Two theories of dynamic semantics

Jeroen Groenendijk
Martin Stokhof
Department of Philosophy
Department of Computer Science, Faculty of Arts
University of Amsterdam[1]

[1]To appear in *Jelia '90. European Workshop on Logics in AI. Edited by Jan van Eyck. Lecture Notes in Computer Science. Springer Verlag: Berlin.* This paper was written while we were engaged on the DYANA-project (EBRA 3715). We would like to thank Frank Veltman and Roel de Vrijer for their help.

Abstract

The dynamic view on the semantics of natural language, though stemming already from the seventies, has developed into a widely studied subject in the second half of the eighties. At present, the unification of various dynamic theories constitutes an important issue. In this paper, two theories are compared, viz. *update semantics*, and *dynamic predicate logic*. In section 1 a general characterization of the idea of a dynamic semantics for natural language is given which subsumes these two theories. Sections 2 and 3 are devoted to short expositions of each of them. In the final section 4 a comparison is made.

1 Dynamic semantics for natural language

The standard approach to modeltheoretic semantics for natural language, to which we will refer in the sequel as *static semantics*, can be characterized as follows. The meaning of a sentence is identified with its truth-conditional content. Consequently, the interpretation of a sentence with respect to some model M is given by a recursive definition of the truth of a sentence with respect to M and certain parameters specified in M (assignments of values to variables, possible worlds, points in time, speaker, hearer, and so on.) We use the term *index* to cover whatever parameters are in use. Doing so, the meaning of a sentence in M can be identified with the set of indices with respect to which it is true in M. Other semantic notions are defined in terms of this one. For example, entailment is defined as meaning inclusion in all M. And the notion of updating an information state with a sentence is defined as taking the conjunction of the information state with the information content, i.e., the truth-conditional content, of the sentence.

By contrast, the dynamic outlook on natural language interpretation, referred to as *dynamic semantics*, starts from a fundamentally different basic notion. Not the information content, but the information change potential of a sentence is regarded as constituting its meaning. Consequently, the notion of the interpretation of a sentence with respect to a model M is given by a recursive definition of the result of updating an information state with the sentence. The meaning of a sentence with respect to M can then be identified with the update function associated with the sentence in M. This already brings out the fundamental difference between a static and a dynamic semantical system. Whereas in the former the notion of information content is the basic recursive notion, in the latter it is the notion of information change that plays this role. Finally, the dynamic notion of meaning brings along new possibilities for defining entailment. One among them is the following: ϕ entails ψ iff whenever we update an information state s with ϕ, we end up with an information state s' in which ψ is accepted or satisfied.

The key notion in the above characterization is that of information. For our present purposes, the following simple notion suffices. Information is a set of possibilities, viz., those situations which are still open. Information may concern different things, hence these possiblities may be diverse in character. For example, information may concern just the values of variables, or it may concern world-time pairs, or whatever parameters we decide to take into account. In the terminology we adopt here, information is about indices.

Information can then be represented as follows. Let I be the set of indices. An information state is a subset of I. Getting better informed is eliminating possibilities, i.e., going from an information state to a subset of that state. The minimal information state is the set of all possible indices I. A state of maximal information is a unit set $\{i\}, i \in I$. The empty set $\emptyset$ is the absurd information state. The power set $\wp(I)$ of I, henceforth denoted by S, is the set of all information states. It is partially ordered by $\subseteq$, which in the present context represents the relation of 'being at least as strong an information state as'. Updates (information state transformers) are functions from information states to information states.

