What's at Stack in Discourse

Stefan Kaufmann, Stanford University

Introduction

This is a paper on modal subordination. I present the idea as an extension of the dynamic logic found in Groenenkijk, Stokhof and Veltman 1996 [2], henceforth referred to as GSV. An information state s is a set of possibilities $i \in I$, where a possibility is a pair of a referent system and a possible world. A referent system is a two-step assignment function: Variables point to *pegs*, unique persistent objects which in turn are assigned to objects in the domain of individuals. This setup makes it possible to keep the number of variables that are active at any point in discourse small: A newly introduced discourse referent is associated with a fresh peg, while the variable referring to that peg may be reassigned from its previous use.

The details of the system are laid out in [2, 185–195]. The definition of *support*, through some intermediate notions, states that a state s supports a proposition ϕ iff the state $s[\phi]$ exists and contains descendants of *all* possibilities in s. A descendant $i\prime$ of a possibility i preserves the possible world of i, and its referent system differs from that of i at most in the introduction of new pegs and the assignment of variables to them.

Temporary Contexts

Consider what it takes to interpret the following well-known example from[2][1]:

(1) a. A wolf might come in.
b. It would eat you first.
c. $\Diamond p;^{\mathbf{w}} q$

The procedure is usually assumed to consist of (i) temporarily adding information to the present state and (ii) examining the outcome. Crucially in cases of modal subordination, the auxiliary state derived in the process is available for the immediately following discourse.

In a brief informal discussion of this kind of procedure, GSV talk about such temporary states as being kept "in memory" or "removed from memory" [2, 204]. This is a nice intuitive metaphor. At the same time it demonstrates the need to think of discourse processing as happening to something bigger than an information state in the usual sense, viz. an environment that has "memory" in which entire states can be stored and retrieved. This also requires a way of distinguishing between and referring to states.

In the above example (1), starting out from an initial state s, the two steps are as follows: First, update s with p, thereby obtaining a state $s\prime$ in which p is supported. Second, examine this new state and check that it is non-empty. Keep the temporary state $s\prime$, which supports p, available for further operations. The interpretation of the next proposition, q, then operates on this temporary state, $s\prime$.

The result of processing (1) is a state in which it is known that

(2) a. a wolf might come in ($\Diamond p$) and
b. *if a wolf comes in*, it eats you first ($p \rightarrow q$)

How does the implication in (2)b come about? The interpreter has derived a temporary state $s\prime$ by applying p. In other words, $s\prime$ is the set of *all* possibilities

[1] I use the operator $^{\mathbf{w}}$ as the translation of *would*.

obtained by updating s with p. Next, the interpreter learns that $s\prime$ supports q. According to the definition of *support* given above, this means that *all* the remaining possibilities in $s\prime$ have descendants in $s\prime[q]$. So the two steps combined result in a state in which is known that *every* possibility in $s[p]$ subsists $s[p][q]$. But this is the definition of implication.

This result provides the intuitive foundation of the treatment proposed here. Processing a modally subordinated statement is a normal update on a temporary state; its interpretation as the consequent of an implication follows from the way the states involved are related.

Returning to the above example of the "main" state s and the auxiliary state $s\prime$, the last piece of information to be applied to s, $[p \rightarrow q]$, is indirectly obtained from a statement about the effect of $[q]$ on $s\prime$. In other words, the information that s supports $p \rightarrow q$ is recovered from the information that $s\prime$ supports q. In general, gaining information in one state through the use of an auxiliary state means learning in the former about the latter.

It will be useful to have a way of talking about this process of learning in one state about another. For this we can capitalize on the conception of propositions as functions from states to states, which makes the relations of implication and support interdefineable: In any given state s, information about a proposition ϕ is information about the state obtained by applying ϕ to s. To know in state s that p implies q means to know that $s[p]$ supports q. To know that $\neg p$ means to know that $s[p]$ is the empty state. And so on.

