Dynamic Logic in Natural Language

van Benthem, J.

Publication date
2012

Document Version
Submitted manuscript

Published in
The Routledge Companion to Philosophy of Language

Citation for published version (APA):
van Benthem, J. (2012). Dynamic Logic in Natural Language. In G. Russell, & D. Graff Fara (Eds.), *The Routledge Companion to Philosophy of Language* (pp. 652-666). (Routledge Philosophy Companions). Routledge.
https://www.taylorfrancis.com/books/e/9781136594083/chapters/10.4324%2F9780203206966-62

DYNAMIC LOGIC IN NATURAL LANGUAGE

1 Dynamic perspectives in natural language

Truth-conditional semantics Standard first-order logic defines truth in a three-part scheme: a language, structures $\boldsymbol{D}$ of objects with relations and operations, and maps from language to structures that drive semantic evaluation. In particular, "interpretation functions" I map predicate letters to real predicates, while variable assignments s map individual variables to objects. Logicians often lump $\boldsymbol{D}$ and I together into a "model" $\boldsymbol{M}$, and then interpret formulas:

formula φ is true in model $\boldsymbol{M}$ under assignment s ($\boldsymbol{M}, s \models \varphi$)

with a recursive definition matching syntactic construction steps with semantic operations for connectives and quantifiers. This pattern has been applied to natural language since Montague 1974, stating under which conditions a sentence is true. Compositional interpretation in tandem with syntactic construction works even beyond logical and natural languages: it is also a well-known design principle for programs (van Leeuwen ed. 1990). And the paradigm finds an elegant mathematical expression in algebra and category theory.

From products to activities Still, the above semantics merely describes a static relationship between sentences and the world. But truth is just one aspect of natural language, perhaps not even its crux. What makes language constitutive of human life seem *dynamic acts* of assertion, interpretation, or communication. In recent years, such acts have entered logical theory, from interpretation to speech acts and discourse. This is often considered pragmatics – but often, the natural semantic meaning *is* the use. In a maxim from mathematics and computer science: 'never study representation without transformation'. We cannot understand the static structure of a language without studying the major processes it is used for. Our very lexicon suggests a duality between product and process views. "Dance" is both a verb and a noun, "argument" is an activity one can pursue and its product that logicians write down.

Concrete dynamic systems There are many sources for current dynamic semantics of natural language. One is the seminal work of Kamp and Heim on anaphoric interpretation of pronouns as creating *discourse representation structures* that get modified as speech proceeds, and that can be matched against reality when the need arises. But the most incisive example has been "dynamification" of existing logics. First-order logic is a pilot for static truth conditions, but it can also model essentials of the process of evaluation. The latter involves shifting relationships between variables and objects (Groenendijk and Stokhof 1991). Consider the truth condition for an existential quantifier:

$$\boldsymbol{M}, s \models \exists x\, \varphi \text{ iff there is an object } d \text{ in } \boldsymbol{M} \text{ such that } \boldsymbol{M}, s[x:=d] \models \varphi.$$

Intuitively, this calls for a search through available objects d in $\boldsymbol{M}$ for the variable x until we hit the first d for which φ holds. The latter object is then available for further reference, as it should, say, in little texts like "$\exists xPx.\ Qx$" ("A man came in. He whistled.") that support anaphoric reference between the two occurrences of x across sentence boundaries.

Dynamic ideas work much more widely, with temporal expressions as a major paradigm (ter Meulen 1995). As for sentence-level processes, sentences can change information states of hearers by elimination of all models from a current set that do not satisfy the formula. This folklore idea of range-with-elimination underlies the account of conversation in Stalnaker 1978, or the "update semantics" of Veltman 1996, where meanings are potentials for changing information states. Thus, dynamic linguistic acts come into the scope of logic, making them amenable to the compositional analysis that has served truth-conditional views in the past.

Richer versions: social dynamics and games Classical semantics has no actors, since it is about bare relationships with reality. Dynamic semantics is about single agents that compute on discourse structures, or change single minds. But language is about speakers and hearers that create shared meanings, and over that channel, engage in meaningful activities, cooperative or competitive. This requires a study of information flow in multi-agent communication where "social" knowledge about what others know and mutual expectations are crucial. And beyond single speech acts, there is a longer-term strategic aspect. I choose my words toward an end, depending on how I think you will take them, and next, so do you. Such behavior over time is the realm of game theory. Dynamic logics and games will be discussed below.

Literature For dynamic semantics, see Dekker 2008, Groenendijk, Stokhof and Veltman 1996, and for discourse representation theory, Kamp and Reyle 1993. Other sources include speech act theory (Searle and Vanderveken 1985) and "score-keeping games" (Lewis 1979). The "Dynamics" chapter in van Benthem and ter Meulen eds. 1997 (updated in 2010 with an appendix on "Information Dynamics") adds links to dynamic logics, artificial intelligence, and computational linguistics. Kamp and Stokhof 2007 has extensive philosophical reflection.

