As time goes by...
Representing ordinary English reasoning in time about time

Alice G.B. ter Meulen

"Thus, the issue of 'static versus dynamic' is rather subtle."
Johan van Benthem, Logic in action, p. 259.

1. Introduction

In dynamic semantics of natural language the objective is to analyze how a given context is affected by incorporating new information, associating with linguistic expressions different context-change potentials (CCP). In reasoning about time the order in which information is presented to the interpretor is a particularly relevant dimension, for it constrains the inferences we draw about what happened when. Competent English speakers share fundamental linguistic intuitions about the ways in which the morphological inflections may be used to indicate in what order the described events took place. Consider (1) and (2):

(1) Jane was patrolling the neighborhood. She noticed a car parked in an alley.
(2) Jane noticed a car parked in an alley. She was patrolling the neighborhood.

Both (1) and (2) makes us answer the question whether Jane was patrolling the neighborhood already before she noticed the car positively. The order of presentation of the two sentences hence does not seem to affect our judgement. Nor does our answer depend on the meaning of the particular predicates involved. Our judgement is based solely on the contextual interaction of the past tense, the aspectual properties of progressive verbal morphology and the order of presentation.
But the order of presentation in (3) and (4) does affect what we answer to the question whether Jane had turned the corner before she noticed the car.

(3) Jane turned the corner. She noticed a car parked in an alley.
(4) Jane noticed a car parked in an alley. She turned the corner.

We judge on the basis of (3) that she did turn the corner first, but given (4), her noticing the car must have preceded her turning the corner. These core data have set the agenda for the dynamic semantics of tense and aspect since the early eighties when Hans Kamp and his colleagues first proposed Discourse Representation Theory as a new toolkit, integrating tense logical results with Reichenbachian reference times as third semantic parameter (see van Eijck and Kamp (1997) for an updated presentation of DRT).
One may be tempted to a quick hypothesis (// denotes order of presentation irrelevant):

Hypothesis 1:
A. PAST PROG (e1) // PAST (e2) = e1 includes e2, part of e1 precedes e2.
B. PAST (e1) + PAST (e2) = e1 precedes e2, all of e1 precedes e 2.

But it is easy to find evidence against B, consider (1a) where the past progressive of (1) is turned into a simple past.

(1a) Jane patrolled the neighborhood. She noticed a car parked in an alley.

If B were valid, we should conclude that Jane patrolled the neighborhood before she noticed the car. But our intuition does not support that conclusion, for from (1a) we may still conclude to the same temporal relations as we did based on (1). So what explains this difference between the two simple past tense discourses?

The fundamental difference is one of aspectual class. If we describe what Jane does with predicates that apply homogeneously to any part of her action, the description is called an activity (ACT). If our description does not apply to a part of her action, it is called an accomplishment (ACC). Our first hypothesis needs refining to make B depend on aspectual class, as below.

Hypothesis 2.

A. Correct

B. PAST (e1**ACC**) + PAST (e2) = e1 precedes e2, all of e1 precedes e 2.

C. PAST (e1 **ACT**) + PAST (e2) = e1 includes e2, part of e1 precedes e2.

This leads to the prediction that aspctual class effects are still overruled by progressive inflections, as it cancels the dynamic force the past tense accomplishments have. Indeed, applying this prediction to (3) and (4) we see it is borne out in our intuitive judgements concerning the temporal relations described in (3a) and (4a).

(3a) Jane was turning the corner. She noticed a car parked in an alley.
(4a) Jane noticed a car parked in an alley. She was turning the corner.

Had Jane started to turn the corner before she noticed the car? Both (3a) and (4a) make us answer positively, as if her turning the corner were portrayed in slow motion.

Now one may wonder whether there is any relevant semantic difference between activities and progressive descriptions, as both seem to lead to the same conclusions about the temporal relations. Contrary to the account of these data in DRT, I would like to argue that indeed there is a significant difference. Consider the three sentences discourses in (5), a sequence of ACC, ACT and PROG and (6), a sequence of ACC and two PROG.

