Towards A Tableaux Machine for Language Understanding

Michael Kohlhase, Alexander Koller
Universität des Saarlandes, Germany
{kohlhase@ags|koller@coli}.uni-sb.de

Abstract

We outline an abstract inference machine for producing discourse models in natural language understanding. This machine has tableaux as its central data structure and can operate in model generation and theorem proving modes. Search spaces are controlled by keeping track of NP saliences and equipping proof rules with costs.

1 Introduction

Model generation is an inference procedure that has recently become a popular approach to language understanding in computational semantics. In general, this is the problem of finding a (Herbrand) model satisfying a given set of formulas of predicate logic; NL understanding is analyzed as a process of generating models for the logical form of the discourse. This ties in nicely with semantics when natural language communication is understood as a process of transporting a "mental model" from the speaker's mind to the hearer's.

Unfortunately, pure model generation has some properties that limit its usefulness for computational semantics. Maybe the most obvious one is that the universe of the constructed model keeps growing monotonically with each analyzed sentence, and all entities and properties are equally available for new rule applications. This leads to combinatorial explosion very quickly. Humans cope with this problem by focusing on a small number of *salient* items; besides, they spend only a limited amount of *resources* on language processing.

In this paper, we outline some ideas about a *tableaux machine* for computational semantics that follows these principles. This machine operates on tableaux and takes commands and data from the syntax-semantics interface. It extends pure model generation with notions of salience and resources (in the form of proof rule costs) and will generally conduct a best-first model search that weights expenses for rule applications against the quality of a (partial) model that can be achieved in this way. This is a way out of the combinatorial explosion of the discourse universe because "old" entities (of low salience) are too expensive to consider. Furthermore, it can switch between model generation and theorem proving modes. This is very natural because both operate on tableaux, and it is useful; e.g. the TP mode can be used to check if the current model entails a certain formula.

The work reported here is very much in progress. In particular, the details of how costs are obtained and handled and how they interact with salience are still open. We list open questions comprehensively in the conclusion.

The structure of the paper is as follows. We will first discuss ordinary tableau refutation and model generation in Section 2; then we will extend model generation with resources, mode switching, (above all) saliences and integrate them into the tableaux machine in Section 3. In Section 4 we will apply the tableaux machine to the analysis of definite NPs, a notorious problem that has been tackled with MG techniques before. Finally, Section 5 concludes and lists open questions.

2 First-Order Tableaux for Refutation and Model Generation

First-order tableaux were originally used as a data structure in refutation procedures. Here a theorem is proved by decomposing its negation into a tree of possible instantiations (the tableau). Each branch in this tree corresponds to a possible model of the formula at the root, so if all branches are closed (inconsistent), the root formula is unsatisfiable, and the theorem is valid. Conversely, model generation procedures [MB88, LP93, FL96, Bau98, Pel99] build a tableau for the root formula and analyze the open branches as possible models, showing that it is indeed satisfiable. The two inference procedures only differ in strat-

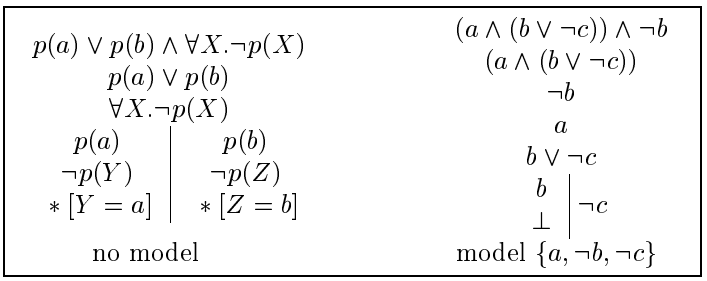


Figure 1: Refutation and Model Generation

egy (see Figure 1) and share both the underlying data structure (the tableau) and the propositional rules[1] in Figure 2. Tableau refutation uses the quantifier rules in Figure 3 and model generation those in Figure 4; furthermore, model generation calculi normally assume a kind of unique name assumption called Herbrand equality: two individual constants are equal, iff they are identical, consequently, a branch can be closed if it contains a literal $c = d^{\mathbf{f}}$.

Definition 2.1 (Tableau Refutation) The tableau refutation calculus uses the propositional rules in Figure 2 and the quantifier rules in Figure 3. The constant[2] f in rule $\mathcal{T}(\exists)$ is called a Skolem function and takes care of the variable dependencies.

Furthermore, it uses a generalized branch closure rule $\mathcal{T}(\bot)$ that allows to close a branch if there are literals $\mathbf{A}^{\mathbf{t}}$ and $\mathbf{B}^{\mathbf{f}}$ and a most general unifier σ of $\mathbf{A}$

[1] We have only shown those for (positive and negative) negated and conjunctive formulas; the other connectives are defined in terms of them as usual.

[2] We denote variables by upper-case letters and constants by lower-case ones.

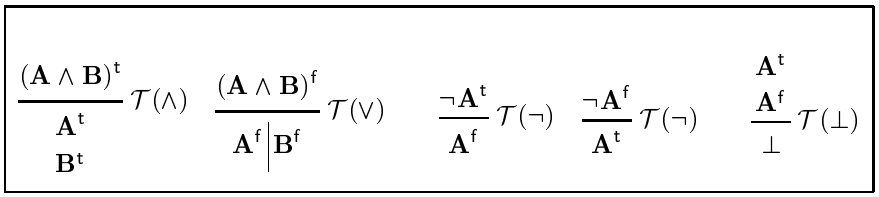


Figure 2: Propositional Tableau Rules

and $\mathbf{B}$. In this situation the rule closes the branch and applies σ to the whole tableau.

