commissioners used to be leading businessmen? (F.45)

Neither commissioner spends time at home.
$\therefore$ ¬Does either commissioner spend a lot of time at home? 5 (F.46)

At least three commissioners spend time at home.
$\not\therefore$ Do at least three commissioners spend a lot of time at home? 5, 3 (*F.47)

At most ten commissioners spend time at home.
$\therefore$ Do at most ten commissioners spend a lot of time at home? 5, 3 (F.48)

D.1.4. Upwards Monotonicity on 1st Argument

A Swede won a Nobel prize.
Every Swede is a Scandinavian. (F.49)
$\therefore$ Did a Scandinavian win a Nobel prize?

Every Canadian resident can travel freely within Europe.
Every Canadian resident is a resident of the North American continent. (F.50)
$\not\therefore$ Can every resident of the North American continent travel freely within Europe?

All Canadian residents can travel freely within Europe.
Every Canadian resident is a resident of the North American continent. (F.51)
$\not\therefore$ Can all residents of the North American continent travel freely within Europe?

Each Canadian resident can travel freely within Europe.
Every Canadian resident is a resident of the North American continent. (F.52)
$\not\therefore$ Can each resident of the North American continent travel freely within Europe?

The residents of major western countries can travel freely within Europe.
All residents of major western countries are residents of western countries. (F.53)
$\not\therefore$ Do the residents of western countries have the right to live in Europe?

No Scandinavian delegate finished the report on time.
$\not\therefore$ Did any delegate finish the report on time? (F.54)

Some Irish delegates finished the survey on time.
$\therefore$ Did any delegates finish the survey on time? (F.55)

Many British delegates obtained interesting results from the survey.
$\not\therefore$ Did many delegates obtain interesting results from the survey? 2 (*F.56)

Several Portuguese delegates got the results published in major national newspapers.
$\therefore$ Did several delegates get the results published in major national newspapers? (F.57)

Most Europeans who are resident in Europe can travel freely within Europe.

$\not\therefore$ Can most Europeans travel freely within Europe? 1 (F.58)

A few female committee members are from Scandinavia.

$\therefore$ Are at least a few committee members from Scandinavia? 3 (F.59)

Few female committee members are from southern Europe.

$\not\therefore$ Are few committee members from southern Europe? 1 (*F.60)

Both female commissioners used to be in business.

$\not\therefore$ Did both commissioners used to be in business? (F.61)

Neither female commissioner spends a lot of time at home.

$\not\therefore$ Does either commissioner spend a lot of time at home? 5 (F.62)

At least three female commissioners spend time at home.

$\therefore$ Do at least three commissioners spend time at home? 3 (F.63)

At most ten female commissioners spend time at home.

$\not\therefore$ Do at most ten commissioners spend time at home? 3 (*F.64)

D.1.5. Downwards Monotonicity on 1st Argument

A Scandinavian won a Nobel prize.

Every Swede is a Scandinavian. (F.65)

$\not\therefore$ Did a Swede win a Nobel prize?

Every resident of the North American continent can travel freely within Europe.

Every Canadian resident is a resident of the North American continent. (F.66)

$\therefore$ Can every Canadian resident travel freely within Europe?

All residents of the North American continent can travel freely within Europe.

Every Canadian resident is a resident of the North American continent. (F.67)

$\therefore$ Can all Canadian residents travel freely within Europe?

Each resident of the North American continent can travel freely within Europe.

Every Canadian resident is a resident of the North American continent. (F.68)

$\therefore$ Can each Canadian resident travel freely within Europe?

The residents of western countries can travel freely within Europe.

All residents of major western countries are residents of western countries. 11 (*F.69)

$\therefore$ Do the residents of major western countries have the right to live in Europe?

No delegate finished the report on time.