(5) ACC + ACT + PROG
Jane noticed a car parked in an alley. She patrolled the neighborhood. She was driving along the Rokin.
(6) ACC + PROG + PROG
Jane noticed a car parked in an alley. She was patrolling the neighborhood. She was driving along the Rokin.

Now was Jane driving along the Rokin when she noticed the car in the alley? If our answer is based on (5), it is not all that clear what we should say. She may have been driving along the Rokin when she noticed the car or she may have been elsewhere. Perhaps she started her patrolling of the Rokin after she had noticed the car elsewhere. But if (6) is the information on which we base our answer, it is perfectly clear that both her patrolling must have started before she noticed the car and her driving along the Rokin. Hence the car she noticed must have been in an alley off the Rokin. So (6) gives us enough information to infer that she must have been driving along the Rokin, when she noticed the car. This inferential difference between (5) and (6) must be caused by the ACT versus PROG in their second clauses.

2. The toolkit: Dynamic Aspect Trees and Chronoscopes.

The dynamic representation of tense and aspect in natural language must make a fundamental distinction between ways of referring to events, i.e. giving dynamic information about change, and ways of describing states, giving static information about what is the case. In Dynamic Aspect Trees (DAT, ter Meulen 1995) dynamic information affects the architecture of the tree representation by adding new nodes, whereas stative information merely adds descriptive labels to existing nodes.

A text provides three kinds of information: (i) the descriptive content, (ii) the aspectual content and (iii) the perspectival content. The descriptive content determines the truth-conditional meaning; it classifies a situation as being of a certain type or supporting that type. It is represented in DATs by labels on nodes, encoding a relation, an appropriate number of arguments and a positive or negative polarity. The aspectual content of a clause tells us how its descriptive content is integrated with the given context. In DATs this is encoded in the open or closed nodes to represent ACT and ACC events, respectively called ‘holes’ and ‘plugs’, and in the stickers for stative information which are appended to nodes without introducing new ones. The perspectival content of a clause or text determines which point in the representation is considered the point-of-view of the evaluation representing the location of the interpretor drawing inferences from the given information. It determines, for instance, the reference of indexicals and demonstratives, but it also affects the form with which available information is reported. E.g. the text may present information in the simple past tense, but later that information may need be reported as a stative sticker using past perfect clauses. The spatio-temporal location of the act of issuing the information is also included in the perspectival content, in DATs represented as the source-node, the unique right-most terminal node. A chronoscope is a connected, a-cyclic path from root to terminal node, representing a cone of simultaneously satisfiable descriptive content, carried in the labels on its nodes. A chronoscope containing the unique current node is a current chronoscope, which is unique in a DAT only when the current node is terminal.

When the initial past tense clause is processed, the past tense rule sets the current node to a newly created left sister open node, labelled PAST, to dominate all of the past tense discourse. Subsequent past tense clauses must be represented at nodes dependent on this PAST node. A DAT is step-wise constructed during the process of interpretation. There may be choice-points at which the interpretor may decide to accept or reject certain inferences or interpretions, based on information that is contained in external sources, called oracles. The DAT system is designed only to determine the conquences of an interpretive choice, but it does not provide any heuristic guidelines in making the choice.

The DAT construction rules are presented here rather informally, for expository easy. A reader who wishes to see a formal presentation of the DAT construction should consult Seligman & ter Meulen (1995) or (ter Meulen 1995). Each DAT has a unique current node, designated with *c* .

DAT construction rules

(i) *Sticker rule:*

for progressive and perfect clauses, simple lexical states (*be*, *have*) =>

(i) stack sticker label on *c*, if *c* is a plug

(ii) sticker label on next new node, otherwise

(ii) *Hole rule:*

for ACT => affix label on new hole: reset *c* to new hole, i.e. next node to be dependent

(iii) *Plug rule:*
for ACC => affix label on new plug: reset *c* to new plug, ie. next node to be sister.