2 Parallels with computer languages and computation

The unity of languages Philosophy has long known a tension between formal and natural language methods, with Russell's "Misleading Form Thesis" claiming that natural language obscures the logical form of statements. By contrast, Montague proclaimed the unity of formal and natural languages in their design principles and theoretical properties. Later authors tested "Montague's Thesis" on *programming languages*: formally designed, but driving a real communicative practice between humans and machines. They found striking analogies with natural language, from category structure to paradoxes of intensionality.

Semantics of programs: computing change A major challenge in logics of computation is giving a meaning to imperative programs. These are not propositions that are true or false, but instructions for changing states of a computer, or indeed any process. As it turns out, first-order assignment semantics is a great fit here, allowing for a compositional definition of the relation of *successful transition* between assignments for program execution:

$s_1 [[\pi]]^M s_2$: there is a successful execution of program π starting in s_1 and ending in s_2.

Here the assignments s, originally an auxiliary device, become important in their own right, as memory states of a computer. A typical case are atomic programs $x:=t$, where the assignment s_1 changes to one with all values for variables the same, except that x is now set to $[[t]]^M{}_{s1}$, the value of the term t in $\boldsymbol{M}$ under the old assignment s_1. In a picture, we now view meanings in terms of transition arrows between states:

$$s_1 \xrightarrow{\quad\pi\quad} s_2$$

Thus, action is identified with pairs <*input state, output state*>. There are richer process views in computer science, but we will use this simple extensional format in what follows.

Compositional structure Like propositions, programs have complex syntactic structure, and their interpretation proceeds inductively: we match them with semantic ones. Here are three basic operations (the textbook Harel 1987 explains them with kitchen recipes):

Sequential composition	$\pi_1 \; ; \pi_2$
Guarded choice	*IF* φ *THEN* π_1 *ELSE* π_2
Iteration	*WHILE* φ *DO* π

For instance, the semi-colon *;* denotes sequential composition of relations: its transitions arise from first making a successful transition for π_1, and then one for π_2. The *WHILE* loop is unbounded, we make a computer run for as long as it takes to achieve $\neg\varphi$.

Logics of programs The oldest computational program logic is *Hoare calculus* of *correctness assertions* that express what a given program does in terms of standard propositions:

$\{\varphi\}\ \pi\ \{\psi\}$ "after *every* successful execution of program π starting from a state where *precondition* φ holds, *postcondition* ψ holds":

Note the co-existence of programs π and propositions φ: there is no conflict between statics and dynamics. Also, non-determinism is allowed: a program may have several executions. Now we can do program logic, much as rules for connectives analyze static propositions:

Example Rules of the Hoare Calculus.

$$\frac{\{\varphi\}\ \pi_1\ \{\psi\} \qquad \{\psi\}\ \pi_2\ \{\chi\}}{\{\varphi\}\ \pi_1\ ;\ \pi_2\ \{\chi\}} \qquad \text{composition}$$

$$\frac{\{\varphi \wedge \chi\}\ \pi_1\ \{\psi\} \qquad \{\varphi \wedge \neg\chi\}\ \pi_2\ \{\psi\}}{\{\varphi\}\ IF\ \chi\ THEN\ \pi_1\ ELSE\ \pi_2\ \{\psi\}} \qquad \text{guarded choice}$$

$$\frac{\{\varphi\}\ \pi\ \{\varphi\}}{\{\varphi\}\ WHILE\ \psi\ DO\ \pi\ \{\varphi \wedge \neg\psi\}} \qquad \text{iteration}$$

Checking soundness of these rules will make you understand a lot about the logic of change.

From programs to general actions The preceding describes a logic of action for any dynamic event, not just shifts in variable assignments or states of a computer. Logics like this have been applied to natural language, as we shall see, but also to conversation, strategies in games, and quantum-mechanical measurements that change a physical system. In this process, a reversal of perspective has occurred. Thinking of all the dynamic effects of using language, we might consider the computational stance as primary. Van Benthem 1996 even claims that *natural language is a programming language* for cognitive actions.

3 Technical background: dynamic logic of action

Behind program semantics lies a logic familiar to philosophers studying intensional notions.

Modal logics of process graphs Consider *process graphs* $\boldsymbol{M} = (S, \{R_a\}, V)$ with a set of states S, a family of binary transition relations R_a for basic actions (sometimes written $\rightarrow_a$), and a valuation V interpreting proposition letters p as local properties of states. Over such models, one can interpret a language with labeled modalities over "action-accessible states":

$$\boldsymbol{M}, s \models [a]\varphi \text{ iff } \boldsymbol{M}, t \models \varphi \text{ for all } t \text{ with } s\, R_a\, t.$$

The dual existential modality $\langle a\rangle\varphi$ is defined as $\neg[a]\neg\varphi$. Hoare correctness statements are modal implications $\varphi \rightarrow [\pi]\psi$. (A good textbook is Blackburn, de Rijke and Venema 2000.) Major uses of modal logic today are *action* and *knowledge*, a notion that will return later.