$\therefore$ ¬Did any Scandinavian delegate finish the report on time? (F.70)

Some delegates finished the survey on time.
$\not\therefore$ Did any Irish delegates finish the survey on time? (F.71)

Many delegates obtained interesting results from the survey.
$\not\therefore$ Did many British delegates obtain interesting results from the survey? 2 (F.72)

Several delegates got the results published in major national newspapers.
$\not\therefore$ Did several Portuguese delegates get the results published in major national newspapers? (F.73)

Most Europeans can travel freely within Europe.
$\not\therefore$ Can most Europeans who are resident outside Europe travel freely within Europe? 1 (*F.74)

A few committee members are from Scandinavia.
$\not\therefore$ Are at least a few female committee members from Scandinavia? 3 (F.75)

Few committee members are from southern Europe.
$\therefore$ Are few female committee members from southern Europe? 1 (*F.76)

Both commissioners used to be in business.
$\therefore$ Did both female commissioners used to be in business? (F.77)

Neither commissioner spends a lot of time at home.
$\therefore$ ¬Does either female commissioner spend a lot of time at home? 5 (F.78)

At least three commissioners spend time at home.
$\not\therefore$ Do at least three male commissioners spend time at home? 3 (F.79)

At most ten commissioners spend time at home.
$\therefore$ Do at most ten female commissioners spend time at home? 3 (*F.80)

D.2. Plurals

D.2.1. Conjoined Noun Phrases

Smith, Jones and Anderson signed the contract.
$\therefore$ Did Jones sign the contract? (F.81)

Smith, Jones and several lawyers signed the contract.
$\therefore$ Did Jones sign the contract? (F.82)

Either Smith, Jones or Anderson signed the contract.
$\not\therefore$ Did Jones sign the contract? (F.83)

Either Smith, Jones or Anderson signed the contract.
∴ If Smith and Anderson did not sign the contract, did Jones sign the contract? 13 (*F.84)

Exactly two lawyers and three accountants signed the contract.
∴ ¬Did six lawyers sign the contract? 6 (*F.85)

Exactly two lawyers and three accountants signed the contract.
∴ ¬Did six accountants sign the contract? 6 (*F.86)

Every representative and client was at the meeting.
∴ Was every representative at the meeting? 8 (*F.87)

Every representative and client was at the meeting.
∴̸ Was every representative at the meeting? 8 (F.88)

Every representative or client was at the meeting.
∴ Were every representative and every client at the meeting? 8 (*F.89)

D.2.2. Definite Plurals

The chairman read out the items on the agenda.
∴ Did the chairman read out every item on the agenda? (F.90)

The people who were at the meeting voted for a new chairman.
∴̸ Did everyone at the meeting vote for a new chairman? 12 (*F.91)

All the people who were at the meeting voted for a new chairman.
∴ Did everyone at the meeting vote for a new chairman? 9 (*F.92)

The people who were at the meeting all voted for a new chairman.
∴ Did everyone at the meeting vote for a new chairman? (F.93)

The inhabitants of Cambridge voted for a Labour MP.
∴̸ Did every inhabitant of Cambridge vote for a Labour MP? 12 (*F.94)

The Ancient Greeks were noted philosophers.
∴̸ Was every Ancient Greek a noted philosopher? 12 (*F.95)

The Ancient Greeks were all noted philosophers.
∴ Was every Ancient Greek a noted philosopher? (F.96)

D.2.3. Bare Plurals

This section of the FraCaS testsuite is outside the scope of our work. (see note 7)

D.2.4. Dependent Plurals

All APCOM managers have company cars.
Jones is an APCOM manager. (F.103)
∴ Does Jones have a company car?

All APCOM managers have company cars.
Jones is an APCOM manager. (F.104)
$\not\therefore$ Does Jones have more than one company car?

D.2.5. Negated Plurals

Just one accountant attended the meeting. (F.105)
∴ ¬Did no accountants attend the meeting?

Just one accountant attended the meeting. (F.106)
∴ ¬Did no accountant attend the meeting?

Just one accountant attended the meeting. (F.107)
∴ Did any accountants attend the meeting?

Just one accountant attended the meeting. (F.108)
∴ Did any accountant attend the meeting?

Just one accountant attended the meeting. 4 (*F.109)
∴ ¬Did some accountants attend the meeting?

Just one accountant attended the meeting. (F.110)
∴ Did some accountant attend the meeting?