Conversely, to know in state s that state $s\prime$ supports q means to know that any proposition that, when applied to s, yields $s\prime$, implies q.[2] I use this correlation to extend the formalism (where "$s \vdash \psi$" reads "s supports ψ):

(3) $a[s \vdash \psi]b \Leftrightarrow \forall\phi\Big(b[\phi]s \Rightarrow b \vdash (\phi \rightarrow \psi)\Big)$

We can start to implement these notions by representing both contexts, the "real" one (which is assumed to contain the "real" world) and the temporarily derived one, in parallel. When (3)a is applied to an initial state s, the second context $s\prime$ is derived as the result of updating s with p. Next, it is asserted that $s\prime$ also supports q. Finally, this results in the update of the original state with $p \rightarrow q$. In short, the processing of (3) should lead through something like the following steps:

(4)

	$\exists s\prime\ s[p]s\prime$	$[q]$
$\exists s\ Init(s)$	$[\Diamond p]$	$([s\prime \models q])$

This eliminates from the original state s those possibilites incompatible with p, and furthermore, from the remaining set of possibilities those are eliminated which do not persist after the application of p and q. The paranthesized operation in the lower right corner of (4) stands for the resulting implication.

Similar effects are observed in other cases. Consider (5) from Roberts [4]:

(5) a. If John bought a book, he'll be home reading it by now.
b. It'll be a murder mystery.
c. $p \rightarrow q;^{\mathbf{w}} r$

Here the condition embedded under *if* in the first sentence should be kept around for use in interpreting the second. Put in the same tabular form as the previous example, this would lead the interpreter through these steps:

[2] For any two states $s, s\prime$, where $s \leq s\prime$ (in the sense of [2, 188]), there is a class of (compositions of) propositions taking s to $s\prime$. Such propositions need not necessarily correspond straightforwardly to natural-language expressions.

(6)

	$\exists s\prime\ s[p]s\prime$	$[q]$	$[r]$
$\exists s\ Init(s)$		$([s\prime \models q])$	$([s\prime[q] \models r])$

Similarly with negation. Adding to a state s the information that $\neg p$ makes it possible to talk (counterfactually) about what would be the case if p:

(7) a. John doesn't own a Porsche.
b. His wife would hate it.
c. $\neg p;^{\mathbf{w}} q$
d.

	$\exists s\prime\ s[p]s\prime$	$[q]$
$\exists s\ Init(s)$	$[\neg p]$	$([s\prime \models q])$

Temporary states are not available across arbitrary stretches of discourse. Switching to plain indicative closes off modal contexts, i.e., any temporary state is abandoned ("removed from memory", in GSV's terms):

(8) a. If John bought a book, he'll be home reading it by now.
b. John works at a gas station.
c. #It'll be a murder mystery.

Finally, the procedure is recursive. From a temporarily derived context, another one can be derived. Consider another continuation of example (1), this time with one more layer of subordination:

(9) a. A wolf might come in.
b. It would eat you first.
c. It might also open the fridge.
d. It would drink the mango juice.

The first temporary state, $s\prime$, however, must be kept and updated. Suppose that (9) is further continued with (10):

(10) a. It might not open the fridge, though.

Here the interpreter should be able to revert to $s\prime$, i.e., abandoning $s\prime\prime$ should not completely throw him out of the modal context.

Stacks

The formal representation I am proposing was already implicit in the tables of the previous section. I assume that processing operates not on states, but on *stacks* of states. In plain indicative mood, the stack has one element and behaves like the usual state; introducing and abandoning temporary contexts corresponds to pushing and popping, respectively.

Definition 1 (Stacks) *A stack σ is a structure $\langle S, < \rangle$, where $S = \{s_1, \ldots, s_n\}$ is a set of states ordered by the transitive, non-reflexive and antisymmetric relation $<$. I write (σ, s_n) for the stack consisting of $(s_1, \ldots, s_n)$. $\emptyset$ is the empty stack, and (s) is shorthand for $(\emptyset, s)$.*

To refer to operations on such stacks, I introduce a set of operators. The three main ones are for pushing a state ($[\cdot]^{\uparrow}$), making an assertion in a temporary state ($[\cdot]_{\downarrow}$), and popping a state from the stack ($[\triangledown]$). Together with the auxiliary operations $[\cdot]_{\cup}$ and $[\triangledown]!$, these are sufficient for the translation of *if-then* clauses.

