Dynamic Context Management

STEFAN KAUFMANN

This paper provides a treatment of *modal subordination*, the interpretation of a string of discourse in a pre-established, but implicit context. The idea is to make a dynamic modal logic sufficiently rich to maintain hypothetical information states across sentences. The interpretation of a sentence as modally subordinated then arises due to the *context* in which it is encountered, rather than the sentence itself or some special processing mechanism.

1 Introduction

The phenomenon of modal subordination has presented a problem for a variety of formal accounts of discourse processing. Well-known examples include (27) and (28) from Roberts (1989):

(27) a. If Edna forgets to fill the birdfeeder, she will feel very bad.
b. The birds will get hungry.

(28) a. If John bought a book, he'll be home reading it by now.
b. It'll be a murder mystery.
c. #It's a murder mystery.

These have in common a first sentence that is understood as "setting the stage", or providing the context, for the interpretation of the second. This dependence of the latter on the former causes a simple, sentence-by-sentence interpretation mechanism either to assign the wrong truth conditions to the sequence as a whole or to predict the sequence to be uninterpretable. (27), for instance, may be true even if the birds do not get hungry. In (28)b, *a book* in the first sentence serves as the

*I am grateful to Stanley Peters, Johan van Benthem, Frank Veltman, David Beaver, Edward Flemming, the audience at the 11th Amsterdam Colloquium and the 7th CSLI Workshop, as well as two anonymous reviewers for helpful discussion and comments.

Formalizing the Dynamics of Information
Martina Faller, Stefan Kaufmann & Marc Pauly (eds.)

antecedent of *it* in the following, even though the former is interpreted as an implication and thus assumed to block outside anaphoric access to its constituents.

Generally in examples like these, letting A, B and C translate into the meaning representations φ, ψ and χ, respectively, an adequate procedure would map expressions like (29)a to translations as in (29)b:

(29) a. "if A then B; will C"
b. $\varphi \to \psi; \varphi \to \chi$

This is only an informal approximation of the desired result, but it is evident that (29) correctly predicts the birds to be hungry *if* Edna forgets to fill the birdfeeder in (27), as well as the anaphoric link between *it* and the book bought by John in (28).

The question that all formal accounts of modal subordination have to answer is how this sharing of a conditional antecedent is brought about and why modally subordinated material is interpreted like the consequent in a conditional. There are two ways of looking at this which differ in a subtle, but important respect: One could assume that the sentences must somehow be interpreted in a special way, for instance by linking them anaphorically among each other or by arranging their representations in a larger structure. Or one could say that the sentences simply mean what they normally mean and that the subordinating effect is due to special properties of the *context* in which they are evaluated.

1.1 Representations

Roberts (1989) discussed—and dismissed—an obvious solution for sentences like those discussed in Section 1, which she dubbed the "insertion approach": allowing for a special mechanism that conjoins the two subordinated clauses in one embedding structure as in (30).

(30) a. "if A then B; will C"
b. $\varphi \to \psi \wedge \chi$

Although (30) does provide an accurate account for examples like (27) and (28), it does not work properly in (31):

(31) a. A thief might come in.
might φ
b. He would take the silver.
would ψ
c. $\Diamond(\varphi \wedge \psi)$
d. $(\Diamond\varphi) \wedge (\varphi \to \psi)$

(31)c, obtained by "insertion", predicts the discourse to hold true even of cases in which a thief breaks in and does not take the silver, which is ruled out by the English mini-discourse. A more appropriate translation would have been like the one given (31)d.

Roberts' conclusion was that examples like (31) make it impossible to find a straightforward structural account for the phenomenon. Her proposal is to appeal to the notion of *accommodation* (of the missing antecedent) for an explanation: In constructing the representation of the discourse, information is used that is not drawn from the actual linguistic material in a systematic way, but obtained by some extralinguistic reasoning process. Once Roberts postulated accommodation for examples like (31), she generalized the treatment to all the cases, even the ones above where the insertion approach would have sufficed.

To many, invoking accommodation amounts to admitting defeat, and since Roberts' proposal, attempts have been made to take control of the phenomena within structural theories of discourse processing. There is a strong sense that the effects of modal subordination are in fact predictable and systematic, and that they are more closely tied to the structure of the discourse than accommodation would predict.

One recent proposal (Frank, 1996) represents a direction of research that regards modal subordination as essentially anaphoric: The symbols representing occurrences of sentences are labeled and can be referred to by variables of the appropriate type ("context referents" in Frank's L[abeled]DRT). Figure 13.1 shows one of Frank's examples (ex. (79), p. 131), here given as (32). In this notation (defined in section 3.2 of her dissertation), upper-case letters represent context referents, + the *update* relation between DRSs and :: the relation between a context referent and the DRS that is associated with it.

(32) a. If a thief breaks into the house, he will take the silver.
b. If in addition he finds the safe, he will try to open it.