D.2.6. Collective and Distributive Plurals

This section of the FraCaS testsuite is outside the scope of our work.

D.3. Notes

reservations, root causes for possibly incorrect decisions:

1. The quantifiers most and few have not been implemented. (10 examples)
2. The quantifier many has not been implemented. (5 examples)
3. The quantifiers "at least X" and "at most X" have not been implemented. (12 examples)
4. Example ∗F.109 uses the quantifier some with a plural `N'`-phrase, the intended meaning being that of 'at least two'. This particular use of some has not been implemented. (1 example)
5. The intended semantics of the phrase "a lot of X" in the testsuite is such that we should have "a lot of X" → "X" but not the converse "X" → "a lot of X". This special semantics of 'a lot of' has not been implemented. (8 examples)
6. A theory of arithmetic has not been implemented. (2 examples)
7. The ERG represents bare plurals by injecting the underspecified definite quantifier UDEF_Q. Even when one restricts attention to cases where it stands with a plural variable (i.e. stands with a plural `N'`), it is still nontrivial to distinguish the bare plurals that can give rise to genericity from situations where a grammar-injected UDEF_Q must be interpreted as an existential quantifier. Thus, bare plurals have not been implemented. Apart from the FraCaS section on bare plurals, this leads to an error only in example ∗F.13. (1 section + 1 additional example)
8. Example F.88 is identical to ∗F.87, despite having a different annotation. Here the testsuite appeals to a different reading for example F.88. Unfortunately, this distributive reading for the quantifiers can only be extracted from the ERG semantics by nontrivial rewrite operations which induce ambiguity. The phenomenon was not implemented, hence example F.88 is not covered either. (3 examples)
9. The particular use of 'all the' in example ∗F.92 ('All the people ...') is not supported by the ERG. It always injects a PART_OF predicate as in 'All the world ...'. (1 example)
10. In example ∗F.14, the ERG returns a fragmented semantic representation. (1 example)
11. Example ∗F.69 is clearly a mistake in the testsuite. (1 example)
12. Plural the is not supported. (4 examples)
13. In example ∗F.84 there are two possible readings for the connective AND_C. The reading which licenses the answer given by the inference machinery, though contrary to the gold standard answer, is as follows: Let's assume that there exists a contract, which was signed by either Smith, Jones, or Anderson, in the sense that it carries at least one signature, which is either that of Smith, that of Jones, or that of

Anderson. Let's also assume that this same contract was not signed by both Smith and Jones, i.e. that it does not carry the two signatures of Smith and Jones; or alternatively, we could assume that there does not exist any such contract. This leaves open both the possibility of that contract (or any contract) to carry or not carry the signature of Jones. (1 example)

statistics:

- In section 1, we can regard all 5 subsections as falling within the scope of our work. These 5 subsections comprise 80 examples altogether.
- In section 2, 2 out of 6 subsections were out of scope. These 6 subsections comprise 30 examples. The 2 subsections which were out of scope account for 6 out of these 30 examples.
- Example *F.69 in section 1 must be discarded due to the gold standard decision being incorrect.
- Among the 79 examples in section 1, our system made incorrect decisions for 16 of them.
- Among the 79 examples in section 1, we have reservations about 34 of them.
- Among the 24 examples in section 2, our system made incorrect decisions for 10 of them.
- Among the 24 examples in section 2, we have reservations about 11 of them.
- All incorrect decisions made by the system can be attributed to a root cause corresponding to one of our reservations.
- Among all the 103 examples in the testsuite, 50 were entailing, 42 were unknowable, and 11 were contradictions. (48.5%/40.8%/10.7%)
- Among the 58 examples about which we had no reservations, 32 were entailing, 21 were unknowable, and 5 were contradictions. (55.2%/36.2%/8.6%)
- Out of the 103 examples, 78 were correct, which yields an accuracy of 75.7%. (constant choice: 48.5%, random choice: 41.31%)
- Out of the 58 examples, all were correct, which yields an accuracy of 100%. (constant choice: 55.2%, random choice: 44.27%)

Bibliography

Bach, K. (1997), 'Do belief reports report beliefs?', *Pacific Philosophical Quarterly* 78, 215–241.