Complex actions: propositional dynamic logic The same formalism can deal with complex actions. The language now has components, one of *programs (P)* and one of *formulas (F)*:

$F \quad := \quad$ *atomic propositions* $\mid \neg F \mid (F \wedge F) \mid [P]F$

$P \quad := \quad$ *atomic actions* $\mid (P;P) \mid (P \cup P) \mid P^* \mid F?$

For elegance and sweep, program operators are now the *regular operations* of composition, Boolean choice, Kleene iteration, and tests for formulas. The semantics matches the mutual recursion in the syntax. $\boldsymbol{M}, s \models \phi$ says that ϕ *is true at state* s, while $\boldsymbol{M}, s_1, s_2 \models \pi$ says the transition from s_1 to s_2 is a *successful execution* of program π. Here are a few key clauses:

- $\boldsymbol{M}, s \models [\pi]\phi$ iff *for all* s' *with* $\boldsymbol{M}, s, s' \models \pi$, *we have* $\boldsymbol{M}, s' \models \phi$
- $\boldsymbol{M}, s_1, s_2 \models a$ iff $(s_1, s_2) \in R_a$

  $\boldsymbol{M}, s_1, s_2 \models \pi_1;\pi_2$ iff *there is an* s_3 *with* $\boldsymbol{M}, s_1, s_3 \models \pi_1$ *and* $\boldsymbol{M}, s_3, s_2 \models \pi_2$

  $\boldsymbol{M}, s_1, s_2 \models \pi_1 \cup \pi_2$ iff $\boldsymbol{M}, s_1, s_2 \models \pi_1$ *or* $\boldsymbol{M}, s_1, s_2 \models \pi_2$

  $\boldsymbol{M}, s_1, s_2 \models \pi^*$ iff *some finite sequence of* π*–transitions in* $\boldsymbol{M}$ *connects the state* s_1 *with the state* s_2

  $\boldsymbol{M}, s_1, s_2 \models \phi?$ iff $s_1 = s_2$ *and* $\boldsymbol{M}, s_1 \models \phi$

Axiomatic system Propositional dynamic logic has a natural proof system *PDL*. We give it here to show how logics of action and change can be designed just like classical logics:

- All principles of the minimal modal logic for all modalities $[\pi]$
- Computation rules for decomposing program structure:

  $[\pi 1;\pi 2]\phi \leftrightarrow [\pi 1][\pi 2]\phi$

  $[\pi 1 \cup \pi 2]\phi \leftrightarrow [\pi 1]\phi \wedge [\pi 2]\phi$

  $[\phi?]\psi \leftrightarrow (\phi \rightarrow \psi)$

  $[\pi^*]\phi \leftrightarrow \phi \wedge [\pi][\pi^*]\phi$
- The Induction Axiom $(\phi \wedge [\pi^*](\phi \rightarrow [\pi]\phi)) \rightarrow [\pi^*]\phi$.

PDL can derive all Hoare rules, while generalizing modal logic and relational algebra. All its theorems are valid, and there is a nice completeness proof. And in all this, *PDL* is *decidable*. *PDL* distils the "essence of computability": cf. Harel, Kozen and Tiuryn 2000.

Process theories: the larger picture The above suggests a study of *process equivalences*, as processes have many natural levels of detail. Also, the decidability of *PDL* suggests a balance between the *expressive power* of a logic and the *computational complexity* of its laws. This balance is also crucial to understanding natural language. Finally, *PDL* only studies sequential operations on programs, while *parallel composition* is the reality in network computing. This deep subject is pursued in Lambda Calculus, a system close to Montague's work, and Process Algebra (Bergstra et al. 2001). Parallel computation is *simultaneous action*, which can also be modeled by games (Abramsky 2008). Interestingly, our linguistic performance, too, is a parallel composition of grasping meanings and engaging in discourse at the same time.

4 Dynamic semantics of natural language sentences

Many systems of interpretation highlight actions that deal with anaphora, temporality, and many other expressions. A major paradigm is Discourse Representation Theory (Kamp and Reyle 1993), but we present an approach due to Groenendijk and Stokhof 1991.

Translation lore *Dynamic predicate logic* (*DPL*) "dynamifies" first-order logic. Here is why. One is usually taught some folklore to make first-order formulas fit actual linguistic forms:

1	A man came in. He whistled.	The two underlined phrases can co-refer.
2	* No man came in. He whistled.	The two underlined phrases cannot co-refer.
3	* He whistled. A man came in.	The two underlined phrases cannot co-refer.
4	If a man came in, he whistled.	The two underlined phrases can co-refer.

The obvious translation $\exists x\, Cx \wedge Wx$ for 1 does not give the right scope, and one uses a bracket trick: $\exists x\, (Cx \wedge Wx)$. The translation for 2: $\neg\exists x\, Cx \wedge Wx$ does give the right scope, the quantifier does not bind the free variable in Wx. The translation for 3: $Wx \wedge \neg\exists x\, Cx$, too, is correct. But the translation for 4: $\exists x\, Cx \rightarrow Wx$ has the wrong scope, and one uses brackets plus a quantifier-change (though the sentence has $\rightarrow$ as its main operator): $\forall x\, (Cx \rightarrow Wx)$.

