Cognitive Modeling for Formal Semantics: The Organization of DRSs in Declarative Memory*

Adrian Brasoveanu[1] and Jakub Dotlačil[2]

[1] Linguistics, UC Santa Cruz, Santa Cruz, CA, USA
abrsvn@gmail.com, abrsvn@ucsc.edu
[2] University of Utrecht, Utrecht, The Netherlands
j.dotlacil@gmail.com

Abstract

One of the key and still outstanding challenges for formal semantics is "how to build formal semantics into real-time processing models – whether psychological or computational – that involve the integration of linguistic and not-specifically linguistic knowledge." [11] In this paper, we outline the structure of a cognitively realistic semantic processor that we have fully implemented elsewhere [5]. The processor is able to incrementally construct DRT semantic representations in response to linguistic stimuli. We show how this semantic parser can be used to account for the reaction time (RT) data in the fan experiment reported in [4], which investigates how propositional information of the kind encoded by atomic DRSs is (stored and) retrieved from declarative memory. We provide the first (to our knowledge) fully implemented and cognitively realistic model of incremental interpretation and semantic evaluation that is systematically informed by formal semantics and can be used to model RT data.

1 Introduction

One of the key and still outstanding challenges for formal semantics is "how to build formal semantics into real-time processing models – whether psychological or computational – that involve the integration of linguistic and not-specifically linguistic knowledge." [11, 4]) In this paper, we outline the structure of a cognitively realistic semantic processor – that is, a parser that incrementally constructs semantic representations in response to linguistic stimuli of the kind presented in self-paced reading or eye-tracking experiments. The parser has been implemented in [5].

Our choice for a processing-friendly semantics framework is Discourse Representation Theory (DRT) [9, 10]. We chose DRT because atomic DRSs and the compositional construction principles used to build them provide meaning representations and elementary compositional operations that are well understood mathematically, widely used in formal semantics, and can simultaneously function as both meaning representations and models (at least when we restrict ourselves to persistent DRSs [10, 96-97]). Thus, atomic DRSs can be thought of as mental models in the sense of [7, 8].

Due to the incremental nature of the semantic parser and its incremental-interpretation friendly cognitive components, the parser can be used to predict reaction time (RT) data. In particular, we show how it can model time latencies in the fan experiment reported in [4]. This experiment investigates how basic propositional information of the kind encoded by

*We would like to thank the Amsterdam Colloquium 2019 anonymous reviewers and the reviewers of [5], on which this paper builds. Jakub Dotlačil was supported by the NWO grant VC.GW17.112. The usual disclaimers apply.

atomic DRSs is stored and retrieved from memory, which is an essential component of real-time semantic interpretation in at least two respects: (*i*) incrementally processing semantic representations involves composing and integrating semantic representations introduced by new sentences or new parts of a sentence with semantic representations of the previous discourse; (*ii*) incremental interpretation also involves evaluating new semantic representations relative to our mental model of the world, and integrating their content into our world knowledge database stored in declarative memory. We provide the first (to our knowledge) cognitively realistic model of incremental interpretation and truth evaluation that is systematically informed by formal semantics and can be used to model RT data.

2 The fan effect: three generalizations

The fan effect "refers to the phenomenon that, as participants study more facts about a particular concept, their time to retrieve a particular fact about that concept increases." ([3, 186]). The original experiment in [4] demonstrated the fan effect in recognition memory. Participants studied 26 facts about people being in various locations, ten of which are exemplified in (1).

(1) **a.** A lawyer is in a cave. **b.** A debutante is in a bank. **c.** A doctor is in a bank.
d. A doctor is in a shop. **e.** A captain is in a church. **f.** A captain is in a park.
g. A fireman is in a park. **h.** A hippie is in a park. **i.** A hippie is in a church.
j. A hippie is in a town.

In the training part of the experiment, participants committed 26 items of this kind to memory. In the test part of the experiment, participants were presented with a series of sentences, some of which they had studied in the training part (targets) and some of which were novel (foils). They had to recognize the targets and had to reject the foils, which were novel combinations of the same people and locations.