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B. & Szpektor, I. (2006), The second PASCAL recognising textual entailment challenge., *in* 'Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment (RTE-2)'.

Barwise, J. & Cooper, R. (1981), Generalized quantifiers and natural language, *in* 'Linguistics and Philosophy', Vol. 4, pp. 159–219.

Beigman-Klebanov, B. & Beigman, E. (2009), 'From annotator agreement to noise models', *Computational Linguistics* **35**(4), 495–503.

Bensley, J. & Hickl, A. (2008), Application of LCC's GROUNDHOG system for RTE-4, *in* 'Workshop Notebook of the Text Analysis Conference (TAC)'.

Bentivogli, L., Dagan, I., Dang, H. T., Giampiccolo, D. & Magnini, B. (2009), The fifth PASCAL recognizing textual entailment challenge, *in* 'Workshop Notebook of the Second Text Analysis Conference (TAC '09)'.

Bergmair, R. (2006*a*), Closed domain question answering using fuzzy semantics, Master's thesis, University of Cambridge Computer Laboratory.

Bergmair, R. (2006*b*), Syntax-driven analysis of context-free languages with respect to fuzzy relational semantics, Technical Report UCAM-CL-TR-663, University of Cambridge, Computer Laboratory.
URL: *http:// www.cl.cam.ac.uk/ TechReports/ UCAM-CL-TR-663.pdf*

Bergmair, R. (2008), Monte carlo semantics: McPIET at RTE4, *in* 'Proceedings of the First Text Analysis Conference'.

Bergmair, R. (2009), A proposal on evaluation measures for RTE, *in* 'Proceedings of the 2009 Workshop on Applied Textual Inference (TextInfer)', Association for Computational Linguistics, Suntec, Singapore, pp. 10–7.

Bergmair, R. & Bodenhofer, U. (2006), Syntax-driven analysis of context-free languages with respect to fuzzy relational semantics, *in* 'Proceedings of the 2006 IEEE International Conference on Fuzzy Systems', pp. 2075–82.

Blackburn, P. & Bos, J. (2005), *Representation and Inference for Natural Language: A First Course in Computational Semantics*, CSLI.

Borkowski, L., ed. (1970), *Selected works of Jan Łukasiewicz*, North-Holland.

Bos, J. (1996), Predicate logic unplugged, *in* P. Dekker & M. Stokhof, eds, 'Proceedings of the Tenth Amsterdam Colloquium', Amsterdam, Netherlands, pp. 133–43.

Bos, J. (2005), Towards wide-coverage semantic interpretation, *in* 'Proceedings of the Sixth International Workshop on Computational Semantics (IWCS-6)', pp. 42–53.

Bos, J. & Markert, K. (2005*a*), Combining shallow and deep NLP methods for recognizing textual entailment, *in* I. Dagan, O. Glickman & B. Magnini, eds, 'Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment (RTE-1)'.

Bos, J. & Markert, K. (2005*b*), Recognising textual entailment with logical inference, *in* 'Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP 2005)', pp. 628–635.

Bos, J. & Markert, K. (2006*a*), Recognising textual entailment with robust logical inference, *in* J. Quinonero-Candela, I. Dagan, B. Magnini & F. d'Alché Buc, eds, 'Machine Learning Challenges, MLCW 2005', Vol. 3944 of *LNAI*, pp. 404–426.

Bos, J. & Markert, K. (2006*b*), When logical inference helps determining textual entailment (and when it doesn't), *in* 'Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment (RTE-2)'.

Carroll, J., Minnen, G. & Briscoe, T. (1999), Corpus annotation for parser evaluation, *in* 'Proceedings of the EACL-99 Post-Conference Workshop on Linguistically Interpreted Corpora', Bergen, Norway, pp. 35–1.

Chambers, N., Cer, D., Grenager, T., Hall, D., Kiddon, C., MacCartney, B., de Marneffe, M.-C., Ramage, D., Yeh, E. & Manning, C. D. (2007), Learning alignments and leveraging natural logic, *in* 'Proceedings of the Workshop on Textual Entailment and paraphrasing (RTE-3)'.