We define the following notions:

Definition 1 Let s be an information state, i an index, and τ an update. Then:

1. τ is successful in s iff $\tau(s) \neq \emptyset$
2. τ is true in i iff τ is successful in $\{i\}$
3. $\downarrow\tau = \{i \mid \tau \text{ is true in } i\}$ ('down')
4. $\uparrow s = \lambda s' \colon s' \cap s$ ('up')
5. $\tau \circ \tau' = \lambda s \colon \tau'(\tau(s))$ ('sequencing')

Bearing in mind that the dynamic meaning of a sentence is an update, 1.1 says that updating with a sentence is successful iff the resulting information state is not the absurd one. According to 1.2 the static notion of truth is still available: a sentence ϕ is true at an index i iff the state of maximal information $\{i\}$ can be successfully updated with ϕ. The $\downarrow$-operator defined in 1.3 retrieves the information content of a sentence from its dynamic meaning. As was remarked above, in the static set-up the notion of update is defined globally: when applied to the information content of ϕ, the $\uparrow$-operator defined in 1.4 returns an update which takes the intersection of an information state and the information content of ϕ. Note that for every s $\downarrow\uparrow s = s$, but not for all τ it holds that $\uparrow\downarrow\tau = \tau$. Finally, the update corresponding to a sequence of two sentences ϕ and ψ is defined in 1.5 as the composition (in reversed order) of the updates of ϕ and ψ.

Getting better informed was described above as going from an information state to a subset of it. Not every update, however, makes you better informed. Those that do we call *eliminative updates*:

Definition 2 τ is eliminative iff for every s, $\tau(s) \subseteq s$

Notice that the following holds:

(a) $\downarrow\tau = \{i \mid \tau(\{i\}) = \{i\}\}$, if τ is eliminative

Another relevant property of updates is that of *distributivity*:

Definition 3 τ is distributive iff for every s, $\tau(s) = \bigcup_{i \in s} \tau(\{i\})$

A distributive update is one which works 'point-wise'. For distributive updates it holds that they can be viewed also as relations between indices, instead of as functions from sets of indices to sets of indices. Generally, the relation R_τ corresponding to τ is defined by $R_\tau = \{\langle i, j\rangle \mid j \in \tau(\{i\})\}$. Then we have:

(b) $\tau(s) = \{j \mid \exists i \in s\colon \langle i, j\rangle \in R_\tau\}$, if τ is distributive

Updates which are both eliminative and distributive we call *classical*:

Definition 4 τ is classical iff τ is eliminative and distributive

Classical updates have the following characteristic. By distributivity, a classical update τ is equivalent to a relation between indices. If τ is also eliminative, this relation can only consist of pairs of identical indices. And such relations can be identified with sets. Now we note the following facts (See also van Benthem (1989)):

(c) $\tau(s) = s \cap {\downarrow}\tau$, if τ is classical

This says that if a sentence expresses a classical update, updating an information state with that sentence comes down to taking the intersection of the information state and the information content of the sentence.

Consequently, we have:

(d) ${\uparrow}{\downarrow}\tau = \tau$, if τ is classical

These observations show that a dynamic semantics which assigns only classical updates to sentences is not really dynamic after all: it is equivalent with a static semantics with a globally defined notion of update. In other words, a truly dynamic semantics will assign at least to some sentences updates which lack at least one of the properties of distributivity and eliminativity.

2 Update semantics

In this section, we sketch the basics of Veltman's system of update semantics by giving a simplified version of the simplest system for which it makes sense: propositional logic with an operator *might*. The reader is referred to Veltman (1990) for the actual system and more substantial applications.

Consider the following two examples:

(1) It might be raining. ... It isn't raining.

(2) It isn't raining. ... *It might be raining.

There is a marked contrast between (1) and (2), and this displays the aspect of the meaning of the modal operator *might* that is at stake here. In examples like these, *might* serves to indicate something about an information state, rather than something about the world. A sentence of the form *might* ϕ requires that ϕ be compatible with the information state. The difference between (1) and (2) is the following. In (1), it is first expressed that the information state is compatible with the possibility that it rains, while subsequently the information state is updated with the information that it does not rain. This is okay. In (2), the information state is first updated with the information that it does not rain, and once this has happened the information that it rains is no longer compatible with the information state. So (2) is out.