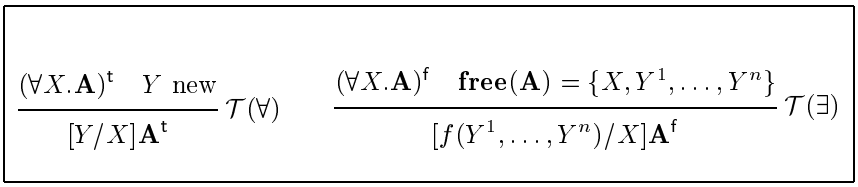


Figure 3: Free-Variable Quantifier Rules for Theorem Proving

Definition 2.2 (Model Generation with $\mathcal{RM}$) The model generation calculus $\mathcal{RM}$ consists of the propositional tableau rules (see Figure 2) and the model generation rules for the quantifiers (see Figure 4).

The $\mathcal{T}(\forall)$ rule tests the scope on all members of the Herbrand universe $\mathcal{H}$ of the current branch; it must be applied exhaustively to obtain a saturated branch. The $\mathcal{T}(\exists)$ rule reuses constants that occur in the current branch and alternatively introduces a new constant c^{new}. In this way, the $\mathcal{T}(\exists)$ minimizes the size of the universe and also avoids Skolemization which would introduce problematic function symbols into the Herbrand universe. For our purpose here, we can get by without function symbols, and thus always have a finite Herbrand universe.

When extending the Herbrand universe of a branch by c^{new}, all $\mathcal{T}(\forall)$ must be re-instantiated with respect to c^{new}.

The $\mathcal{RM}$ model generation calculus has been originally developed for a certain form of non-monotonic reasoning called minimal entailment [Lor94]. $\mathcal{RM}$ is a refutation complete first-order tableau calculus where each open saturated branch is a Herbrand model. [Kon00] proves that $\mathcal{RM}$ is complete for finite satisfiability, i.e. $\mathcal{RM}$ is a decision procedure for theories that either are unsatisfiable or have a finite model. Additionally, $\mathcal{RM}$ is complete for finite minimal models that also are domain-minimal: if a theory is finitely satisfiable, then one of the models generated by $\mathcal{RM}$ will be minimal with the smallest possible universe.

Model generation calculi have been used as an inference-based approach to natural language understanding (see [BK00, GK00, Kon00]). For instance

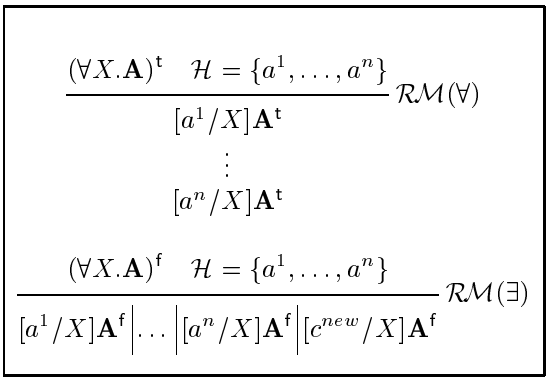


Figure 4: Quantifier rules for model generation ($\mathcal{RM}$)

Gardent and Konrad interpret the definite description "the" as $\lambda P, Q.\mathsf{uniq}(P) \wedge \mathsf{every}(P, Q)$, where $\mathsf{uniq}(P)$ stands for $\exists X.P(X) \wedge \forall Y\, P(Y) \Rightarrow X = Y$ and every for $\forall X P(X) \Rightarrow Q(X)$. In a sentence like "the rabbit is cute", we obtain the logical form $\mathsf{uniq}(rab) \wedge rab \subseteq cute$. Due to domain minimality, the $\mathcal{RM}$ calculus generates the model $\{\ldots, rab(c), cute(c)\}$ in situations with at most one rabbit. The specific form of the $\mathcal{T}(\exists)$ rule of $\mathcal{RM}$ gives the intended behavior that a rabbit is reused if one exists in the discourse model, and if there is none, one (c^{new}) is accommodated.

In situations with more than one rabbit, this analysis fails. Therefore, Gardent and Konrad refine the analysis of "the" to $\lambda P^1, P^2, Q.\mathsf{uniq}(P^1 \cap P^2) \wedge \mathsf{every}(P^1 \cap P^2, Q)$, where P^2 is an "identifying property", which they claim is determined by the context. For instance in

(1) Bugs and Bunny are rabbits. Bugs is in the hat. Jon removes **the rabbit** from the hat.

the determining property P^2 is "being in the hat" (which is induced by world knowledge about the action of removing), which makes Bugs unique and the discourse acceptable.

The big advantage of the model-generation approach is that it can use the world knowledge and thus e.g. account for bridging references. A problem of this approach is that it seems difficult to determine the identifying properties at all. Gardent and Konrad give examples like their (1) but no general theory. [HSAM93] face similar problems in their related abductive approach, and push the problem into the world model by using a generic relation (rel for relatedness) and simply postulate world knowledge e.g. to infer $\mathsf{rel}(x, y)$ from $\mathsf{part_of}(x, y)$. This approach seems to lead to difficult ontological problems; e.g. is there a general theory of relatedness? We believe that some of these problems can be alleviated by incorporating saliences into the logical analysis (see below); but clearly, saliences alone will not solve all problems with definites all by themselves.

