Natural Language Understanding as First-Order Abduction via Stable Models

Petr Homola
Codesign Ltd.
phomola@codesign.cz

Abstract

We present a method for finding abductive proofs in first-order Horn theories by using an answer-set solver. We illustrate our solution with examples from the domain of natural language understanding. Furthermore we describe a novel way of ranking abductive proofs of logical forms of sentences.

1 Introduction

From a pragmatic perspective, natural language understanding (NLU) can be thought of as first-order abduction [Hobbs *et al.*, 1993; Ovchinnikova *et al.*, 2014]. The sentence we want to interpret is what is "observed" and the "best" abductive proof tells us what the sentence actually means. Abductive reasoning is ampliative, that is, what we conclude cannot be proved deductively, but it extends our background knowledge in a coherent way.

Abduction is intractable (cf. [Appelt and Pollack, 1992], "even in the case of propositional Horn clause theories, the problem of computing an explanation for an arbitrary proposition is NP hard"). Propositional abduction is known to be implementable as stable model enumeration [Gelfond and Kahl, 2014] but in the case of NLU the underlying theory is first-order. In early experiments with interpretation as (weighted, i.e., cost-based) abduction, a Prolog-based abductive theorem prover was used [Hobbs *et al.*, 1993]. Later a more efficient algorithm based on integer linear programming was developed [Inoue *et al.*, 2012; Ovchinnikova *et al.*, 2014]. But neither approach allows for seamless integration of abduction with full-fledged deduction. In this contribution we present a method for comparatively efficient first-order abduction based on answer-set solving. We use the solver described in [Gebser *et al.*, 2012].

In Section 2 we give an overview of the problem illustrated with a simple example. Section 3 describes the translation of a first-order abductive problem into an answer-set problem. Section 4 briefly describes possible integrity constraints that significantly constrain the proof search space. Section 5 presents a method for selecting the best proof in the domain of NLU. Section 6 concludes.

2 Preliminaries

Abduction is the reasoning process of finding explanations for observations. In NLU, the "observations" are (logical forms of) sentences and the interpretation of a sentence is the "best" (i.e., most coherent) abductive proof that explains it with respect to a background theory. Formally, I is an interpretation of observations O with respect to a background theory T if[1]

(1) $$T \cup I \vDash O \wedge T \cup I \nvDash \bot$$

that is, $T \cup I$ entails O and is consistent. The sentence *John is an elephant* may mean that there is an actual elephant whose name is John, i.e., I can be the literal meaning of the sentence. But if we know (from context) that John is a person, $T \cup I$ will be inconsistent, hence I cannot be the literal meaning of the sentence and we have to find some other (nonliteral) interpretation that makes sense in the given context.

We use the logical representation proposed by Hobbs (1985), which is a "conjunctivist" scope-free first-order approach to linguistic meaning. Consider the sentence

(2) John sees Mary.

Irrelevant details aside, its logical representation is[2]

(3) $$(\exists e, x, y) see'(e, x, y) \wedge John(x) \wedge Mary(y)$$

that is, there is an eventuality e which is a seeing, John does it and Mary undergoes it. The logical representation of

(4) John doesn't see Mary.

would be

(5) $$(\exists e_1, e_2, x, y) not'(e_1, e_2) \wedge see'(e_2, x, y) \wedge \\ \wedge John(x) \wedge Mary(y)$$

that is, the negation of the eventuality expressed in (3) is asserted. Hobbs (2005) showed that an appropriately rich and

[1] T is a set of first-order formulae and I and O are sets of positive literals.

[2] The relation between the primed and unprimed predicates is given by the following axiom schema (see [Hobbs, 1985]):