This account allows for multiple uses of information drawn from a single occurrence of a linguistic expression, and it does so in a manner similar to the treatment of pronominal coreference.

It is not clear, however, that modal subordination does indeed have all the properties of anaphora. In particular, it seems to be more *local* than the latter: intervening material "closes off" the modal context.[1]

[1] This claim is a bit strong. Apparent counterexamples are sometimes discussed in which modally subordinated contexts can span across statements in factual mood; see Roberts (1995); Kibble (1998). I cannot discuss these cases at length here, but based on those examples, I would maintain that the intervening material is tolerated only if it is relevant to and plays a particular rhetorical role—generally

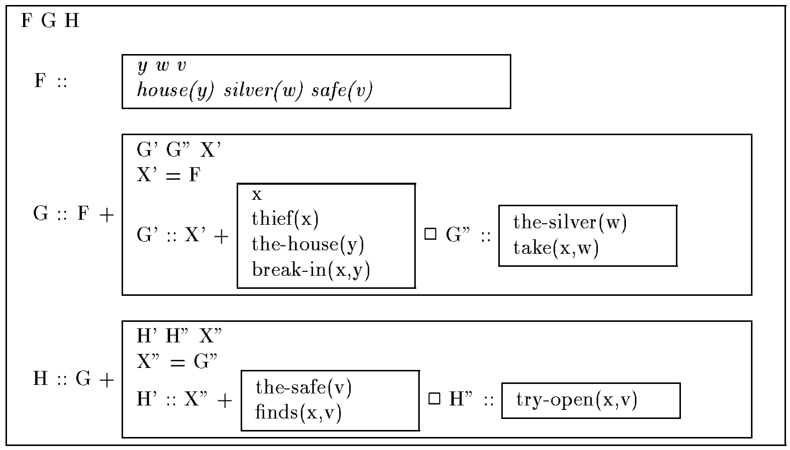


Figure 13.1: Frank-style Labeled DRS for (32)

Nor is it clear that processing a modally subordinated utterance relies on the same kind of procedure of "finding a suitable antecedent" as pronominal expressions.

In the next section I will step back and take a different approach, starting from common intuitions about the way these constructions are understood.

2 Dynamics

I will present the main idea in a simple propositional fragment.

Let a set I of possibilities be given. Possibilities correspond to total truth assignments to atomic propositions. The set of information states S is the powerset of I. Update of information states is interpreted as a function on S:

(33)
$$\begin{aligned} s[\varphi] &= \{i \in s | i(\varphi) = 1\} \text{ for atomic } \varphi \\ s[\varphi; \psi] &= s[\varphi][\psi] \\ s[\neg\varphi] &= s - s[\varphi] \\ s[\varphi \to \psi] &= \{i \in s | i \notin s[\varphi] \text{ or } i \in s[\varphi; \psi]\} \\ s[\Diamond\varphi] &= \{i \in s | \exists i' \in s . i' \in s[\varphi]\}^2 \end{aligned}$$

In this simple system, consistency and support are defined as expected:

explanation—in the conditional that arises from the modal subordination. If that is correct, than the stack formalism could be loosened somewhat to account for those configurations, without abandoning the general right-frontier effect.

[2] The somewhat non-standard definition for $\Diamond\varphi$ needs an explanation. Stating

(35) Let s be an information state and φ a proposition.
a. φ is *consistent* with s iff $s[\varphi] \neq \emptyset$.
b. φ is *supported* by s iff $s[\varphi] = s$.

2.1 Temporary States

Consider what it takes to interpret the following example from Groenendijk et al. (1996, 203):

(36) a. A wolf might come in.
b. It would eat you first.
$\Diamond\varphi$; would ψ

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 operation, Groenendijk et al. (1996, 204) talk about such temporary states as being kept "in memory" or "removed from memory". 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 (36), starting out from an initial state s, the two steps are as follows: First, update s with φ, thereby obtaining a state s' in which φ is supported. Second, examine this new state and check that it is non-empty. Keep the temporary state s', which supports φ, available for further operations. The interpretation of the next proposition ψ then operates on this "secondary" state, s'.

The result of processing (36) is a state in which it is known that

---

that φ is possible in a state s is equivalent to stating that φ is not known to be false. If φ were known to be false, then an update with $\neg\varphi$ would not add any new information, i.e., s would *support* φ (cf. Def. (35).) So the update with $\Diamond\varphi$ is successful if s demonstrably does not support $\neg\varphi$. The condition that there be a possibility in s which would be eliminated by an update with $\neg\varphi$ does precisely that.

The definition for implication is similar to the one in Groenendijk et al. (1996); it can easily be verified that the equality in (34) with an alternative formulation similar to the one in Zeevat (1992) holds.

