Resource-Adaptive Model Generation as a Performance Model

MICHAEL KOHLHASE

Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213, USA. E-mail: `kohlhase+@cs.cmu.edu`

ALEXANDER KOLLER

Computational Linguistics, Saarland University, Saarbrücken, Germany. E-mail: `koller@coli.uni-sb.de`

ABSTRACT: Model generation calculi, close relatives of tableau calculi for theorem proving, can be used as competence models for semantic natural language understanding. Unfortunately, existing model generation calculi are not yet plausible as performance models *of actual human processing, since they fail to capture computational aspects of human language processing.*

We outline an extended model generation calculus that solves the most unpleasant computational inadequacy; In the extended calculus, tableau expansion rules are equipped with costs, and model construction is a process that optimizes model quality under resource constraints with respect to these costs. We embed the new calculus into an abstract inference machine and illustrate the possibilities of this approach by presenting a partial theory of definite descriptions in this setting.

In this case study, the constants in the universe are given saliences, that are maintained across the model generation process. This additional data serves as one important source of information for model quality and resource cost estimation.

KEYWORDS: *model generation, natural language understanding, bounded optimality, performance models*

1 Introduction

Model generation [21, 18, 7, 3, 22] is an inference procedure that has recently been applied to language understanding in computational semantics. In general, model generation is the problem of finding a (Herbrand) model satisfying a given set of formulas of predicate logic. Model generation procedures are usually based on tableau calculi that analyze the satisfiability of sets of input formulae in an and/or tree. These procedures saturate a tableau by applying expansion rules; branches that do not contain contradictions correspond to models of the input formulae.

In the application to natural language understanding, model generation can be used to generate a first-order model from the logical form of the discourse [4, 9, 16]. If we understand natural language communication as a process of transporting a "mental model" from the speaker's mind to the hearer's, the function of model generation is to reconstruct the model on the hearer's side. A central advantage of model generation as a model of language understanding is that world knowledge can be taken into account very easily.

Unfortunately, pure model generation has some properties that limit its usefulness for computational semantics. Maybe the two most unpleasant ones are what we will call the *omniscience problem* and the *universe explosion problem*. The former is the problem that a model generation procedure will eventually derive all logical consequences of the input sentences and the world knowledge, which humans certainly don't do. On the computational side, this means that because of the undecidability of predicate logic, saturation need never finish. Universe explosion is the problem that the universe of the constructed models grows monotonically throughout the discourse, and all constants and properties are equally available for new rule applications. This leads to combinatorial explosion very quickly, and is again certainly unlike the language processing of humans, who forget about old discourse entities that are not kept active by talking about them.

The root cause of both problems is that pure model generation can only serve as a *competence model* for inference-based discourse understanding: It delimits the extent of what can be inferred *at most*. To model human language understanding correctly, however, it is necessary to restrict it to a *performance model*, which computes what people really infer with their limited computational power.

In this paper, we will discuss a way to turn the model generation approach to natural language understanding into a performance model by making it resource-sensitive. Every rule application has certain costs, and we only apply rules as long as the "progress" we make in building a model justifies their costs. This resource-sensitive variant of model generation is embedded into a computational model we call a *tableaux machine*, which gives us high-level control over the expansion process. Natural language understanding in this approach becomes a process of bounded optimization of the hearer's world model; a hearer will try to maximize the quality of his model, as long as the inference costs are not prohibitively high. Thus the omniscience problem is solved.

This general framework has many open parameters, some of which we will instantiate in a sample application to the interpretation of definite descriptions. In our analysis of definite descriptions , we will add a notion of *saliences* of constants to the model generation system. Saliences measure the availability of an entity for reference, and are thus obviously useful in an analysis of definite reference. At the same time, they interact with rule costs in a way that solves the universe explosion problem: While all constants are still available for instantiations, most will be so unsalient that using them will be too expensive in practice.

We will first discuss pure model generation in Section 2. Then we will introduce our general framework in Section 3, and present the instantiation to the definite reference application in Section 4. The conclusion in Section 5 compares our work with other approaches and comprehensively lists open questions, of which there are many.

2 First-Order Tableaux for Refutation and Model Generation

First-order tableaux were originally used as a data structure in refutation procedures. A theorem is proved by decomposing its negation into a tree of possible instantiations (the tableau; see e.g. [8]). 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 [21, 18, 7, 3, 22] build a tableau for the root formula and read the open branches as possible models, showing that it is indeed satisfiable.

$$
\begin{array}{c}
p(a) \lor p(b) \land \forall X. \neg p(X) \\
p(a) \lor p(b) \\
\forall X. \neg p(X) \\
\begin{array}{c|c}
p(a) & p(b) \\
\neg p(Y) & \neg p(Z) \\
*\,[Y = a] & *\,[Z = b]
\end{array} \\
\text{no model}
\end{array}
\qquad\qquad
\begin{array}{c}
(a \land (b \lor \neg c)) \land \neg b \\
(a \land (b \lor \neg c)) \\
\neg b \\
a \\
b \lor \neg c \\
\begin{array}{c|c}
b & \\
\bot & \neg c
\end{array} \\
\text{model } \{a, \neg b, \neg c\}
\end{array}
$$

FIG. 1. Refutation and Model Generation

The two inference procedures only differ in strategy (see Figure 1) and share both the underlying data structure (the tableau) and the propositional inference 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 $c = d^{\mathsf{t}}$.

$$\frac{(\mathbf{A} \wedge \mathbf{B})^{\mathsf{t}}}{\begin{array}{c}\mathbf{A}^{\mathsf{t}}\\ \mathbf{B}^{\mathsf{t}}\end{array}}\mathcal{T}(\wedge) \qquad \frac{(\mathbf{A} \wedge \mathbf{B})^{\mathsf{f}}}{\mathbf{A}^{\mathsf{f}} \mid \mathbf{B}^{\mathsf{f}}}\mathcal{T}(\vee) \qquad \frac{\neg\mathbf{A}^{\mathsf{t}}}{\mathbf{A}^{\mathsf{f}}}\mathcal{T}(\neg) \qquad \frac{\neg\mathbf{A}^{\mathsf{f}}}{\mathbf{A}^{\mathsf{t}}}\mathcal{T}(\neg) \qquad \frac{\begin{array}{c}\mathbf{A}^{\mathsf{t}}\\ \mathbf{A}^{\mathsf{f}}\end{array}}{\bot}\mathcal{T}(\bot)$$