$$P(x) \equiv (\exists e) P'(e, x) \wedge Rexist(e)$$

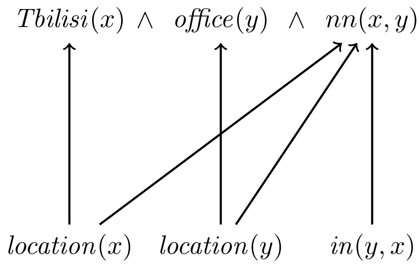

Figure 1: Proof of *the Tbilisi office*

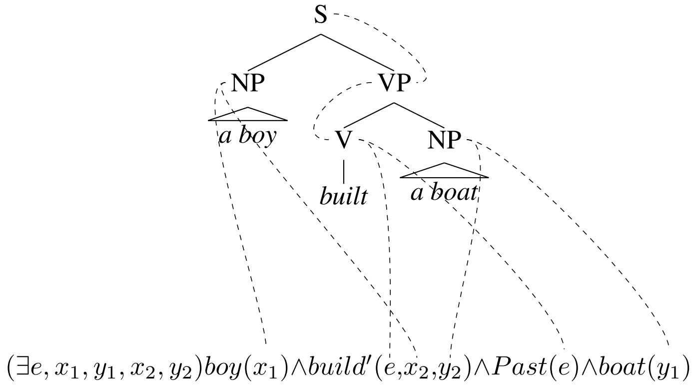

Figure 2: Parse tree and variable bindings for *A boy built a boat*

precise theory of commonsense reasoning can be expressed in this way.

The intended real-world meaning of the logical forms (3) and (5) is their literal meaning. But the logical representation can contain linguistic predications that need be interpreted in order to be given a real-world meaning. The noun phrase (NP)

(6) the Tbilisi office

is parsed as

(7) $(\exists x, y)\, \mathit{Tbilisi}(x) \wedge \mathit{office}(y) \wedge nn(x, y)$

that is, there are two entities and an nn (i.e., N-N compound) relation between them, but the predicate nn only tells us that x and y are adjacent NPs and x precedes y. If we know from context that there is an office in Tbilisi, we want to interpret (7) as $in(\mathit{office}, \mathit{Tbilisi})$. If we have no such information, but our background knowledge contains the facts that Tbilisi is a city and that cities and offices are locations, we can still draw the (defeasible) conclusion that there is an in relation between the entities. Whichever the case, though, we need axiom (8). We use $\rightharpoonup$ to denote (possibly defeasible) implication in abductive rules and $\supset$ to signify implication in "hard" rules, i.e., $P(x) \rightharpoonup Q(x) \equiv P(x) \wedge etc(x) \supset Q(x)$ where etc is a literal specific to the rule.

(8) $$(\forall x, y) location(x) \wedge location(y) \wedge in(x, y) \rightharpoonup \\ \rightharpoonup nn(y, x)$$

that is, the fact that x is located in y can be expressed by a compound nominal at the lexical level. Of course, axiom (8) is only defeasible, for an in relation can also be expressed by other linguistic constructions. Thus, to interpret (7) we have to backchain on axiom (8) and unify both variables. The proof of (7) is depicted in Figure 1. As can be seen, the abduction process is first-order even for simple examples.

3 Translation of deductive and abductive rules

To get the logical form of a sentence we first have to parse it. Since this paper focuses on interpretation, we will only sketch very briefly how the parser is implemented. We have an LFG grammar [Kaplan and Bresnan, 1982; Dalrymple *et al.*, 1995a; Dalrymple, 2001; Bresnan, 2001] whose rules are augmented with annotations that incrementally build up the logical form. Consider the sentence

(9) A boy built a boat.

whose parse tree and variable bindings are given in Figure 2. The lexicon contains a logical expression for every preterminal. For example, the morpholexical entry for *boy* provides $boy(x_1)$, the entry for *built* provides $build'(e, x_2, y_2) \wedge Past(e)$, etc. The syntactic rules unify the variables provided by the nodes they operate on. The rule S $\rightarrow$ NP VP, for example, will unify x_1 and x_2, thus expressing the fact that the *boy* is the agent of e, which is a building event. This way of semantic parsing is easy to implement if one already has a rule-based grammar (lexicon and rules).

Alternatively, we could use so-called "glue semantics" [Dalrymple *et al.*, 1993; 1995b], though this would be less straightforward, since glue semantics produces higher-order formulae, thus we would have to "flatten" them and introduce eventualities into the predicates. Yet another possibility is to implement the complete grammar in the abductive framework itself, as suggested by Hobbs (2003), which has the advantage that parsing and interpretation can be seamlessly integrated and carried out in one step. However if a wide-coverage rule-based grammar is available, it is probably better to use it as a "preprocessor".