Dynamifying standard first-order semantics *DPL* assigns dynamic meanings without tricks. It reinterprets first-order formulas ϕ as *evaluation procedures*, transition relations between assignments like with programs:

Atoms as tests

$\boldsymbol{M}, s_1, s_2 \models Px$ iff $s_1 = s_2$ *and* $I^{\boldsymbol{M}}(P)(s_1(x))$

Conjunction as composition

$\boldsymbol{M}, s_1, s_2 \models \phi \wedge \psi$ iff *there is* s_3 *with* $\boldsymbol{M}, s_1, s_3 \models \phi$ *and* $\boldsymbol{M}, s_3, s_2 \models \psi$

Negation as failure test

$\boldsymbol{M}, s_1, s_2 \models \neg\phi$ iff $s_1 = s_2$ *and there exists no* s_3 *with* $\boldsymbol{M}, s_1, s_3 \models \phi$

Existential quantification as random reset

$\boldsymbol{M}, s_1, s_2 \models \exists x$ iff $s_2 = s_1[x{:=}d]$ *for any object d in the domain.*

Example Dynamic evaluation and bindings explained.

(1) Evaluating $\exists x\ Cx \wedge Wx$ composes a random reset with two successive test actions. This moves from states s to states $s[x{:=}d]$ where both $C(d)$, $W(d)$ hold. (2) $Wx \wedge \exists x\ Cx$ composes a test, a random reset, and one more test. This moves from states s where $W(s(x))$ holds to states $s[x{:=}d]$ where $C(d)$ holds: no binding achieved. (3) The non-binding is explained by the negation test, which leaves no new value for x to co-refer. (4) To get the implications right, we define $\phi \rightarrow \psi$ as $\neg(\phi \wedge \neg\psi)$. This works out to a new test: *every successful execution of* ϕ *can be followed by one for* ψ. This does what it should for both implications. ■

Logic as evaluation algebra Conceptually, *DPL* makes predicate logic a theory of two basic actions: variable resets and atomic tests. “Standard logic” then becomes a mix of general relation algebra, at the level of the dynamic logic *PDL,* plus specific laws for reset actions on first-order models. Thus we see an intriguing fact. The *basic process logic* of evaluation is decidable – the undecidability of first-order logic arises from debatable special mathematical features of assignments (van Benthem 1996). *DPL* views also apply to discourse, suggesting notions of “dynamic inference”: cf. the mentioned sources.

We leave the reader with a thought. If meaning is dynamic, what computational process drives natural language? Do typical program structures like *WHILE* and * iteration make sense?

5 Logical dynamics of conversation

We now move from sentences to discourse, and information flow in communication.

Example Cooperative questions and answers.

I ask you in Amsterdam: “Is this building the Rijksmuseum?”. You answer: “Yes”. This is a simple thing we all do all the time, but subtle information flows. By asking the question, I tell you that I do not know the answer, and that I think you may know. And by answering, you do not just convey a topographical fact – you also make me know that you know, and as a result, you know that I know that you know, etc. ■

Common knowledge, at every depth of iteration, mixes factual information and social information about what others know. The latter is the glue of communication, according to philosophers, economists, and psychologists. Hence we need logics that treat information flow with actors on a par. We do this by "dynamifying" the static logic of knowledge:

Epistemic logic The *epistemic language EL* extends propositional logic with modal operators $K_i\phi$ (*i knows that* ϕ), for each agent i in a total group I, and $C_G\phi$: ϕ *is common knowledge* in the subgroup G. The inductive syntax rule is as follows:

$$p \mid \neg\phi \mid \phi \vee \psi \mid K_i\phi \mid C_G\phi$$

This language describes the Question/Answer scenario with formulas like

(i) $\neg K_Q\varphi \wedge \neg K_Q\neg\varphi$ (***Q*** does not know whether φ),

(ii) $\neg K_Q\neg(K_A\varphi \vee K_A\neg\varphi)$ (***Q*** thinks that ***A*** may know the answer).

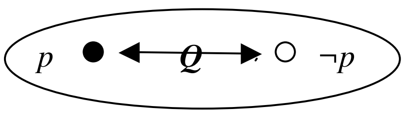
After communication, we have $K_A\varphi \wedge K_Q\varphi$, $K_QK_A\varphi \wedge K_AK_Q\varphi$, and even $C_{\{Q, A\}}\varphi$.

Formally, consider models $\boldsymbol{M} = (W, \{\sim_i \mid i \in G\}, V)$, with worlds W, accessibility relations $\sim_i$ for agents $i \in G$ between worlds, and V a valuation as usual. *Pointed models* $(\boldsymbol{M}, s)$ have an actual world s for the true state of affairs (perhaps unknown to the agents). Here accessibility no longer encodes actions, but *information ranges*: the options agents see for the actual world. Further conditions on $\sim_i$ encode special assumptions about agents' powers of observation and introspection. Very common are *equivalence relations*: reflexive, symmetric, and transitive. Such "information diagrams" interpret the epistemic language. Here are the key clauses:

$\boldsymbol{M}, s \models K_i\phi$ iff *for all t with* $s \sim_i t$: $\boldsymbol{M}, t \models \phi$