FIG. 2. Propositional Tableau Rules

Definition 2.1 (Tableau Refutation) The tableau refutation calculus $\mathcal{T}$ 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. $\mathcal{T}$ uses a generalized branch closure rule $\mathcal{T}(\bot)$ that allows to close a branch if it contains literals $\mathbf{A}^{\mathsf{t}}$ and $\mathbf{B}^{\mathsf{f}}$ such that there is a most general unifier σ of $\mathbf{A}$ and $\mathbf{B}$. In this situation the rule closes the branch and applies σ to the whole tableau.

$$\frac{(\forall X.\mathbf{A})^{\mathsf{t}} \quad Y \text{ new}}{[Y/X]\mathbf{A}^{\mathsf{t}}}\mathcal{T}(\forall) \qquad \frac{(\forall X.\mathbf{A})^{\mathsf{f}} \quad \mathbf{free}(\mathbf{A}) = \{X, Y^1, \ldots, Y^n\}}{[f(Y^1, \ldots, Y^n)/X]\mathbf{A}^{\mathsf{f}}}\mathcal{T}(\exists)$$

FIG. 3. Free-Variable Quantifier Rules for Theorem Proving

[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.

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 $\mathcal{RM}(\forall)$ rule tests the scope of a universal formula on all members of the Herbrand universe $\mathcal{H}$ of the current branch (i.e. the set of individual constants occurring on it); it must be applied exhaustively to obtain a saturated branch. The $\mathcal{RM}(\exists)$ rule reuses constants that occur in the current branch and alternatively introduces a new constant c^{new}. In this way, $\mathcal{RM}(\exists)$ minimizes the size of the universe and also avoids Skolemization, which would introduce problematic function symbols into the Herbrand universe[3]. 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 $\mathcal{RM}(\forall)$ must be re-instantiated with respect to c^{new}.

$$\frac{(\forall X.\mathbf{A})^{\mathsf{t}} \quad \mathcal{H} = \{a^1, \ldots, a^n\}}{\begin{array}{c}[a^1/X]\mathbf{A}^{\mathsf{t}} \\ \vdots \\ [a^n/X]\mathbf{A}^{\mathsf{t}}\end{array}} \mathcal{RM}(\forall) \qquad \frac{(\forall X.\mathbf{A})^{\mathsf{f}} \quad \mathcal{H} = \{a^1, \ldots, a^n\}}{[a^1/X]\mathbf{A}^{\mathsf{f}} \Big| \ldots \Big| [a^n/X]\mathbf{A}^{\mathsf{f}} \Big| [c^{new}/X]\mathbf{A}^{\mathsf{f}}} \mathcal{RM}(\exists)$$

FIG. 4. Quantifier rules for model generation ($\mathcal{RM}$)

The $\mathcal{RM}$ model generation calculus was originally developed for a certain form of non-monotonic reasoning called minimal entailment [20]. $\mathcal{RM}$ is a refutation complete first-order tableau calculus where each open saturated branch is a Herbrand model. [?] 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.

$\mathcal{RM}$ and other model generation calculi have also been applied to natural language understanding [4, 9, 16]. One popular application area is the interpretation of definite descriptions. Gardent and Konrad [9], for example, 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 $\mathsf{every}(P, Q)$ for

[3]The existence of a function together with an individual constant immediately makes the Herbrand universe (the set of closed terms) infinite

$\forall X.P(X) \Rightarrow Q(X)$. In a sentence like "the rabbit is cute", we obtain the logical form $\mathsf{uniq}(\mathsf{rabbit}) \wedge \mathsf{rabbit} \subseteq \mathsf{cute}$. Due to domain minimality, the $\mathcal{RM}$ calculus generates the model $\{\ldots, \mathsf{rabbit}(c), \mathsf{cute}(c)\}$ in situations with at most one rabbit. The specific form of the $\mathcal{RM}(\exists)$ 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

(2.1) 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.

One of the big advantages of the model-generation approach is that it can make use of 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 (2.1) but no general theory. [12] 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.

In addition, we can observe the universe explosion and omniscience problems mentioned in the introduction. If we try to apply a model generation calculus to e.g. the first few pages of a novel, we will have to introduce a large number of entities into the discourse universe. To saturate a branch, we must instantiate every quantified formula with each such entity, which quickly leads to combinatorial explosion. Domain minimality as in [9] helps for toy discourses, but is no general solution where a text keeps introducing new objects into the discourse universe. On the other hand, an instance of the omniscience problem comes up already if the world knowledge tells us that every human has a father. The model generator has to go off and generate infinite chains of ancestors in order to saturate the tableau. Both types of behavior pose serious computational problems in the search for a model, and do not correspond to the way that humans seem to process language.

3 Towards a Performance Model

We propose to solve these problems in a new framework that combines a variant of the $\mathcal{RM}$ calculus with *resources*. Operational matters will be taken care of by a *tableaux machine*, which operates on tableaux as its central data structure, performs rule applications, and has non-local control over the inference process. This framework can then be instantiated to fit different applications. By way of example, we shall show in Section 4 how it supports the resolution of definite NPs.

3.1 A Refined Calculus

The calculus we use is based on $\mathcal{RM}$, but replaces the quantifier rules from Fig. 4 by the more fine-grained rules in Fig. 5, and is extended by the derived rules shown in Fig. 6.

