

PDF version of the entry
The Logic of Action
<http://plato.stanford.edu/archives/fall2016/entries/logic-action/>
from the FALL 2016 EDITION of the

STANFORD ENCYCLOPEDIA OF PHILOSOPHY



Edward N. Zalta Uri Nodelman Colin Allen R. Lanier Anderson
Principal Editor Senior Editor Associate Editor Faculty Sponsor

Editorial Board
<http://plato.stanford.edu/board.html>

Library of Congress Catalog Data
ISSN: 1095-5054

Notice: This PDF version was distributed by request to members of the Friends of the SEP Society and by courtesy to SEP content contributors. It is solely for their fair use. Unauthorized distribution is prohibited. To learn how to join the Friends of the SEP Society and obtain authorized PDF versions of SEP entries, please visit <https://leibniz.stanford.edu/friends/>.

Stanford Encyclopedia of Philosophy
Copyright © 2016 by the publisher
The Metaphysics Research Lab

Center for the Study of Language and Information
Stanford University, Stanford, CA 94305

The Logic of Action
Copyright © 2016 by the authors
Krister Segerberg, John-Jules Meyer, and Marcus Kracht

All rights reserved.

Copyright policy: <https://leibniz.stanford.edu/friends/info/copyright/>

The Logic of Action

First published Tue Mar 31, 2009; substantive revision Tue Apr 2, 2013

In this article we provide a brief overview of the logic of action in philosophy, linguistics, computer science, and artificial intelligence. The logic of action is the formal study of action in which formal languages are the main tool of analysis.

The concept of action is of central interest to many disciplines: the social sciences including economics, the humanities including history and literature, psychology, linguistics, law, computer science, artificial intelligence, and probably others. In philosophy it has been studied since the beginning because of its importance for epistemology and, particularly, ethics; and since a few decades it is even studied for its own sake. But it is in the logic of action that action is studied in the most abstract way.

The logic of action began in philosophy. But it has also played a certain role in linguistics. And currently it is of great importance in computer science and artificial intelligence. For our purposes it is natural to separate the accounts of these developments.

- 1. The Logic of Action in Philosophy
 - 1.1 Historical overview
 - 1.2 The stit saga
 - 1.3 Intentions
 - 1.4 Logics of special kinds of action
- 2. The Logic of Action in Linguistics
 - 2.1 Speech acts
 - 2.2 Action sentences
 - 2.3 Dynamic semantics
- 3. The Logic of Action in Computer Science

- 3.1 Reasoning about programs
- 4. The Logic of Action in Artificial Intelligence
 - 4.1 Representing and reasoning about actions
 - 4.2 Description and specification of intelligent agents
- Conclusion
- Bibliography
- Academic Tools
- Other Internet Resources
- Related Entries

1. The Logic of Action in Philosophy

1.1 Historical overview

Already St. Anselm studied the concept of action in a way that must be classified as logical; had he known symbolic logic, he would certainly have made use of it (Henry 1967; Walton 1976). In modern times the subject was introduced by, among others, Alan Ross Anderson, Frederick B. Fitch, Stig Kanger, and Georg Henrik von Wright; Kanger's work was further developed by his students Ingmar Pörn and Lars Lindahl. The first clearly semantic account was given by Brian F. Chellas (1969). (For a more detailed account, see Segerberg 1992 or the mini-history in Belnap 2001.)

Today there are two rather different groups of theories that may be described as falling under the term logic of action. One, the result of the creation of Nuel Belnap and his many collaborators, may be called *stit theory* (a term that will be explained in the next paragraph). The other is *dynamic logic*. Both are connected with modal logic, but in different ways. Stit theory grew out of the philosophical tradition of modal logic. Dynamic logic, on the other hand, was invented by computer scientists in

order to analyse computer action; only after the fact was it realized that it could be viewed as modal logic of a very general kind. One important difference between the two is that (for the most part) actions are not directly studied in stit theory: the ontology does not (usually) recognize a category of actions or events. But dynamic logic does. Among philosophers such ontological permissiveness has been unusual. Hector-Neri Castañeda, with his distinction between propositions and practitions, provides one notable exception.

The stit tradition is treated in this section, the dynamic logic one in the next.

1.2 The stit saga

The term “stit” is an acronym based on “sees to it that”. The idea is to add, to an ordinary classical propositional language, a new propositional operator *stit*, interpreting $stit_i\phi$, where i stands for an agent and ϕ for a proposition, as *i sees to it that ϕ* . (The official notation of the Belnap school is more laborious: $[i\ stit : \phi]$.) Note that ϕ is allowed to contain nestings of the new operator.

In order to develop formal meaning conditions for the stit operator *stit* a semantics is defined. A *stit frame* has four components: a set T , the nodes of which are called moments; an irreflexive tree ordering $<$ of T ; a set of agents; and a choice function C . A maximal branch through the tree is called a history.

The tree $(T, <)$ seems to correspond to a naïve picture familiar to us all: a moment m is a temporary present; the set $\{n : n < m\}$ corresponds to the past of m , which is unique; while the set $\{n : m < n\}$ corresponds to the open future of m , each particular maximal linear subset of which corresponds to a particular possible future.

To formalize the notion of action, begin with two general observations:

- i. usually an agent is not able to select one possible future to become the unique actual future, but
- ii. by his action he can make sure that certain futures, which before his action are possible, are no longer possible after his action.

This is where the choice function C comes in: for each moment m and agent i , C yields a partitioning C_i^m of the set H_m of all histories through m . An equivalence class in C_i^m is called a choice cell. (Note that two histories belonging to the same choice cell agree up to the moment in question but not necessarily later on.) If h is a history running through m we write $C_i^m(h)$ for the choice cell of which h is a member. It is natural to associate C_i^m with the set of actions open to the agent i at m , and to think of the choice cell $C_i^m(h)$ as representing the action associated with h .