Another problem that all of the model generation approaches share is that they are computationally intractable except on relatively small discourses and axiomatizations of the world knowledge. For instance the number of persons

introduced in the first pages of a typical novel (and their properties) will already lead to combinatorial explosion. Domain minimality as in [GK00] helps for toy discourses, but is no general solution where a text keeps introducing new objects into the discourse universe. Moreover, world knowledge such as that every human has a father may cause a model generator to go off and generate huge chains of ancestors.

3 Towards a Tableaux Machine

In this situation, we propose to embed the model generation process into a more general **Tableaux Machine**, which has more control over the inference process. More concretely, we model natural language understanding as an incremental process as consisting of i) syntactic analysis, which interprets sentences one-by-one, ii) semantics construction, which generates logical forms for them and passes them as input to iii) the Tableaux machine. The latter is a search process that encapsulates a tableau whose branches correspond to a set of models for the interpreted discourse; we will discuss it in more detail in section 3.2

At the heart of the model generation in the tableau machine is a notion of *salience*, which allows to distinguish how "important" the various objects in the discourse universe are right now; because importance of an object decays if it is not mentioned for a long time, this also allows to "forget" about such objects. Furthermore, the inference rules are associated with **costs**, and the inference process takes these costs into account (it is **resource-sensitive**). So generation of long ancestor chains as above is still possible (and completeness is retained), but strongly dispreferred. We change some rules of $\mathcal{RM}$ to take saliences and resources into account; the new calculus thus obtained is called RAMG.

We speak about a tableaux **machine** for two reasons: We want to suggest a "black box" architecture that is fed input by the syntax-semantics interface and then goes off to work independently, and we want to stress that the machine has non-local control over the inference process (e.g. expansion strategies, cost optimization) unavailable in a calculus.

Below, we first spend some time on introducing salience, then we outline the tableaux machine. This is the central section of the paper, and it is where its "in progress" nature is most obvious. In particular, the salience section makes various assumptions that should be tested carefully, and the discussion of resources is rather sketchy so far.

3.1 Salience

The **salience** of an object is defined as the availability of an entity in the discourse universe for (anaphoric or definite) reference. Our treatment of salience is inspired by theories of anaphoric reference (e.g. [Als87, LL94]): It is at first approximated by syntactic criteria (e.g. the type of NP, its grammatical and thematic role), decays over time, and is raised whenever the entity is referred to.

Originally, each NP is assigned a **salience expression** (an arithmetic expression in terms of universal variables, constants, and natural numbers) by the syntax-semantics interface. This salience is annotated to the existential quantifiers (occurrences of $\forall$ under a negative polarity in the logical form of the input)

and constants in the formula fed to the tableaux machine. For instance, the sentence

(2) Mary chases a rabbit.

could be assigned the following output from the syntax-semantics interface:

$$\exists X_{60}.\mathsf{rabbit}(X) \wedge \mathsf{chase}(\mathsf{mary}_{100}, X)$$

Mary's salience is significantly higher than the rabbit's because she is the subject of the sentence. For this article, we simply assign subjects salience 100 and objects salience 60, but one would ultimately employ a more intricate system, as in [LL94].

Since salience decays, each branch in the tableau is stratified into **salience levels** (one per analyzed sentence), each of which is associated with a **salience function** assigning natural numbers (saliences) to constants. When the tableau branches, each of the two choices gets a salience function of its own, but we assume by default that they are equal and only mention deviations from this general rule. We indicate the (unique) salience function of a level on the right hand column of the tableaux in the examples.

Each level is initialized with the salience function that maps each term to half the salience it had in the previous level (this intuition also comes from [LL94]). Then saliences for new terms can be added, and saliences for existing terms can be raised – either by incorporating a term from the syntax-semantics interface (using a constant boosts its salience) or by application of an inference rule.

$$\frac{a, b \in \mathcal{H}}{a = b^{\mathbf{t}} \,\big|\, a \neq b^{\mathbf{t}}}\,\mathcal{R}(id)^{1} \qquad \frac{}{(\mathbf{A} \vee \neg\mathbf{A})^{\mathbf{t}}}\,\mathcal{R}(tnd)$$

$$\frac{(\forall X.\mathbf{A})^{\mathbf{t}} \quad a \in \mathcal{H}}{[a/X]\mathbf{A}^{\mathbf{t}}}\,\mathcal{R}(\forall) \qquad \frac{(\forall X_S.\mathbf{A})^{\mathbf{f}} \quad c^{new} \notin \mathcal{H}}{[c^{new}/X]\mathbf{A}^{\mathbf{f}}}\,\mathcal{R}(\exists)^{2}$$

+ standard equality rules $\mathcal{R}(=), \mathcal{R}(sym), \mathcal{R}(refl), \mathcal{R}(trans)$

[1] The saliences of a and b are replaced by their sum in the left branch.
[2] c is introduced with salience $S{\downarrow}$.