The ten items in (1) form a minimal network of facts that instantiates the 9 experimental conditions in [4]. Each condition uniquely specifies how often a person or location concept has been used in (1). In case a person concept has been used only once (in one propositional fact), we say it has a fan of 1, if it has been used twice, we say it has a fan of 2 etc. In (1), and in the experiment as a whole, person or location concepts can have a fan of 1, 2 or 3. For example, *lawyer* and *cave* have a fan of 1, *captain* and *bank* have a fan of 2, and *hippie* and *park* have a fan of 3. The mean RTs for target recognition and foil rejection in this fan experiment, measured in seconds (s), are provided in Table 1 (from [3, 187]).

Several generalizations become apparent based on this data and various follow-up experiments, as discussed in [3]. **Generalization (i)**: averaging over targets and foils, the effect of 1-fan (both person and location) was about 1.2 s and increased by about 50 ms for each additional fan. **Generalization (ii)** (the min effect): retrieval latency is a function of the minimum fan associated with a probe, e.g., participants tend to respond more slowly to the 2-2 fan items than to the 1-3 or 3-1 items. **Generalization (iii)**: the fan effects are approximately equal for targets and foils, suggesting that foil rejection is not done by a serial (possibly exhaustive) search of the facts one knows about a concept.

3 The account in a nutshell: DRSs in declarative memory

We can reformulate the notion of fan as a relation between the main DRS contributed by a sentence and the sub-DRSs contributed by its three parts: the person indefinite, the location

Target RTs		location fan **1**	**2**	**3**
person fan	**1**	1.11	1.17	1.15
	2	1.17	1.20	1.23
	3	1.22	1.22	1.36

Foil RTs		location fan **1**	**2**	**3**
person fan	**1**	1.20	1.25	1.26
	2	1.22	1.36	1.47
	3	1.26	1.29	1.47

Table 1: The mean RTs (in seconds) for target recognition and foil rejection in the fan experiment in [4], as reported in [3].

indefinite, and the relational predicate *in*. Consider the fan example in (1-j). The DRSs (meaning representations) of the three major components of the sentence – the indefinites *a hippie* and *a town* and the binary predicate *in* – are composed / combined together to form the DRS / meaning representation for the full sentence. The content of *A hippie is in a town* ends up represented in memory as the attribute-value matrix (AVM) below: the DREF attribute specifies the new discourse referent (dref) introduced by indefinites, PRED specifies the predicate, and ARG1 and ARG2 specify which drefs fill the first and second argument slots.

$${}_{\text{MAIN-DRS}}\left[\begin{array}{ll} \text{SUB-DRS}_1: & \left[\begin{array}{ll}\text{DREF:} & 1\\ \text{PRED} & \text{hippie}\\ \text{ARG1:} & 1\end{array}\right] \\ \text{SUB-DRS}_2: & \left[\begin{array}{ll}\text{DREF:} & 2\\ \text{PRED} & \text{town}\\ \text{ARG1:} & 2\end{array}\right] \\ \text{SUB-DRS}_3: & \left[\begin{array}{ll}\text{PRED} & \text{in}\\ \text{ARG1:} & 1\\ \text{ARG2:} & 2\end{array}\right] \end{array}\right]$$

This partitioning into 3 sub-DRSs matches the basic compositional skeleton generally assumed in the formal semantics literature for this type of sentences, as well as the real-time incremental comprehension process the ACT-R cognitive architecture [1, 2] imposes on us. ACT-R, a widely used architecture that includes an explicit model of declarative memory, among other things, provides the cognitive infrastructure for our incremental semantic processor / parser.

The evaluation of the sentence as true (target) or false (foil) is accomplished by recalling from declarative memory the DRS with the highest activation, and checking whether the DRS incrementally constructed for the current sentence matches the components (sub-DRSs) of the DRS retrieved from memory. In principle, any DRS could be recalled from memory, but sub-DRSs in the currently constructed DRS spread activation to the same sub-DRSs in declarative memory, increasing the activation of the main DRS with matching information.