(34) $$\begin{aligned} s[\varphi \to \psi] &= s[\neg(\varphi; \neg\psi)] \\ &= (s - s[\varphi]) \cup s[\varphi; \psi] \end{aligned}$$

(37) a. a wolf might come in ($\Diamond\varphi$) and
b. *if a wolf comes in,* it eats you first ($\varphi \rightarrow \psi$)

How does the implication in (37)b come about? The interpreter has derived a temporary state s' by applying φ. In other words, s' is the set of *all* possibilities obtained by updating s with φ. Next, the hearer updates s' with ψ. This amounts to ruling out *all* the possibilities in s' where ψ does not hold. But this must have an effect on s as well: The hearer now knows that those possibilities that were removed from s' are not available, so those possibilities of which they were descendants are removed from s.

This result provides the intuitive foundation of the treatment proposed here. Processing a modally subordinated statement is a normal update on a state; its interpretation as the consequent of an implication follows from the fact that that state is an "auxiliary" one and updates on it are reflected in updates on the "main" state.

So the last update, with $[\varphi \rightarrow \psi]$, is indirectly obtained from a statement about the effect of $[\psi]$ on s'. In other words, the information that $\varphi \rightarrow \psi$ holds is recovered from the information that the auxiliary state supports ψ. 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 inter-define the relations of implication and support: 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 φ implies ψ means to know that $s[\varphi]$ supports ψ. To know that $\neg\varphi$ means to know that $s[\varphi]$ is the empty state. And finally, to know that $\Diamond\varphi$ means to know that $s[\varphi]$ is non-empty.[3]

Conversely, to know in state s that state t supports ψ means to know that any proposition that, when applied to s, yields t, implies ψ. We can use this correspondence to define a helpful piece of notation. Given two states s and t and a proposition $\varphi_{s,t}$ such that $s[\varphi_{s,t}]t$, we can define an expression $s[t \vdash \psi]$ which may be read as "learning in state s that state t supports ψ" as in (38)a and obtain the equivalence in (38)b:

(38) a. $s[t \vdash \psi] = \{i \in s \mid i \notin t \text{ or } i \in t[\psi]\}$
b. $s[t \vdash \psi] = s[\varphi_{s,t} \rightarrow \psi]$[4]

[3]Note that the definitions given in (33) for the relevant formulae in fact encode this intuition and even mention $s[\varphi]$ explicitly.

Note that since this paper is only dealing with monotonic updates, if there is a proposition $\varphi_{s,t}$ as defined, we can be sure that $t \subseteq s$. To facilitate a uniform handling of implication and negation, the latter can also be rewritten in this way. I will use two more symbols to refer to two special propositions. The first, $\bot$, reduces any state to the empty state $\emptyset$. Updates involving it are defined in (39)a:

(39) a. $s[\bot] = \emptyset$

b. $t \vdash \bot \Leftrightarrow t = \emptyset$

c. $s[t \vdash \bot] = \{i \in s | i \notin t \text{ or } i \in t[\bot]\}$
$= s[\neg\varphi_{s,t}]$

Thus given the obvious consequence in (39)b, we can define negation as implication as in (39)c. In this definition, since $t[\bot] = \emptyset$ for any t, the latter part of the descriptor becomes trivially false and the result of the update is precisely the set of possibilities in s that do not survive in t, i.e. $s - t$.

Something similar can be defined for the possibility operator $\Diamond$. From the definition of support, we have (40)a; (40)b follows, and (40)c is just for notational convenience.

(40) a. $t \nvdash \bot \Leftrightarrow t \neq \emptyset$

b. $s[t \nvdash \bot] = \{i \in s | \exists i' \in s.i' \in t\}$
$= s[\Diamond\varphi_{s,t}]$

c. $[t \vdash \top] \stackrel{def}{\equiv} [t \nvdash \bot]$

There is still a sense in which the definitions in (39) and (40) are fundamentally different: The former can actually add information to the input state by eliminating some, but not all possibilities. The latter can only return the input itself upon success or the empty state, indicating failure. These different behaviors are a consequence of the set-up of the current system: Since the only operation on states is elimination of possibilities, a statement that t is empty can be made true by eliminating all possibilities that are in t. A statement that t is not empty, on the other hand, cannot be made true if it is false, since there is no way of "putting back" possibilities.

3 Stacks

Consider two information states s and t and a proposition φ such that $s[\varphi]t$. The last section gave a definition of the expression $s[t \vdash \psi]$ as "learning in state s that state t supports ψ", which amounts to "learning in state s that the proposition φ, which leads to t, implies ψ."

This also provides as a special case a way of rewriting the standard expression $s[\varphi \to \psi]$ as $s[s[\varphi] \vdash \psi]$.

Modal subordination is like implication in that it involves "learning in state s that state t supports ψ". It is different in that state t is not explicitly introduced, but assumed to be present in the context, introduced at an earlier point. This can be modeled by keeping both s and t intact and available.

The *context* in which a proposition is interpreted is a *stack* of states, defined, for this example, as the sequence $\langle t, \langle s \rangle \rangle$. Given that a stack element is invariably derived by normal update from the one underneath it and update in the system is a monotonic elimination procedure, the order of the states in the sequence implies that $t \subseteq s$.[5] In such a setting we will call t an *auxiliary* or *temporary* state. Let us call the stack σ.