Figure 5: RAMG rules

The latter is achieved by the revised expansion rules in Figure 5. The most interesting rule in this respect is $\mathcal{R}(\exists)$. This rule introduces a new constant c whose salience, $S{\downarrow}$, is determined from the salience expression S for the existential variable.[3] At the time when the existential quantifier is the main connective of the formula, all universal variables in S must have been instantiated to constants by applications of the $\mathcal{R}(\forall)$ rule. We obtain $S{\downarrow}$ from this instantiation of S by replacing all constants by their current saliences and evaluating. This is

[3] This is an important difference to $\mathcal{RM}$ motivated by salience and resource considerations: The old $\mathcal{RM}(\exists)$ rule was explosive, since it created branches for all members of the Herbrand base, whereas $\mathcal{R}(\exists)$ simply introduces one new constant, which can be identified (by need) with an existing one by the new $\mathcal{R}(id)$ rule.

a very common pattern of salience calculation, which capture by the *chaining rule* $\mathcal{R}(\Rightarrow)$, shown in Figure 6. As we have seen, it can be derived from the other rules.

$$\frac{\begin{array}{c}\forall X^1\mathbf{A}^1 \Rightarrow \ldots \forall X^n\mathbf{A}^n \Rightarrow \exists Y_S.\mathbf{B}^{\mathsf{t}} \\ \mathbf{A}^1(a^1)^{\mathsf{t}} \\ \ldots \\ \mathbf{A}^n(a^1,\ldots,a^n)^{\mathsf{t}} \qquad\qquad c \notin \mathcal{H}\end{array}}{\mathbf{B}[a^i/X^i, c/Y]^{\mathsf{t}}}\ \mathcal{R}(\Rightarrow)^1$$

[1] c is introduced with salience $S(a^1,\ldots,a^n)\downarrow$.

Figure 6: The chaining rule

To get an intuition for this mechanism, let us go through a simple analysis of bridging inferences, as in the following small discourse.

(3) Mary could not start her car. The battery was empty.

We write

$$\mathbf{S}^1 = \exists X_{60}.\mathsf{car}(X) \wedge \mathsf{own}(\mathsf{mary}_{100}, X) \wedge \neg\mathsf{start}(X)$$

for the semantics of the first sentence and (without going into the analysis of definites here – but see Section 4) simply $\mathbf{S}^2$ for that of the second.

In the first salience level, we expand $\mathbf{S}^1$. This introduces Mary and her car into the discourse universe with appropriate saliences. Now we can apply the chaining rule $\mathcal{R}(\Rightarrow)$ with the following rule from the world knowledge:

$$\mathbf{R} = \forall X.\mathsf{car}(X) \Rightarrow \exists Y_{X/2}\mathsf{bat}(Y) \wedge \mathsf{part_of}(Y, X)$$

This introduces the battery of Mary's car into the universe, with salience 30 (because the car has salience 60). This completes the analysis of the first sentence. Now the second salience level is initialized by giving each individual half the salience they had in the first level; but the battery in Mary's car is still much more salient than any other battery in the world, and the definite reference can succeed. This argument is shown in more detail in Figure 7. Note that it would still work if there had been more objects in the universe, as long as their (decayed) saliences were low; and that if we had known that Mary is a robot, which also has batteries (by a variant of $\mathbf{R}$), inference would have correctly given Mary's battery more salience (50) than the car's, and the definite would have been resolved to denote Mary's battery.

To speak about saliences within our calculus, we assume a special binary predicate $\prec$, the **salience ordering**. $c \prec d$ is true iff the salience of c is much lower than that of d at the current salience level; that is, if $c : n$ and $d : m$ at this salience level and $n \geq (1+\delta)m$ for some predefined threshold $\delta \in \mathbb{R}$.[4] Correspondingly, we extend the set of rules in Figure 5 by a rule $\mathcal{R}(\prec)$ that closes a branch ending in a literal $c \prec d^{\mathsf{t}}$ if $c \prec d$ is false, and a branch ending in $c \prec d^{\mathsf{f}}$ if it is true.

Of course, an exact specification of the salience expressions in the syntax-semantics interface and the world knowledge is still an open problem, which we

[4] As a consequence, the relative saliences measured by this are invariant under salience decay, and can only be changed by inference.

#	formula	Justification		salience	
		by	from	level	*value*
0	$\mathbf{R}^{\mathsf{t}}$	*World*			
1	$\mathbf{S}^{1\,\mathsf{t}}$	*Given*			
2	$\mathsf{car}(c)^{\mathsf{t}}$	$\mathcal{R}(\exists)$	1	1	$c : 60$
3	$\mathsf{own}(c, \mathsf{mary})^{\mathsf{t}}$	$\mathcal{R}(\exists)$	1	1	$\mathsf{mary} : 100$
4	$\mathsf{start}(c)^{\mathsf{f}}$	$\mathcal{R}(\exists)$	1	1	
5	$\mathsf{bat}(d)^{\mathsf{t}}$	$\mathcal{R}(\Rightarrow)$	0, 2	1	$d : 30$
6	$\mathsf{part_of}(d, c)^{\mathsf{t}}$	$\mathcal{R}(\Rightarrow)$	0, 2	1	
7	$\mathbf{S}^{2\,\mathsf{t}}$	*Given*		2	$c : 30$
				2	$\mathsf{mary} : 50$
	...	...		2	$d : 15$

Figure 7: A tableau generated by RAMG

have not solved in this paper. Essential to the design of such a specification is that there should be no cyclic relations which allow expansions of infinitely increasing salience (from a son to a father to a son etc.).