The aspects of update semantics which are relevant for the comparison we are to carry out, can be captured by the following definitions. The language discussed is that of propositional logic with a sentential operator $\diamond$ which can be 'outscoped' only by conjunction. A model is a pair $\langle W, V \rangle$, with W a set of possible worlds, and V a function assigning a subset of W to the atomic sentences. The set of possible worlds constitutes the set of indices of this system, so the set of information states $S = \wp(W)$. The interpretation function $[\ ]_M$ assigns update functions to sentences. We leave out reference to M whenver this does not lead to confusion. We write $s[\phi]$ for the result of updating an information state s with ϕ, and the definition runs as follows:

Definition 5

1. $s[p] = s \cap V(p)$
2. $s[\neg\phi] = s - s[\phi]$
3. $s[\phi \wedge \psi] = s[\phi][\psi]$
4. $s[\diamond\phi] = \begin{cases} s & \text{if } s[\phi] \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$

The update function associated with an atomic sentence is classical, as is the one assigned to a negation. As was to be expected, conjunction is interpreted as sequencing: updating s with $\phi \wedge \psi$ means updating s with ϕ first, and then updating the resulting state $s[\phi]$ with ψ. The clause for $\diamond\phi$ says that updating s with $\diamond\phi$ leaves s as it is, if updating s with ϕ would be successful, and results in the absurd information state if it would not be. This update is eliminative, but not distributive, hence it is not classical. So it is here that the dynamics of the system resides. That $[\diamond\phi]$ is not distributive follows from the fact that something may be compatible with an information state s without being compatible with all non-empty subsets of s. Since composition preserves properties of updates, all updates are eliminative, but not all are distributive.

For an explication of the contrast between (1) and (2), we need the following definition:

Definition 6 ϕ is consistent iff ϕ is successful for some s

Then we can observe that $\Diamond p \wedge \neg p$ is consistent, whereas $\neg p \wedge \Diamond p$ is not.

Another basic notion of update semantics is that of acceptance:

Definition 7 ϕ is accepted in s, written as $s \Vdash \phi$, iff $s \subseteq s[\phi]$

Note that since all (sequences of) sentences in this system express eliminative updates, we have that $s \Vdash \phi$ iff $s[\phi] = s$, which is the way Veltman defines acceptance.

In terms of acceptance the following notion of entailment is defined:

Definition 8 $\phi_1, \ldots, \phi_n \models \psi$ iff for all M and s: $s[\phi_1]_M \ldots [\phi_n]_M \Vdash \psi$

A conclusion ψ follows from a sequence of premises $\phi_1, \ldots, \phi_n$ if whenever an information state s is updated with $\phi_1, \ldots, \phi_n$ consecutively, the result is an information state which accepts ψ. There are alternative notions of entailment available in update semantics, see Veltman (1990), van Benthem (1990).

3 Dynamic predicate logic

In this section we sketch the system of dynamic predicate logic (*DPL*). In order to facilitate comparison, we give the system in the format of update semantics. See Groenendijk & Stokhof (1989) for the original formulation and more details.

DPL is concerned mainly with a compositional analysis of anaphoric relations. The following simple examples will explain what is at stake:

(3) A man walks in the park. He whistles

(4) Every farmer who owns a donkey, beats it

If we were to translate these sentences in a compositional, 'on-line' manner, what we would end up with are the following:

(5) $\exists x(\mathtt{man}(x) \wedge \mathtt{walk\_in\_the\_park}(x)) \wedge \mathtt{whistle}(x)$

(6) $\forall x((\mathtt{farmer}(x) \wedge \exists y(\mathtt{donkey}(y) \wedge \mathtt{own}(x, y))) \rightarrow \mathtt{beat}(x, y))$

Of course, given the ordinary static semantics of predicate logic these formulae do not express what the sentences mean, since the final occurrence of x in (5) and of y in (6) is not bound by the corresponding existential quantifier. The idea is to provide a dynamic semantics which enables the existential quantifier to bind free occurrences of the variables it quantifies over, in such a way that (5) and (6) become equivalent with (7) and (8):

(7) $\exists x(\texttt{man}(x) \wedge \texttt{walk\_in\_the\_park}(x) \wedge \texttt{whistle}(x))$