If we apply a proposition ψ to σ, the output is a stack $\tau = \langle t', \langle s' \rangle \rangle$, where t' and s' are the results of updating t and s, respectively. The update consists of two parts: $t[\psi]t'$ and $s[t \vdash \psi]s'$. The operation on t is a standard update and if it is informative, it will remove some possibilities from t; crucially, these possibilities are removed *along with their ancestors* in s. Both these steps combined have the effect that in t' it is known that ψ and in s' it is known that the proposition that leads to t, viz. φ, implies ψ.

Some notational conventions will make it easier to formulate these procedures for the general case.

Definition 1 (Stacks) *The set of stacks is defined as follows:*

a. *If s is an information state, then $\langle s \rangle$ is a stack.*
b. *If s is an information state and σ is a stack, then $\langle s, \sigma \rangle$ is a stack.*
c. *Nothing else is a stack.*

To simplify the definitions, we can define the cardinality of a stack to be the number of its members and refer to its elements by counting from the top down, thus for a stack $\sigma = \langle s, \langle t \rangle \rangle$, $|\sigma| = 2$, $\sigma_0 = s$ and $\sigma_1 = t$.

Stacks are used in this paper with the assumption that the bottom state represents the indicative, "original" state, while all others are "temporary." An empty stack is not defined, nor can the operations below produce one, since all interpretation by definition takes place in some state.

[5] This is, of course, only true for factual conditionals. Counterfactuals are not dealt with in this paper, although an extension of this formalism to include a semantics of modal base and ordering source in the spirit of Kratzer (1991) should not be too difficult.

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 ($[\cdot]_{\downarrow}$), and popping the top state from the stack ($[\bigtriangledown]$). These are sufficient for the translation of *if-then* clauses.

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

a. Assume: $\langle s, \sigma\rangle\,[\varphi]^{\uparrow}\,\langle s[\varphi], \langle s, \sigma\rangle\rangle$

b. Conclude: $\langle s, \sigma\rangle\,[\varphi]_{\downarrow}\,\langle s[\varphi], \tau\rangle$,
where $|\sigma| = |\tau| = n$
and $\sigma_i[s \vdash \varphi]\tau_i$ for all $i,\ 0 \leqslant i < n$

c. Pop: $\langle s, \sigma\rangle\,[\bigtriangledown] = \begin{cases} \langle s\rangle \text{ if } \sigma = \langle\rangle \\ \sigma \ \text{ otherwise} \end{cases}$

These definitions determine the result of applying the operations on the input stack. $[\varphi]^{\uparrow}$ adds a new element at the top which corresponds to the result of updating s, the top element of the input stack, with φ.

$[\varphi]_{\downarrow}$ updates the top element with φ. As discussed above, this makes the information that φ is implied by the proposition leading to the top element available to all elements further down.

Similarly for the "pop" operator $[\bigtriangledown]$: If there are two or more elements on the stack, it removes the top element. It has no effect if there is only one state left in σ.

It is straightforward to build a procedure that enforces these conditions on the output, for example by recursion on the stack, but that is not the concern of this paper.

3.1 Illustration

We can now use the operators defined above to demonstrate how the logic works, keeping in mind its application to modal subordination. First, implication, commonly translated as in (41)a, can now be redefined as a two-step process of setting up a hypothetical context and making a statement about it, as in (41)b:

(41) a. $s[\varphi \to \psi]$
b. $s\,[\varphi]^{\uparrow}\,[\psi]_{\downarrow}$

The update in (41)a consists in removing those possibilities from s which have descendants in $s[\varphi]$, but not in $s[\varphi][\psi]$. In (41)b, the effect on s is the same, but in addition to that a new state is introduced and remains in the environment. We can visualize this as in (42).

(42) $\langle\ s\ \rangle\ [\varphi]^{\uparrow} \left\langle \begin{array}{c} s[\varphi] \\ s \end{array} \right\rangle [\psi]_{\downarrow} \left\langle \begin{array}{c} s[\varphi][\psi] \\ s[s[\varphi] \vdash \psi] \end{array} \right\rangle$

The result is, as in (41)a, the removal from s of all those possibilities that have descendants in $s[\varphi]$ but not in $s[\varphi][\psi]$. And indeed, if we apply the "pop" operator $[\bigtriangledown]$ to the output state once, the result will be a single state equivalent to the one in (41)a, as the reader may verify. Now, however, the state $s[\varphi][\psi]$ is still in the stack, and subsequent updates will yield different results. Before showing this, it is useful to note the following:

(43) a. If $s[\varphi]t$, $t[\psi]t'$ and $s[t \vdash \psi]s'$, then $s'[\varphi][\psi]t'$.[6]