[3, 188-189] account for these generalizations in ACT-R. **Generalization (iii)** above is captured because memory search in ACT-R is not serial. **Generalizations (i) and (ii)** follow from properties of spreading activation. More specifically, in ACT-R, the total activation A_i of a fact/main DRS (a.k.a. chunk) i like the AVM above is:

$$A_i = B_i + \sum_j W_j S_{ji} \tag{1}$$

The base activation B_i in eq. (1) is determined by the history of usage of fact i: how many times i was retrieved and how much time elapsed since each of those retrievals. The spreading activation component $\sum_j W_j S_{ji}$ boosts the activation of fact i based on the concepts j that fact i is associated with, e.g., the person and location concepts/sub-DRSs in the AVM above.

For each concept j, W_j is the extra activation flowing from concept j, and S_{ji} is the strength of the connection between concept j and fact i that modulates how much of the extra activation W_j actually ends up flowing to fact i. Source activation W (for simplicity, let's assume it is the same for all concepts j and drop the subscript) is inversely proportional to the number of concepts j in fact i. For example, $W = 1/3$ for a fact/main DRS with 3 concepts/sub-DRSs.

The account of the fan effect crucially relies on the strengths of association S_{ji} between concepts j and facts i. The strengths of association are taken to be:

$$S_{ji} = S - \log(\text{fan}_j) \tag{2}$$

In eq. (2), S is a constant (baseline strength) to be estimated for specific experiments, and fan_j is the fan of concept j. Thus, strength of association, and therefore fact activation, decreases as a logarithmic function of concept fan (see Chapter 8 in [5] for a detailed justification).

The time T it takes to retrieve fact i from memory, i.e., its retrieval latency, is a function of fact activation A_i, as specified in eq. (3), where F, the latency factor, is a free parameter.

$$T = Fe^{-A_i} \tag{3}$$

In an experiment, we usually do not observe the retrieval time directly, rather, we observe the latency in carrying out the task. We can represent this latency as T_{total}, such that $T_{total} = I + T$, where I is an intercept, i.e., the time needed to carry out cognitive tasks other than retrieval from declarative memory. We can simplify T_{total} as follows ():

$$T_{total} = I + F' \prod_j \text{fan}_j^W, \qquad \text{where } F' = Fe^{-B_i - S} \tag{4}$$

This latency equation shows how the ACT-R model captures the fan effect, i.e., **Generalization (i)** that recognition latency increases with fan. The retrieval latency for an AVM like the one above is: $T_{total} = I + F'(\text{fan}_{person} \cdot \text{fan}_{location} \cdot \text{fan}_{in})^{\frac{1}{3}} = I + F' \sqrt[3]{\text{fan}_{person} \cdot \text{fan}_{location} \cdot \text{fan}_{in}}$. The predicate *in* is connected to all the sentences/facts, so it has a constant fan and contributes a constant amount of latency across all conditions in the experiment. But the person and location fan values fan_{person} and $\text{fan}_{location}$ are manipulated in the experiment, and the equation correctly predicts that as they increase, the corresponding latency increases.

This equation also provides an account of **Generalization (ii)** (the min effect). The reason is that the product of a set of numbers with a constant sum, specifically the product $\text{fan}_{person} \cdot \text{fan}_{location}$, is maximal when the numbers are equal, e.g., $2 \times 2 > 3 \times 1$.

The remainder of the paper discusses in more detail how DRSs like the AVM above are constructed incrementally by means of production rules stored in procedural memory (Section §4) and how semantic (truth) evaluation is modeled as DRS/mental model retrieval from declarative memory (Section §5). In Section §6, we show how the complete cognitive model is fit to the fan-experiment data by embedding it into a Bayesian model, for which it provides the likelihood function; we also compare the posterior predictions of the Bayes+ACT-R model with the RTs observed in the fan experiment and discuss some limitations of the model. Finally, we outline some prospects for cognitive modeling in formal semantics (Section §7).