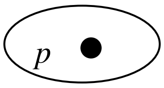
$\boldsymbol{M}, s \models C_G\phi$ iff *for all t that are reachable from s by some finite sequence of arbitrary* $\sim_i$ *steps* ($i \in G$): $\boldsymbol{M}, t \models \phi$

We draw one model for a simple question answer episode (omitting reflexive arrows). Agent ***Q*** does not know if p, but ***A*** is informed about it:



In the actual world (the black dot), the following formulas are true: p, K_Ap, $\neg K_Qp \wedge \neg K_Q\neg p$, $K_Q(K_Ap \vee K_A\neg p)$, $C_{\{Q, A\}} (\neg K_Qp \wedge \neg K_Q\neg p)$, $C_{\{Q, A\}}(K_Ap \vee K_A\neg p)$. This is a good reason for ***Q*** to ask ***A*** about p: he knows that she knows the answer.

After the answer "Yes", intuitively, this model changes to the following one-point model:



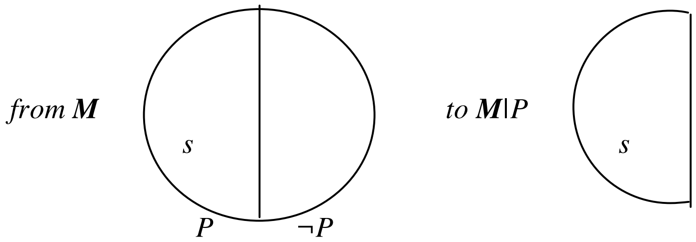
.

Now, common knowledge $C_{\{Q, A\}}p$ holds at the actual world.

Epistemic logic sharpens intuitive distinctions about information, especially levels of group knowledge. Communication often turns *implicit* group knowledge into explicit knowledge.

Axiom systems for epistemic inference Complete logics capturing epistemic reasoning about oneself and others are known (Fagin et al. 1995). The base system is a minimal modal logic. Structural restrictions to equivalence relations add *S5* axioms of introspection, while the complete logic of common knowledge can be axiomatized with *PDL*-techniques.

A dynamic turn: public update by elimination Now for the logical dynamics of information flow. A pilot system for exploring this starts from a folklore view: an event $!P$ yielding the information that P is true *shrinks the current model* to just those worlds that satisfy P. This is called *public hard information*. More precisely, for any epistemic model $\boldsymbol{M}$, world s, and P true at s, the model $(\boldsymbol{M}|P, s)$ ('***M*** *relativized to P at s*') is the sub-model of $\boldsymbol{M}$ whose domain is the set $\{t \in \boldsymbol{M} \mid \boldsymbol{M}, t \models P\}$. Drawn in a simple picture, an update step then goes



This mechanism models communication, but also acts of public *observation*. It has been applied to games and other social scenarios. Crucially, truth values of formulas may change after update: agents who did not know that P now do. We need a logic to keep things straight.

Dynamic logic of public announcement The language of *public announcement logic PAL* adds action expressions to *EL*, plus matching dynamic modalities, defined by the syntax rules:

Formulas	P:	$p \mid \neg\phi \mid \phi\vee\psi \mid K_i\phi \mid C_G\phi \mid [A]\phi$
Action expressions	A:	$!P$

Here the semantic clause for the dynamic action modality "looks ahead" between models:

$$\boldsymbol{M}, s \models [!P]\, \phi \qquad \text{iff} \qquad \textit{if } \boldsymbol{M}, s \models P, \textit{ then } \boldsymbol{M}|P, s \models \phi$$

A typical assertions here is $[!P]K_i\phi$, which states what agent i will know after receiving hard information that P. Reasoning about information flow revolves around a dynamic *recursion equation* that relates new knowledge to old knowledge an agent had before:

The following equivalence is valid for *PAL*: $[!P]K_i\phi \leftrightarrow (P \rightarrow K_i(P \rightarrow [!P]\phi))$.

The reader may find it helpful to prove this. The complete and decidable logic for knowledge under public communication is well-understood. *PAL* is axiomatized by any complete epistemic logic over static models plus *recursion axioms*

$[!P]q$	$\leftrightarrow$	$P \rightarrow q$	*for atomic facts* q
$[!P]\neg\phi$	$\leftrightarrow$	$P \rightarrow \neg[!P]\phi$	
$[!P]\phi\wedge\psi$	$\leftrightarrow$	$[!P]\phi \wedge [!P]\psi$	
$[!P]K_i\phi$	$\leftrightarrow$	$P \rightarrow K_i(P \rightarrow [!P]\phi)$	

There is more here than meets the eye. The logic *PAL* uncovers many subtleties of natural language. Suppose that in our question-answer episode, ***A*** had not said *!P*, but the equally true "You don't know that *P*, but it is true" ($\neg K_Q P \wedge P$). The latter "Moore sentence" achieves the same update, but it has become *false* afterwards! Statements switching their own truth values are essential in conversation, puzzles, and games (Geanakoplos and Polemarchakis 1982). For much more about *PAL* and related systems, including links to epistemology, cf. Baltag, Moss and Solecki 1998, van Ditmarsch, van der Hoek and Kooi 2007, van Benthem 2010.