3. Illustrations of DATs

The construction rules are now applied to the data discussed in section 1. The first observation, based on (1) and (2), was that the order of presentation of a progressive and a simple past is irrelevant in determining the temporal relation between what they describe. So (1) and (2) should produce the same DAT, as in fig. 1.
(1) Jane was patrolling the neighborhood (Prog sticker). She noticed a car parked in an alley (plug).
(2) Jane noticed a car parked in an alley (plug). She was patrolling the neighborhood (Prog sticker).

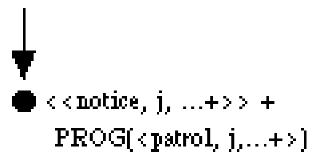

Fig. 1. DAT for (1) and (2)

The PROG-sticker awaits the introduction of the notice-plug for (1), and for (2) it is appended to the notice-plug after its introduction. For transparency, the labels are not always fully specified in these illustrations and the source node is not displayed, when it is obvious where its location is. From fig. 1 we infer that (i) she was patrolling the neighborhood when she noticed the car as a matter of DAT logic, as both labels are attached to the current node, and (ii) she must have started her patrolling before she noticed the car, by an inference rule relating the start of an event as presupposition of its progressive. Her starting the patrolling cannot be situated, as all we know is that it was earlier, so the current node should only carry a modal label representing a possible DAT update with a prior start. We cannot infer anything about her ending patrolling, other than that if it occurs, it occurs later.
A text consisting of two ACC, both represented as plugs, is given in (3), where the introduction of the PASTnode is illustrated in fig. 2.
(3) Jane turned the corner (plug). She noticed a car parked in an alley (plug).

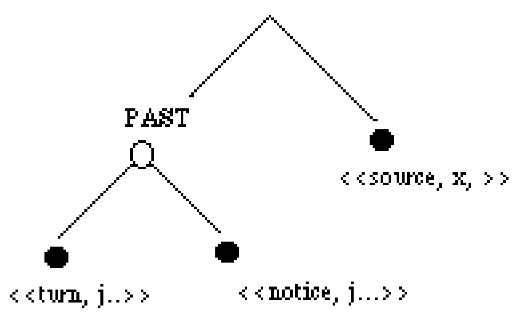

Fig. 2. DAT for (3)

Since the order of presentation was here considered relevant for temporal reasoning, reversing the order as in (4):

(4)Jane noticed a car parked in an alley (plug). She turned the corner (plug).

would produce an isomorphic DAT, but its labels would be reversed. The left-to-right order on nodes is used in precedence inferences, i.e. from fig. (2) we would conclude she first turned the corner, before she noticed the car.
To represent differences between ACT and PROG-stickers in DATs we discussed (1a),

(1a) Jane patrolled the neighborhood (hole). She noticed a car parked in an alley (plug).

starting with a simple past ACT, which creates a hole for Jane's patrolling with a dependent plug for her noticing the car.

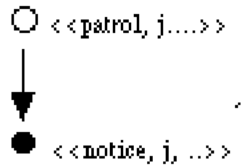

Fig. 3 DAT for (1a).

The intended meaning of the dependency is that the noticing event took place within the time her patrolling took, i.e. the down-arrow represents temporal inclusion of events. This relation is not reversable, as it would be if there was a sticker on the plug, as in fig. 1. We can also infer from fig. 3 that she was patrolling the neighborhood when she noticed the car, introducing a PROG-sticker on any node dependent on a node carrying the corresponding label. In other words, fig 1 is entailed by fig. 3, but not vice versa.
The same point is illustrated in texts (3a) and (4a), both creating fig. 4.

(3a) Jane was turning the corner (Prog sticker). She noticed a car parked in an alley (plug).

(4a) Jane noticed a car parked in an alley (plug). She was turning the corner (Prog sticker).

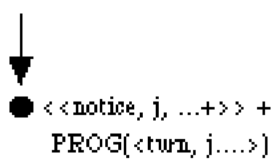

Fig. 4. DAT for (3a) or (4a).