A *stit model* has an additional component: a valuation. A valuation in a frame, it turns out, is a function that assigns to a variable and each index either 1 (truth) or 0 (falsity), where an index is an ordered pair consisting of a history and a moment on that history. The notion of truth or falsity of a formula with respect to an index can now be defined. If V is the valuation we have the following basic truth-condition for atomic ϕ :

$$(h, m) \models \phi \text{ iff } V(\phi, (h, m)) = 1.$$

The truth-conditions for the Boolean connectives are as expected; for example,

$$\begin{aligned} (h, m) \models \neg\phi &\text{ iff } (h, m) \not\models \phi, \\ (h, m) \models \phi \wedge \psi &\text{ iff } (h, m) \models \phi \text{ and } (h, m) \models \psi. \end{aligned}$$

Let us write $\llbracket \phi \rrbracket_m$ for the set $\{h \in H_m : (h, m) \models \phi\}$, that is, the set of histories agreeing with h at least up to m and such that ϕ is true with

respect to the index consisting of that history and m . Defining formal truth-conditions for the stit operator there are at least two possibilities to be considered:

1. $(h, m) \models \text{stit}_i\phi$ iff $C_i^m(h) \subseteq \llbracket \phi \rrbracket_m$.
2. $(h, m) \models \text{stit}_i\phi$ iff $C_i^m(h) \subseteq \llbracket \phi \rrbracket_m$ and $\llbracket \phi \rrbracket_m \neq H_m$.

To distinguish the two different operators that those conditions define, the former operator is called the *Chellas stit*, written *cstit*, while the latter operator is called the *deliberative stit*, written *dstit*.

In words, *cstit* $_i\phi$ is true at an index (h, m) if ϕ is true with respect to h' and m , for all histories h' in the same choice cell at m as h ; this is called the positive condition. The truth-condition for *dstit* $_i\phi$ is more exacting; not only the positive condition has to be satisfied but also what is called the negative condition: there must be some history through m such that ϕ fails to be true with respect to that history and m .

Both *cstit* and *dstit* are studied; it is claimed that they capture important aspects of the concept “sees to it that”. The two operators become interdefinable if one also introduces the concept “it is historically necessary that”. Using \Box for historical necessity, define

$$(h, m) \models \Box\phi \text{ iff, for all } h' \in H_m, (h', m) \models \phi.$$

Then the formulas

$$\begin{aligned} \text{dstit}_i\phi &\leftrightarrow (\text{cstit}_i\phi \wedge \neg\Box\phi) \text{ and} \\ \text{cstit}_i\phi &\leftrightarrow (\text{dstit}_i\phi \vee \Box\phi) \end{aligned}$$

are true with respect to all indices.

One advantage of stit theory is that the stit analysis of individual action can be extended in natural ways to cover group action.

A number of the initial papers defining the stit tradition are collected in the volume Belnap 2001. One important later work is John F. Horty's book (2001). The logic of stit was axiomatized by Ming Xu (1998).

1.3 Intentions

Michael Bratman's philosophical analysis of the notion of intention has had a significant influence on the development of the logic of action within computer science. It will be discussed below.

1.4 Logics of special kinds of action

In a series of papers Carlos Alchourrón, Peter Gärdenfors and David Makinson created what they called a logic of theory change, later known as the AGM paradigm. Two particular kinds of change inspired their work: change due to deontic actions (Alchourrón) and change due to doxastic actions (Gärdenfors and before him Isaac Levi). Examples of deontic actions are derogation and amendment (laws can be annulled or amended), while contraction and expansion are analogous doxastic actions (beliefs can be given up, new beliefs can be added). Later the modal logic of such actions has been explored under the names *dynamic deontic logic*, *dynamic doxastic logic* and *dynamic epistemic logic*. (For the classic paper on AGM, see AGM 1985. For an introduction to dynamic deontic logic and dynamic doxastic logic, see Lindström and Segerberg 2006. We will return to this topic in Section 4, where it is viewed from the perspective of the field of artificial intelligence.

2. The Logic of Action in Linguistics

In linguistics, there are two ways in which actions play a role: on the one hand, utterances *are* actions and on the other they can be used to talk *about* actions. The first leads to the study of speech acts, a branch of

pragmatics, the second to the study of the semantics of action reports, hence is of a distinctly semantic nature. In addition to this, there is a special type of semantics, dynamic semantics, where meanings are not considered as state descriptions but as changes in the state of a hearer.

2.1 Speech acts

The study of speech acts goes back to Austin (1957) and Searle (1969). Both emphasise that using language is to perform certain acts. Moreover, there is not just one act but a whole gamut of them (Austin himself puts the number in the magnitude of 10^3). The classification he himself gives involves acts that are nowadays not considered as part of a separate science: the mere act of uttering a word (the *phatic act*) or sentence is part of phonetics (or phonology) and only of marginal concern here. By contrast, the *illocutionary* and *perlocutionary* acts have been the subject of intense study. An *illocutionary act* is the linguistic act performed by using that sentence; it is inherently communicative in nature. By contrast, the *perlocutionary act* is an act that needs surrounding social contexts to be successful. The act of naming a ship or christening a baby, for example, are perlocutionary. The sentence "I hereby pronounce you husband and wife" has the effect of marrying two people only under certain well-defined circumstances. By definition, perlocutionary acts take us outside the domain of language and communication.

Searle and Vanderveken (1985) develop a logic of speech acts which they call *illocutionary logic*. This was refined in Vanderveken 1990 and Vanderveken 1991. Already, Frege used in his *Begriffsschrift* the notation " $\vdash \phi$ ", where ϕ denotes a proposition and " \vdash " the judgment sign. So, " $\vdash \phi$ " says that ϕ is provable, but other interpretations of \vdash are possible (accompanied by different notation; for example, " $\models \phi$ " says that ϕ is true (in the model), " $\neg \phi$ " says that ϕ is refutable, and so on). An elementary speech act is of the form $F(\phi)$, where F denotes an illocutionary point and

ϕ a proposition. In turn, an illocutionary force is identified by exactly seven elements:

- a. a *point*,
- b. the mode of achievement of the illocutionary point,
- c. the degree of strength of the illocutionary point,
- d. the propositional content conditions,
- e. the preparatory conditions,
- f. the sincerity conditions,
- g. the degree of strength of the sincerity conditions.

There are exactly five points according to Searle and Vanderveken (1985):

- The *assertive* point is to say how things are.
- The *commissive* point is to commit speaker to doing something.
- The *directive* point is to get other people to do things.
- The *declarative* point is to change the world by saying so.
- The *expressive* point is to express feelings and attitudes.

Later treatments of this matter tend to disregard much of the complexity of this earlier approach for the reason that it fails to have any predictive power. Especially difficult to handle are “strengths”, for example. Modern models try to use update models instead (see Section 2.3 below). Van der Sandt 1991 uses a discourse model with three different slates (for each speaker, and one common slate). While each speaker is responsible for maintaining his own slate, changes to the common slate can only be made through communication with each other. Merin 1994 seeks to reduce the manipulations to a sequential combination of so-called *elementary social acts*: *claim*, *concession*, *denial*, and *retraction*.