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

$$\frac{\begin{matrix} a = b \\ \mathbf{A} \end{matrix} \quad a \in \mathcal{W},\ \ b \in \mathcal{H}}{[b/a]\mathbf{A}}\,\mathcal{R}(=) \qquad \frac{a \in \mathcal{H}}{a = a^{\mathsf{t}}}\,\mathcal{R}(\mathit{refl})$$

$$\frac{a, b \in \mathcal{H}}{a = b^{\mathsf{t}} \,\big|\, a \neq b^{\mathsf{t}}}\,\mathcal{R}(id) \qquad \frac{a = b^{\mathsf{t}} \quad a, b \in \mathcal{U}}{\bot}\,\mathcal{R}(una)$$

FIG. 5. RAMG rules

Definition 3.1 (RAMG**: Resource-Adaptive Model Generation)** The calculus RAMG consists of the propositional rules in Figure 2 augmented with the rules in Figure 5.

The key differences between the two calculi are that RAMG atomizes the treatment of quantifiers and does not have a unique name assumption. $\mathcal{RM}$ does not have to treat equality, since the underlying Herbrand semantics regards different constants as necessarily different. In RAMG, we partition the

Herbrand base $\mathcal{H} = \mathcal{U} \cup \mathcal{W}$ into subsets $\mathcal{U}$ for constants with a unique name assumption, and *parameters* $\mathcal{W}$ without (the new witness constants w^{new} from $\mathcal{R}(\exists)$ end up here). In addition, RAMG adds explicit rules for treatment of equality in a tableau. Note that the $\mathcal{R}(=)$ rule (Figure 5) is directional; it only allows the substitution for a constant without the unique name assumption. Finally, $\mathcal{R}(una)$ mechanizes the unique name assumption by allowing a branch to close if two different constants with unique names are claimed to be equal. As a consequence of the introduction of equality into the calculus, the notion of an Herbrand model has to be generalized to that of a canonical model, i.e. a model where the universe is a set of equivalence classes of closed first-order terms. As these models can also be represented by the set of literals on a branch, this change of semantics does not affect our use of model generation as an inference procedure (see [8] for details).

The fact that RAMG uses a rule $\mathcal{R}(\exists)$ for existential quantification that only introduces a new witness constant instead of making a case distinction like the $\mathcal{RM}(\exists)$ rule from $\mathcal{RM}$ does not affect theoretical completeness, since $w^{new} \in \mathcal{W}$ and thus we can apply $\mathcal{R}(id)$ followed by $\mathcal{R}(=)$ (repeatedly in the right branch) to the result to obtain the branches of $\mathcal{RM}(\exists)$. The difference is that $\mathcal{R}(\exists)$ allows a finer-grained (atomized) modeling of the reasoning processes. We will take advantage of this in Section 4, where we study model generation in the presence of salience.

The *derived* rules in Figure 6 are all not theoretically necessary in the calculus since their effects can also be achieved by combinations of the other rules. They are still useful in a resource-sensitive setting, as we shall see below. The rule *R(tnd)* introduces a case distinction over an arbitrary formula by appealing to the law of the excluded middle ("tertium non datur"; it is called a "cut" rule in [8]). The rule $\mathcal{R}(\Rightarrow)$ encapsulates a very common reasoning pattern called *chaining*, which combines existential and universal reasoning.

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

FIG. 6. The chaining rule

3.2 Resources

Now we equip each rule in the RAMG calculus with *rule application costs*: Each application of a rule has certain costs associated with it that have to be "paid" in each of its applications. Application costs allow us to employ a resource-sensitive control regime for the expansion of a tableau. We can, for example, stop expansion once some cost-sensitive termination criterion is met. In addition, we can traverse the search space spanned by the possible rule applications in a more goal-directed way by employing e.g. best-first search. The generation of long chains of ancestors is still possible in principle; but they will become increasingly expensive to compute, so in practice there will always be an upper limit on the chains that are generated at all, and the search strategy will prefer to look at other things first before creating more ancestors than necessary.

The exact selection of rule costs depends on the application. 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 $\mathcal{R}(tnd)$ or $\mathcal{R}(id)$, should be extremely expensive. Cheap operations include conjunction expansion and the existential expansions $\mathcal{R}(\exists)$ and $\mathcal{R}(\Rightarrow)$. We can now see why it is useful to have specialized derived rules. An application of $\mathcal{R}(\Rightarrow)$ does not have the same character of blind guessing as an application of $\mathcal{R}(\forall)$ because all universal variables are instantiated to constants for which the respective antecedent is present on the branch. It is therefore much less important to restrict this rule's application, and it can be cheaper.

3.3 The Tableaux Machine

The process of resource-sensitive tableau expansion in this framework is handled by a *tableaux machine*. A tableaux machine is an automaton whose states are tableaux. In normal operation, the machine will in each step pick a branch and a formula on it to expand, according to some expansion strategy, and perform the expansion. This proceeds until some termination criterion is met, at which point the machine stops.

In addition to this basic processing model, the tableaux machine supports the following operations:

1. *New formulas*: As new sentences come in, their logical form can be added to all branches of the tableau.
2. *Error recovery*: When the current branch closes, the machine can backtrack to some earlier point and explore a different branch to avoid closure.
3. *Theorem-proving sub-tableaux*: Specially marked sub-tableaux can be processed in "theorem-proving mode". A branch with such a subtableau can

only be expanded further once the theorem-proving subtableau closes.

Ultimately – in the spirit of bounded optimization [24] – the machine continues expanding formulas until the costs of all possible rule applications outweigh the expected gain in model quality. For instance, we would assume small models to be "better" than large models, and the machine should spend some of its budget on deriving equalities. The definition of what constitutes a good model, however, is still an open question; first ideas in this direction are discussed in [5, 15].

A simplified version of this approach would be to assume that each sentence in the discourse comes with a fixed budget that the machine can use to perform inferences on it; then computation simply stops when the budget is exhausted. This has the same basic effect of limiting the search space and circumventing the omniscience problem, but it is not exactly what we want. For consider for instance a sentence like "Mary likes a rabbit," and assume that the machine has already established distinct constants for Mary and the rabbit. Now the machine could apply the $\mathcal{R}(\exists)$ rule again to introduce a second rabbit. This would not happen under the bounded optimization regime because it would definitely not improve the model quality. But as long as the fixed budget is not yet exhausted, the machine would happily go off and create more rabbits.