To appreciate the different behavior of holes and stickers in a slightly more complex content, we discussed (5) and (6), differing only in the second clause.

(5) Jane noticed a car parked in an alley (plug). She patrolled the neighborhood (hole).
She was driving along the Rokin (Prog sticker).

(6) Jane noticed a car parked in an alley (plug). She was patrolling the neighborhood (Prog sticker). She was driving along the Rokin (Prog sticker).

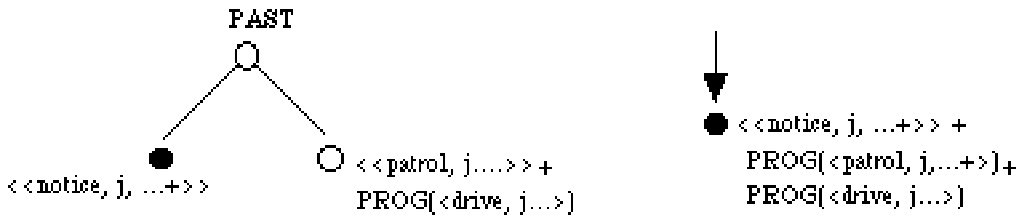

Fig. 5 DATs for resp. (5) and (6)

A difference was observed in degree of reliability of the judgement whether Jane was already driving along the Rokin, when she noticed the car in the alley. Given (6) it would be definitely clearer that she was, than given (5). This is accounted for in a DAT representation as follows. From the right DAT for (6) in fig. 5 we infer, as the *drive*-sticker is on the *notice*-plug, that she must have been driving along the Rokin at the time she noticed the car. But in the left DAT in fig. 5 additional information is needed to draw that conclusion. There is no DAT inference rule that copies PROG stickers to left or right sister nodes, but there may always be other sources, serving as oracles to the temporal reasoning component taht provide the information to supports this sticker portability.

4. Dynamic inference as DAT construction

Since the three construction rules of the previous section only create right-downward branching, two new dynamic rules are defined to allow backing up into a chronoscope when new information cannot be accommodated in the current one, or when the new information carries temporal simultaneity presuppositions.

(7) Plug-up rule:
if extending current chronoscope introduces a node with labels incompatible with the labels of its ancestors:

(i) back up to lowest node *n* in the current chronoscope with incompatible labels,
(ii) plug *n* up, if it is not already a plug
(iii) reset current node c =*n*
(iv) apply plug rule to *n*

(8) Unplugging rule:
if temporal presuppositions of new information require a new node to be temporally dependent on the current node even when it is a plug, substitute for the current node a hole with the same labels, apply the hole rule.

To illustrate the application of plugging up and unplugging nodes, consider the story in (9) and its DAT in fig 6.

(9) After dinner (plug), Jane did her homework (hole). She was sitting on the sofa (PROG sticker). The cat slept on her lap (hole). Suddenly the doorbell rang (plug). She got up to open the door (plug). It was John (sticker). He wanted her to come with him (sticker). He did not realize she was doing her homework (negative polarity sticker). She started to explain he better leave (plug). First she said her homework was not done yet (plug).

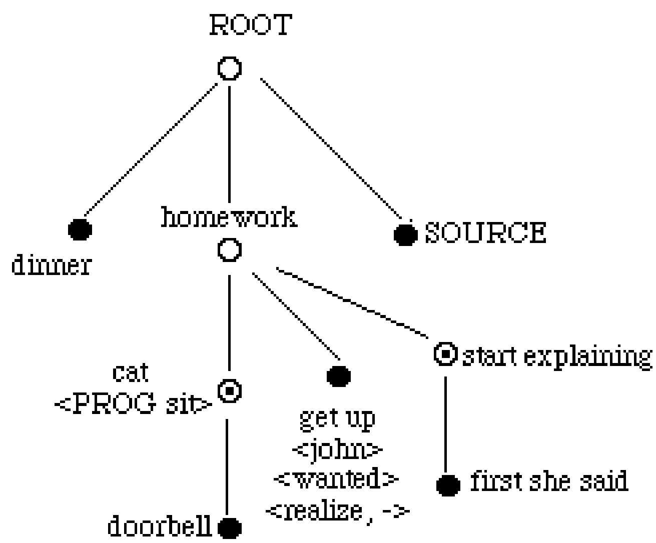

Figure 6. DAT for (9).