4 Incremental DRS construction and procedural memory

The AVM in Sect. 3 is an ACT-R friendly representation of the DRS $[x, y|\text{hippie}(x), \text{town}(y), \text{in}(x, y)]$, and we will assume here that the intuitive correspondence between these alternative representations is obvious (see Chapter 8 in [5] for a much more detailed discussion).

With these kind of representations in place, we can move to the core component of our cognitive model, which is an eager left-corner parser that parses syntactic trees and DRSs simultaneously and in parallel. As a new word is incrementally read in the usual left-to-right order for English, the model eagerly and predictively builds as much of the syntactic and semantic representation as it can, before deciding to move the eyes to the next word. This process of syntactic and semantic representation building is the process of comprehending the new word and integrating it into the currently available, partial syntactic and semantic structure. The comprehension and integration result in a new partial syntactic and semantic structure, which, in turn, provides the context relative to which the next word is interpreted.

The overall dynamics of parsing is thus very similar to dynamic semantics: a sequence of parsing actions or a sequence of dynamic updates charts a path through the space of information states. Info states serve a dual function: they provide the context relative to which a semantic update is executed, and they encode the result of executing that update. Information states are simpler for dynamic semantics: they are basically variable assignments or similar structures. For our ACT-R incremental parser, information states are complex entities consisting of the partial syntactic and DRS structures built up to that point, as well as the state of the buffers and modules of the ACT-R mind at that point in the comprehension process.

Implicit in the above description of the parsing process is the fact that an ACT-R mind is composed of modules, which include declarative and procedural memory, but also visual and motor modules etc. Modules are not directly accessible: they can only be accessed through associated buffers, e.g., the retrieval buffer is associated with declarative memory. Buffers serve a dual purpose: individually, they provide the input/output interface to specific modules; as a whole, however, buffers represent the current cognitive state of the mind. Crucially, productions fire based on the current cognitive state, i.e., conditioned on the contents of various buffers. The ACT-R architecture constrains cognitive behavior in various ways, two of which are that buffers can hold only one chunk (AVM), and only one production can fire at any given time.

Production rules are condition-action pairs stored in procedural memory. These rules trigger a cognitive action if the cognitive context (the buffers) satisfy a range of conditions. Let us see some of these rules in action by showing how our model incrementally interprets our running example *A hippie is in a town*. Assume that the first word *A* has already been read and recognized as a determiner. At that point, the PROJECT `NP` rule below is selected and fired:

PROJECT: `NP ==> Det N`

conditions			$\Longrightarrow$	actions		
goal>	TASK:	parsing		goal>	TASK:	move_peg
	STACK1:	S			STACK1:	N
	DREF_PEG:	=peg			STACK2:	NP
					STACK3:	S
retrieval>	ISA:	word				
	CAT:	Det		+imaginal>	ISA:	parse_state
					NODE_CAT:	NP
					DAUGHTER1:	Det
				+context>	ISA:	drs
					DREF:	=peg
					ARG1:	=peg

The conditions of this rule involve the goal buffer, whose contents drive the cognitive process, and the retrieval buffer, which is the interface to the declarative memory module. The lexical entry for the determiner *a* has to be available in the retrieval buffer, and our current top parsing goal is to parse a sentence (S). Furthermore, the rule assumes that a fresh discourse referent index, a.k.a. DREF_PEG, is available in the goal buffer. The value of the DREF_PEG attribute is assigned to an ACT-R variable =peg (in this context, '=' indicates that we're dealing with a variable). The variable =peg (which stands for a specific value, 1 in this case) will be used to specify the cognitive actions triggered by this rule. The index =peg is fresh in the sense that no discourse referent with that index was introduced in the semantic representation up to this point (it was never the argument of a predicate, it couldn't have served as the antecedent for a pronoun etc.) The term 'peg' and its specific usage here originates in [12].