The actual output of the tableaux machine is application dependent; some options are

- the information whether there are models at all – "*no*" might lead to rejection of the discourse;
- the preferred 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.

We indicate sub-tableaux in *theorem-proving mode* by drawing boxes around these sub-tableaux. In this mode, the machine operates exactly like a tableaux theorem prover: It employs the theorem-proving quantifier rules from Figure 3 and the RAMG equality rules and attempts to close all branches. Only if this is possible does it continue to expand the branch that contained the theorem-proving subtableau. For the purposes of this paper, we will ignore resources in theorem proving mode, but maybe it will eventually turn out that resources are useful in the theorem-proving mode as well. This would be easy to accommodate in our model.

A major use of theorem-proving sub-tableaux is to check whether a branch entails a certain formula. The idea here is that a formula φ is entailed by the branch iff $\neg\varphi$ is inconsistent with it. That is, we can simply add a theorem-proving subtableau with root φ^{f}, and expansion of the branch will only continue if φ is really entailed by the branch. This will be very useful in Section 4 to check the uniqueness condition for definite reference.

3.4 The Tableaux Machine as a Performance Model

Now let us look a bit more closely at how the RAMG tableaux machine operates. In the initial state of the machine, the tableau has one branch, which contains the world knowledge and the logical form of an input sentence. We have already said that if we remove the resource limitations, RAMG is a sound and complete calculus for predicate logic. Given infinite resources, it will thus *saturate* the tableau; that is, it will only stop when no inference rule can add new formulas to any branch. In a saturated tableau, all formulas have been expanded and all deductive consequences have been computed, and the literals on the open branches represent full models of the current discourse. We call these models *epistemically closed* (ECMs). ECMs are almost always infinite and therefore cannot be computed in practice. They are nevertheless interesting as a (theoretical) limit case and correspond to a competence model of language understanding in a certain sense.

In actual processing, model search stops before reaching the ECM, so the branches of the tableau only constitute partial potential models of the discourse; we call them *approximative* models (AMs) because they are the closest approximations of ECMs that we can get with finite resources. They are *partial* since there may still be unexpanded formulas or inferences that have not been drawn; we cannot say for certain what tuples must in the denotation of each predicate symbol. Furthermore, they are only *potential* models, since the formulae present in this branch could actually be inconsistent (with the world knowledge), but the machine has not yet managed to deduce an explicit contradiction, which would result in a closed branch and a rejection of the potential model. In other words, the RAMG tableaux machine makes exactly the same tradeoff as a human language user: It can guarantee that the computation terminates, but not that it will find all consequences and inconsistencies. Therefore it can serve as the basis of a performance model of human language understanding – which of course still needs to be instantiated for a specific application.

In particular, our processing model allows us to implement a wide variety of best-first search strategies for models. This opens the possibility of going down a promising computation path which later turns out to be wrong – i.e. exactly what happens in garden path phenomena. The preferred AM which is explored because it is first evaluated as best corresponds to the garden-path interpretation, while the correct interpretation is on one of the branches which aren't explored. If now the preferred branch closes, or further computation becomes prohibitively expensive, the tableaux machine can backtrack and choose to expand the other branch further.

4 Interpreting Definite Noun Phrases

In this section, we present an instantiation of the tableaux machine framework designed to support the resolution of definite noun phrases. The key mechanism we introduce here are *saliences*, which among other things allow us to control the tableau expansion through their interaction with resources. The analysis presented here differs from previous analyses in [12, 9] and heavily exploits the extensions to plain model generation afforded by the resource-sensitive framework.

The account of the interpretation of definite NPs 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. We first introduce the formal machinery, then go through some examples, and ultimately show how to deal with accommodation.

4.1 Adding Saliences to RAMG

In linguistics, the *salience* of an object is a measure for its availability in the discourse universe for (anaphoric or definite) reference. The treatment of salience in RAMG is inspired by theories of anaphoric reference (e.g. [2, 17]): Salience 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.

The function of saliences in our framework will be twofold. For one thing, they will interact with resources, which will provide a way to solve the universe explosion problem. Secondly, they will interact with the resolution of definites more directly, allowing us to resolve definites even when they don't refer uniquely in a strict sense.

We obtain initial saliences by assigning each NP 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[4] in the logical form of the input) and constants in the formula fed to the tableaux machine. For instance, the sentence

[4] For the expository purposes of this paper we will neglect the dynamic differences between the linguistic quantifiers $\neg\forall\neg$ and $\exists$. A full-fledged theory of definites would of course either deal with this or explain why it does not matter.

(4.1) Mary chases a rabbit.

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

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

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, e.g. as in [17].

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 numeric 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 salience function of a level on the right hand column of the tableaux in the examples.

As in [17], each level is initialized with the salience function that maps each term to half the salience it had in the previous level. 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.

Now let us look more closely at the tableau-expansion rules in Figure 5 and extend them with a mechanism for manipulating saliences; the variant rules are shown in Figure 7. The most interesting rule with respect to salience is $\mathcal{R}(\exists)$. This rule introduces a new constant w which we have to assign a salience. Since we have assumed a salience expression S (which is now ground) associated with existential variables, we can just give the new constant the salience $S{\downarrow}$, i.e. we replace the constants in S by their own saliences and evaluate the resulting arithmetic expression. So $\mathcal{R}(\exists)$ would yield the tableau fragment in Figure 8. In this example, the new individual w is introduced with salience 60. Since there is only one sentence, we do not have to distinguish salience levels.

To get an intuition for the full mechanism, let us go through a simple analysis of bridging inferences, as in the small discourse in (4.3).[5] We write $\mathbf{S}^1$ for the semantics of the first sentence (4.4) and simply $\mathbf{S}^2$ for that of the second. To obtain the bridging reference, we will have to use world knowledge such as the fact that all cars have batteries, which we formalize by (4.5).