Compared to dynamic semantics, logical dynamics of information flow has a discourse focus. This is "pragmatics" – but from a logical point of view, the border with semantics is thin.

6 The logical dynamics of agency

Public announcement is just the start of a dynamics of interactive agency. We mention a few dimensions of a richer picture of what language users are and do.

From knowledge to belief Language users do not just have knowledge, but also *beliefs*. What they hear involves belief revision (Gaerdenfors 1988), and this process can be triggered by information that is "soft" rather than hard, depending on the reliability of the source. Dynamic logics for belief revision "dynamify" static doxastic logics, where agents believe what is true in their "most plausible" worlds. This time, update does not eliminate worlds: it transforms the *relative plausibility* that agents assign to worlds. Forming and correcting beliefs is a learning ability that is more essential to human intelligence than just recording hard information. Rationality is not being correct all the time, but having a talent for *correction*.

Private information Information flow is driven by a differential: we do not all know the same things. Dynamic epistemic logics can also model *private* communication in a group (think of emails with *bcc* rather than *cc*), a phenomenon of high complexity. They even deal with lying and cheating, a central topic in real language, since most communication is unreliable to some extent. The usual Gricean focus on helpful truthful communication seems otherwordly.

Questions and issue management Questions do not just convey information, they also *direct discourse* by raising and modifying topics. This is crucial to language, communication and inquiry. A logical dynamics of questions must represent *issues* and actions modifying these. Two recent flavors are Groenendijk 2009, van Benthem and Minica 2009.

Preference and evaluation dynamics Agency is not just information dynamics. Rational decision and strategic interaction involve a balance of information and *evaluation*, encoded in our preferences. Entanglement of information and evaluation, and preference change pervade deontic logic (Gruene Yanoff and Hansen 2010) and games. But again natural language remains close. We often change our evaluation of situations by speech acts such as suggestions or *commands* from some moral or esthetic authority. Philosophers have also drawn attention to the normative character of "discourse obligations" (Brandom 1994).

And beyond Further relevant features of agency studied in logic include *trust*, *intentions* and *commitments*, but we the picture should be clear by now. Dynamic analysis of language may at the same time have to be an analysis of the agents using that language.

7 Games in logic and natural language

Beyond logics of agency lies a world of *games*. Dynamic logics describe single update steps, but the next level of language use is strategic behaviour over time. What I say now responds to what you say, but it is usually directed toward a future goal, and part of a long-term plan. We also saw that language involves iterated social knowledge of agents about each other and multi-agent equilibrium. Both have received sophisticated treatments in game theory.

Evaluation games and "game-theoretic semantics" We ilustrate games for natural language with a simple pilot system, again in terms of first-order logic (Hintikka and Sandu 1997).

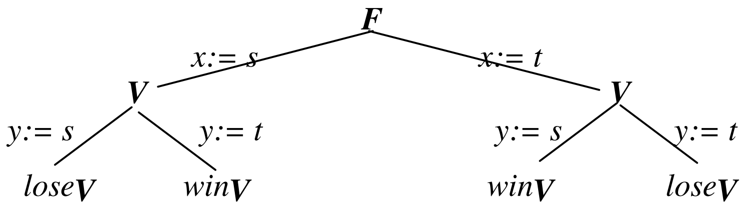
Evaluation of formulas ϕ can be cast as a game for two players. *Verifier* ***V*** claims that ϕ is true in a model ***M****, s*, *Falsifier* ***F*** that it is false. Here are natural moves of defense and attack:

atoms *Pd, Rde,…*	***V*** wins if the atom is true, ***F*** if it is false
disjunction $\phi \vee \psi$	***V*** *chooses* which disjunct to play
conjunction $\phi \wedge \psi$	***F*** *chooses* which conjunct to play
negation $\neg\phi$	*role switch* between the players, play continues with ϕ

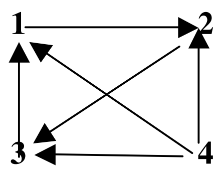
Next, the quantifiers make players look inside ***M***'s domain of objects:

$\exists x\ \phi(x)$	***V*** *picks an object d*, and play continues with $\phi(d)$
$\forall x\ \phi(x)$	the same move, but now for ***F***

The game schedule is determined by the form of ϕ. Consider a model ***M*** with two objects *s, t*. Here is the complete game for $\forall x\ \exists y\ x \neq y$ in ***M***, pictured as a tree of possible moves:



This is a game of perfect information: players know throughout what has happened. Branches are the possible plays, with 2 wins for each player. But ***V*** is the player with a *winning strategy*, she has a rule for always winning. For a more exciting example, look at a network with arrows for directed communication links (all self-loops are present but not drawn):



The formula $\forall x\ \forall y\ (Rxy \vee \exists z(Rxz \text{ and } Rzy))$ says that any two nodes can communicate in two steps. Just analyze the game and see who can win.