Definition 2 (Stack Operations) *The set of stack operators includes the following:*

a.	Assume:	$(\sigma, s)[\phi]^{\uparrow}(\tau, t) \Leftrightarrow \tau = (\sigma, s) \wedge s[\phi]t$
b.	Conclude:	$(\sigma, s)[\phi]_{\downarrow}(\tau, t) \Leftrightarrow \sigma[s \vdash \phi]_{\cup}\tau \wedge s[\phi]t$
c.	Trickle:	$(s)[\phi]_{\cup}(t) \Leftrightarrow s[\phi]t$
		$(\sigma, s)[\phi]_{\cup}(\tau, t) \Leftrightarrow \sigma[\phi]_{\cup}\tau \wedge s[\phi]t$
d.	Pop:	$(\sigma, s)[\bigtriangledown]\sigma$
e.	Popout:	$(s)[\bigtriangledown]!(s)$
		$\sigma[\bigtriangledown]!\tau \Leftrightarrow \sigma[\bigtriangledown] \circ [\bigtriangledown]!\tau$

The "trickle" operator $[\cdot]_{\cup}$ propagates an assertion down the stack if the stack has several elements. It ensures that the bottom and all intermediate elements are updated.

These operations are triggered by certain linguistic expressions:

Definition 3 (Translations) *A translation* $[\![\cdot]\!]$ *maps linguistic expressions to stack operations:*

a. $[\![p]\!] = [\bigtriangledown]! \circ [p]$
b. $[\![\textit{if } p \textit{ then } q]\!] = [p]^{\uparrow} \circ [q]_{\downarrow}$
c. $[\![^{\mathbf{w}}p]\!] = [p]_{\downarrow}$

Existential modality and negation require a slightly different operation. Consider negation: What is stated about the temporary state is an assertion not about a property common to all possibilities it contains, but about its cardinality. Let us call this (meta-level) statement "0" and run the definitions as in (11):

(11) a. $s \vdash 0 \Leftrightarrow s = \emptyset$
b. $[\neg p] = [p \rightarrow 0]$
c. $[\![\text{not } p]\!] = [p]^{\uparrow} \circ [0]_{\downarrow}$

Applying the sequence in (11)c to a state s will indeed result in the elimination of those possibilities in s that have descendants in $s[p]$, but it will also render $s[p]$ useless for any further modalized updates.

To avoid this unwelcome result, negation and possibility are processed as in (12)a, and the corresponding linguistic expressions are translated as in (12)b,c:

(12) a. $(\sigma, s)[p]_{\downarrow\downarrow}(\tau, t) \Leftrightarrow t = s \wedge \sigma[s \vdash p]_{\downarrow}\tau$
b. $[\![\text{not } p]\!] = [p]^{\uparrow} \circ [0]_{\downarrow\downarrow}$
c. $[\![\text{might } p]\!] = [p]^{\uparrow} \circ [1]_{\downarrow\downarrow}$

This has the desired effect of leaving the derived context intact for later retrieval.

A sample dialogue

A simple example will illustrate this and show that the discourse method of keeping track of temporary states is not limited to the examples discussed above. Consider the following simple dialogue between Mary and Beth, arriving at Mary's home late at night:

(13) a. M: My husband isn't home yet.
b. B: How do you know?
c. M: Well, the light would be turned on!

Dialogue is aimed at levelling out differences between the knowledge states of its participants. Here, Mary transfers one fact and one rule from her own state to Beth's.

As they approach Mary's home, they can both see that the light is turned off. Let's call this fact $\neg q$. Next, Mary utters (13)a:

(14) "$\neg p$"

Beth is curious to find out how $\neg p$ follows from $\neg q$. So she asks (13)b, repeated as (15)a:

(15) "How does $\neg p$ follow from what we know?"

Mary's reaction makes sense only if we assume that p is available for modal subordination. Her utterance of (13)c, given here as (16)a, is interpreted by Beth as (16)b:

(16) a. "$\mathbf{w}q$"
b. $p \rightarrow q$
c. $$\frac{\neg q \qquad p \rightarrow q}{\neg p}$$

Mary's utterance of $\neg p$ in (13)a makes p available as an antecedent for the implication $p \rightarrow q$, which is the missing link that Beth needed in order to replicate Mary's modus tollens.