Uttering a sentence is acting. This action can have various consequences, partly intended partly not. The fact that utterances as actions are embedded in a bigger scheme of interaction between humans has been put into focus

recently (see, for example, Clark 1996). Another important aspect that has been highlighted recently is the fact that by uttering a sentence we can change the knowledge state of an entire group of agents, see Balbiani *et al.* 2008. After publicly announcing ϕ , ϕ becomes common knowledge among the entire group. This idea sheds new light on a problem of Gricean pragmatics, where certain speech acts can only be successful if certain facts are commonly known between speaker and hearer. It is by means of an utterance that a speaker can establish this common knowledge in case it wasn't already there.

2.2 Action sentences

Davidson (1967) gave an account of action sentences in terms of what is now widely known as *events*. The basic idea is that an action sentence has the form $(\exists e)(\dots)$, where e is a variable over acts. For example, “Brutus violently stabbed Caesar” is translated (ignoring tense) as $(\exists e)(\text{stab}(e, \text{Brutus}, \text{Caesar}) \wedge \text{violent}(e))$. This allows to capture the fact that this sentence logically entails that Brutus stabbed Caesar. This idea has been widely adopted in linguistics; moreover, it is now assumed that basically all verbs denote events (Parsons 1990). Thus action sentences are those that speak about special types of events, called *eventualities*.

Vendler (1957) classified verbs into four groups:

- a. *states* (“know”, “sit”),
- b. *activities* (“run”, “eat”),
- c. *accomplishments* (“write a letter”, “build a house”), and
- d. *achievements* (“reach”, “arrive”).

Moens and Steedman (1988) add a fifth category:

- e. *points* (“flash”, “burst”).

The main dividing line is between states and the others. The types (b)–(e) all refer to change. This division has been heavily influential in linguistic theory; mostly, however, research concentrated on its relation to aspect. It is to be noted, for example, that verbs of type (c) can be used with the progressive while verbs of type (d) cannot. In an attempt to explain this, Krifka 1986 and Krifka 1992 have introduced the notion of an *incremental theme*. The idea is that any eventuality has an underlying activity whose progress can be measured using some underlying participant of the event. If, for example, I write a letter then the progress is measured in amounts of words. The letter is therefore the incremental theme in “I write a letter” since it defines the progress. One implementation of the idea is the theory of aspect by Verkuyl (1993). Another way to implement the idea of change is constituted by a translation into propositional dynamic logic (see Naumann 2001). Van Lambalgen and Hamm (2005) have applied the event calculus by Shanahan (1990) to the description of events.

2.3 Dynamic semantics

The idea that propositions can not only be viewed as state descriptions but also as updates has been advocated independently by many people. Consider the possible states of an agent to be (in the simplest case) a theory (that is, a deductively closed set of sentences). Then the *update* of a theory T by a proposition ϕ is the deductive closure of $T \cup \{\phi\}$.

Gärdenfors 1988 advocates this perspective with particular attention to belief revision. Veltman 1985 develops the update view for the treatment of conditionals. One advantage of the idea is that it is possible to show why the mini discourse “It rains. It may not be raining.” is infelicitous in contrast to “It may not be raining. It rains.”. Given that an update is felicitous only to a consistent theory, and that “may ϕ ” (with epistemic “may”) simply means “it is consistent” (written $\diamond\phi$), the first is the sequence of updates with ϕ and $\diamond\neg\phi$. The second step leads to

inconsistency, since ϕ has already been added. It is vital in this approach that the context is constantly changing.

Heim 1983 contains an attempt to make this idea fruitful for the treatment of presuppositions. In Heim’s proposal, a sentence has the potential to change the context, and this is why, for example, the sentence “If John is married his wife will be happy.” does not presuppose that John is married. Namely, the second part of the conditional (“his wife will be happy”) is evaluated against the the context incremented by the antecedent (“John is married”). This of course is the standard way conditions are evaluated in computer languages. This parallel is exploited in Van Eijck 1994, see also Kracht 1993.

The idea of going dynamic was further developed in Dynamic Predicate Logic (DPL, see Groenendijk and Stokhof 1991), where all expressions are interpreted dynamically. The specific insight in this grammar is that existential quantifiers have a dynamically growing scope. This has first been noted in Kamp 1981, where a semantics was given in terms of intermediate representations, so-called Discourse Representation Structures. Groenendijk and Stokhof replace these structures by introducing a dynamics into the evaluation of a formula, as proposed in Dynamic Logic (DL). An existential quantifier is translated as a random assignment “ $x \leftarrow ?$ ” of DL, whose interpretation is a relation between assignments: it is the set of pairs $\langle\beta, \gamma\rangle$ such that $\beta(y) = \gamma(y)$ for all $y \neq x$ (in symbols $\beta \sim_x \gamma$). The translation of the sentence “A man walks.” is

$$(1) \quad \langle x \leftarrow ? \rangle \text{man}'(x) \wedge \text{walk}'(x)$$

This is a proposition, hence interpreted as a set. One can however, push the dynamicity even lower, and make all meanings relational. Then “A man walks.” is interpreted by the ‘program’

$$(2) \quad x \leftarrow ?; \text{man}'(x)?; \text{walk}'(x)?$$

Here, $\text{man}'(x)?$ uses the test constructor “?”: $\phi?$ is the set of all $\langle\beta, \beta\rangle$ such that β satisfies ϕ . The meaning of the entire program (2) therefore also is a relation between assignments. Namely, it is the set R of all pairs $\langle\beta, \gamma\rangle$ where $\beta \sim_x \gamma$, and $\gamma(x)$ walks and is a man. The meaning of (1) by contrast is the set of all β such that some $\langle\beta, \gamma\rangle \in R$. Existential quantifiers thus have ‘side effects’: the change in assignment is never undone by a quantifier over a different variable. Hence the open-endedness to the right of the existential. This explains the absence of brackets in (1). For an overview of dynamic semantics see Muskens *et al.* 1997.

3. The Logic of Action in Computer Science

The logic of action plays an important role in computer science. This becomes evident once one realizes that computers perform actions in the form of executing program statements written down in some programming language, changing computer internals and, by interfaces to the outside world, also that outside world. As such a logic of action provides a means to reason about programs, or more precisely, the execution of programs and their effects. This enables one to prove the correctness of programs. In principle, this is something very desirable: if we could prove all our software correct, we would know that they would function exactly the way we designed them. This was already realized by pioneers of computer programming such as Turing (1949) and Von Neumann (Goldstein and Von Neumann 1963). Of course, this ideal is too hard to establish in daily practice for all software. Verification is a nontrivial and time-consuming occupation, and there are also theoretical limitations to it. However, as the alternative is “just” massive testing of programs experimentally, with no 100% guarantee of correctness, it has remained an active area of research to this day.