Now suppose, for instance, that the next update is $[\chi]_{\downarrow}$. Let us abbreviate $s[s[\varphi] \vdash \psi]$ as s' and $s[\varphi][\psi]$ as t'. The stack undergoes the changes shown in (44):

(44) $\left\langle \begin{array}{c} t' \\ s' \end{array} \right\rangle [\chi]_{\downarrow} \left\langle \begin{array}{c} t'[\chi] \\ s'[t' \vdash \chi] \end{array} \right\rangle$

The operation that takes s' to $s'[t' \vdash \chi]$ is the removal of all those possibilties that have descendants in t' but not in $t'[\chi]$. But since $t' = s'[\varphi][\psi]$, this is equivalent to removing all those possibilities that have descendants in $s'[\varphi][\psi] = s'[\varphi;\psi]$, but not in $s'[\varphi][\psi][\chi] = s'[\varphi;\psi][\chi]$. From this and the definition of implication it is obvious that the end result is the equivalent of obtaining $s[t' \vdash \chi]$ by an update with $[(\varphi;\psi) \rightarrow \chi]$.

This last result is what makes this logic suitable for modal subordination. The next section will elaborate on this.

3.2 Other modes

For the treatment of statements expressing possibility, I define a variant of the "Conclude" operator, "Conclude2":

(45) Conclude2: $\langle t, \langle s, \sigma\rangle\rangle\, [\varphi]_{\Downarrow}\, \langle t, \langle s[\varphi], \tau\rangle\rangle$,
where $|\sigma| = |\tau| = n$
and $\sigma_i[s \vdash \varphi]\tau_i$ for all $i,\ 0 \leqslant i < n$

The only difference between the one in Definition 2 and this is that this leaves the top element of the stack untouched. This is necessary because in the current setup an update of the form $s[\top]$ is not defined and cannot easily be defined analogously to the other updates. Note also that $[\cdot]_{\Downarrow}$ is not defined on singleton stacks. One can argue that this is intuitively reasonable, but in the present system it has little significance since the operator is always invoked immediately after a pushing operation (see Section 4 below.)

As an example, consider the update in (46).

$$(46)\quad \langle\ s\ \rangle\ [\varphi]^{\uparrow} \left\langle \begin{array}{c} s[\varphi] \\ s \end{array} \right\rangle [\top]_{\Downarrow} \left\langle \begin{array}{c} s[\varphi] \\ s[s[\varphi] \vdash \top] \end{array} \right\rangle$$

Given the definitions, this returns s in the bottom element just in case $s[\varphi]$ is not empty, and the empty state otherwise. $s[\varphi]$ is then left in the environment for modal subordination. A more concrete example follows below.

4 Application

With the stack operations defined in the previous section, we can now translate proposition-embedding linguistic expressions. For simplicity, I collapse *would* φ and *will* φ and replace them with the expression ${}^{\mathbf{w}}\varphi$, because even though they cannot be used interchangeably in all cases, their meaning under modal subordination is essentially the same.

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

$$\begin{array}{lrcl}
\text{a.} & [\![\varphi]\!] & = & [\bigtriangledown]^{\star}\,[\varphi]_{\downarrow} \\
\text{b.} & [\![\text{anyway}]\!] & = & [\![\text{however}]\!] = \ldots = [\bigtriangledown]^{+} \\
\text{c.} & [\![\text{then } \varphi]\!] & = & [\![{}^{\mathbf{w}}\varphi]\!] = [\varphi]_{\downarrow} \\
\text{d.} & [\![\text{if } \varphi]\!] & = & [\varphi]^{\uparrow} \\
\text{e.} & [\![\text{not } \varphi]\!] & = & [\varphi]^{\uparrow}\,[\bot]_{\Downarrow} \\
\text{f.} & [\![\text{might } \varphi]\!] & = & [\varphi]^{\uparrow}\,[\top]_{\Downarrow}
\end{array}$$

These definitions mostly just reiterate what was developed above. The first two highlight a point where this formalism does not make any predictions: Discourse connectives such as those in the rules, and indeed the simple indicative to some extent, induce the popping of some indefinite number of stack elements. Saying that either exactly one or all but one are popped would be too rigid; it seems, rather, that these items mark junctions at which the whole stack is open to re-evaluation, and the hearer chooses the state to which the following material is most sensibly applied.[7]

[7] How exactly such a choice between candidate states is to be made is an open question at this point; presumably some measure of utility of update or value of information will be needed. This goes beyond the scope of this paper.