Deduction poses no problem to the solver. We can have rules such as[3]

(10) $$\begin{array}{c} elephant_1(x) \supset mammal_1(x) \\ mammal_1(x) \supset animal_1(x) \end{array}$$

and we can use strong negation and disjunction, as in

(11) $$\begin{array}{l} person_1(x) \supset \neg animal_1(x) \\ person_1(x) \supset man_1(x) \vee woman_1(x) \end{array}$$

The main result reported in this paper is a method for converting abduction with observations that contain variables into an answer-set problem. We represent observations and assumptions as follows:

(12)
$$\begin{array}{lcl} \text{observations:} & & \\ elephant(x) & \Rightarrow & pred(elephant, obsrv, var_x) \\ \text{assumptions:} & & \\ elephant_1(x) & \Rightarrow & pred(elephant1, asmpt, var_x) \end{array}$$

that is, variables are encoded as individuals. An abductive rule is encoded as follows:[4]

(13) $$\begin{array}{c} elephant_1(x) \rightharpoonup elephant(x) \Rightarrow \\ \Rightarrow pred(elephant, x, y) \wedge \\ \wedge x \in \{obsrv, asmpt\} \supset_0^1 \\ \supset_0^1 pred(elephant_1, asmpt, y) \wedge \\ \wedge explainedBy(elephant, y, rule_1(y)) \wedge \\ \wedge assumedBy(elephant_1, y, rule_1(y)) \end{array}$$

that is, if there is a predication we want to explain that can be unified with the consequent of an abductive rule, we may assume the antecedent (but we may ignore the rule because the cost of assuming the antecedent may be higher than the cost of assuming the consequent, thus we have to consider both cases). The auxiliary predications are used in the following rules, which help us guarantee that the computed stable model is a correct abductive proof (in the sense of [Hobbs *et al.*, 1993]):[5]

(14) $$\begin{array}{c} assumedBy(p, x, r) \supset assumed(p, x) \\ pred(p, asmpt, x) \wedge \sim assumed(p, x) \supset \bot \\ explainedBy(p, x, r_1) \wedge explainedBy(p, x, r_2) \wedge \\ \wedge r_1 \neq r_2 \supset \bot \end{array}$$

that is, a predication can be assumed only if it can be derived by backchaining on an abductive rule and a predication can be explained by no more than one rule.

The most important aspect is how variables in observations are handled. Since an answer-set program (ASP) has to be effectively propositional, we have to "emulate" equality. If a predication containing a "reified" variable (e.g., var_x) can be unified with another predication, we can bind the variable. For example, if the knowledge base contains $person_1(John)$ and we observe or assume $person_1(var_x)$, we may add $eq(John, var_x)$ to the stable model. We may also decide not to bind a variable, in which case a new individual has to be added to the knowledge base. Of course, we need axioms that guarantee that eq is an equivalence relation. We will not list all of them here but let us mention the most important axiom schema. If P is a predicate, we need $P(x) \wedge eq(x, y) \supset P(y)$ in order for deduction to work. Thus whenever we bind a variable, the knowledge base grows (in the worst case exponentially). There is no way around this problem since answer-set solving is propositional. Luckily for us, the logical form of a sentence contains only few variables (around a dozen), hence the presented method is viable for NLU. In actual fact, we are sacrificing space for time, since we need new literals for each variable assignment.

The rules described above correctly define all and only the stable models that correspond to an abductive proof in our NLU framework. We use defeasible rules to infer what might be true (thus extending our knowledge) based on what we already now. We process one sentence at a time in order to keep the proof search space as small as possible. Of course, in a connected discourse this may lead to a contradiction. Generally we try to assume as little as possible (see Section 5), but if we arrive at a contradiction, we have to backtrack and reprocess part of the discourse. We plan on using a truth maintenance system in the future, but for the time being, we simply reinterpret the sentences which might be affected by the false assumption.

We will now illustrate with an example how the solver finds proofs. Consider the following observation, abductive rules, and known facts (as usual, x is a variable and a, b are constants):

(15) $$\begin{array}{ll} \text{observation:} & q(x) \\ \text{rule:} & p(x) \rightharpoonup q(x) \\ \text{rule:} & r(x) \rightharpoonup q(x) \\ \text{facts:} & p(a), r(b) \end{array}$$

The complete proof graph for (15) is given in Figure 3. The labelled edges are possible "merges" (variable bindings by unification) and the unlabelled edges are licensed by backchaining on abductive rules. A proof is equivalent to a

[3] In the remainder of the paper, we use subscripted predicates to represent real-world meaning and unsubscripted predicates to represent lexical meaning. This distinction is necessary in order to accommodate lexical ambiguity and figurative speech such as metaphors and metonymy. For example, the literal real-world meaning of *John is an elephant* (whose logical form is $John(x) \wedge elephant(x)$) is $John_1(x) \wedge elephant_1(x)$, but if we already know that John is a person ($person_1(John)$) we are forced into a figurative meaning such as $John_1(x) \wedge clumsy_1(x)$. Thus in this sense, to interpret a sentence is to steer clear of contradictions in the knowledge base.