3.1 Reasoning about programs

Program verification has a long history. Already since the inception of the computer and its programming researchers started to think of ways of analyzing programs to be sure they did what they were supposed to do. In the 60s the development of a true mathematical theory of program correctness began to take serious shape (de Bakker 1980, 466). Remarkably, the work of John McCarthy who we will also encounter later on when we turn to the field of artificial intelligence played an important role here, distinguishing and studying fundamental notions such as ‘state’, McCarthy 1963a. This led on the one hand to the field of semantics of programming languages, and on the other to major advances in program correctness by Floyd (1967), Naur (1966), Hoare (1969) and Dijkstra (1976) (de Bakker 1980). Floyd and Naur used an elementary stepwise induction principle and predicates attached to program points to express invariant properties of imperative-style programs (Cousot 1990, 859), programs that are built up from basic assignment statements (of arithmetical expressions to program variables) and may be composed by sequencing, conditionals and repetitions. While the Floyd-Naur approach—called the *inductive assertion method*—giving rise to a systematic construction of verification conditions, was a method to prove the correctness of programs by means of logic, it was not a logic itself in the strict sense of the word. The way to a proper logic of programs was paved by Hoare, whose *compositional* proof method led to what is now known as Hoare logic. By exploiting the syntactic structure of (imperative-style) programs, Hoare was able to turn the Floyd-Naur method into a true logic with as assertions so-called Hoare triples of the form $\{P\} S \{Q\}$, where P and Q are first-order formulas and S is a program statement in an imperative-style programming language as mentioned above. The intended reading is if P holds before execution of the statement S then Q holds upon termination of (execution of) S . (The issue whether the execution of S terminates can be put in the reading of this Hoare triple either conditionally (partial correctness) or nonconditionally (total correctness),

giving rise to different logics, see Harel *et al.* 2000). To give an impression of Hoare-style logics, we give here some rules for a simple programming language consisting of variable assignments to arithmetic expressions, and containing sequential (;), conditional (IF) and repetitive (WHILE) composition.

$$\frac{\{P\} S_1 \{Q\}, \{Q\} S_2 \{R\}}{\{P\} S_1 ; S_2 \{Q\}}$$

$$\frac{\{P \wedge B\} S_1 \{Q\}, \{P \wedge \neg B\} S_2 \{Q\}}{\{P\} \text{ IF } B \text{ THEN } S_1 \text{ ELSE } S_2 \{Q\}}$$

$$\frac{\{P \wedge B\} S \{P\}}{\{P\} \text{ WHILE } B \text{ DO } S \{P \wedge \neg B\}}$$

Later Pratt and Harel generalized Hoare logic to dynamic logic (Pratt 1976, Pratt 1979a, Harel 1979, Harel 1984, Kozen and Tiuryn 1990, Harel *et al.* 2000), of which it was realized^[1] that it is in fact a form of modal logic, by viewing the input-output relation of a program S as an accessibility relation in the sense of Kripke-style semantics.^[2] A Hoare triple $\{P\} S \{Q\}$ becomes in dynamic logic the following formula: $P \rightarrow [S]Q$, where $[S]$ is the modal box operator associated with (the accessibility relation associated with) the input-output relation of program S . The propositional version of Dynamic Logic, PDL, was introduced by Fischer and Ladner (1977), and became an important topic of research in itself. The key axiom of PDL is the induction axiom

$$[S^*](P \rightarrow [S]P) \rightarrow (P \rightarrow [S^*]P)$$

where $*$ stands for the iteration operator, S^* denoting an arbitrary (finite) number of iterations of program S . The axiom expresses that if after any number of iterations of S the truth of P is preserved by the execution of S , then, if P is true at the current state, it will also be true after any number of iterations of S . A weaker form of PDL, called HML, with only an atomic

action box and diamond and propositional connectives, was introduced by Hennessy & Milner to reason about concurrent processes, and in particular analyze process equivalence (Hennessy and Milner 1980).

It is also worth mentioning here that the work of Dijkstra (1976) on weakest precondition calculus is very much related to dynamic logic (and Hoare's logic). In fact, what Dijkstra calls the *weakest liberal precondition*, denoted $\mathbf{wlp}(S, Q)$, is the same as the box operator in dynamic logic: $\mathbf{wlp}(S, Q) = [S]Q$, while his *weakest precondition*, denoted $\mathbf{wp}(S, Q)$, is the total correctness variant of this, meaning that this expression also entails the termination of statement S (Cousot 1990).

It was later realized that the application of dynamic logic goes beyond program verification or reasoning about programs. In fact, it constitutes a logic of general action. In Meyer 2000 a number of other applications of dynamic logic are given including deontic logic (see also Meyer 1988), reasoning about database updates, the semantics of reasoning systems such as reflective architectures. As an aside we note here that the use of dynamic logic for deontic logic as proposed in Meyer 1988 needed an extension of the action language, in particular the addition of the 'action negation' operator. The rather controversial nature of this operator triggered work on action negation in itself (see *e.g.*, Broersen 2004). Below we will also encounter the use of dynamic logic in artificial intelligence when specifying intelligent agents.

The logics thus far are adequate for reasoning about programs that are supposed to terminate and display a certain input/output behavior. However, in the late seventies one came to realize that there are also programs that are not of this kind. Reactive programs are designed to react to input streams that in theory may be infinite, and thus show ideally nonterminating behavior. Not so much input-output behavior is relevant here but rather the behavior of programs over time. Therefore Pnueli

(1977) proposed a different way of reasoning about programs for this style of programming based on the idea of a logic of time, *viz.* (*linear-time temporal logic*). (Since reactivity often involves concurrent or parallel programming, temporal logic is often associated with this style of programming. However, it should be noted that a line of research continued to extend the use of Hoare logic to concurrent programs (Lamport 1977, Cousot 1990, de Roever *et al.* 2001).) Linear-time temporal logic typically has temporal operators such as next-time, always (in the future), sometime (in the future), until and since.