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

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

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

[5]This analysis oversimplifies in several respects and is only intended here as an illustration of saliences. We discuss a better analysis below.

$$\frac{(\forall X_S.\mathbf{A})^{\mathsf{f}} \quad w^{new} \notin \mathcal{H}}{[w^{new}/X]\mathbf{A}^{\mathsf{f}}}\,\mathcal{R}(\exists)^1 \qquad \frac{a, b \in \mathcal{H} \quad salient}{a = b^{\mathsf{t}} \mid a \neq b^{\mathsf{t}}}\,\mathcal{R}(id)^2$$

$$\frac{c \prec^{\mathsf{t}} c \quad c : n \quad d : m < (1+\delta)n}{\bot}\,\mathcal{R}(\prec^{\mathsf{t}})$$

$$\frac{c \prec^{\mathsf{f}} d \quad c : n \quad d : m \geq (1+\delta)n}{\bot}\,\mathcal{R}(\prec^{\mathsf{f}})$$

$$\frac{\begin{array}{c} \forall X^1.\mathbf{A}^1 \Rightarrow \ldots \forall X^n.\mathbf{A}^n \Rightarrow \exists Y_S.\mathbf{B}^{\mathsf{t}} \\ \mathbf{A}^1(a^1)^{\mathsf{t}} \\ \ldots \\ \mathbf{A}^n(a^1, \ldots, a^n)^{\mathsf{t}} \qquad c \notin \mathcal{H} \end{array}}{[a^i/X^i, c/Y]\mathbf{B}^{\mathsf{t}}}\,\mathcal{R}(\Rightarrow)^3$$

[1] c is introduced with salience $S\downarrow$.
[2] The saliences of a and b are replaced by their sum in the left branch.
[3] c is introduced with salience $S(a^1, \ldots, a^n)\downarrow$.

FIG. 7. Variants of RAMG Rules with Saliences

#	formula	Justification by	from	salience
0	$(\exists X_{60}.\mathsf{rabbit}(X) \wedge \mathsf{chase}(\mathsf{mary}_{100}, X))^{\mathsf{t}}$	*input*		
1	$(\mathsf{rabbit}(w) \wedge \mathsf{chase}(\mathsf{mary}_{100}, w))^{\mathsf{t}}$	$\mathcal{R}(\exists)$	0	$w : 60$
2	$\mathsf{rabbit}(w)^{\mathsf{t}}$	$\mathcal{T}(\wedge)$	1	
3	$\mathsf{chase}(\mathsf{mary}_{100}, w)^{\mathsf{t}}$	$\mathcal{T}(\wedge)$	1	

FIG. 8. A tableau generated by RAMG

Note that the salience expression $X/2$ on the existential variable Y contains the (universal) variable X, since the salience of the battery depends on that of the car. The model generation process will now produce a tableau as in Figure 9.

In the first salience level, we expand $\mathbf{S}^1$. This introduces Mary and her car c into the discourse universe with appropriate saliences. Now we can apply the

#	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	$\mathsf{bat}(d) \wedge \mathsf{part\_of}(d, c)^{\mathsf{t}}$	$\mathcal{R}(\Rightarrow)$	$0, 2$	1	$d : 30$
6	$\mathsf{bat}(d)^{\mathsf{t}}$	$\mathcal{T}(\wedge)$	5	1	
7	$\mathsf{part\_of}(d, c)^{\mathsf{t}}$	$\mathcal{T}(\wedge)$	5	1	
8	${\mathbf{S}^2}^{\mathsf{t}}$	*Given*		2	$c : 30$
				2	$\mathsf{mary} : 50$
	...	...		2	$d : 15$

FIG. 9. A tableau generated by RAMG

chaining rule $\mathcal{R}(\Rightarrow)$ with the $\mathbf{R}$; this introduces the battery d of Mary's car into the universe, with salience 30 (because the car has salience 60). To understand this better, let us think through the derivation summarized in the $\mathcal{R}(\Rightarrow)$ rule. The first step is to instantiate the universal variable X in $\mathbf{R}$ with c, which also gives us the salience expression $c/2$, which can be evaluated, since it is ground now. We do this by replacing c with its salience (60) and obtain the value 30.

Generally, in the $\mathcal{R}(\Rightarrow)$ rule in Figure 7, the witness constant w is introduced with salience $S(a^1, \ldots, a^n)\downarrow$: At the time 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.

But back to our example: line 7 completes the analysis of the first sentence, and we have a battery b that is part of the car c explicitly represented in our model. Now the second salience level is initialized by giving each individual half the salience they had in the first level; As we will see in the next section the battery in Mary's car is still much more salient than any other battery in the world, and the definite reference can succeed.

Note that this argument 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.

The only other rule we have to change to add saliences to add saliences to *RaMG* is the $\mathcal{R}(id)$ rule. As the rule introduces an equality that can be used to eliminate (via $\mathcal{R}(=)$) an individual constant in the left branch, the saliences of a and b are replaced by their sum.

Finally, saliences should have an influence on rule costs: It should be more expensive to expand a formula with low-salience constants than one with highly salient ones. This immediately solves the universe explosion problem. While the discourse universe will contain all objects that have ever been mentioned, most of them will have low salience, and the machine will avoid working with them because this would be too expensive.

In addition, we would like to be able to speak directly about saliences in our calculus. To this end, 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 $m \geq (1 + \delta)n$ for some predefined threshold $\delta \in \mathbb{R}$.[6] 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.

4.2 The τ Connective

For our analysis of definite reference, we introduce a new type of quantified formula into the language of logical forms: $\tau(P, Q, S)$. 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 4.4). 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 10. Note that the boxes around the "$\mathsf{uniq}_{\prec}$" formulas indicate that these subtableaux are in theorem-proving mode.