[4] $p \supset_0^1 q$ means `0{q}1 :- p`, that is, the consequent may or may not be included in the stable model.

[5] $\sim$ denotes default negation.

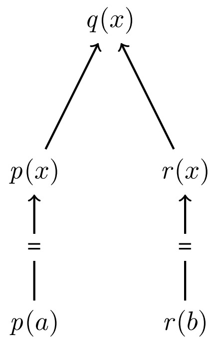

Figure 3: Complete proof graph for (15)

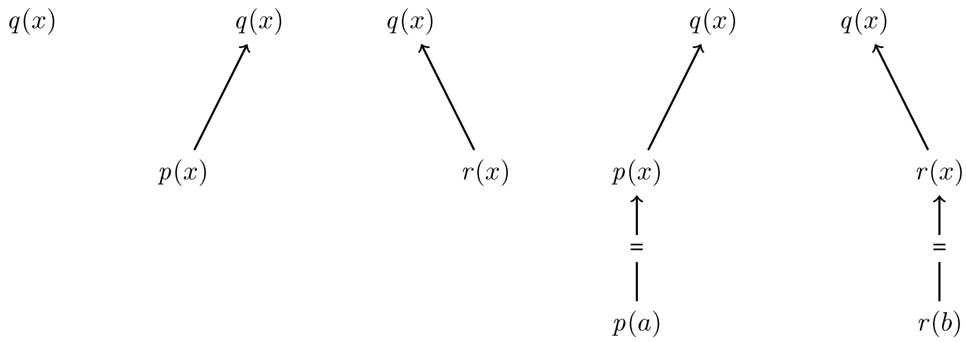

Figure 4: Proofs of (15)

subgraph of the complete proof graph complying with the following conditions:

1. An observation or assumption is explained by no more than one rule.[6]
2. There is a path from any assumption to an observation (i.e., we eliminate assumptions that do not contribute to the explanation of an observation).
3. Variable assignments conform to the usual constraints on equivalence.

All the proofs of (15) are depicted in Figure 4. The corresponding hypotheses (including the observation) are:[7]

(16)
$$\begin{array}{c} q(x) \\ p(x) \wedge q(x) \\ r(x) \wedge q(x) \\ q(a) \\ q(b) \end{array}$$

4 Constraining the proof search space

In modern answer-set solvers, aggregate functions such as $count$, sum, or max can be used. We can thus define a predicate whose argument tells us the size of (that is, the number of assumptions in) a proof and rule out any proof that is too big:

(17)
$$\begin{array}{c} numberOfAssumptions(x) = |\{p : assumed(p)\}| \\ numberOfAssumptions(x) \wedge \\ \wedge x > maxNumberOfAssumptions \supset \bot \end{array}$$

We can also have a predicate that tells us the length of the proof path from p_1 to p_2 and an integrity constraint that rules out any proof whose length is greater than an integer constant, $maxProofLength$:

(18) $proofLength(p_1, p_2, l) \wedge l > maxProofLength \supset \bot$

These two simple integrity constraints can significantly constrain the proof search space, thus speeding up the enumeration of proofs.

5 Ranking abductive proofs

Even a relatively simple background theory, as in our experiments, will have at least hundreds of abductive and deductive rules, which means that the logical form of an average sentence can have many different proofs (since both deduction and abduction are explosive). Moreover in a long discourse consisting of many sentences, the knowledge base will contain many individuals available for unification, which can multiply the number of proofs. The method of weighted-abduction [Hobbs *et al.*, 1993; Hobbs, 2001; 2003] seems to yield good results, but it cannot be used in

[6]This condition does not mean that a literal cannot be implied by more rules, it only says that only one (defeasible) rule is taken to be its explanation.

[7]In the domain of NLU, the logical form of a sentence is an existential closure so we would have to introduce a new individual for every free variable.