An interesting difference between temporal logic on the one hand, and dynamic logic and Hoare logic on the other, is that the former is what in the literature is called an *endogenous* logic, while the latter are so-called *exogenous* logics. A logic is exogenous if programs are explicit in the logical language, while for endogenous logics this is not the case. In an endogenous logic such as temporal logic the program is assumed to be fixed, and is considered part of the structure over which the logic is interpreted (Harel *et al.* 2000, 157). Exogenous logics are *compositional* and have the advantage of allowing analysis by structural induction. Later Pratt (1979b) tried to blend temporal and dynamic logic into what he called *process logic*, which is an exogenous logic for reasoning about temporal behavior.

At the moment the field of temporal logic as applied in computer science has developed into a complete subfield on its own, including techniques and tools for (semi-)automatic reasoning and model-checking (*cf.* Emerson 1990). Also variants of the basic linear-time models have been proposed for verification, such as *branching-time temporal logic* (and, in particular the logics CTL (computation tree logic) and its extension CTL* (Emerson 1990), in which one can reason explicitly about (quantification over) alternative paths in nondeterministic computations, and more recently also an extension of CTL, called *alternating-time temporal logic*

(ATL), with a modality expressing that a group of agents has a joint strategy to ensure its argument, to reason about so-called open systems. These are systems, the behavior of which depends also on the behavior of their environments, see Alur *et al.* 1998.

Finally we mention still alternative logics to reason about programs, *viz.* fixpoint logics, with as typical example the so-called μ -calculus, dating back to Scott and de Bakker (1969), and further developed in Hitchcock and Park 1972, Park 1976, de Bakker 1980, and Meyer 1985. The basic operator is the least fixed point operator μ , capturing iteration and recursion: if $\phi(X)$ is a logical expression with a free relation variable X , then the expression $\mu X\phi(X)$ represents the least X such that $\phi(X) = X$, if such an X exists. A propositional version of the μ -calculus, called propositional or modal μ -calculus comprising the propositional constructs \rightarrow and **false**, together with the atomic (action) modality $[a]$ and μ operator is completely axiomatized by propositional modal logic plus the axiom $\phi[X/\mu X\phi] \rightarrow \mu X\phi$, where $\phi[X/Y]$ stands for the expression ϕ in which X is substituted by Y , and rule

$$\frac{\phi[X/\psi] \rightarrow \psi}{\{\mu X\phi \rightarrow \psi\}}$$

(Kozen 1983, Bradfield and Stirling 2007). This logic is known to subsume PDL (*cf.* Harel *et al.* 2000).

4. The Logic of Action in Artificial Intelligence

In the field of artificial intelligence (AI), the aim is to devise intelligently behaving computer-based artifacts (with the purpose of understanding human intelligence or just making intelligent computer systems and programs). In order to achieve this, there is a tradition within AI to try and construct these systems based on symbolic representations of all relevant factors involved. This tradition is called symbolic AI or ‘good old-

fashioned' AI (GOFAI). In this tradition the sub-area of knowledge representation (KR) obviously is of major importance: it played an important role since the inception of AI, and has developed to a substantial field of its own. One of the prominent areas in KR concerns the representation of actions, performed by either the system to be devised itself or the actors in its environment. Of course, besides their pure representation also reasoning about actions is important, since representation and reasoning with these representations are deemed to be closely connected within KR (which is sometime also called KR&R, knowledge representation & reasoning). A related, more recent development within AI is that of basing the construction of intelligent systems on the concept of an (*intelligent*) *agent*, an autonomously acting entity, regarding which, by its very nature, logics of action play a crucial role in obtaining a logical description and specification.

4.1 Representing and reasoning about actions

As said above, the representation of actions and formalisms/logics to reason with them are very central to AI and particularly the field of KR. One of the main problems that one encounters in the literature on reasoning about actions in AI, and much more so than in mainstream computer science, is the discovery of the so-called *frame problem* (McCarthy and Hayes 1969). Although this problem has been generalized by philosophers such as Dennett (1984) to a general problem of relevance and salience of properties pertaining to action, the heart of the problem is that in a 'common-sense' setting as one encounters in AI, it is virtually impossible to specify all the effects by the actions of concern, as well as, notably, all *non-effects*. For instance, given an action, think about what changes if the action is performed and what does not—generally the latter is much more difficult to produce than the former, leading to large, complex attempts to specify the non-effects. But there is of course also the problem of *relevance*: what aspects are relevant for the problem at hand;

which properties do we need to take into consideration? In particular, this also pertains to the preconditions of an action that would guarantee the successful performance/execution of an action. Again, in a common-sense environment, these are formidable, and one can always think of another (pre)condition that should be incorporated. For instance, for successfully starting the motor of a car, there should be a charged battery, sufficient fuel, ..., but also not too cold weather, or even sufficient power in your fingers to be able to turn the starting key, the presence of a motor in the car, ... etc. etc. In AI one tries to find a solution for the frame problem, having to do with the smallest possible specification. Although this problem gave rise to so-called *defeasible* or *non-monotonic* solutions such as defaults ('normally a car has a motor'), which in itself gave rise to a whole new a realm within AI called *nonmonotonic or commonsense reasoning*, this is beyond the scope of this entry (we refer the interested reader to the article by Thomason (2003) in this encyclopedia). We focus here on a solution that does not appeal to nonmonotonicity (directly).

Reiter (2001) has proposed a (partial) solution within a framework, called the *situation calculus*, that has been very popular in KR especially in North America since it was proposed by John McCarthy, one of the founding fathers of AI (McCarthy 1963b, McCarthy 1986). The situation calculus is a dialect of first-order logic with some mild second-order features, especially designed to reason about actions. (One of its distinctive features is that of the so-called *reification* of semantic notions such as states or possible worlds (as well as truth predicates) into syntactic entities ('situations') in the object language.) For the sake of conformity in this entry and reasons of space, we will try rendering Reiter's idea within (first-order) dynamic logic, or rather, a slight extension of it. (We need action variables to denote action expressions and equalities between action variables and actions (or rather action expressions) as well as (universal) quantification over action variables).