Assuming its preconditions are satisfied, the rule triggers several actions. Most importantly, it simultaneously builds a syntactic representation in the imaginal buffer and a semantic representation in the (discourse) context buffer. The '+' sign (+imaginal, +context) indicates that new representations are added to these buffers. In the imaginal buffer (a standard buffer in ACT-R that maintains an internal representation of the current stage of the cognitive process), we build the unary branching node NP dominating the Det node (which in turn dominates the word *A*). In the context buffer, we start a new DRS that will be further specified by the upcoming N (*hippie*). This DRS introduces a new discourse referent with the index =peg and requires this discourse referent to be the first argument (ARG1) of the still-unspecified predicate contributed by the upcoming N.

The context buffer is not a standard buffer in ACT-R. We postulate it in our model for pedagogical purposes, so that we can neatly separate the syntactic and semantic components of our parse states and be able to focus on the semantic aspects of the incremental interpretation process. The DRS contributed by the PROJECT `NP` rule to the context buffer is provided below. Note that the value of the =peg index is specified as 1, since this is the first discourse referent introduced in the sentence.

AVM format:

ISA:	drs
DREF:	1
ARG1:	1

DRS format:

1
STILL-UNSPECIFIED-PREDICATE(1)

In addition to adding chunks to the imaginal and (discourse) context buffers, the PROJECT `NP` production rule also specifies a new task in the goal buffer, which is to move_peg. This will trigger a rule that advances the 'fresh discourse referent' index to the next number – in our case, it will advance it to 2. This way, we ensure that we have a fresh discourse referent available for any future expression that might introduce a new discourse referent, e.g., another indefinite.

We also have a new stack of expected syntactic categories in the goal buffer: we first expect an N, at which point we will be able to complete the subject NP. Once that is completed, we can return to our overarching goal of parsing an S.

Once the discourse referent peg is advanced, a sequence of rules fires that leads to reading the next word, namely *hippie*, and retrieving its lexical entry from declarative memory. At that point, the PROJECT AND COMPLETE `N` rule below is fired.

PROJECT AND COMPLETE: `N`

	conditions				actions	
goal>	STACK1:	N		goal>	STACK1:	NP
	STACK2:	NP			STACK2:	S
	STACK3:	S				
				+imaginal>	ISA:	parse_state
retrieval>	ISA:	word	$\Longrightarrow$		NODE_CAT:	NP
	CAT:	N			DAUGHTER1:	Det
	PRED:	=p			DAUGHTER2:	N
				context>	ISA:	drs
					PRED:	=p

This rule requires the lexical entry for an N (*hippie*, in our case) to be in the retrieval buffer. It also requires the top of the goal stack, that is, the most immediate syntactic expectation, to be N, followed by NP and S in the subsequent stack positions.

Once these preconditions are satisfied, the rule triggers several actions. First, N is popped off the goal stack. Second, a new part of the tree is added to the imaginal buffer: the binary branching node NP with two daughters, a Det on the left branch and an N on the right. Finally, the DRS in the context buffer is updated with a specification of the PRED attribute: the new value is the predicate =p, contributed by the word *hippie* in the retrieval buffer.

After this rule fires, the DRS in the context buffer becomes SUB-DRS$_1$ of the MAIN-DRS for the sentence *A hippie is in a town*:

AVM format:

$$\begin{bmatrix} \textsc{isa:} & \text{drs} \\ \textsc{dref:} & 1 \\ \textsc{pred} & \text{hippie} \\ \textsc{arg1:} & 1 \end{bmatrix}$$

DRS format:

$$\begin{array}{|c|}\hline \mathit{1} \\ \hline \text{hippie}(1) \\ \hline \end{array}$$

At this point, we fire the production rule PROJECT AND COMPLETE `S ==> NP VP` (not listed here; see Chapter 8 in [5] for details), which eagerly discharges the expectations for an NP and an S from the goal stack and replaces them both with a VP expectation. At the same time, a new part of the syntactic tree is built in the imaginal buffer, namely the top node S and its two daughters NP and VP. This rule also performs an important semantic operation: it transfers the discourse referent 1 introduced by the subject NP to the goal buffer, so that it is available as the first argument of the VP we are about to parse.