Logic and games The crucial fact about evaluation games is this equivalence:

A formula is true iff Verifier has a winning strategy,

while it is false iff Falsifier has a winning strategy. This follows from *Zermelo's Theorem* on "determined games", a tepping stone toward solution procedures for games with richer preferences than winning and losing ("Backward Induction"). In evaluation games, logical constants change from "control expressions" for procedures to *game actions* like making a choice or performing a role switch. Thus, at the heart of natural language, there is a multi-agent game algebra of its users. Actually, in game theory, the norm is *imperfect information*: players need not know exactly where they are in a game tree (think of card games). Hintikka and Sandu 1997 claim that "branching patterns" of independence between quantifiers in natural langage involve imperfect information about objects chosen by one's opponent.

Evaluation games are not a realistic account of discourse, which is usually about consistency rather than truth. But modern logic has many further games that are relevant.

"Logic games" Games in logic analyze argumentation (Lorenzen 1955), compare models (Ehrenfeucht 1957), construct models (Hodges 1985), etc. In each case, winning strategies for various agents encode basic notions. A winning argumentation strategy is a proof for the claim, or if the opponent wins, a counter-model. Games are at the heart of modern logic, and quantification is deeply tied to dependent action.

Signaling games for meaning A quite different use of games in language has emerged in Parikh 2000, Jaeger and van Rooij 2007, and Gaerdenfors and Warglien 2006. These start from the *signaling games* in Lewis 1969 that analyze basic lexical meanings. We have situations and linguistic objects that can represent them. Agents might choose any association, but stable conventions are Nash Equilibria in a game where a Sender chooses a coding scheme and Receiver a decoding, with some plausible assumptions on their utility functions. Thus meanings become *equilibria in language games*. Richer infinite "evolutionary games" can even explain diachronic phenomena, or emergence of linguistic conventions, using thought experiments in terms of fitness and stability against invaders.

Integrating different games Signaling games are very different from logic games, where meanings are given. Integration of these perspectives on natural language is an open problem.

Logic and game theory We have discussed special games for linguistic and logical activities. But there is also an interface of logic and general game theory, in the study of strategies, information and reasoning of agents. This involves epistemic, doxastic and dynamic logics for analysing rational play and game solution. This relates to the sense in which computer scientists have embraced games in *multi-agent systems* (Shoham and Leighton-Brown 2008), and philosophers in epistemology (Stalnaker 1999). While this interface is not disjoint from language and logic games, we will not pursue it here: cf. van der Hoek and Pauly 2006.

Coda: temporal perspective Lexical meaning assignment, evaluation, or argumentation are special-purpose short-term processes. These run against the backdrop of an infinite linguistic process over time: the “operating system” of natural language. Here language dynamics meets with temporal logics (Parikh and Ramanujam 2003, Belnap, Horty and Xu 2001), learning theory (Kelly 1996), and infinite computational processes (Graedel, Thomas and Wilke 2003).

8 Discussion: putting the dynamics together

Dynamic semantics versus dynamic logic Dynamics in this article has two different strands. “Dynamic semantics” is a *new account of meaning*, replacing truth-conditional accounts, and generating “nonstandard logics”. By contrast, dynamic logics of information update keep the old language with its semantics and logic, but *add dynamic events* as a new layer. The former approach is implicit: the dynamics “loads” the meaning of the old language, while the latter approach is explicit, the dynamics occurs in operators extending the old language. The implicit/explicit contrast occurs widely in logic – but what fits natural language best?

Combined architectures How can we turn our carrousel of dynamic activities and games into one integrated story of language use? Combining logics can be tricky. Simple decidable logics for knowledge and time combine into highly undecidable logics for agents with perfect memory (Halpern and Vardi 1989). No integration is known for dynamics, and we may first need an account of “linguistic agents”, the way Turing analyzed “computing agents”.

Cognitive realities Finally, natural language is an interface where logic meets reality – and so dynamic logics meet cognitive science. Van Benthem 2010 proposes studying language in a broad sense here, including “successful insertions” of new logic-inspired behavior.

9 Conclusion

We have shown how natural language meets with dynamic logics of meaning and agency, leading to new interfaces between logic, linguistics, computer science and game theory.

10 References

Abramsky, S. (2008) ‘Information, Processes and Games’, in P. Adriaans and J. van Benthem (eds) Handbook of the Philosophy of Information, Amsterdam: Elsevier, 483–549.

Baltag, A, L. Moss and S. Solecki (1998), “The Logic of Public Announcements, Common Knowledge and Private Suspicions”, in Proceedings TARK 1998, 43–56, Los Altos: Morgan Kaufmann.

Belnap, N., M. Perloff and M. Xu (2001) Facing the Future, Oxford: Oxford University Press.

Bergstra, J., A. Ponse and S. Smolka (eds) (2001) Handbook of Process Algebra, Amsterdam: Elsevier.

Blackburn, P., M. de Rijke and Y. Venema (2000) Modal Logic, Cambridge UK: Cambridge University Press.

Brandom, R. (1994) Making It Explicit: Reasoning, Representing, and Discursive Commitment, Cambridge MA: Harvard University Press.

Dekker, P. (2008) "A Guide to Dynamic Semantics", Amsterdam: Institute for Logic, Language and Computation.

van Ditmarsch, H., W. van der Hoek and B. Kooi (2007) Dynamic Epistemic Logic, Dordrecht: Springer.