What is known as Reiter’s solution to the frame problem assumes a so-called closed system, that is to say, a system in which all (relevant) actions and changeable properties (in this setting often called ‘fluents’ to emphasize their changeability over time) are known. By this assumption it is possible to express the (non)change as a consequence of performing actions as well as the issue of the check for the preconditions to ensure successful performance in a very succinct and elegant manner, and coin it in a so-called successor state axiom of the form

$$(\forall A)[\text{Poss}(A) \rightarrow (([A]f(\mathbf{x})) \leftrightarrow (\gamma_f^+(\mathbf{x}, A) \vee (f(\mathbf{x}) \wedge \neg\gamma_f^-(\mathbf{x}, A)))))]$$

where A is an action variable, and $\gamma_f^+(\mathbf{x}, A)$ and $\gamma_f^-(\mathbf{x}, A)$ are ‘simple’ expressions without action modalities expressing the conditions for ϕ becoming true and false, respectively. So the formula is read informally as, under certain preconditions pertaining to the action A at hand, the fluent (predicate) f becomes true of arguments \mathbf{x} , if and only if either the condition $\gamma_f^+(\mathbf{x}, A)$ holds or $f(\mathbf{x})$ holds (before the execution of A) and the condition $\gamma_f^-(\mathbf{x}, A)$ (that would cause it to become false) does not hold. Furthermore, the expression $\text{Poss}(A)$ is used schematically in such axioms, where the whole action theory should be complemented with so-called precondition axioms of the form $\phi_A \rightarrow \text{Poss}(A)$ for concrete expressions ϕ_A stating the actual preconditions needed for a successful execution of A .

To see how this works out in practice we consider a little example in a domain where we have a vase v which may be broken or not (so we have “broken” as a fluent), and actions drop and repair. We also assume the (non-changeable) predicates fragile and held-in-hand of an object. Now the successor state axiom becomes

$$\begin{aligned} (\forall A)[\text{Poss}(A) \rightarrow \\ ([A]\text{broken}(v) \leftrightarrow \\ ((A = \text{drop}(v) \wedge \text{fragile}(v)) \vee (\text{broken}(v) \wedge A \neq \text{repair}(v)))))] \end{aligned}$$

and as precondition axioms we have $\text{held-in-hand}(x) \rightarrow \text{Poss}(\text{drop}(x))$ and $\text{broken}(x) \rightarrow \text{Poss}(\text{repair}(x))$. This action theory is very succinct: one needs only one successor state axiom per fluent and one precondition axiom per action.

Finally in this subsection we must mention some other well-known approaches to reasoning about action and change. The *event calculus* (Kowalski and Sergot 1986, Shanahan 1990, Shanahan 1995) and *fluent calculus* (Thielscher 2005) are alternatives to the situation-based representation of actions in the situation calculus. The reader is also referred to Sandewall and Shoham 1994 for historical and methodological issues as well as the relation with non-monotonic reasoning. These ideas have led to very efficient planning systems (*e.g.*, TALplanner, Kvarnström and Doherty 2000) and practical ways to program robotic agents (*e.g.*, the GOLOG family (Reiter 2001) of languages based on the situation calculus, and FLUX (Thielscher 2005) based on the fluent calculus).

4.2 Description and specification of intelligent agents

In the last two decades the notion of an intelligent agent has emerged as a unifying concept to discuss the theory and practice of artificial intelligence (*cf.* Russell and Norvig 1995, Nilsson 1998). In short, agents are software entities that display forms of intelligence/rationality and autonomy. They are able to take initiative and make decisions to take action on their own without direct control of a human user. In this subsection we will see how logic (of action) is used to describe / specify the (desired) behavior of agents (*cf.* Wooldridge 2002). First we focus on single agents, after which we turn to settings with multiple agents, called multi-agent systems (MAS) or even agent societies.

4.2.1 Single agent approaches

Interestingly, the origin of the intelligent agent concept lies in philosophy.

First of all there is a direct link with *practical reasoning* in the classical philosophical tradition going back to Aristotle. Here one is concerned with reasoning about action in a syllogistic manner, such as the following example taken from Audi 1999, p. 729:

Would that I exercise.

Jogging is exercise.

Therefore, I shall go jogging.

Although this has the form of a deductive syllogism in the familiar Aristotelian tradition of theoretical reasoning, on closer inspection it appears that this syllogism does not express a purely logical deduction. (The conclusion does not follow logically from the premises.) It rather constitutes a representation of a decision of the agent (going to jog), where this decision is based on mental attitudes of the agent, *viz.* his/her beliefs (jogging is exercise) and his/her desires or goals (would that I exercise). So, practical reasoning is reasoning directed toward action, the process of figuring out what to do, as Wooldridge (2000) puts it. The process of reasoning about what to do next on the basis of mental states such as beliefs and desires is called deliberation.

Dennett (1971) has put forward the notion of *the intentional stance*: the strategy of interpreting the behaviour of an entity by treating it as if it were a rational agent that governed its choice of action by a consideration of its beliefs and desires. As such it is an anthropomorphic instance of the so called design (functionality) stance, contra the physical stance, towards systems. This stance has been proved to be extremely influential, not only

in cognitive science and biology/ethology (in connection with animal behavior), but also as a starting point of thinking about artificial agents.

Finally, and most importantly, there is the work of the philosopher Michael Bratman (1987), which, although in the first instance aimed at human agents, lays the foundation of the BDI approach to artificial agents. In particular, Bratman makes a case for the incorporation of the notion of *intention* for describing agent behavior. Intentions play the important role of selection of actions that are desired, with a distinct commitment attached to the actions thus selected. Unless there is a rationale for dropping a commitment (such as the belief that an intention has been achieved already or the belief that it is impossible to achieve) the agent should persist / persevere in its commitment, stick to it, so to speak, and try realizing it,

After Bratman's philosophy was published, researchers tried to formalize this theory using logical means. We mention here three well-known approaches. Cohen and Levesque (1991) tried to capture Bratman's theory in a linear-time style temporal logic where they added primitive operators for belief and goal as well as some operators to cater for actions, such as operators for expressing that an action is about to be performed ($HAPPENS\alpha$), has just been performed ($DONE\alpha$) and what agent is the actor of a primitive action ($ACT_i \alpha$: agent i is the actor of α). From this basic set-up they build a framework in which ultimately the notion of intention is defined in terms of the other notions. In fact they define two notions: an *intention_to_do* and an *intention_to_be*. First they define the notion of an achievement goal (A-Goal): an A-Goal is something that is a goal to hold later, but is believed not to be true now. Then they define a persistent goal (P-Goal): a P-Goal is an A-Goal that is not dropped before it is believed to be achieved or believed to be impossible. Then the intention to do an action is defined as the P-Goal of having done the action, in a way such that the agent was aware of it happening. The intention to achieve a state

satisfying ϕ is the P-Goal of having done some action that has ϕ as a result where the agent was aware of something happening leading to ϕ , such that what actually happened was not something that the agent explicitly had *not* as a goal.