To illustrate, consider again the case of example (27), repeated here as (47), and let us represent it as in (47)c. According to the definitions above, this translates into the sequence of stack operations in (47)d:

(47) a. If Edna forgets to fill the birdfeeder, she will feel very bad.
b. The birds will get hungry.
c. if φ then ψ ; $^{\mathbf{w}}\chi$
d. $[\varphi]^{\uparrow}$; $[\psi]_{\downarrow}$; $[\chi]_{\downarrow}$

We have seen above that the problem with this example is that the second sentence does not assert that the birds will get hungry, although it would do so in isolation. Now we see that the sequence of updates in (47)d, which is similar to the example in Section 3.1, does predict the correct inference patterns. Given an initial stack $\langle s \rangle$, we saw in the previous section that s is updated with $[t \vdash \psi]$ and $[t' \vdash \chi]$, where $s[\varphi]t$ and $t[\psi]t'$, and that these updates are equivalent to $[\varphi \rightarrow \psi][(\varphi;\psi) \rightarrow \chi]$. The whole process is depicted below (spread over two lines for readability).

(48) a. $\left\langle\ s\ \right\rangle [\varphi]^{\uparrow} \left\langle \begin{array}{c} s[\varphi] \\ s \end{array} \right\rangle [\psi]_{\downarrow} \left\langle \begin{array}{c} s[\varphi][\psi] \\ s[s[\varphi] \vdash \psi] \end{array} \right\rangle$

b. $\left\langle \begin{array}{c} s[\varphi][\psi] \\ s[s[\varphi] \vdash \psi] \end{array} \right\rangle [\chi]_{\downarrow} \left\langle \begin{array}{c} s[\varphi][\psi][\chi] \\ s[s[\varphi] \vdash \psi][s[\varphi][\psi] \vdash \chi] \end{array} \right\rangle$

Now suppose that s has a possibility i_0 in which Edna fills the birdfeeder and the birds do not get hungry. Clearly, i_0 is not removed by any of the update operations, since both are conditional on φ, viz. that Edna fails to fill the birdfeeder. Therefore the resulting bottom element of the stack, $s[\varphi \rightarrow \psi][(\varphi;\psi) \rightarrow \chi]$, does not support χ (i.e., it does not follow that the birds will get hungry.)

Examples involving possibility are handled as well. Here the "Conclude2" operator $[\cdot]_{\Downarrow}$, whose sole difference from "Conclude1" ($[\cdot]_{\downarrow}$) is that it leaves the top element of the stack unchanged, is put to work. Consider again example (31), repeated here as (49):

(49) a. A thief might come in.
b. He would take the silver.
c. might φ ; $^{\mathbf{w}}\psi$
d. $[\varphi]^{\uparrow}$; $[\top]_{\Downarrow}$; $[\psi]_{\downarrow}$

The choice of the two "Conclude" operators in this order is dictated by the translations in Definition 3 above. According to the definitions in 2, then, this translates into the following sequence:

(50) a. $\left\langle\ s\ \right\rangle\ [\varphi]^{\uparrow} \left\langle \begin{array}{c} s[\varphi] \\ s \end{array} \right\rangle\ [\top]_{\Downarrow} \left\langle \begin{array}{c} s[\varphi] \\ s[s[\varphi] \vdash \top] \end{array} \right\rangle$

b. $\left\langle \begin{array}{c} s[\varphi] \\ s[s[\varphi] \vdash \top] \end{array} \right\rangle\ [\psi]_{\downarrow} \left\langle \begin{array}{c} s[\varphi][\psi] \\ s[s[\varphi] \vdash \top][s[\varphi] \vdash \psi] \end{array} \right\rangle$

The final result of this is a state in which it is known that (i) a thief might come in and (ii) *if a thief comes in*, he takes the silver. This is the desired result.

4.1 Referent systems

Nothing has been said so far about the way a corresponding first-order system would work. A previous version of this paper was implemented fully in the referent-system model of Groenendijk et al. (1996), but as two reviewers rightfully pointed out, that was a complication which contributed little to the main point. The interested reader can verify that by implementing possibilities not as worlds, but as pairs of a referent system and a world, the present system works without much further modification. The only required additional change is that the subset relation that orders the elements in a stack has to be redefined as the relation of *extension:* While updates always decrease the number of *worlds* in the state, an update involving an existential quantifier may increase the number of *possibilities.* For a complete description of the system, see Groenendijk et al. (1996, 185–195).

4.2 An Amendment: xor

It was already pointed out by Roberts (1989, 702–704) that the possible anaphoric relationships between the disjuncts of exclusive disjunction pose a challenge to simple DRT approaches: In (51), it is clear that the pronoun *it* is interpreted as coreferential with the *bathroom* in the first disjunct—yet the latter's embedding under negation would seem to render it unaccessible.

(51) a. Either there is no bathroom in the house, or it's in a funny place.
b. φ xor ψ

Kamp and Reyle (1993, 185–190) agree with Roberts (1989) and provide for this construction an essentially elliptic account, stipulating that the second part of the sentence is given an accommodated copy of the material embedded under the negative operator in the first part—that which is referred to by the implicit *otherwise* in such sentences. More generally, they suggest to "add" (not "copy") the negation of the first disjunct if the latter is not already negated, and if it is negated, to add whatever is embedded under the negation. This rule is, however, only briefly touched upon in the text and not elaborated further.