We assume, for simplicity, that the copula (*is*) is semantically vacuous. After that, the rule PROJECT AND COMPLETE `PP ==> P NP` parses the preposition *in*. Once again, we refer the reader to Chapter 8 of [5] for details. The sub-DRS contributed by the subject NP is cleared from the context buffer and the partially constructed sub-DRS below is placed there:

AVM format:

$$\begin{bmatrix} \textsc{isa:} & \text{drs} \\ \textsc{pred} & \text{in} \\ \textsc{arg1:} & 1 \end{bmatrix}$$

DRS format:

$$\begin{array}{|c|}\hline \\ \hline \text{in}(1,\ \_) \\ \hline \end{array}$$

The fact that the previous sub-DRS is cleared from the context buffer means that we never have the full view of the main DRS we are constructing. During the incremental interpretation process, we hold only parts of it in our cognitive state (working memory). This is a consequence of essential constraints imposed by the ACT-R architecture, which reflect experimentally established constraints on human cognitive processes.

We can now move on to parsing the location NP *a town*. Once the indefinite determiner *a* is read and lexically retrieved, the PROJECT AND COMPLETE `NP ==> Det N` rule is fired, which sets the second argument slot of the predicate *in* to a newly introduced discourse referent 2. The result is SUB-DRS$_3$ of the main DRS we need to construct for our running example:

AVM format:

$$\begin{bmatrix} \textsc{isa:} & \text{drs} \\ \textsc{pred} & \text{in} \\ \textsc{arg1:} & 1 \\ \textsc{arg2:} & 2 \end{bmatrix}$$

DRS format:

$$\begin{array}{|c|}\hline \\ \hline \textsc{in}(1,\ 2) \\ \hline \end{array}$$

We then clear this sub-DRS from the context buffer and start constructing a new DRS in there that, once the subsequent word *town* is parsed, will be SUB-DRS$_2$ of the main DRS for our running example *A hippie is in a town*. This is the final sub-DRS we needed to construct:

AVM format:

$$\begin{bmatrix} \textsc{isa:} & \text{drs} \\ \textsc{dref:} & 2 \\ \textsc{pred:} & \text{town} \\ \textsc{arg1:} & 2 \end{bmatrix}$$

DRS format:

$$\begin{array}{|c|}\hline \mathit{2} \\ \hline \text{town}(2) \\ \hline \end{array}$$

5 Truth evaluation as retrieval from declarative memory

Our model has completely parsed the sentence *A hippie is in a town.* With the DRS for this sentence in hand, we move on to establishing whether the sentence was studied in the training phase or not. To put it differently, the training phase presented a set of facts, i.e., a *model* in formal semantics terms, and now we have to evaluate whether the test sentence is true relative to the model. That is, we view *semantic evaluation as memory retrieval.*

Thus, the bigger picture behind our DRT-based model of the fan effect is that the process of semantic interpretation proceeds in two stages, similar to the way interpretation proceeds in DRT. In the initial stage, we incrementally construct the semantic representation (DRS, mental discourse model) for the sentence to be evaluated. In DRT [10], this first stage involves a step-by-step transformation of a complete syntactic representation of the sentence into a DRS by means of a series of construction-rule applications. In our processing model, this stage consists of applying an eager, left-corner, syntax-and-semantics parser to the current test sentence.

The DRS/mental discourse model that is the result of the first stage encodes constraints that the set of facts/actual model in declarative memory has to satisfy. In simple cases like the one we are considering, the mental model is homomorphic to a fact in declarative memory.

We can therefore move to the second stage, in which we evaluate the truth of this DRS by connecting it to the actual, 'real-world' model, which is our background database of facts stored in declarative memory. In DRT, the second stage involves constructing an embedding function (a partial variable assignment) that verifies the DRS relative to the model. In our processing model, truth/falsity evaluation involves retrieving – or failing to retrieve – a fact from declarative memory that has the same structure as the DRS we have just constructed.

Once again, the full details of the process of semantic evaluation as memory retrieval are provided in Chapter 8 of [5]. The main idea is that the semantic evaluation process sets up a cognitive state that provides the correct configuration for spreading activation from sub-DRSs to main DRSs stored in memory, so that **Generalizations (i)**, **(ii)** and **(iii)** above are captured.