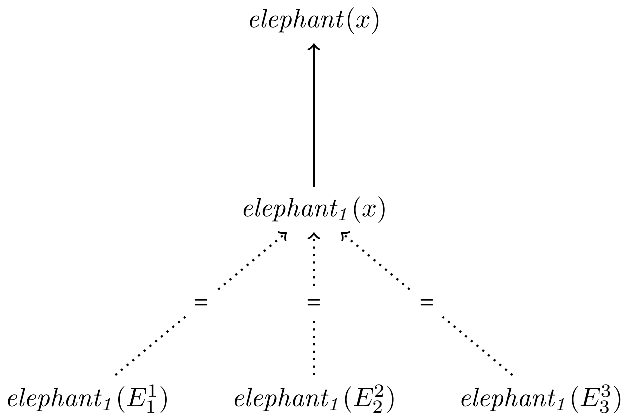

Figure 5: Literal interpretations of *elephant(x)*

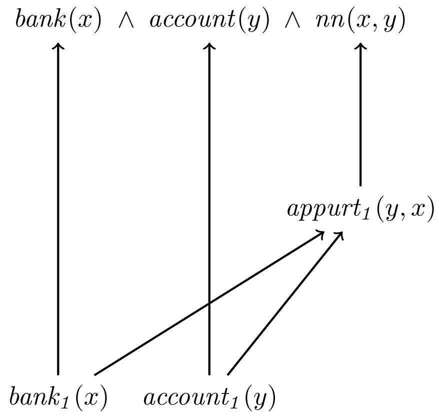

Figure 6: Interpretation of *bank account*

an answer-set program because it works with real numbers. While it is possible to enumerate all the abductive proofs and evaluate them later, we decided to try to solve the ranking problem within ASP. In this section, we describe how to rank proofs using the answer-set solver.

The basic idea, coming from Hobbs *et al.* (1993), is that one should prefer proofs that unify assumed predications with what we already know and assume as little as possible. In other words, maximally coherent (with respect to the context) and minimally ampliative proofs are preferred. We add one more criterion: salience. Informally, individuals that occurred recently in the discourse have higher salience. If the pronoun *he* is used in a sentence, it is interpreted by backchaining on the following abductive rule:

(19) $person_1(x) \wedge male_1(x) \rightharpoonup he(x)$

that is, *he* refers to a male person. To bind the variable, we have to find an individual which conforms to the selectional constraints. But there can be many such individuals in the knowledge base. Thus we rank the proofs with respect to their salience, which is the sum of the saliences of all individuals unified with a variable. Newly introduced constants are assigned the highest salience, as in the case of c for the indefinite NP in

(20) He bought a book.
$he(x) \wedge buy'(e, x, c) \wedge book(c)$

If there is $elephant(x)$ among the observations and the knowledge base contains $elephant_1(E_1^1), elephant_1(E_2^2), elephant_1(E_3^3)$,[8] there are three literal interpretations in which the variable x is unified, as illustrated in Figure 5. Based on a linguistic insight, we want to prefer the proof which unifies the variable x with the most salient individual.

Unification can help us resolve lexical ambiguity even when there are no individuals in the knowledge base which could be unified with the variables. Consider the compound nominal

(21) bank account
$bank(x) \wedge account(y) \wedge nn(x, y)$

and assume that we have the following background theory

[8] The upper index expresses the salience of the individual.

(with five lexical rules and one commonsense rule):

(22)
$$\begin{aligned} bank_1(x) &\rightharpoonup bank(x) \\ bank_2(x) &\rightharpoonup bank(x) \\ account_1(x) &\rightharpoonup account(x) \\ account_2(x) &\rightharpoonup account(x) \\ appurt_1(x,y) &\rightharpoonup nn(y,x) \\ account_1(x) \wedge bank_1(y) &\rightharpoonup appurt_1(x,y) \end{aligned}$$

that is, accounts (defeasibly) appertain to banks and the relation *appurt(enance)* can be expressed by a compound nominal (nn).[9] We see in Figure 6 that two predications are unified. If we interpreted $bank(x)$ as $bank_2(x)$ and/or $account(y)$ as $account_2(y)$, there would be no unification of predications and hence the proof would be less coherent.

As we have just shown, abduction can be used to lexically disambiguate phrases even if there is no additional context or previous discourse. Of course, in order for this method to work background theories are needed that capture relations between entities that occur in the sentence. Creating commonsense knowledge bases is generally a very complex task but it is feasible at least for smaller closed domains.

There can be many proofs with the same number of unified predications, thus we need a criterion that will help us distinguish them. The simplest criterion is the size of the proof, i.e., the number of assumptions made. Intuitively, we do not want to assume more than is necessary; if an assumption does not help us arrive at a more coherent proof (that is, a proof with more unifications), is should be omitted. This simple idea seems to be good enough to rule out most undesired proofs.