Frank (1996), in a variation on this theme, proposes a rule that generally makes two instances of the first disjunct available and embeds the second occurrence under negation.These two occurrences are represented by two context-referents. An external condition stipulates that they refer to the same antecedent. This is combined with a second rule cancelling double negation along the lines of Kamp and Reyle (1991) (a later, related treatment can be found in Krahmer and Muskens (1995)). This procedure disembeds, as it were, the copy of the first disjunct and makes the occurrence of *bathroom* in it accessible to the pronoun.

These examples are also related to cases of *split modality* as discussed by Frank (based on earlier discussions by Landman): In a phenomenon that is closely related to typical cases of modal subordination, more than one thread of discourse can be maintained in parallel:

(52) a. You will stay unmarried, or you will marry a tramp.
b. You'll become a nun, or the tramp will beat you regularly.
c. Either way you'll have a miserable life.

It is easy to see intuitively how these examples should be dealt with in the present framework: Where Frank uses discourse referents with the same antecedent and an added "structural/rhetoric constraint" called $parallel(G_1, G_2)$ (G_1, G_2 being context referents), the stack formalism should keep both of the relevant contexts in store for later use, forking off, as it were, two threads of discourse. Implementing this, however, requires a substantial change to the system, viz. adding a new dimension: The stack elements will have to be able to hold more than one state, and those states have to be ordered. Thus a stack element can itself be a sequence of states:

(53) a. $s[\varphi \text{ xor } \psi] = \langle s[\varphi], s[\psi] \rangle$
b. $\langle s, t \rangle [\varphi \text{ xor } \psi] = \langle s[\varphi], t[\psi] \rangle$
c. $[\varphi \text{ xor } \psi]_\downarrow \equiv [\varphi]_\downarrow \text{ xor } [\psi]_\downarrow$

For "bathroom" sentences, assuming the equivalence in (53)c and assuming that the interpreter knows how to make the choice between the operators $[\cdot]_\downarrow$ and $[\cdot]_\Downarrow$, one would obtain (54):

$$(54)\quad \left\langle\ s\ \right\rangle\ [\varphi]^{\uparrow} \left\langle \begin{array}{c} s[\varphi] \\ s \end{array} \right\rangle [\bot \text{ xor } \psi]_{\downarrow} \left\langle \begin{array}{c} s[\varphi] \\ \langle s[s[\varphi] \vdash \bot], s[s[\varphi] \vdash \psi] \rangle \end{array} \right\rangle$$

To be sure, although such an alteration would work for the examples above, it might turn out that as it stands it would be no less stipulative than previous proposals. It remains to be seen whether the framework proposed here can offer a substantial improvement in these cases.

5 Further Issues

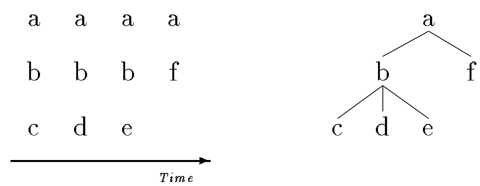
Using stacks in dynamic treatments of discourse is not a new idea. An example is Zeevat's implementation of DRT in a stack-based framework (Zeevat, 1992). The basic idea is very similar to the one pursued here. Zeevat's system pushes auxiliary states during the processing of a sentence and removes them immediately after they are used. The main difference is that in the present system, they stay in place until they are cleared by an explicit additional operation. There is also a more subtle difference in the way implication is encoded: If we used Zeevat's system, the output stack would contain two additional states, one corresponding to the antecedent, the other to the consequent (see the definition in Note 2.) Thus assuming an appropriate definition of the stack operations, and simplifying the representation somewhat, we would have the following picture for a simple update with $\varphi \rightarrow \psi$:

$$(55)\quad \left\langle\ s\ \right\rangle \text{ if } \varphi \left\langle \begin{array}{c} s[\varphi] \\ s \end{array} \right\rangle \text{ then } \psi \left\langle \begin{array}{c} s[\varphi][\psi] \\ s[\varphi][\neg\psi] \\ s[\neg(\varphi;\neg\psi)] \end{array} \right\rangle$$

While one could claim that this conceptually simplifies the treatment of conditionals (the downward proliferation of the update is a simple difference operation), it is not clear how the parallel behavior of existential modality could be implemented in an equally appealing way. One would have to stipulate that an update with the information that $\Diamond\varphi$ would involve two pushing operations, too, and leave *two* copies of $s[\varphi]$ on top of the stack. Furthermore, unless we introduce revision or non-monotonic updates, the state in the middle of the output stack in (55) is empty, thus it would block the path of downward percolation of any subsequent update with information of the form ${}^{\mathbf{w}}\chi$. Therefore the present system seems more suitable for the purpose at hand.