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

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

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

$$\frac{\tau(P,Q,S)^\alpha \quad \mathcal{H}=\{a^1,\ldots,a^n\}}{\begin{array}{c|c|c} Q(a^1)^{\mathsf{t}} & & Q(a^n)^{\mathsf{t}} \\ \boxed{\mathsf{uniq}_\prec(a^1,Q)^{\mathsf{f}}} \quad P(a^1)^\alpha & \ldots & \boxed{\mathsf{uniq}_\prec(a^n,Q)^{\mathsf{f}}} \quad P(a^n)^\alpha \end{array}} \; \mathcal{R}(\tau)^1$$

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

FIG. 10. The RAMG rule for definite descriptions.

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.[7] 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.

4.3 Simple Definites

Now let us go through some examples to illustrate all this.

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

(4.7) 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 11, 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 11 for $\mathcal{H} = \{\mathsf{a}, \mathsf{b}\}$, and close a subtableau in each branch. This is easy for a: The

[7] We really have to use a conjunction over all members of the universe. A conjunction over all constants c for which Qc is in the current branch is not sufficient because it is possible that we simply have not derived Qc yet, e.g. because we ran out of resources. Inside the theorem-proving subtableau, where resources are ignored, we may still be able to derive Qc. For practical applications, we would of course capture tractable fragments of the inference with unary properties Q by using a sorted logic as e.g. in [?] as basis instead of our unsorted first-order logic, as sorted logics yield more tractable overall inference procedures.

$$\begin{array}{|c|c|}\hline
\begin{array}{c}
\exists X.\mathsf{lstus}(X) \wedge \forall Y.\mathsf{lstus}(Y) \Rightarrow X{=}Y^{\mathsf{t}} \\
\mathsf{lstus}(\mathsf{a})^{\mathsf{t}} \\
\forall Y.\mathsf{lstus}(Y) \Rightarrow \mathsf{a}{=}Y^{\mathsf{t}} \\
\tau(\mathsf{hasror}, \mathsf{lstus}, 100)^{\mathsf{t}} \\
\hline
\begin{array}{c|c}
\begin{array}{c} \mathsf{lstus}(\mathsf{a})^{\mathsf{t}} \\ \boxed{1}\quad \mathsf{hasror}(\mathsf{a})^{\mathsf{t}} \end{array} &
\begin{array}{c} \mathsf{lstus}(\mathsf{b})^{\mathsf{t}} \\ \boxed{2}\quad \mathsf{hasror}(\mathsf{b})^{\mathsf{t}} \\ \bot \end{array}
\end{array}
\end{array} & \mathsf{a}, \mathsf{b} : 100 \\ \hline
\end{array}$$

$$\boxed{1} := \begin{array}{|c|}\hline
\mathsf{uniq}_{\prec}(\mathsf{a}, \mathsf{lstus})^{\mathsf{f}} \\ \hline
\begin{array}{c|c}
\begin{array}{c} \mathsf{lstus}(\mathsf{a}) \Rightarrow \mathsf{a}{=}\mathsf{a} \vee \mathsf{a} \prec \mathsf{a}^{\mathsf{f}} \\ \mathsf{lstus}(\mathsf{a})^{\mathsf{t}} \\ \mathsf{a}{=}\mathsf{a}^{\mathsf{f}} \\ \mathsf{a} \prec \mathsf{a}^{\mathsf{f}} \\ \bot \end{array} &
\begin{array}{c} \mathsf{lstus}(\mathsf{b}) \Rightarrow \mathsf{a}{=}\mathsf{b} \vee \mathsf{b} \prec \mathsf{a}^{\mathsf{f}} \\ \mathsf{lstus}(\mathsf{b})^{\mathsf{t}} \\ \mathsf{a}{=}\mathsf{b}^{\mathsf{f}} \\ \mathsf{b} \prec \mathsf{a}^{\mathsf{f}} \\ \mathsf{a}{=}\mathsf{b}^{\mathsf{t}} \\ \bot \end{array}
\end{array} \\ \hline
\end{array}$$

$$\boxed{2} := \begin{array}{|c|}\hline
\mathsf{uniq}_{\prec}(\mathsf{b}, \mathsf{lstus})^{\mathsf{f}} \\ \hline
\begin{array}{c|c}
\begin{array}{c} \mathsf{lstus}(\mathsf{a}) \Rightarrow \mathsf{b}{=}\mathsf{a} \vee \mathsf{a} \prec \mathsf{b}^{\mathsf{f}} \\ \mathsf{lstus}(\mathsf{a})^{\mathsf{t}} \\ \mathsf{b}{=}\mathsf{a}^{\mathsf{f}} \\ \mathsf{a} \prec \mathsf{b}^{\mathsf{f}} \\ \text{CAN'T CLOSE!} \end{array} &
\begin{array}{c} \mathsf{lstus}(\mathsf{b}) \Rightarrow \mathsf{b}{=}\mathsf{b} \vee \mathsf{b} \prec \mathsf{b}^{\mathsf{f}} \\ \mathsf{lstus}(\mathsf{b})^{\mathsf{t}} \\ \mathsf{b}{=}\mathsf{b}^{\mathsf{f}} \\ \mathsf{b} \prec \mathsf{b}^{\mathsf{f}} \\ \bot \end{array}
\end{array} \\ \hline
\end{array}$$

FIG. 11. The largest state in the US has rich oil reserves.

left branch closes by reflexivity of =, and the right one by application of the uniqueness condition in the world knowledge. However, the b subtableau will not close. Hence, we continue computation on the a branch, where a's salience has now been increased. Note that we cannot recognize in general that we cannot close b's theorem-proving subtableau while expanding the tableau; but as we know beforehand that we will be able to continue computation only on the a or on the b branch, we do not have to worry about b after closing a's subtableau.

A more interesting example involves arguing about salience. The constant js (for "Joe Smith") in the example (4.7) 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. Structures like this are very common in newspaper texts. 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.