Chang, C. C. (1958*a*), 'Algebraic analysis of many valued logics', *Transactions of the American Mathematical Society* 88(2), pp. 467–490.

Chang, C. C. (1958*b*), 'Proof of an axiom of Łukasiewicz', *Transactions of the American Mathematical Society* **87**(1), pp. 55–56.

Chang, C. C. (1959), 'A new proof of the completeness of the Łukasiewicz axioms', *Transactions of the American Mathematical Society* **93**(1), pp. 74–80.

Cooper, R., Crouch, D., van Eijck, J., Fox, C., van Genabith, J., Jaspars, J., Kamp, H., Milward, D., Pinkal, M., Poesio, M. & Pulman, S. (1996), Using the framework, Technical Report D16, FraCaS project deliverable.

Copestake, A. (2007), Semantic composition with (robust) minimal recursion semantics, *in* 'Proceedings of the Workshop on Deep Linguistic Processing (DeepLP '07)', Association for Computational Linguistics, Morristown, NJ, pp. 73–0.

Copestake, A. (2009), Slacker semantics: why superficiality, dependency and avoidance of commitment can be the right way to go, *in* 'EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 1–9.

Copestake, A., Flickinger, D., Pollard, C. & Sag, I. A. (2005), 'Minimal recursion semantics: An introduction', *Research on Language & Computation* **3**(4), 281–332.

Copestake, A., Lascarides, A. & Flickinger, D. (2001), An algebra for semantic construction in constraint-based grammars, *in* 'Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)', Toulouse, France.

Crabbé, M. (2001), 'The formal theory of syllogisms', *Review of Modern Logic* **9**(1–2), 29–52.

Crouch, R., Karttunen, L. & Zaenen, A. (2006), 'Circumscribing is not excluding: A reply to Manning', manuscript.

Curran, J. R., Clark, S. & Bos, J. (2007), Linguistically motivated large-scale NLP with C&C and Boxer, *in* 'Proceedings of the Demonstrations Session of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)'.

Dagan, I., Glickman, O. & Magnini, B. (2005), The PASCAL recognising textual entailment challenge, *in* I. Dagan, O. Glickman & B. Magnini, eds, 'Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment (RTE)'.

Dalrymple, M., Lamping, J. & Saraswat, V. (1993), LFG semantics via constraints, *in* 'Proceedings of the Sixth Meeting of the European ACL', pp. 97–105.

Davidson, D. (1968), 'On saying that', *Synthese* **19**, 130–146.

De Finetti, B. (1974), *Theory of probability: a critical introductory treatment*, Wiley, London. Translation of Teoria delle probabilita.

Elkan, C. (1994), ‘The paradoxical success of fuzzy logic’, *IEEE Expert: Intelligent Systems and Their Applications* **9**(4), 3–8.

Fellbaum, C. (1998), *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA.

Flickinger, D. (2000), ‘On building a more efficient grammar by exploiting types’, *Natural Language Engineering* **6 (1) (Special Issue on Efficient Processing with HPSG)**, 15–28.

Font, J., Rodriguez, A. J. & Torrens, A. (1984), ‘Wajsberg algebras’, *Stochastica* pp. 5–31.

Frege, G. (1879), Begriffsschrift, *in* I. Angelelli, ed., ‘Begriffsschrift und andere Aufsätze’, Georg Olms.

Frege, G. (1892), Über Sinn und Bedeutung, *in* M. Textor, ed., ‘Funktion, Begriff, Bedeutung’, Vol. 4 of *Sammlung Philosophie*, Vandenhoek & Ruprecht, pp. 23–47.

Fuchss, R., Koller, A., Niehren, J. & Thater, S. (2004), Minimal recursion semantics as dominance constraints: Translation, evaluation, and analysis, *in* ‘Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL-04)’, Barcelona, Spain.

Giampiccolo, D., Dang, H. T., Magnini, B., Dagan, I. & Dolan, B. (2008), The fourth PASCAL recognizing textual entailment challenge, *in* ‘Workshop Notebook of the Text Analysis Conference (TAC)’.