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 becomes easier to see when considering its relation to other formalisms, especially trees. The fact that a sequence of stacks can alternatively be viewed as a sequence of "snapshots" of different stages in the constructions of a tree has been exploited variously (Grosz and Sidner, 1986):

(56)



In fact, non-accommodationist treatments of modal subordination in DRT and its derivatives (cf. Frank (1996)) typically build the equivalent of tree structures, with DRSs labeling nodes and (DRS-)subordination as the ordering relation between them. However, at any point in a discourse, the stack representation introduced here contains no more than one stack, i.e., one path in the tree. The tree that would correspond to an entire discourse is therefore not an essential level of representation, but epiphenomenal, encoding merely the *history* of the processing. This view is motivated by a desire to separate the information *content* conveyed in the discourse from more accidental and idiosyncratic properties of the *protocol*: The temporary contexts are kept only as long as needed to contextualize incoming portions of incomplete information (in this sense, modal subordination is a form of underspecification), and the final result of the processing does not reveal the way it came about.

Note that the limitation to one stack at a time introduces a strong *right frontier* constraint. Such a generalization has been discussed and motivated before (Grosz and Sidner, 1986; Webber, 1991), among others. In tree formalisms it has to be stated explicitly, while it is a natural consequence of the present system. This is not to say that reference to earlier parts of the discourse is impossible, but only that the *implicit* kind of conditionalization observed in modal subordination is restricted to local contexts and thus adequately treated here (cf. Note 1 for locality.)

Furthermore, 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. By virtue of the fact that every newly added piece of information is immediately propagated down the stack, the bottom element is up-to-date after every sentence. In this respect the system proposed here differs also from "Zeinstra's Logic" (Muskens et al., 1996, 610–612), which otherwise has some similarity to this framework:

There an update at the top is not complete and has no consequences further down in the stack until Zeinstra's right-bracket operator] is applied. The right bracket corresponds roughly to the Pop-operator $[\bigtriangledown]$ of the present formalism with respect to its effect on the shape of the stack. In addition, however, it ensures the downward propagation of updates, a task which here is fulfilled in mid-discourse by the $[\cdot]_{\downarrow}$ operation.

Other phenomena for which analyses in terms of trees have been proposed 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 by ter Meulen (1995) in terms of "Dynamic Aspect Trees". Another application would be quantification in the domain of individuals, rather than worlds. But the pursuit of these ideas must be left for another occasion.

References

Frank, A. 1996. *Context Dependence in Modal Constructions.* PhD thesis, Institut für maschinelle Sprachverarbeitung, Stuttgart.

Groenendijk, J., M. Stokhof, and F. Veltman. 1996. Coreference and modality. In Lappin, S., editor, *The Handbook of Contemporary Semantic Theory*, pages 179–213. Blackwell, Oxford.

Grosz, B. J. and C. L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Kamp, H. and U. Reyle. 1991. *A Calculus for First Order Discourse Representation Structures.* Arbeitsbericht des Sonderforschungsbereich 340, Nr. 16. Universität Stuttgart.

Kamp, H. and U. Reyle. 1993. *From Discourse to Logic.* Kluwer, Dordrecht.

Kaufmann, S. 1998. What's at stack in discourse. In Dekker, P., M. Stokhof, and Y. Venema, editors, *Proceedings of the 11th Amsterdam Colloquium*, pages 187–192. ILLC.

Kibble, R. 1998. Modal subordination, focus and complement anaphora. In Ginzburg, J., Z. Khasidashvili, C. Vogel, J.-J. Lévi, and E. Vallduví, editors, *The Tbilisi Symposium on Logic, Language and Computation: Selected Papers*, pages 71–84. CSLI.

Krahmer, E. and R. Muskens. 1995. Negation and disjunction in discourse representation theory. *Journal of Semantics*, 12:357–376.

Kratzer, A. 1991. Modality / Conditionals. In von Stechow and Wunderlich (1991), pages 639–656.

ter Meulen, A. 1995. *Representing Time in Natural Language*. MIT Press, Cambridge, MA.

Muskens, R., J. van Benthem, and A. Visser. 1996. *Dynamics*. Report and Technical Note Series, LP-96-04. ILLC, Amsterdam.

Prüst, H., R. Scha, and M. van den Berg. 1994. Discourse grammar and verb phrase anaphora. *Linguistics and Philosophy*, 17:261–327.

Roberts, C. 1989. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy*, 12:683–721.

Roberts, C. 1995. Domain restriction in dynamic semantics. In Bach, E. et al., editors, *Quantification in Natural Language*. Kluwer, Dordrecht.

von Stechow, A. and D. Wunderlich, editors. 1991. *Semantik: ein internationales Handbuch der zeitgenössischen Forschung = Semantics*. Walter de Gruyter.

Vermeulen, K. 1994. *Explorations of the Dynamic Environment*. PhD thesis, Utrecht.

Webber, B. L. 1991. Structure and ostension in the interpretation of discourse deixis. *Language and Cognitive Processes*, 6:2:107–135.

Zeevat, H. 1992. Presupposition and accommodation in update semantics. *Journal of Semantics*, 9(4):379–412.