(8) $\forall x \forall y((\texttt{farmer}(x) \wedge \texttt{donkey}(y) \wedge \texttt{own}(x,y))) \rightarrow \texttt{beat}(x,y))$

The relevant aspects of *DPL* are the following. The language is that of ordinary first-order predicate logic. Models M are also ordinary first-order models, consisting of a domain D and an interpretation function F. The set of indices is the set G of assignments of values to variables: information is information about values of variables, information states are sets of assignments, i.e., $S = \wp(G)$, and updates are functions from sets of assignments to sets of assignments. By $g \approx_x h$ we mean that the assignments g and h differ at most with respect to the value they assign to x.

We define the update $[\approx_x]$ as follows:

Definition 9 $s[\approx_x] = \bigcup_{g \in s} \{h \mid g \approx_x h\}$

This update works as follows. If s is an information state, $s[\approx_x]$ is the information state we get from s by 'forgetting' all information s contains about the value of x: $s \subseteq s[\approx_x]$. So, the update function $[\approx_x]$ is non-eliminative, and it behaves rather like a 'downdate'. Notice that since it is defined pointwise, it is distributive.

We now define a dynamic meaning assignment to the language of first-order predicate logic using the format of update semantics:

Definition 10

1. $s[Pt_1 \ldots t_n] = s \cap \{g \mid \langle [t_1]_g, \ldots, [t_n]_g \rangle \in F(P)\}$
2. $s[t_1 = t_2] = s \cap \{g \mid [t_1]_g = [t_2]_g\}$
3. $s[\neg\phi] = s - {\downarrow}[\phi]$
4. $s[\phi \wedge \psi] = s[\phi][\psi]$
5. $s[\exists x \phi] = s[\approx_x][\phi]$

The updates associated with atomic sentences, identities, and negations constitute classical updates. Notice that the negation of ϕ is defined as the difference of the information state with the truth conditional content ${\downarrow}[\phi]$ of ϕ. So, when s is updated with $\neg\phi$ we get those assignments in s with respect to which ϕ is false. Conjunction is defined as sequencing: $[\phi \wedge \psi] = [\phi] \circ [\psi]$. The crucial clause is the last one. Updating an information state s with a formula $\exists x \phi$ amounts to first forgetting the information s contains about the possible values of x, and then updating the resulting state $s[\approx_x]$ with ϕ. In other words, the update $[\exists x\phi]$ is equivalent with the sequence of updates $[\approx_x] \circ [\phi]$.

Since atomic formulae, identities, and negations constitute classical updates; $[\approx_x]$ is a distributive, but non-eliminative update; and conjunction and existential quantification are defined in terms of sequencing of updates, the system as a whole is distributive and non-eliminative. So, update semantics and *DPL* are both non-classical, but in different ways.

In the original formulation of *DPL* meanings were relations between assignments, instead of, as in the present format, functions from sets of assignments to sets of assignments. In view of fact (b) mentioned in section 1, this makes no difference. Consequently, what was said in Groenendijk & Stokhof (1989) about the logic of *DPL* carries over to the present formulation. For our present purposes it suffices to discuss some basic notions and facts.

We first return to the two examples discussed above. The first example concerns sequences of sentences which, when translated compositionally turn up as formulae of the form $\exists x\phi \wedge \psi$. The interpretation of such a formula is the sequence of updates $[\exists x\phi] \circ [\psi]$, i.e., $([\approx_x] \circ [\phi]) \circ [\psi]$. Since sequencing is associative, this is the same as $[\approx_x] \circ ([\phi] \circ [\psi])$. So, from the associativity of sequencing, it immediately follows that:

(e) $\exists x\phi \wedge \psi = \exists x(\phi \wedge \psi)$

The essential dynamic feature of *DPL* is that the binding force of an existential quantifier is not restricted to occurrences of variables inside its syntactic scope, but extends to occurrences of variables further on. This shows that a dynamic interpretation of the language of predicate logic makes it possible to translate examples such as (3) in a compositional way.