In the framework presented here, Mary's two utterances translate into

(17) $[p]^{\uparrow} \circ [0]_{\downarrow\downarrow}; [q]_{\downarrow}$

This provides Beth with exactly the information she needs: $\neg p$ and $p \rightarrow q$.

Referents

The *wolf* example in GVS in fact allows for two interpretations depending on the relative order of the modal and the existential operator associated with the indefinite NP. A specific reading, in which a particular wolf is referred to, is obtained by processing the existential first. Then the animal is available in the original state, say s, and by inheritance also in s'. On this reading, the discourse in (1) can felicitously be continued with reference to the wolf in the indicative (18):

(18) It looks hungry.

If, however, the order is reversed, then the wolf is present only in the modal context, and the continuation with (18) fails to find an antecedent for the pronoun.

Definites, however, can be introduced in a modal context and remain available. Thus regardless of the position of the existential, the reference in (19) is unproblematic:

(19) a. A wolf might come in.
b. It would eat the pizza$_i$.
c. It$_i$ smells so good.

Such cases of accommodation can be handled by providing different mechanisms at the subsentential level. While an NP of the form *a wolf* is translated into the sequence $[x]; wolf(x)$, which is applied only at the top level of the stack, definites introduce a change to the referent system that is propagated down through the stack: *the pizza* is introduced at every level.

(20) a. indefinites: $[\![\text{a } P]\!] = [x]; P(x)$
b. definites: $[\![\text{the } P]\!] = [[x]; P(x)]_{\cup}$

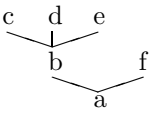
This is, to be sure, only a rough outline. A related idea using stacks of DRSs can be found in [6].

Further Issues

The stack framework is presented here as a way of handling modal expressions involving quantification over dependent domains. It can be put to other uses as well. This is easy to see when considering its relation to other formalisms, especially trees. A sequence of stacks can alternatively be viewed as a sequence of paths in a tree:

(21)

	c	d	e	
	b	b	b	f
	a	a	a	a



In fact, non-accommodational treatments of modal subordination in DRT and its derivatives (cf. [1]) typically build the equivalent of tree structures. However, at any point in a discourse, the stack representation contains no more than one path in the tree. The tree that would correspond to a discourse is therefore not an essential level of representation, but epiphenomenal, showing merely the *history* of the processing.

Furthermore, by virtue of the fact that every newly added piece of information is immediately propagated down the stack, this formalism provides a more realistic picture of the incremental nature of discourse processing than representations in which the result of the processing is available only after some larger structure is completed. This holds also for "Zeinstra's Logic" (cf. [3]), which otherwise has some similarity to this framework.

Other phenomena that can be analyzed in terms of trees would lend themselves to this more dynamic stack treatment. For instance, the interpretation of events with respect to temporal inclusion and precedence has been dealt with in [5] in terms of "Dynamic Aspect Trees". Another application would be quantification in the domain of individuals, rather than worlds. But this is not the right place to pursue these extensions.

References

[1] Anette Frank. *Context Dependence in Modal Constructions.* PhD thesis, IMS, Stuttgart, 1996.

[2] Jeroen Groenendijk, Martin Stokhof, and Frank Veltman. Coreference and modality. In S. Lappin, editor, *The Handbook of Contemporary Semantic Theory*, pages 179–213. Blackwell, Oxford, 1996.

[3] Reinhard Muskens, Johan van Benthem, and Albert Visser. *Dynamics.* Report and Technical Note Series, LP-96-04. ILLC, Amsterdam, 1996.

[4] Craige Roberts. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy*, 12:683–721, 1989.

[5] Alice ter Meulen. *Representing Time in Natural Language.* MIT Press, Cambridge, MA, 1995.

[6] Henk Zeevat. Presupposition and Accommodation in Update Semantics. *Journal of Semantics*, 9:4:379–412, 1992.