We can use the predicate $numberOfAssumptions$ defined in Section 4 and an analogously defined predicate $numberOfUnifications$ to select the best proof. Our method for ranking abductive proofs yields slightly better results than that proposed by Hobbs *et al.* (1993) in our evaluation, but this does not mean that it is better because the difference is not statistically significant and because Hobbs' method relies on probabilistic weights which are hard to "get right" empirically. Nevertheless our method is relatively simple and can be implemented in ASP (for it uses only natural numbers).

6 Conclusions

We have presented a method for translating (a fragment of) first-order abduction into an answer-set program in the context of NLU. The heretofore used algorithms do not allow for seamless integration of the process of abduction with deduction. Our method helps the solver confine the search space by ruling out logically impossible proofs (with respect to a background theory). We have also suggested how to rank proofs within the answer-set program, since the original framework of weighted abduction would require an additional step to evaluate the proofs. An evaluation has shown that in conjunction with a state-of-the-art answer-set solver, our method is an order of magnitude faster than the approach based on a general automated theorem prover.

There is no doubt that first-order Horn abduction is useful in many areas of artificial intelligence. Our future work will investigate how the proposed method for ranking proofs can be applied to automated goal-driven planning with incomplete information.

[9] $bank_x$ and $account_x$ are different lexical meanings of $bank$ and $account$, respectively.

References

[Appelt and Pollack, 1992] Douglas E. Appelt and Martha E. Pollack. Weighted Abduction for Plan Ascription. *User Modeling and User-Adapted Interaction*, 2(1–2):1–25, 1992.

[Bresnan, 2001] Joan Bresnan. *Lexical-Functional Syntax*. Blackwell Textbooks in Linguistics, New York, 2001.

[Dalrymple *et al.*, 1993] Mary Dalrymple, John Lamping, and Vijat Saraswat. LFG Semantics via Constraints. In *Proceedings of Sixth Meeting of the European ACL*, 1993.

[Dalrymple *et al.*, 1995a] Mary Dalrymple, Ronald M. Kaplan, John T. Maxwell III, and Annie Zaenen. *Formal Issues in Lexical-Functional Grammar*. CSLI Publications, 1995.

[Dalrymple *et al.*, 1995b] Mary Dalrymple, John Lamping, Fernando Pereira, and Vijat Saraswat. Linear logic for meaning assembly. In *Proceedings of Computational Logic for Natural Language Processing*, 1995.

[Dalrymple, 2001] Mary Dalrymple. *Lexical Functional Grammar*, volume 34 of *Syntax and Semantics*. Academic Press, 2001.

[Gebser *et al.*, 2012] Martin Gebser, Benjamin Kaufmann, and Torsten Schaub. Conflict-Driven Answer Set Solving: From Theory to Practice. *Artificial Intelligence*, 187–188:52–89, 2012.

[Gelfond and Kahl, 2014] Michael Gelfond and Yulia Kahl. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents. The Answer-Set Programming Approach*. Cambridge University Press, 2014.

[Hobbs *et al.*, 1993] Jerry R. Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin. Interpretation as Abduction. *Artificial Intelligence*, 63:69–142, 1993.

[Hobbs, 1985] Jerry R. Hobbs. Ontological Promiscuity. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pages 61–69, 1985.

[Hobbs, 2001] Jerry R. Hobbs. Syntax and Metonymy. In P. Bouillon and F. Busa, editors, *The Language of Word Meaning*, pages 290–311. Cambridge University Press, 2001.

[Hobbs, 2003] Jerry R. Hobbs. Discourse and Inference. MS, Information Sciences Institute, University of Southern California, Marina del Rey, 2003.

[Hobbs, 2005] Jerry R. Hobbs. Encoding Commonsense Knowledge. MS, Information Sciences Institute, University of Southern California, Marina del Rey, 2005.

[Inoue *et al.*, 2012] Naoya Inoue, Ekaterina Ovchinnikova, Kentaro Inui, and Jerry Hobbs. Coreference Resolution with ILP-based Weighted Abduction. In *Proceedings of COLING 2012*, pages 1291–1308, 2012.

[Kaplan and Bresnan, 1982] Ronald M. Kaplan and Joan Bresnan. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *Mental Representation of Grammatical Relations*. MIT Press, Cambridge, 1982.

[Ovchinnikova *et al.*, 2014] Ekaterina Ovchinnikova, Ross Israel, Suzanne Wertheim, Vladimir Zaytsev, Niloofar Montazeri, and Jerry Hobbs. Abductive Inference for Interpretation of Metaphors. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 33–41, 2014.