Next there is Rao & Georgeff's formalization of BDI agents using the branching-time temporal logic CTL (Rao and Georgeff 1991, Rao and Georgeff 1998, Wooldridge 2000). On top of CTL they introduce modal operators for Belief (BEL), Goal (GOAL) (sometimes replaced by Desire (DES)) and Intention (of the to_be kind, INTEND) as well as operators to talk about the success (succeeded(e)) and failure (failed) of elementary actions e . So they do not try to define intention in terms of other notions, but rather introduce intention as a separate operator, of which the meaning is later constrained by 'reasonable' axioms. The formal semantics is based on Kripke models with accessibility relations between worlds for the belief, goal and intention operators. However, possible worlds here are complete time trees (modeling the various behaviors of the agent) on which CTL formulas are interpreted in the usual way. Next they propose a number of postulates/axioms that they find reasonable interactions between the operators, and constrain the models of the logic accordingly so that these axioms become validities. For example, they propose the formulas $\text{GOAL}(\alpha) \rightarrow \text{BEL}(\alpha)$ and $\text{INTEND}(\alpha) \rightarrow \text{GOAL}(\alpha)$, for a certain class of formulas α , of which $\alpha = \mathbf{E}(\psi)$ is a typical example. Here \mathbf{E} stands for the existential path quantifier in CTL. Rao and Georgeff also show that one can express commitment strategies in their logic. For example, the following expresses a 'single-minded committed' agent, that keeps committed to its intention until it believes it has achieved it or thinks it is impossible (which is very close to what we saw in the definition of intention in the approach of Cohen and Levesque):

$$\text{INTEND}(\mathbf{A} \diamond \phi) \rightarrow (\text{INTEND}(\mathbf{A} \diamond \phi) \text{ UNTIL } (\text{BEL}(\phi) \vee \neg \text{BEL}(\mathbf{E} \diamond \phi)))$$

where \mathbf{A} stands for the universal path quantifier in CTL.

Finally there is the KARO approach by Van Linder et al. (Van der Hoek *et al.* 1998, Meyer *et al.* 1999), which takes dynamic logic as a basis instead of a temporal logic. First a core is built, consisting of the language of propositional dynamic logic augmented with modal operators for knowledge (\mathbf{K}), belief (\mathbf{B}) and desire (\mathbf{D}) as well as an operator (\mathbf{A}) that stands for ability to perform an action. Next the language is extended mostly by abbreviations (definitions in terms of the other operators) to get a fully-fledged BDI-like logic. The most prominent operators are:

- *opportunity* to do an action (meaning that there is a way the action can be executed leading to a next state)
- *practical possibility* to do an action with respect to an assertion (the conjunction of ability and opportunity of doing the action, together with the statement that the execution of the action leads to the truth of the assertion)
- *can* do an action with respect to an assertion (knowing to have the practical possibility to do the action with respect to the assertion at hand)
- *realizability* of an assertion (the existence of a plan, i.e. a sequence of atomic actions, which the agent has the practical possibility to perform with respect to the assertion at hand)
- *goal* with respect to an assertion (the conjunction of the assertion being desirable, not true yet, but realizable)
- *possibly intend* to do an action with respect to an assertion (expressing that the agent can do the action with respect to the assertion of which he knows it to be a goal of his)

The framework furthermore has special actions *commit* and *uncommit* to control the agent's commitments. The semantics of these actions is such that the agent obviously can only commit to an action α if there is good

reason for it, *viz.* that there is a possible intention of α with a known goal ϕ as result. Furthermore the agent cannot uncommit to a certain action α that is part of the agent's commitments, as long there is a good reason for it to be committed to α , i.e. as long as there is some possible intention where α is involved. This results in having the following validities in KARO: (Here $\mathbf{I}(\alpha, \phi)$ denotes the possibly intend operator and $\mathbf{Com}(\alpha)$ is an operator that expresses that the agent is committed to the action α , which is similar to Cohen & Levesque's intention-to-do operator INTEND_1 in Cohen and Levesque 1990.)

$$\begin{aligned} &\models \mathbf{I}(\alpha, \phi) \rightarrow \langle \text{commit}(\alpha) \rangle \mathbf{Com}(\alpha) \\ &\models \mathbf{I}(\alpha, \phi) \rightarrow \neg \mathbf{Auncommit}(\alpha) \\ &\models \mathbf{Com}(\alpha) \rightarrow \langle \text{uncommit}(\alpha) \rangle \neg \mathbf{Com}(\alpha) \\ &\models \mathbf{Com}(\alpha_1 ; \alpha_2) \rightarrow \mathbf{KCom}(\alpha_1) \wedge \mathbf{K}[\alpha_1] \mathbf{Com}(\alpha_2) \end{aligned}$$

Informally these axioms say the following: if the agent possibly intends an action for fulfilling a certain goal then it has the opportunity to commit to this action, after which it is recorded on its agenda; as long as an agent possibly intends an action it is not able to uncommit to it (this reflects a form of persistence of commitments: as long as there is a good reason for a plan on the agenda it will have to stay on!); if the agent is committed to an action it has the *opportunity* to uncommit to it (but it may lack the *ability* to do this, *cf.* the previous axiom); if an agent is committed to a sequence of two actions then it knows that it is committed to the first and it also knows that after performing the first action it will be committed to the second.

Besides this focus on motivational attitudes in the tradition of agent logics in BDI style, the KARO framework also provides an extensive account of epistemic and doxastic attitudes. This is worked out most completely in Van Linder *et al.* 1995. This work hooks into a different strand of research in between artificial intelligence and philosophy, *viz.* *Dynamic Epistemic*

Logic, the roots of which lie in philosophy, linguistics, computer science and artificial intelligence! Dynamic Epistemic Logic (DEL) is the logic of knowledge change; it is not about one particular logical system, but about a whole family of logics that allow us to specify static and dynamic aspects of knowledge and beliefs of agents (*cf.* Van Ditmarsch *et al.* 2007). The field combines insights from philosophy (about belief revision, AGM-style (AGM 1985), as we have seen in Section 1), dynamic semantics in linguistics and the philosophy of language (as we have seen in Section 2), reasoning about programs by using dynamic logic (as we have seen in Section 3) with ideas in artificial intelligence about how knowledge and actions influence each other (Moore 1977). More generally we can see the influence of the logical analysis of information change as advocated by van Benthem and colleagues (van Benthem 1989, van Benthem 1994, Faller *et al.* 2000). Also Veltman's update semantics of default reasoning (Veltman 1996), an important reasoning method in artificial intelligence (Reiter 1980, Russell and Norvig 1995), can be viewed as being part of this paradigm.