The plugging up rule is here triggered by the incompatibility of Jane's sitting on the sofa and her getting up. Of course, any interpretor thinking that you cannot do your home work when interrupted by the doorbell should back up to the homework node. DATs won't tell you what is or is not compatible information, and different people may make different pragmatic assumptions. The hole carrying the label for Jane's sitting on the sofa is plugged up and the plug rule is applied which introduces the plug for her getting up, onto which three stickers are subsequently stacked. The unplugging rule is triggered by accommodating the presupposition of the adverbial 'first' in the final clause, which begins a list of actions constituting her starting to explain. Typically part-whole relations between arguments of a ACC predicate trigger the unplugging rule, so the sublattice structure on the domain of individuals is inherited in the event-structures in which DATs are eventually interpreted by embeddings. The reasoning required in resolving incompatibility of labels and accommodating temporal presuppositions is a cognitively more demanding task, requiring possibly appeal to an external oracle. Spreading stickers in a DAT is cheap and automatic, so a cognitively lighter task. This form of reasoning is less open to differences of opinionon what constitutes compatible information among interpretors, and hence leads to fewer misunderstandings.

3. Situated inference and logical entailment

Reasoning with DATs is situated, as the current node must support the label corresponding to the conclusion, as illustrated informally above. The current node is the one arrived at after constructing a DAT for all premises in the order given. But people may construct different DATs for the same sequence of premises, and hence may hence draw different conclusions. Stickers may be imported to the current node according to the rules here informally stated, easily formalized as structural rules in a deductive system.

1) All stickers may be copied to all dependent nodes.
2) A PERF sticker may be copied onto any right sister node.

3) A PROG sticker imports a left-directed weak modal START of taht label at that node and a right-directed weak modality for END of that label.
4) A PERF sticker imports a left-directed weak modal with that label at that node.
5) Any label may be copied onto a right sister as PERF sticker of that label.
6) A label may be imported as PROG sticker of that label on any dependent node.
In (Kamp & Reyle 1996, p. 305.) the notion of logical consequence is defined in Discourse Representation Theory as follows: a DRS K' is a logical consequence of DRS K iff any verifying embedding of the conditions in K can be extended to a verifying embedding of the conditions in K'. Hence conclusions cannot add new reference markers to the domain of K, but state just new facts about known ones. But there is no context dependent parameter relative to which the notion of inference is defined, as any free reference marker is existentially closed by the top level existential closure condition.
In DAT logic the notion of situated inference is defined as follows. DATs are provided with a proper modelthory by embeddings, into event-structures with minimal structure, much like in DRT. Given a DAT D for the premisses L1... Ln, with *c* as current node, then L is a situated inference from the premisses, when *c* verifies L for any verifying embedding of that DAT into the possible event-structures E.

DATs predict the situated entailments (11) and (12) from text (10), which in DRT would not be logical entailments but possible default inferences, a notion which requires specification of 'normal courses of events' (cf. Lascarides and Asher (1993))

(10)Jane patrolled the neighborhood. She noticed a car parked in an alley. She gave it a ticket.

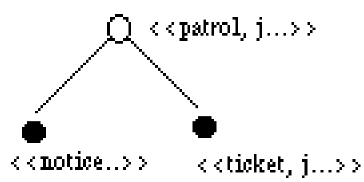

Fig. 7. DAT for (10).

To infer (11) from (10) the DAT in fig. 7 was first constructed, where the ticketing node is the current one. By PROG import, rule 6 above, we import a PROG sticker representing that Jane was patrolling the neighborhood at the ticketing node. The PROG sticker by rule 3 makes us infer that for any end of her patrolling, it can only have occurred after her ticketing, for (12).