Before we can indicate how the second kind of example, that of so-called 'donkey-sentences', can be dealt with, we first note that implication, disjunction and universal quantification can be defined in terms of negation, conjunction and existential quantification in the usual way:

Definition 11

1. $\phi \rightarrow \psi = \neg(\phi \wedge \neg\psi)$
2. $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$
3. $\forall x\phi = \neg\exists x\neg\phi$

The definition of implication makes it 'internally dynamic': an existential quantifier in the antecedent can bind occurrences of variables in the consequent. So we have unconditionally:

(f) $\exists x\phi \rightarrow \psi = \forall x(\phi \rightarrow \psi)$

This feature of *DPL* makes it possible to treat donkey-sentences, such as (4), in a compositional way.

We end this section with some remarks about truth and entailment in *DPL*. The definition of the interpretation of a formula does not proceed in terms of its truth conditions. But as we saw, a global notion of truth is available: the truth conditional content of ϕ is given by $\downarrow[\phi]$. In terms of it we define the notion of satisfaction:

Definition 12 ϕ is satisfied by s, written as $s \models\!\!\mid \phi$, iff $s \subseteq \downarrow[\phi]$

An information state s satisfies ϕ iff ϕ is true with respect to every assignment $g \in s$.

The following notion of entailment can be defined in terms of the notion of satisfaction:

Definition 13 $\phi_1, \ldots, \phi_n \models \psi$ iff for all M and s: $s[\phi_1]_M \ldots [\phi_n]_M \models\!\!\mid \psi$

A conclusion ψ follows from a sequence of premisses $\phi_1, \ldots, \phi_n$ if whenever an information state s is updated with $\phi_1, \ldots, \phi_n$ consecutively, the result is an information state which satisfies ψ. Alternative notions of entailment can be defined for *DPL*, but the present notion has the nice feature of being dynamic itself. For example, we have that:

(g) $\exists x Px \models Px$

More generally we have the following deduction theorem:

(h) $\phi_1, \ldots \phi_n \models \psi$ iff $\models (\phi_1 \wedge \ldots \wedge \phi_n) \rightarrow \psi$

For extensive discussion of the properties of this notion of entailment see Groenendijk & Stokhof (1989).

4 A comparison

The two theories of dynamic semantics which we presented in the previous sections are similar in their general set-up, but also differ at important points. In section 1, we showed that if all updates which a semantics assigns to sentences have both the property of distributivity and that of eliminativity, such a semantics is not essentially dynamic. Veltman's update semantics is eliminative, but lacks the property of distributivity: $\diamond\phi$ tests globally, and not pointwise, whether an information state can be succesfully updated with ϕ. *DPL*, on the other hand, is distributive, but lacks eliminativity: $\exists x$ reinitializes an information state s with respect to the information it contains about the possible values of x.

The difference between the two systems also shows in that they employ different notions of entailment. In both cases an information state is updated consecutively with the premisses, but whereas in update semantics we check whether the resulting information state accepts the conclusion, in *DPL* we require that the resulting state satisfies the conclusion.

In many cases, it makes no difference which of the two notions we use. More in particular, we have the following:

(i) $s \Vdash \phi$ iff $s \models \phi$, if $[\phi]$ is a classical update

But with respect to non-classical updates the two notions are different.

For example, in *DPL* the following holds:

(j) $\exists x Px \models \exists y Py$

If *DPL*-entailment were defined in terms of acceptance, instead of by means of satisfaction, this would not go through. All g in an information state s which is the result of an update with $\exists x Px$ satisfy $g(x) \in F(P)$. For some such g it might hold that $g(y) \notin F(P)$. But such a state cannot accept $\exists y Py$, because after updating s with it, g will no longer be in the resulting state. However, s does satisfy $\exists y Py$ precisely because we can forget all about y. The fact that for all $g \in s\colon g(x) \in F(P)$ guarantees that $\exists y Py$ comes out true with respect to all elements in s. It is the non-eliminative nature of existential quantification that prevents a definition in terms of acceptance.