Giampiccolo, D., Magnini, B., Dagan, I. & Dolan, B. (2007), The third PASCAL recognizing textual entailment challenge, *in* ‘Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing (RTE-3)’, Association for Computational Linguistics, Prague, pp. 1–9.
URL: *http://www.aclweb.org/anthology/W/W07/W07-1401*

Gottwald, S. (2001), *A Treatise on Many-Valued Logics*, Research Studies Press Ltd., Baldock, UK.

Hájek, P. (1998), *Metamathematics of Fuzzy Logic*, Kluwer.

Harabagiu, S. & Hickl, A. (2006), Methods for using textual entailment in open-domain question answering, *in* ‘ACL ’06: Proceedings of the 21st International Conference on

Computational Linguistics and the 44th annual meeting of the ACL', Association for Computational Linguistics, Morristown, NJ, USA, pp. 905–912.

Harris, Z. (1982), *A Grammar of English on Mathematical Principles*, John Wiley & Sons.

Harris, Z. (1991), *A Theory of Language and Information*, Claredon Press.

Hickl, A. (2008), Using discourse commitments to recognize textual entailment, *in* 'Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)', Coling 2008 Organizing Committee, Manchester, UK, pp. 337–344.
URL: *http://www.aclweb.org/anthology/C08-1043*

Iftene, A. (2008), UAIC participation at RTE-4, *in* 'Workshop Notebook of the Text Analysis Conference (TAC)'.

Iftene, A. (2009), Textual Entailment, PhD thesis, Al. I. Cuza University, Iasi, Romania.

Kamp, H. & Reyle, U. (1993), *From Discourse to Logic: An Introduction to Modeltheoretic Semantics, Formal Logic and Discourse Representation Theory*, Kluwer Academic Publishers.

Kant, I. (1781), *Kritik der reinen Vernunft*, Vol. 505 of *Philosophische Bibliothek*, Felix Meiner Verlag, Hamburg.

Koller, A., Niehren, J. & Thater, S. (2003), Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints, *in* 'Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL '03)', Budapest, pp. 195–202.

Koller, A. & Thater, S. (2005), Efficient solving and exploration of scope ambiguities, *in* 'Proceedings of the ACL-05 Demo Session'.

Koller, A. & Thater, S. (2006), An improved redundancy elimination algorithm for underspecified representations, *in* 'ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 409–416.

Koller, A., Thater, S. & Pinkal, M. (2009), Scope underspecification with tree descriptions: Theory and practice, *in* M. W. Crocker & J. Siekmann, eds, 'Resource-Adaptive Cognitive Processes', Cognitive Technologies, Springer Berlin Heidelberg, pp. 337–364.

Lev, I. (2007), Packed Computation of Exact Meaning Representations, PhD thesis, Stanford University.

Li, F., Zheng, Z., Tang, Y., Bu, F., Ge, R., Zhu, X., Zhang, X. & Huang, M. (2008), THU QUANTA at TAC 2008 QA and RTE track, *in* 'Workshop Notebook of the Text Analysis Conference (TAC)'.

Łukasiewicz, J. (1951), *Aristotle's Syllogistic from the Standpoint of Modern Formal Logic*, Claredon Press, Oxford.

Łukasiewicz, J. & Tarski, A. (1930), 'Untersuchungen über den aussagenkalkül', *Comptes rendus des séances de la Société des Sciences et des Lettes de Varsovie* **23**, 39–50.

MacCartney, B. (2009), Natural language inference, PhD thesis, Stanford University.

MacCartney, B. & Manning, C. D. (2007), Natural logic for textual inference, *in* 'Proceedings of the Workshop on Textual Entailment and paraphrasing (RTE-3)'.

Manning, C. (2006), 'Local textual inference: It's hard to circumscribe, but you know it when you see it – and NLP needs it'.
URL: *http://nlp.stanford.edu/manning/papers/LocalTextualInference.pdf*

Màrquez, L., Carreras, X., Litkowski, K. C. & Stevenson, S. (2008), 'Semantic role labeling: An introduction to the special issue', *Computational Linguistics* **34**(2), 145–159.