(11) Jane was patrolling the neighborhood when she ticketed the car in the alley.
(12) Jane must have ended patrolling the neighborhood after she ticketed the car in the alley.

This must suffice as necessarily brief illustration of how situated inference are obtained by importing appropriate stickers to the current node of a DAT constructed for the premisses. The interested reader is referred to the papers on DATs listed below for further discussion and illustration of other examples.
In considering the logical behavior of DAT inferences, it may be remarked here that the classical rules Monotonicity, Permutation and Cut must be applied with great caution.

$$\text{PERM} \quad \frac{X, A, B, Y \mid\!- C}{X, B, A, Y \mid\!- C} \qquad \text{only when A, B are stickers of the same kind}$$

MON $\dfrac{X, Y \mid\!- A}{X, B, Y \mid\!- A}$ only when B is a sticker

CUT $\dfrac{X, A, Y \mid\!- B \qquad Z\mid\!-A}{X, Z, Y\mid\!- B}$ only when Z and A are PERF stickers

In CUT A is already a conclusion, so A is itself a sticker. If Y is empty, of course CUT holds for any sticker. But only PERF stickers are rightward portable, so, if Y introduces new nodes and the current node may be a later one, CUT holds only for if Z and A are PERF stickers that can be taken along to any right sister node. If Y introduces nodes that depend on the node at which A was introduced, A and Z are copied downward onto these new nodes and so CUT holds again no matter what kind of sticker A and Z are. Stickers are not generally freely permutable as labels of a node in a DAT : two adjacent perfective stickers are permutable as well as two adjacent progressive stickers.

4. Summary

Temporal reasoning in time about time is paradigmatic of situated reasoning with partial information in natural language. The tree structures that the DAT system designs are visually direct representations of temporal dependencies, which facilitates inference as geometric form manipulation (prooftheory). They integrate descriptive, aspectual and perspectival information and make all three dimensions affect the situated inferences supported in DATs. We have seen that acitivities extend the current chronoscope and preserve the contextual settings, but they behave quite differently from stickers which represent the various kinds of stative information and may spread in a DAT. The order of presentation of even stative information should be encoded in the representation, hence stickers constitute a stack at a node, which admit local permutation only when the stickers are of the same kind. Conflicting information and accommodation of presuppositons are two forms of dynamic inference with context change potential, i.e. structure building rules in DATs. This paper has been written in the belief that the logical analysis of natural language always creates new insights for both logic and linguistics, while searching for empirical cognitive constraints on human information processing.

References

van Benthem, J. 1995. *Language in Action. Categories, lambdas and dynamic logic.* Cambridge (Mass.), MIT Press.

van Eijck, J. and H. Kamp. 1997. Representing discourse in context. in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language.* MIT/Elsevier. 179-237.

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

Kamp, H. and U. Reyle. 1996. A calculus for first-order discourse representation structures. *Journal of Logic, Language and Information* 5, 297-348.

Lascarides, A. and N. Asher. 1993. Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and Philosophy* 16, 437-594.

Seligman, J. and L. Moss. 1997. Situation Theory. in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*. MIT & Elsevier. 239-309.

Seligman, J. and A. ter Meulen. 1995. Dynamic aspect trees. In L. Pólos et al. (eds.), *Applied Logic: How, What and Why*. Dordrecht: Kluwer Publishers. 287-320.

ter Meulen, A. 1995. *Representing Time in Natural Language. The dynamic interpre–tation of tense and aspect.* Cambridge (Mass.). Bradford Books, MIT Press.

ter Meulen, A. 1996 Perspektiven in die logische Semantik, in : C. Umbach, et al. (eds.), *Perspektive in Sprache und Raum*, Studien zur Kognitionswissenschaft, Wiesbaden, Germany, Deutscher Universitaetsverlag.

ter Meulen, A. 1999. Chronoscopes: dynamic tools for temporal reasoning. in J. Higgin–botham and F. Pianesi (eds.), *Speaking about events.* Oxford U. Press. (to appear).