3.2 Tableaux Machine

A **tableaux machine** is a machine whose states are tableaux with saliences. It takes the logical forms (with salience expressions) generated by semantics construction module as input, adds them to all open branches of the current tableau, and then performs inferences until some resource criterion is met (see below).

The actual output of the tableau machine is application dependent; some choices are

- the preferred (minimal) model given as all the (positive) literals of the preferred branch;
- the literals augmented with all non-expanded formulas (from the discourse); since our tableaux machine is resource-bounded, there might be formulas in the branch that have not been fully expanded to literals which could have contributed to the model or prohibited it;
- the user can give queries to the machine, and it decides whether a model entails truth of the query.

Subtableaux may be marked as being in **theorem-proving mode**; we draw boxes around these subtableaux. In this mode, the machine operates exactly like a tableau theorem prover: It employs the theorem-proving quantifier rules and the RAMG equality rules (but not $\mathcal{R}(tnd)$, $\mathcal{R}(\forall)$, $\mathcal{R}(\exists)$) and attempts to close all branches. For the purposes of this paper, we will ignore resources in TP mode. It may eventually turn out that solving TP side-conditions is actually a resource-adaptive process as well.

In model-generation mode, inference is guided by two factors: resources and strategies. To deal with resources, each proof rule comes with **rule costs**. Ultimately, we envision a bounded optimization regime: Expansion continues only as long as the expected gain in model quality outweighs the cost of any

possible rule application, in the spirit of [RW91]. For instance, small models would be better than large models. For the purposes of this paper, however, we make the highly simplifying assumption that each sentence in the discourse comes with a fixed budget that we can use to perform inferences on it, and that computation simply stops when the budget is exhausted. The basic effect is the same: Expensive rules are rarely applied.

Finding appropriate values for the rule costs is a major open problem of our approach. Qualitatively, however, it should be expensive to instantiate universal formulas, except in the context of the chaining rule. Application of rules that perform unmotivated case distinctions, such as Tertium Non Datur ($\mathcal{R}(tnd)$) and $\mathcal{R}(id)$, should be extremely expensive. Cheap operations include conjunction expansion and the existential expansions $\mathcal{R}(\exists)$ and $\mathcal{R}(\Rightarrow)$. Costs should depend on the saliences of the constants that occur in a rule instance – it should be more expensive to handle formulas with low-salience constants than with highly salient ones. This circumvents the computational problems usually caused by growing discourse universes: Most constants will generally be of low salience, and application of rules to these constants will be dispreferred due to their costs.

Strategies constitute non-local control knowledge above the level of single inference rules. For instance, we assume that existential quantifier rules ($\mathcal{R}(\exists), \mathcal{R}(\Rightarrow)$) are applied as early as possible, in order to keep the discourse universe up to date. More interestingly, when all branches of a MG subtableau are closed, the machine will perform "error recovery" in order to repair this deficiency. We will see this at work in the analysis of accommodation in Section 4.

The result of an inference rule is generally added as highly in the tableau as possible within the current salience level. This technique is well-known in tableau theorem proving as **folding up** (see e.g [LMG94]).

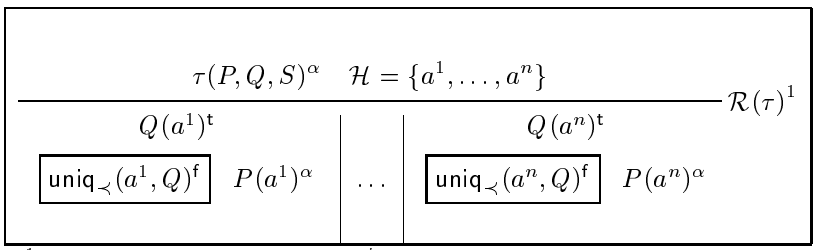
4 An Application: Definites

In this section we sketch an application of the tableaux machine that utilizes salience even beyond its role in controlling the search space: the interpretation of definite NPs. The analysis presented here differs from previous analyses in [HSAM93, GK00]; it exploits all three extensions to plain model generation that we defined in the previous section. It is not (yet) intended as a theory with complete linguistic coverage, but as a test case to see how far the present approach will carry without further extensions.

4.1 The τ Connective

For our analysis of definite reference, we introduce a new type of quantified formula into the language: $\tau(P, Q, S)$. This formula is produced by the syntax-semantics interface as the semantics of a sentence like "the Q P", with salience S (a salience expression as for existential quantifiers which will become important in Section 4.2). It does not receive a semantics of its own because we do not want to be bogged down with an explicit account of the inherently context-dependent nature of definites in the semantics. Rather, the tableaux machine takes it as a request for instantiation with the correct referent.

If we set aside accommodation for a minute, this referent must be one of the constants in the Herbrand universe. The referent for a definite with restriction Q must be unique in the sense that either there is only one entity in the discourse universe that can have the property Q, or that a unique entity is most salient among them. This motivates the expansion rule for the τ connective, shown in Figure 8. Note that the boxes around the "$\mathsf{uniq}_{\prec}$" formulas indicate that these subtableaux are in theorem-proving mode.



[1] For each i, the salience of a^i is raised by $S{\downarrow}$ in the i-th branch.

Figure 8: The RAMG rule for definite descriptions.