6 Fitting cognitive models to data: Bayes+ACT-R

We fit our cognitive model to the fan-experiment data by embedding the ACT-R-based DRS parser into a Bayesian model. We estimate four model parameters: (*i*) source activation (weight) W; (*ii*) baseline strength of association S; (*iii*) rule firing r, which is the time required for a rule to fire; and, finally, (*iv*) the latency factor F. The structure of the Bayesian model is provided in the left panel of Fig. 1: low-information priors for the parameters are listed at the top and the entire ACT-R model provides the likelihood function, which outputs semantic evaluation (main DRS retrieval) latencies.

The discussion in Section §3 above provides the motivation for estimating the W, S and F parameters. We have also decided to estimate the r parameter, which can be seen as a way to capture the intercept I in eq. (4).

The posterior estimates for these four parameters are provided and discussed in Chapter 8 of [5]. The plot of the posterior predictions of the model is given in the right panel of Fig. 1. The plot compares the RTs predicted by our model (y-axis) against the observed, experimentally-obtained RTs. The blue dots represent the means, the blue whiskers are 95% credible intervals.

The model captures the data well: the dots are close to the red diagonal line, where predicted and observed RTs are identical. There are slight discrepancies between the predictions of the model and the actual observations, which are due to variance in the data that our fan model ignores from the start. For instance, when we inspect the Target sentence RTs in Table 1, we

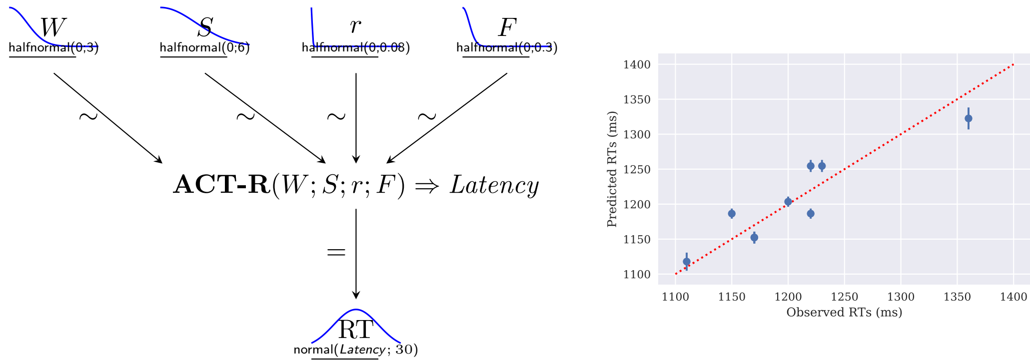

Figure 1: Bayesian model that embeds the ACT-R semantic parser (left-side), and fit of the model to the fan experiment (right-side).

see that the mean RT for targets with a fan of 3 for person and 1 for location is 1.22 s, while the mean RT for targets with a fan of 3 for location and 1 for person is 1.15 s. This difference of 70 ms cannot be captured in our model, which treats the two cases as identical.

7 Prospects for cognitive modeling in formal semantics

We modeled the fan experiment in [4], bringing together the ACT-R account of the fan effect in [3] and formal semantics theories of natural language meaning and interpretation. The resulting incremental semantic processor is the first one to integrate, in a computationally explicit way, dynamic semantics in its DRT incarnation on one hand, and mechanistic processing models formulated within an independently motivated cognitive architecture on the other.

In developing this cognitive model, we argued that the fan effect provides fundamental insights into the memory structures and cognitive processes that underlie semantic interpretation and evaluation, which is the process of determining whether a given sentence is true relative to a database of known facts, i.e., relative to a model in the sense of model-theoretic semantics.

Future directions for this line of research include investigating whether a partial match of known facts is considered good enough for language users in comprehension. This could provide an integrated account of a variety of interpretation-related phenomena, e.g., Moses illusions, the interpretation of plural definites like *The reporters asked questions*, where the sentence is true without every single reporter asking a question, and 'partial presupposition resolution' cases, where part of the presupposition is resolved and part of it is accommodated.