With respect to update semantics we observe that:

(k) $\Diamond\phi \models \Diamond\phi$

This fact would not hold if entailment in update semantics were defined in terms of satisfaction, instead of by means of acceptance. An information state which is updated with $\Diamond\phi$ may very well not satisfy $\Diamond\phi$. For notice that satisfaction is defined pointwise. Consequently, if we apply the definition of truth to $\Diamond\phi$ it says that $\Diamond\phi$ is true with respect to w iff ϕ is true with respect w. So the following holds:

(l) $\downarrow[\Diamond\phi] = \downarrow[\phi]$

On the other hand, any s which is updated with $\Diamond\phi$ accepts $\Diamond\phi$, though it need not accept ϕ. Evidently, the non-distributive character of the modal operator prohibits a definition of entailment in terms of satisfaction.

What this means is that update semantics interprets modal sentences not as assertions about the world, but rather takes them as saying something about information states: $\Diamond\phi$ checks whether ϕ is compatible with s, i.e., whether s can be succesfully updated with ϕ. It is only consistency with information, and not truth that makes sense for modal notions interpreted along these lines. This means that modal assertions are not considered to contribute genuine information. Whenever we update an information state s with $\Diamond\phi$, it will never lead to a more informative state. It either leaves s as it is, or it leads to error, i.e., to the absurd information state.

There are cases, however, where modal statements seem to have real update effects. Consider the following example (taken from Roberts ((1989))):

(9) A wolf might come in. It would eat you first.

This sequence is informative in the following sense. An information state s could be such that it contains a possible world w in which a wolf comes in that doesn't eat you first. This possible world should be eliminated by updating s with this sequence of sentences. So modal assertions should have real updating effects also.

Another interesting feature of (9) is that it exhibits an anaphoric relation between expressions which are in the scope of distinct modal operators: the indefinite term *a wolf*, which is in the scope of *might*, and the pronoun *it*, which is in the scope of *would*. Evidently, in order to be able to deal with phenomena such as these, we need a dynamic semantics of both modal operators and the existential quantifier which is more than just the sum of the two, and which allows us to account for interactions between modal operators and quantifiers. So what we need is a dynamic semantics for modal predicate logic.

However, as we have seen above, the dynamic treatment of modalities in update semantics results in a non-distributive, eliminative system, whereas dynamic predicate logic is non-eliminative and distributive. Each of these two systems has an adequate notion of entailment, but for each this is a different one. If we were to combine update semantics and dynamic predicate logic in a straightforward way, the result would be a non-eliminative and non-distributive system. The question then arises what notion of entailment naturally belongs to such a system. Although the general answer to this question is not yet clear, we may observe that in such a modal predicate logic information states will concern two separate aspects, possible worlds and assignments to variables. It might very well turn out that these two aspects, though linked in certain ways, can nevertheless be treated separately to such an extent that updates can be characterized as eliminative and non-distributive with respect to the one parameter, and as distributive but non-eliminative with respect to the other. If this turns out to be the case, it might also be possible to define a notion 'in between' acceptance and satisfaction, which can be used in defining an adequate notion of entailment.

References

Benthem, J. v.: 1989, Semantic parallels in natural language and computation, in H. Ebbinghaus et al. (eds.), *Logic Colloquium '87*, Elsevier, Amsterdam

Benthem, J. v.: 1990, *General Dynamics*, Prepublication LP–90–11, ITLI, University of Amsterdam, Amsterdam, To appear in *Theoretical Linguistics*

Groenendijk, J. and Stokhof, M.: 1989, *Dynamic Predicate Logic*, Prepublication LP–89–02, ITLI, University of Amsterdam, Amsterdam, To appear in *Linguistics and Philosophy*

Roberts, C.: 1989, Modal subordination and pronominal anaphora in discourse, *Linguistics and Philosophy* 12(4)

Veltman, F.: 1990, Defaults in update semantics, in H. Kamp (ed.), *Conditionals, Defaults, and Belief Revision*, Edinburgh, Dyana deliverable R2.5.A