Consider an arbitrary branch of the rule, say for a^i. The branch closes immediately if we can derive within the tableau that $\neg Q(a^i)$, i.e. for all entities that we *know* not to have the property Q. Supposing that this is not the case, we must next close a subtableau starting with the negation of the formula

$$\mathsf{uniq}_{\prec}(a^i, Q) := \bigwedge_{c \in \mathcal{H}} Qc \Rightarrow a^i = c \vee c \prec a^i$$

In order to close the theorem-proving subtableau, we must prove (within the given resources) for each object in the Herbrand universe that has the property Q that it is either equal to a^i or much less salient. This formalizes the above intuition that the referent of a definite must be unique among the salient entities. The branch for a^i can remain open only if the subtableau can be closed; so up to equality of constants, only one branch can ever be open, and if it is, we are guaranteed that it has the correct referent for the definite. The salience of a^i is raised in this branch because the object has just been mentioned in the discourse.

The cost for the new rule $\mathcal{R}(\tau)$ is set to zero, to the effect that τ formulas are always expanded in the current salience level. Some examples will help to make this clear.

(4) The largest state of the US has rich oil reserves.

(5) Joe Smith gave a press conference today. The CEO of Westinghouse announced budget cuts.

The first example can be interpreted by using world knowledge alone (Figure 9, where we have extracted the theorem-proving subtableaux $\boxed{1}$ and $\boxed{2}$). We know from world knowledge that there is a unique largest state of the US, for which we introduce the Skolem constant a. Now we must expand the τ formula for each member of the Herbrand universe, as shown in Figure 9 for $\mathcal{H} = \{\mathsf{a}, \mathsf{b}\}$, and close a subtableau in each branch. This is easy for a: The left branch closes by reflexivity of $=$, and the right one by application of the uniqueness condition

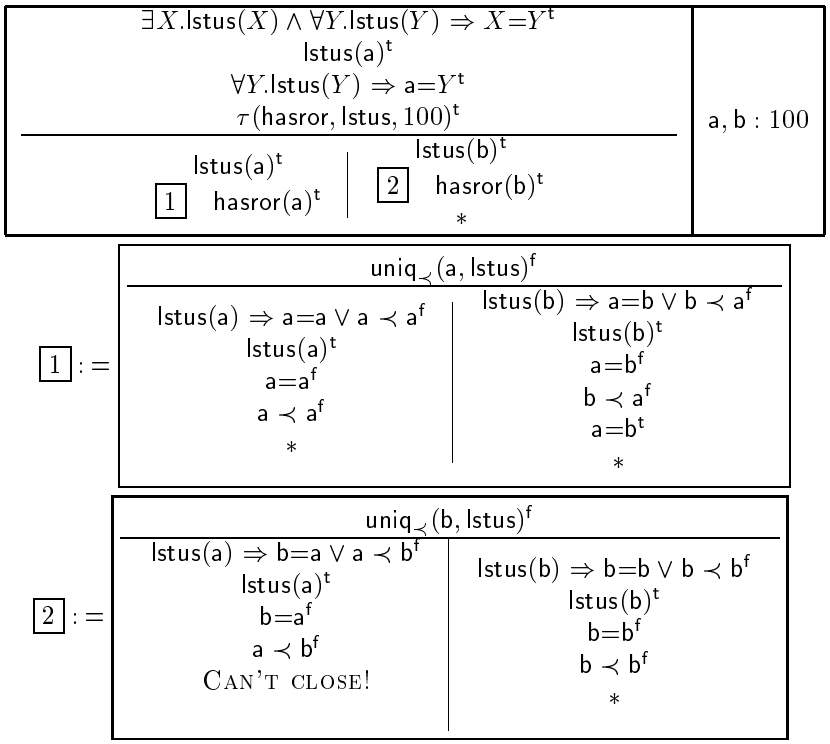


Figure 9: The largest state in the US has rich oil reserves.

in the world knowledge. However, the b subtableau will not close. Hence, we can continue computation only on the a branch, where a's salience has now been increased.

A more interesting example involves arguing about salience: (5) has a structure that is common in newspaper texts. In this example, the constant js (for "Joe Smith") is much more salient than any other because it has been the subject of the preceding sentence. So although we do not know of anyone to be a unique CEO (which is why the subtableau for a does not close), we can still close the subtableau for js by $\mathcal{R}(\prec)$ because a is much less salient at the time. Note that we have simplified the presentation by analyzing e.g. "gave a press conference" as the predicate gpc. However, it would not have hurt to introduce a specific constant for the press conference: There would have been a third branch in the τ expansion, but it could have been closed immediately because we know that CEOs are persons and hence a press conference cannot be a CEO.

The analysis of τ as a quantifier solves examples such as the following:

(6) The president of every New York City authority is a crook.