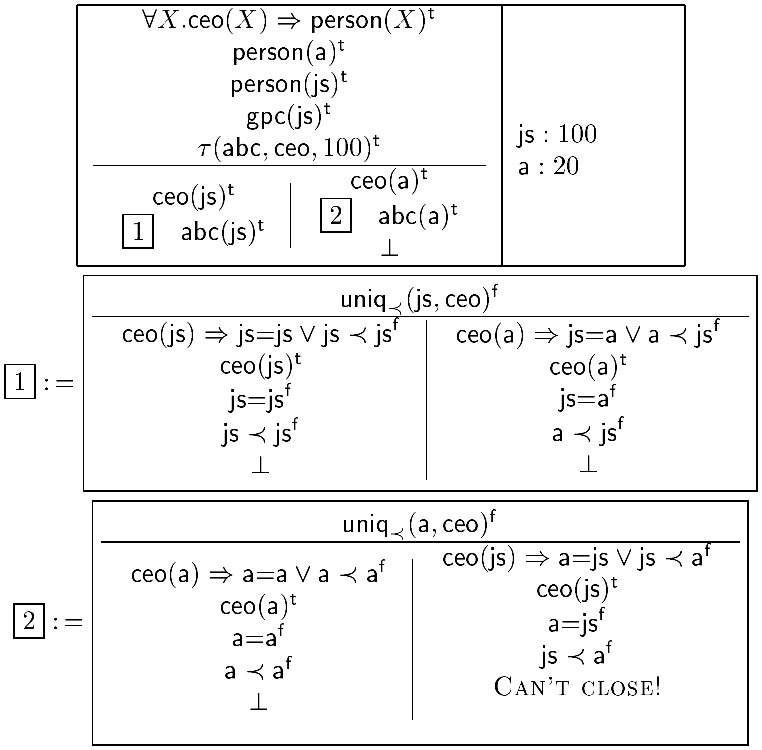

FIG. 12. The CEO of Westinghouse announced budget cuts.

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

(4.8) 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 30), however, we can apply a chaining rule which looks exactly like $\mathcal{R}(\Rightarrow)$, but with τ instead of $\exists$.[8] 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 (4.3) will go through with this analysis without any problem.

4.4 Accommodation

Throughout the discussion so far, we have assumed that the referent for a definite NP is already present in the discourse universe. Indeed, many examples that appear to require accommodation can be analyzed without, by making heavy use of world knowledge. For instance, even the famous

(4.9) The president of Uganda has curly hair.

does not really require accommodation

A less educated reader may not know whether Uganda has a president. However, world knowledge *does* tell us that Uganda (probably) has *some* unique highest political representative a. So we can proceed as with example (4.6) above, simply adding the information that this representative is a president.

But of course there *are* examples that properly require accommodation, e.g. the following sentence:

(4.10) 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. Even if we have derived some potential rabbits from world knowledge, we may still fail because they are of low salience and we will not be able to close any uniqueness subtableau.

In this situation, the "error recovery" process mentioned in Section 3.3 takes place. To evade closure of the top-level tableau, the machine backtracks to 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,

[8] Note that although the chaining rules can logically be derived from the other rules, we can obtain interesting computational behaviour by giving them lower costs.

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.

Note that applications of the $\mathcal{R}(\tau)$ rule have the potential of blowing up the size of the search space tremendously. We avoid this problem, however, by giving this rule prohibitively high application costs. Thus it is never applied in ordinary search; it is only invoked by the special error recovery mechanism if nothing else can help in the expansion of a uniqueness proof.

4.5 Definites and Models

Let us now evaluate this approach to definite descriptions with respect to the ECM/AM distinction. Given the discussion in Section 3.4, it is natural to express the following hypothesis:

> Definites *always* refer in epistemically closed models and are perceived by a hearer as a request for further expansion of the current (approximative) mental model to make them referring as well.

The first part of this claim expresses that the referent of a well-formed definite must already exist in an ECM; ECMs that have been consistent with the discourse so far, but which do not contain an appropriate referent, are discarded. This is exactly what the $\mathcal{R}(\tau)$ inference rule does. In a similar fashion, it discards any ECM in which the definite does not describe a unique entity (up to salience). This means that the ECM universe never grows in understanding a definite.[9]

The same need not be true in any AM (as expressed in the second part of the hypothesis about the control behaviour of definites). The principle is rather that an AM may be consistent with many ECMs, in some of which the reference will be resolved correctly. Let's re-examine our examples above to get a feeling for this. At first, (4.9) seems to call for accommodation; the sentence can be understood even if the hearer had no idea that Uganda is a republic right now. However, there is world knowledge that countries generally have a highest representative, whose type depends on the type of government. So if we do not have previous knowledge about the government of Uganda, there will be an ECM in which Uganda is a republic and has a president, and another ECM in which Uganda is a monarchy and has a king, and a third one in which Uganda has an unspecified witness for highest-representative, all of which are

[9] In contrast, the universe *always* grows in understanding an *in*definite – but we might be justified to collapse the new object with an existing one in the next inference step using the $\mathcal{R}(id)$ rule.

consistent with the discourse. The definite refers in the first and the third ECM, but has no referent in the second one, so this ECM is discarded.

Of course, the definite does *not* refer in the hearer's AM. The definite is perceived as a request for further approximation of the ECM, which performs the inferences that introduce the ECM referent. This is what is usually called accommodation.

In (4.8), we start out with an infinite number of ECMs – one for each n, where n is the number NYC authorities – most of which we don't know the first thing about, except that they have a president (like countries, public authorities tend to have a uniquely determined highest representative). Now whichever ECM we pick, the preferred reading of (4.8) tells us that all of these presidents are crooks. Again, subsequent information that there are exactly, say, twenty NYC authorities will only select the relevant ECMs and close off all other branches.

Finally, the real accommodation example (4.10) seems to break the claim, since we had to apply the law of the excluded middle $P \vee \neg P$ in a case distinction to "artificially create" an antecedent for the definite description. But even though $\mathcal{R}(tnd)$ is extremely expensive in RAMG, and thus is only executed as an error recovery process in the tableau machine (at the AM level), at the ECM level, where resources are infinite there is an ECM that has already applied $\mathcal{R}(tnd)$, where the definite refers (and our hypothesis is saved). In fact it is that ECM that is approximated by the error recovery process in the tableau machine.