McAllester, D. A. & Givan, R. (1992), 'Natural language syntax and first-order inference', *Artificial Intelligence* **56**(1), 1–20.

McKay, T. & Nelson, M. (2005), Propositional attitude reports, *in* E. N. Zalta, ed., 'The Stanford Encyclopedia of Philosophy', spring 2010 edn.
URL: *http://plato.stanford.edu/archives/spr2010/entries/prop-attitude-reports/*

Meredith, C. A. (1958), 'The dependence of an axiom of Łukasiewicz', *Transactions of the American Mathematical Society* **87**(1), p. 54.

Metcalfe, G., Olivetti, N. & Gabbay, D. (2008), *Proof Theory for Fuzzy Logics*, Vol. 36 of *Applied Logic Series*, Springer.

Montague, R. (1970*a*), 'English as a formal language', *Linguaggi nella e nella Tecnica* pp. 189–24.

Montague, R. (1970*b*), 'Universal grammar', *Theoria* **36**, 373–98.

Montague, R. (1973), The proper treatment of quantification in ordinary english, *in* J. Hintikka, J. Moravcsik & P. Suppes, eds, 'Approaches to Natural Language', pp. 221–42.

Morgan, C. G. & Pelletier, F. J. (2004), 'Some notes concerning fuzzy logics', *Linguistics and Philosophy* **1**(1), 79–97.

Moss, L. S. (2007*a*), 'Completeness theorems for syllogistic fragments'.
URL: *http://www.indiana.edu/ iulg/moss/uwe.pdf*

Moss, L. S. (2007*b*), 'Syllogistic logic with complements'.
URL: *http://www.indiana.edu/ iulg/moss/comp.pdf*

Niehren, J. & Thater, S. (2003), Bridging the gap between underspecification formalisms: Minimal recursion semantics as dominance constraints, *in* 'Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL-03)', Sapporo, Japan, pp. 195–202.

Padó, S., de Marneffe, M.-C., MacCartney, B., Rafferty, A. N., Yeh, E. & Manning, C. D. (2008), Deciding entailment and contradiction with stochastic and edit distance-based alignment, *in* 'Proceedings of the First Text Analysis Conference'.

Pavelka, J. (1979), 'On fuzzy logic', *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **25**, pp. 45–52, 119–134, 447–464.

Peñas, A., Rodrigo, Á., Sama, V. & Verdejo, F. (2007), Overview of the answer validation exercise 2006, *in* C. Peters, P. Clough, F. C. Gey, J. Karlgren, B. Magnini, D. W. Oard, M. de Rijke & M. Stempfhuber, eds, 'Evaluation of Multilingual and Multi-modal Information Retrieval: 7th Workshop of the Cross-Language Evaluation Forum (CLEF 2006)', Vol. 4730 of *Lecture Notes in Computer Science*, Springer, Alicante, Spain, pp. 257–264.

Peñas, A., Rodrigo, Á. & Verdejo, F. (2008), Overview of the answer validation exercise 2007, *in* C. Peters, V. Jijkoun, T. Mandl, H. Müller, D. W. Oard, A. Peñas, V. Petras & D. Santos, eds, 'Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum (CLEF 2007)', Vol. 5152 of *Lecture Notes in Computer Science*, Springer, Budapest, Hungary, pp. 237–248.

Pinkal, M. (1985), *Logik und Lexicon: Die Semantik des Unbestimmten*, Walter de Gruyta & Co, Berlin.

Pratt-Hartmann, I. (2003), 'A two-variable fragment of english', *Journal of Logic, Language and Information* **12**(1), 13–45.

Pratt-Hartmann, I. (2004), 'Fragments of language', *Journal of Logic, Language and Information* **13**(2), 207–23.

Pratt-Hartmann, I. & Moss, L. S. (2009), 'Logics for the relational syllogistic', *The Review of Symbolic Logic* **2**(4), 647–83.

Pratt-Hartmann, I. & Third, A. (2006), 'More fragments of language', *Notre Dame Journal of Formal Logic* **47**(2), 151–77.