Another direction for future research is reexamining the decisions we made when developing our incremental DRT parser that were not based on cognitive plausibility, but instead were made for pedagogical reasons – in an effort to ensure that the contributions made by semantic theories were still recognizable in the final syntax-and-semantics parser. These decisions led to posterior estimates of the parameters in the model that differ from the previous literature. For example, rule firing time r, traditionally set to 50 ms in ACT-R, was estimated to be 15 ms in our parser.

We also oversimplified the model in various ways, e.g., we initiated the semantic evaluation process (the search for a matching fact in declarative memory) only when the sentence was fully parsed syntactically and semantically. This is unrealistic: parsing, disambiguation and semantic evaluation are most probably interspersed processes, and searches for matching facts/main

DRSs in declarative memory are probably launched eagerly after every parsed sub-DRS, if not even more frequently. See [6] for a similar proposal, and for an argument that such an approach, coupled with a judicious use of spreading activation, might explain the preference to provide given (topic) information earlier in the sentence, and new (focused) information later.

Another oversimplification was the decision to add sub-DRSs to the goal buffer to initiate spreading activation only after the entire sentence was parsed. It is likely that some, possibly all, sub-DRSs are added to the goal buffer and start spreading activation to matching main DRSs as soon as they are parsed. It might even be that such sub-DRSs are added to the goal buffer and start spreading activation even before they are completely parsed. This could be the case for NPs with a relative clause, where the partial sub-DRS obtained by processing the nominal part before the relative clause is added to the goal buffer, and then it is revised once the relative clause is processed.

We have barely scratched the surface with the model introduced in this paper. There are many semantic phenomena for which we have detailed formal semantics theories, but no similarly detailed and formalized theories and models on the processing side. The variety of semantic phenomena to be investigated and the variety of semantic frameworks and processing hypotheses that can be formulated and computationally implemented offer a rich and largely unexplored theoretical, empirical and modeling territory.

References

[1] John R. Anderson. *How can the human mind occur in the physical universe?* Oxford University Press, 2007.

[2] John R. Anderson and Christian Lebiere. *The Atomic Components of Thought*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1998.

[3] John R. Anderson and Lynne M. Reder. The Fan Effect: New Results and New Theories. *Journal of Experimental Psychology: General*, 128(2):186–197, 1999.

[4] John Robert Anderson. Retrieval of propositional information from long-term memory. *Cognitive Psychology*, 6(4):451 – 474, 1974.

[5] Adrian Brasoveanu and Jakub Dotlačil. *Computational Cognitive Modeling and Linguistic Theory*. Language, Cognition, and Mind (LCAM) Series. Springer, Dordrecht, 2020. In production. The current version of the *pyactr* library (Python3 ACT-R) is available here: `https://github.com/jakdot/pyactr`; the frozen version used in the book, together with the code for all the models, is available here: `https://github.com/abrsvn/pyactr-book`.

[6] Raluca Budiu and John R. Anderson. Interpretation-based processing: A unified theory of semantic sentence processing. *Cognitive Science*, 28:1–44, 2004.

[7] Philip N Johnson-Laird. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Harvard University Press, 1983.

[8] Philip N. Johnson-Laird. The history of mental models. *Psychology of reasoning: Theoretical and historical perspectives*, pages 179–212, 2004.

[9] Hans Kamp. A theory of truth and semantic representation. In Jeroen Groenendijk, Theo Janssen, and Martin Stokhof, editors, *Formal Methods in the Study of Language*, pages 277–322. Mathematical Centre Tracts, Amsterdam, 1981.

[10] Hans Kamp and Uwe Reyle. *From Discourse to Logic. Introduction to Model theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht, 1993.

[11] Barbara Partee. The semantics adventure. ms., 2011.

[12] C.F.M. Vermeulen. Merging without mystery: Variables in dynamic semantics. *Journal of Philosophical Logic*, 24:405–450, 1995.