5 Conclusion

In this paper, we have presented an approach intended to turn model generation into a performance model for inference-based natural language understanding. It extends current model-generation approaches by rule costs (an idea taken from weighted abduction [12]), and puts the tableau expansion under the control of a tableaux machine, which affords the definition of high-level control strategies. In a sample application to definite reference, we have instantiated this processing model by adding saliences to the tableau. Saliences interact with rule application costs, and are independently interesting for definite reference. Through these extensions to pure model generation, we have solved two of its most pressing problems, which we have termed the *omniscience problem* and the *universe explosion problem.*

Comparison to related work There are many parallels between weighted abduction [12] and our approach: Both employ rule costs, and abduction can be reconstructed in our (deductive) framework as application of the $\mathcal{R}(tnd)$ rule. However, while abduction is a central computational device in [12], we use it only as a last resort because it has the potential to blow up the search space tremendously. A further difference 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. [12] 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 [2, 17], 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 the beginning of a *semantic theory of salience*.

We have exhibited a variety of examples that suggest that the incremental resource-adaptive model generation procedure can serve as a plausible performance model for natural language understanding. While we have focused on technical aspects here, the psycholinguistic literature supports the claim that the model-based analysis is a valid model of human processing. Studies have shown that during discourse comprehension readers or listeners not only represent the logical form of a text but also construct a representation of the states of affairs described by the text, i.e. a representation, the elements of which are mental tokens standing for the referents of linguistic expressions (e.g. [13, 28, 25]). These representations are constructed on-line during discourse comprehension in an incrementally manner (e.g. [1, 6]) they are enriched by a large amount of background knowledge (e.g. [26, 27]) and their major function is to provide the basis for the resolution of anaphora and definite descriptions (e.g. [10, 11, 23]).

Perspectives

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 and a quality measure for (partial) models 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. We have developed a first implementation of the tableaux machine, which will be 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. Finally, we are 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 [24].

On the other hand, several design decisions are only motivated by introspection so far and need to be substantiated (or revised) by real psycholinguistic evidence to obtain a faithful performance model. 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 it would be simple to restore order-independence by employing a strategy called *folding up*, which is well-known in tableau theorem proving [19]. This strategy requires the result of an inference rule to be added as highly in the tableau as possible. 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, a question which even touches upon the interaction of acquiring language and knowledge.

In this paper we have totally ignored the fact that many of the inferences that are performed in the process of natural language understanding are defeasible, even though the incremental variant of model generation we use has a notion of defeasible inference and truth maintenance built into the processing model. In fact, the resource-unlimited tableaux machine described in this paper constitutes a AGM process [**?**] (see [**?**] for details). We conjecture that the resource-bounded model generation process presented in this paper will yield an even more plausible (performance) model for belief change in natural language understanding.

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 (2.1) can be analyzed because Bugs is more salient than Bunny, it is easy to construct variants where our analysis breaks down, e.g. if we add "Bunny is hopping around on the floor", since this would make Bunny more salient than Bugs. 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 the following example, where the CEO should *not* be Joe Smith.

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

It will also be interesting to see whether our analysis of definites can carry over to other types of presupposition.

Finally, we would like to understand better the relation of model generation to dynamic semantics (e.g. DRT and DPL see [29]). 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 [14]) 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 [14] with the approach in this paper. Given such mechanisms, we hope that "uniqueness under scope" examples [9] such as the following sentence can be analyzed.

(5.2) 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. We would like to thank Peter Baumgartner, Aljoscha Burchardt, Markus Egg, Eleni Miltsakaki, Mandy Simons, Stephan Walter, the participants of the ICOS-2 workshop, and the anonymous referees of an earlier version of this paper for helpful discussions, suggestions, criticism, and examples.

References

[1] J. E. Albrecht and O'Brien. Updating a mental model: Maintaining both local and global coherence. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 19:1061–1070, 1993.

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

[3] 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.

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

[5] Aljoscha Burchard and Stephan Walther. BuGS a tableau machine for language understanding. Master's thesis, Computational Linguistics, Saarland University, 2002.

[6] M. de Vega. Backward updating of mental models during continuous reading of narratives. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21:373–385, 1995.

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

[8] Melvin Fitting. *First-Order Logic and Automated Threorem Proving.* Springer Verlag, 1990.

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

[10] S. C. Garrod and T. J. Sanford. Resolving sentences in a discourse context. In M. A. Gernsbacher, editor, *Handbook of Psycholinguistics*, pages 675–698. Academic Press, San Diego, 1994.

[11] A. M. Glenberg, M. Meyer, and K. Lindem. Mental models contribute to foregrounding during text comprehension. *Journal of Memory and Language*, 26:69–83, 1987.

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

[13] Philip N. Johnson-Laird. *Mental Models.* Harvard Univ. Press, Cambridge, Massachusetts, 1983.

[14] 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.

[15] Michael Kohlhase and Mandy Simons. Interpreting negatives in discourse. Technical Report CMU-PHIL-127, Philosophy, Carnegie Mellon University, 2002. submitted.

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

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

[18] 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.

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

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

[21] 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.

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

[23] M. Rinck and G. H. Bower. Anaphora resolution and the focus of attention in situation models. *Journal of Memory and Language*, 34:110–131, 1995.

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

[25] A. J. Sanford and S. C. Garrod. *Understanding written language*. John Wiley & Sons, Chichester, 1981.

[26] T. J. Sanford and S. C. Garrod. Selective processing in text understanding. In M. A. Gernsbacher, editor, *Handbook of Psycholinguistics*, pages 699–719. Academic Press, San Diego, 1994.

[27] M. Singer. Discourse inference processes. In M. A. Gernsbacher, editor, *Handbook of Psycholinguistics*, pages 479–515. Academic Press, San Diego, 1994.

[28] T. A. van Dijk and W. Kintsch. *Strategies of Discourse Comprehension*. Academic Press, Orlando, Florida, 1983.

[29] 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.

Received 31 August 2002