A. Ehrenfeucht, A. (1957) "Application of Games to Some Problems of Mathematical Logic", Bulletin Polish Academy of Sciences Cl. III, 35–37.

Fagin, R., J. Halpern, M. Vardi and Y. Moses (1995) Reasoning about Knowledge, Cambridge MA: MIT Press.

Gaerdenfors, P. (1988) Knowledge in Flux, Cambridge MA: MIT Press.

Gaerdenfors, P. and M. Warglien (2006) "Cooperation, Conceptual Spaces, and the Evolution of Semantics", in: P. Vogt et al. (eds) Symbol Grounding and Beyond, Heidelberg: Springer, 16–30.

Geanakoplos, J. and H. Polemarchakis (1982) "We Can't Disagree Forever", Journal of Economic Theory 28, 192–200.

Grädel, E., W. Thomas and Th. Wilke (eds) (2003) Automata, Logics, and Infinite Games, Heidelberg: Springer.

Groenendijk, J. (2009) "Inquisitive Semantics: Two Possibilities for Disjunction", in P. Bosch, D. Gabelaia and J. Lang (eds) Proceedings Seventh Tbilisi Symposium on Language, Logic and Computation, Heidelberg: Springer, 80–94.

Groenendijk, J. and M. Stokhof (1991) "Dynamic Predicate Logic", Linguistics and Philosophy 14, 39–100.

Groenendijk, J., M. Stokhof and F. Veltman (1996), "Coreference and Modality", in S. Lappin (ed.) Handbook of Contemporary Semantic Theory, Oxford: Blackwell, 179–216.

Grüne Yanoff, T. and S-O Hansen (2010) Preference Change, Dordrecht: Springer.

Halpern, J. and M. Vardi (1989) "The Complexity of Reasoning about Knowledge and Time", Journal of Computer and System Sciences 38, 195–237.

Harel, D. (1987) Algorithmics: The Spirit of Computing, Reading MA: Addison-Wesley.

Harel, D., D. Kozen, and J. Tiuryn (2000) Dynamic Logic, Cambridge MA: MIT Press.

Hintikka, J. and G. Sandu (1997) "Game-Theoretical Semantics", in: J. van Benthem and A. ter Meulen (eds) Handbook of Logic and Language, 361–410.

Hodges, W. (1985) Building Models by Games, Cambridge UK: Cambridge University Press.

Jaeger, G and R. van Rooij (2007) "Language Structure: Psychological and Social Constraints", Synthese 159, 99–130.

Kamp, H. and U. Reyle (1993) From Discourse to Logic, Dordrecht: Kluwer.

Kamp, H. and M. Stokhof (2008) "Information in Natural Language," in P. Adriaans and J. van Benthem (eds) Handbook of the Philosophy of Information, Amsterdam: Elsevier, 49–111.

Kelly, K. (1996) The Logic of Reliable Inquiry, Oxford: Oxford University Press.

Lewis, D. (1969) Convention, Cambridge MA: Harvard University Press.

Lewis, D. (1979) "Scorekeeping in a Language Game", Journal of Philosophical Logic 8, 339–359.

Meulen, A. ter (1995) Representing Time in Natural Language: The Dynamic. Interpretation of Tense and Aspect, Cambridge MA: MIT Press.

Montague, R. (1974) Formal Philosophy, New Haven: Yale University Press (edited by R. H. Thomason).

Parikh, P. (2000) The Use of Language, Stanford: CSLI Publications.

Parikh R. and R. Ramanujam (2003) "A Knowledge-Based Semantics of Messages", Journal of Logic, Language and Information 12, 453–467.

Searle, J. and D. Vanderveken (1985) Foundations of Illocutionary Logic. Cambridge UK: Cambridge University Press.

Shoham Y. and K. Leyton-Brown (2008) Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations, Cambridge UK: Cambridge University Press.

Stalnaker, R. (1978) "Assertion", in P. Cole (ed.) Syntax and Semantics 9, New York: Academic Press, 315–32.

Stalnaker, R. (1999) "Extensive and Strategic Form: Games and Models for Games", Research in Economics 53, 293-291.

Van Benthem, J. (1996) Exploring Logical Dynamics, Stanford: CSLI Publications.

Van Benthem, J. (1999) Logic in Games, Amsterdam: Institute for Logic, Language and Computation. To appear in Texts in Logic and Games, Springer.

Van Benthem, J. (2010) Logical Dynamics of Information and Interaction, Cambridge: Cambridge University Press.

Van Benthem, J. and A. ter Meulen, eds. (1997) Handbook of Logic and Language, Amsterdam: Elsevier Science Publishers.

Van der Hoek, W. and M. Pauly (2006) “Modal Logic for Games and Information”, in P. Blackburn, J. van Benthem and F. Wolter (eds) Handbook of Modal Logic, Amsterdam: Elsevier Science Publishers, 1077–1148.

Van Leeuwen, J. (ed.) (1990) Handbook of Theoretical Computer Science, Amsterdam: Elsevier.

Veltman, F. (1996), “Defaults in Update Semantics”, Journal of Philosophical Logic 25 221–261.