Rodrigo, Á., Peñas, A. & Verdejo, F. (2009), Overview of the answer validation exercise 2008, *in* C. Peters, T. Deselaers, J. N. F. Gonzalo, G. J. F. Jones, M. Kurimo, T. Mandl, A. Peñas & V. Petras, eds, 'Evaluating Systems for Multilingual and Multimodal Information Access: 9th Workshop of the Cross-Language Evaluation Forum (CLEF 2008)', Vol. 5706 of *Lecture Notes in Computer Science*, Springer, Aarhus, Denmark, pp. 296–313.

Romano, L., Kouylekov, M., Szpektor, I., Dagan, I. & Lavelli, A. (2006), Investigating a generic paraphrase-based approach for relation extraction, *in* 'EACL '06: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics'.

Rose, A. & Rosser, J. B. (1958), 'Fragments of many-valued statement calculi', *Transactions of the American Mathematical Society* 87(1), pp. 1–53.

Siblini, R. & Kosseim, L. (2008), Using ontology alignment for the TAC RTE challenge, *in* 'Workshop Notebook of the Text Analysis Conference (TAC)'.

Siblini, R. & Kosseim, L. (2009), AORTE for recognizing textual entailment, *in* A. F. Gelbukh, ed., 'Computational Linguistics and Intelligent Text Processing, 10th International Conference (CICLing 2009)', Vol. 5449 of *Lecture Notes in Computer Science*, Springer, Mexico City, Mexico, pp. 245–255.

Siddharthan, A. (2004), Syntactic Simplification and Text Cohesion, PhD thesis, University of Cambridge.

Tarski, A. (1930), 'Fundamentale Begriffe der Methodologie der deduktiven Wissenschaften I', *Monatfshefte für Mathematik und Physik* **37**, 361–404.

Tarski, A. (1935), 'Der Wahrheitsbegriff in den formalisierten Sprachen', *Studia Philosophica* **1**, 261–405.

van Benthem, J. (1986), *Essays in logical semantics*, Reidel, Dordrecht.

van Benthem, J. (1991), *Language in Action: Categories, Lambdas and Dynamic Logic*, North Holland, Amsterdam.

van Deemter, K. (1995), The sorites fallacy and the context-dependence of vague predicates, *in* M. Kanazawa, C. Pinon & H. de Swart, eds, 'Quantifiers, Deduction, and Context', CSLI Publications, pp. 59–86.

van Deemter, K. (2010*a*), *Not Exactly: In Praise of Vagueness*, Oxford University Press.

van Deemter, K. (2010*b*), Vagueness facilitates search, *in* 'Proceedings of the 2009 Amsterdam Colloquium'. to appear.

van Eijk, J. (2007), Natural logic for natural language, *in* 'Logic, Language, and Computation; 6th International Tbilisi Symposium on Logic, Language, and Computation', Batumi, Georgia.

Voorhees, E. M. (2008), Contradictions and justifications: Extensions to the textual entailment task, *in* 'Proceedings of ACL-08: HLT', Association for Computational Linguistics, Columbus, Ohio, pp. 63–1.
URL: *http://www.aclweb.org/anthology/P/P08/P08-1008*

Voorhees, E. M. & Harman, D. K., eds (2005), *TREC: Experiment and Evaluation in Information Retrieval*, MIT Press.

Wang, R. & Neumann, G. N. (2008), An accuracy-oriented divide-and-conquer strategy, *in* 'Workshop Notebook of the Text Analysis Conference (TAC)'.

Wang, R. & Zhang, Y. (2009), Recognizing textual relatedness with predicate-argument structures, *in* 'Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Singapore, pp. 784–792.
URL: *http://www.aclweb.org/anthology/D/D09/D09-1082*

Zaenen, A., Karttunen, L. & Crouch, R. (2005), Local textual inference: Can it be defined or circumscribed?, *in* 'Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment', Association for Computational Linguistics, Ann Arbor, Michigan, pp. 31–36.
URL: *http://www.aclweb.org/anthology/W/W05/W05-1206*

Zamansky, A., Francez, N. & Winter, Y. (2006), 'A 'natural logic' inference system using the Lambek calculus', *J. of Logic, Lang. and Inf.* **15**(3), 273–95.