We ignore the strong reading (which could be taken care of by accommodation, see below) here and focus on the preferred reading,

$$\forall X.\mathsf{nyca}(X) \Rightarrow \tau(\lambda Y.\mathsf{presof}(Y, X), \mathsf{crook}, X/2)$$

Expansion of this formula will be suspended until the tableau contains an entity known to be a NYC authority because liberal expansion of universal formulas is expensive. Once a constant a is known to be a NYC authority (say with salience

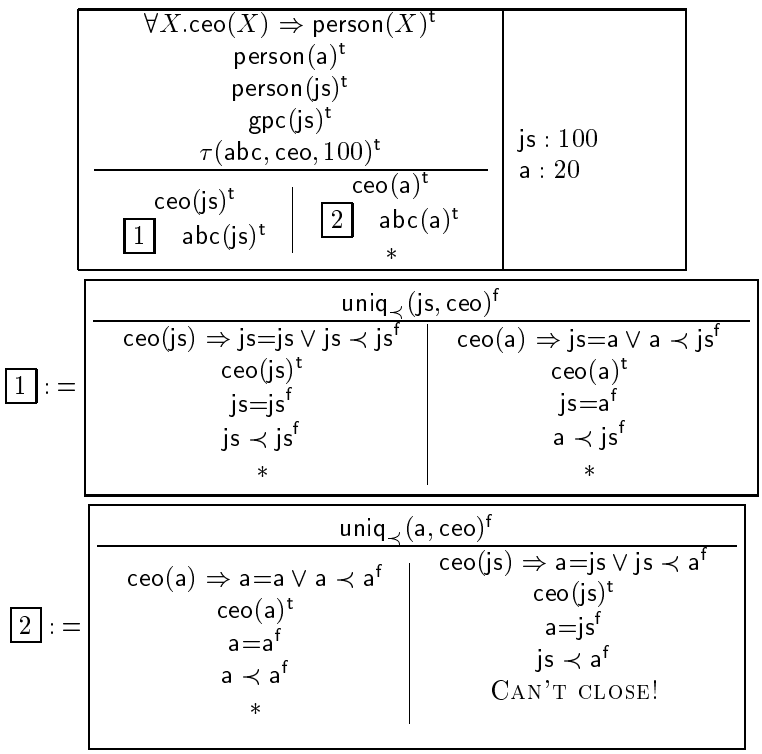


Figure 10: The CEO of Westinghouse announced budget cuts.

30), however, we can apply a chaining rule which looks exactly like $\mathcal{R}(\Rightarrow)$, but with τ instead of $\exists$.[5] This adds the following formula to the tableau:

$$\tau(\lambda Y.\mathsf{presof}(Y, \mathsf{a}), \mathsf{crook}, 15)$$

Now we can apply the $\mathcal{R}(\tau)$ rule as above. The λ-term is only a notational convenience: It is always applied to a constant or variable immediately.

Bridging as discussed in Section 3 is taken care of by salience inheritance. Example (3) will go through with this analysis without any problem.

4.2 Accommodation

Throughout the discussion so far, we have assumed that the referent for a definite NP is already present in the discourse universe. But of course, this is not always true, e.g. in the following sentence:

(7) Mary feeds the rabbit in her garden.

If there are no objects which could be rabbits in our discourse model, computation as above will fail because each branch of the τ expansion closes immediately.

In this situation, the "error recovery" process mentioned in Section 3.2 takes place. To evade closure of the top-level tableau, the machine backtracks to

[5] Note that although the chaining rules can logically be derived from the other rules, we can obtain interesting computational behavior by introducing them with different (lower) costs.

immediately before the application of $\mathcal{R}(\tau)$; then it applies $\mathcal{R}(tnd)$ once with the formula

$$\exists X_{60}.\mathsf{rihg}(X) \vee \neg\exists X.\mathsf{rihg}(X)$$

To obtain the salience attached to the existential quantifier, we evaluate the S parameter of the τ formula. This always produces a number; in this case, 60 because "the rabbit in her garden" is the direct object. Then the machine continues operation as usual. On the positive branch of the new disjunction, it immediately expands the quantifier, introducing a new Skolem constant with appropriate salience into the Herbrand universe. Now $\mathcal{R}(\tau)$ can succeed because there is a new, salient potential rabbit. Nothing new happens on the negative branch.

Our application of the $\mathcal{R}(tnd)$ rule is similar in effect to weighted abduction [HSAM93]: Both introduce a "helpful" formula, at a cost. Note that $\mathcal{R}(tnd)$ is only applied as a last resort because it has the potential to blow up the search space tremendously. So $\mathcal{R}(tnd)$ essentially reconstructs abduction in a deductive framework.

It will be an interesting test for our approach, whether this accomodation for uniquenes can be extended to other presuppositions.

5 Conclusion

In this paper, we have presented the first ideas for a tableaux machine, which is intended as the core of a general inference-based framework for natural language understanding. It combines techniques from weighted abduction [HSAM93] and current model-generation approaches and extends them by a treatment of salience. Salience, usually considered in connection to anaphora resolution, serves as a mechanism for controlling the search space, together with rule costs. Furthermore, it is of independent interest in an analysis of definite reference.

Comparison to related work. A difference between our framework and weighted abduction [HSAM93] is that assigning saliences to (individual) constants (as we do) allows a more fine-grained modeling of computational behavior than considering only rule costs on the level of literals. [HSAM93] use increased assumption costs for literals that contain salient constants to prefer abductions that infer these literals from the world knowledge. However, it is impossible to trace the influence of specific constants, e.g. that of a as opposed to that of b in a literal $p(a, b)$. In this respect weighted abduction can *simulate* some aspects of salience, while our approach that *models* saliences can be seen as an explanation to some of their rule costs.

In contrast to approaches to salience for pronominal anaphora [Als87, LL94], where saliences are used as a special device to disambiguate anaphora resolution from which it takes many intuitions, RAMG makes salience a ubiquitous mechanism that interacts with world knowledge. Thus our approach can be seen as a semantic theory of salience.

Perspectives. The theory of definites as presented in Section 4 is far from complete (and is not intended that way), even if the combination of model-generation, model-entailment, and salience consideration seems to cover a lot of ground. For instance, while (1) can be analyzed because Bugs is more salient than Bunny, it is easy to construct variants where our analysis breaks down. It seems likely, though, that it can be extended to cover such cases. Furthermore,

it would be interesting to think about the interaction of saliences and discourse relations, as in

(8) Joe Smith gave a press conference today. *In contrast*, the CEO of Westinghouse took the day off.

Here the CEO should *not* be Joe Smith.

More generally, the framework laid down so far is only a first step and leaves many open questions. Most pressing among these are the details of rule costs and saliences: The costs must be determined, salience expressions must be further specified, produced by the syntax-semantics interface, and incorporated into world knowledge, and the influence of salience on rule costs must be spelled out. An implementation of the tableaux machine is indispensable for experimentation, to verify that there is a consistent choice of costs at all, and also to show that the new mechanisms really manage to limit search space sizes. We are currently looking at various corpora as case studies, and consider investigating the use of learning techniques from theories of bounded rationality, such as the ones sketched in [RW91].

On the other hand, several design decisions are only motivated by introspection so far and need to be substantiated (or changed) by real psycholinguistic evidence. One example is the present order-dependence of computation: the time when a formula involving saliences is expanded makes a difference to the saliences its constants receive. This seems plausible, but restoring order-independence, e.g. by a modification of the folding-up strategy, is feasible as well. Another example is the decision whether universally quantified sentences in the discourse should be treated in exactly the same way as universal rules in the world knowledge, which even touches upon the interaction of acquiring language and knowledge.

Finally, we would like to understand better the relation of the model generation approach to dynamic semantics (e.g. DRT and DPL see [vEK96]). Like dynamic logics, model generation dynamically introduces discourse referents (in our case new constants by the $\mathcal{R}(\exists)$ rule) for quantified expressions. This structural similarity is highlighted by the fact that an extension of model generation to DRT (see [Koh00]) is conceptually very simple. Both logical systems make make a difference in treating universal and existential quantifiers and make predictions about the visibility of these referents; e.g. DRT by an explicit accessibility relation, and RAMG by decaying saliences. It will be one of the next steps to combine the techniques from [Koh00] with the approach in this paper. Given such mechanisms, we hope that "uniqueness under scope" examples [GK00] such as the following sentence can be analyzed.

(9) Every rabbit who sees a carrot, eats the carrot.

Here our approach has problems, since both the rabbit and the carrot are universal, and do not receive saliences, making the definite difficult to resolve in our approach.

Acknowledgments. The authors would like to thank Markus Egg, Peter Baumgartner, and an anonymous referee for helpful suggestions and examples.

References

[Als87] Hiyan Alshawi. *Memory and Context for Language Interpretation.* Cambridge University Press, Cambridge, 1987.

[Bau98] Peter Baumgartner. Hyper Tableaux — The Next Generation. In Harry de Swaart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, Lecture Notes in Aritificial Intelligence. Springer, 1998.

[BK00] Peter Baumgartner and Michael Kühn. Abducing Coreference by Model Construction. *Journal of Language and Computation*, 1(2):175–190, 2000.

[FL96] Christian Fermüller and Alexander Leitsch. Hyperresolution and automated model building. *Journal of Logic and Computation*, 6(2):173–230, 1996.

[GK00] Claire Gardent and Karsten Konrad. Interpreting definites using model generation. *Journal of Language and Computation*, 1(2):193–209, 2000.

[HSAM93] J. Hobbs, M. Stickel, D. Appelt, and P. Martin. Interpretation as abduction. *Artificial Intelligence*, 63:69–142, 1993.

[Koh00] Michael Kohlhase. Model generation for discourse representation theory. In Werner Horn, editor, *Proceedings of of the 14th European Conference on Artifical Intelligence*, Brighton, UK, 2000. John Wiley & Sons, Chichester, UK. forthcoming.

[Kon00] Karsten Konrad. *Model Generation for Natural Language Semantics.* PhD thesis, Universität des Saarlandes, 2000.

[LL94] Shalom Lappin and Herbert Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.

[LMG94] R. Letz, K. Mayr, and C. Goller. Controlled Integrations of the Cut Rule into Connection Tableau Calculi. *Journal of Automated Reasoning*, 13, 1994.

[Lor94] Sven Lorenz. A tableau prover for domain minimization. *Journal of Automated Reasoning*, 13:375–390, 1994.

[LP93] Shie-Jue Lee and David A. Plaisted. Problem solving by searching for models with a theorem prover. *Artificial Intelligence*, 69(1-2):205–233, 1993.

[MB88] Rainer Manthey and Franccois Bry. SATCHMO: A theorem prover implemented in Prolog. In Ewing L. Lusk and Ross A. Overbeek, editors, *Proceedings of the 9th Conference on Automated Deduction*, number 310 in LNCS, pages 415–434, Argonne, Illinois, USA, 1988.

[Pel99] N. Peltier. Pruning the search space and extracting more models in tableaux. *Logic Journal of the IGPL*, 7(2):217–251, 1999.

[RW91] S. J. Russell and E. Wefald. *Do the Right Thing — Studies in limited Rationality.* MIT Press, 1991.

[vEK96] Jan van Eijck and Hans Kamp. Representing discourse in context. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 179–238. Elsevier Science B.V., 1996.