For the purpose of this entry, it is interesting to note that the general approach taken is to apply a logic of action, *viz.* dynamic logic, to model information change. This amounts to an approach in which the epistemic (or doxastic) updates are reified into the logic as actions that change the epistemic/doxastic state of the agent. So, for example in Van Linder *et al.* 1995 we encounter the actions such as $\text{expand}(\phi)$, $\text{contract}(\phi)$, $\text{revise}(\phi)$, referring to expanding, contracting and revising, respectively, one's belief with the formula ϕ . These can be reasoned about by putting them in dynamic logic boxes and diamonds, so that basically extensions of dynamic logic are employed for reasoning about these updates. It is further shown that these actions satisfy the AGM postulates so that this approach can be viewed as a modal counterpart of the AGM framework. Very similar in spirit is the work of Segerberg (1995) on Dynamic Doxastic Logic (DDL), the modal logic of belief change. In DDL modal operators

of the form $[+\phi]$, $[*\phi]$ and $[-\phi]$ are introduced with informal meanings: “after the agent has expanded/ revised/ contracted his beliefs by ϕ ”, respectively. Combined with the ‘standard’ doxastic operator B , where $B\phi$ is interpreted as “ ϕ is in the agent’s belief set”, one can now express properties like $[+\phi]B\psi$ expressing that after having expanded its beliefs by ϕ the agent believes ψ (also *cf.* Hendricks and Symons 2006).

Finally in this subsection we mention recent work where the KARO formalism is used as a basis for describing also other aspects of cognitive behavior of agents, going ‘beyond BDI’, *viz.* attitudes regarding emotions (Meyer 2006, Steunebrink *et al.* 2007, Steunebrink *et al.* 2012). The upshot of this approach is that an expressive logic of action such as KARO can be fruitfully employed to describe how emotions such as joy, gratification, anger, and remorse, are triggered by certain informational and motivational attitudes such as certain beliefs and goals (‘emotion elicitation’) and how, once elicited, the emotional state of an agent may influence its behavior, and in particular its decisions about the next action to take.

4.2.2 Multi-agent approaches

Apart from logics to specify attitudes of single agents, also work has been done to describe the attitudes of multi-agent systems as wholes. First we mention the work by Cohen & Levesque in this direction (Levesque *et al.* 1990, Cohen and Levesque 1991). This work was a major influence on a multi-agent version of KARO (Aldewereld *et al.* 2004). An important complication in a notion of joint goal involves that of persistence of the goal: where in the single agent case the agent pursues its goal until it believes it has achieved it or believes it can never be achieved, in the context of multiple agents, the agent that realizes this, has to inform the others of the team about it so that the group/team as a whole will believe that this is the case and may drop the goal. This is captured in the

approaches mentioned above. Related work, but not a logic of action in the strict sense, concerns the logical treatment of collective intentions (Keplicz and Verbrugge 2002).

It must also be mentioned here that inspired by several sources among which the work on knowledge and belief updates for individual agents as described by DEL and DDL, combined with work on knowledge in groups of agents such as common knowledge (see, e.g., Meyer and Van der Hoek 1995), a whole new subfield has arisen, which can be seen as the multi-agent (counter)part of Dynamic Epistemic Logic. This deals with matters such as the logic of public announcement, and more generally actions that have effect on the knowledge of groups of agents. This has generated quite some work by different authors such as Plaza (1989), Baltag (1999), Gerbrandy (1998), Van Ditmarsch (2000), and Kooi (2003). For example, public announcement logic (Plaza 1989) contains an operator of the form $[\phi]\psi$, where both ϕ and ψ are formulas of the logic, expressing “after announcement of ϕ , it holds that ψ ”. This logic can be seen as a form of dynamic logic again, where the semantic clause for $[\phi]\psi$ reads (in informal terms): $[\phi]\psi$ is true in a model-state pair iff the truth of ϕ in that model-state pair implies the truth of ψ in a model-state pair, where the state is the same, but the model is transformed to capture the information contained in ϕ . Also in the other approaches the transformation of models induced by communicated information plays an important role, notably in the approach by Baltag *et al.* on action models (Baltag 1999, Baltag and Moss 2004). A typical element in this approach is that in action model logic one has both epistemic and action models and that the update of an epistemic model by an epistemic action (an action that affects the epistemic state of a group of agents) is represented by a (restricted) modal product of that epistemic model and an action model associated with that action. (See Van Ditmarsch *et al.* 2007, p. 151; this book is a recent comprehensive reference to the field.)

Finally we mention logics that incorporate notions from game theory to reason about multi-agent systems, such as game logic, coalition logic (Pauly 2001) and alternating temporal logic (ATL, which we also encountered at the end of the section on mainstream computer science!), and its epistemic variant ATEL (Van der Hoek and Wooldridge 2003, Van der Hoek *et al.* 2007). For instance, game logic is an extension of PDL to reason about so-called determined 2-player games. Interestingly there is a connection between these logics and the stit approach we have encountered in philosophy. For instance, Broersen, partially jointly with Herzig and Troquard, has shown several connections such as embeddings of Coalition Logic and ATL in forms of stit logic (Broersen *et al.* 2006a,b) and extensions of stit (and ATL) to cater for reasoning about interesting properties of multi-agent systems (Broersen 2009, 2010). This area currently is growing fast, also aimed at the application of verifying multi-agent systems (*cf.* Van der Hoek *et al.* 2007), *viz.* Dastani *et al.* 2010. The latter constitutes still somewhat of a holy grail in agent technology. On the one hand there are many logics to reason about both single and multiple agents, while on the other hand multi-agent systems are being built that need to be verified. To this day there is still a gap between theory and practice. Much work is being done to render logical means combining the agent logics discussed and the logical techniques from mainstream computer science for the verification of distributed systems (from section 3), but we are not there yet...!

Conclusion

In this entry we have briefly reviewed the history of the logic of action, in philosophy, in linguistics, in computer science and in artificial intelligence. Although the ideas and techniques we have considered were developed in these separate communities in a quite independent way, we feel that they are nevertheless very much related, and by putting them together in this

entry we hope we have contributed in a modest way to some cross-fertilization between these communities regarding this interesting and important subject.

Bibliography

- Alchourrón, C., Gärdenfors, P., and Makinson, D., 1985, “On the Logic of Theory Change: Partial Meet Contraction and Revision Functions”, *Journal of Symbolic Logic*, 50: 510–530.
- Aldewereld, H. M., Van der Hoek, W., and Meyer, J.-J. Ch., 2004, “Rational Teams: Logical Aspects of Multi-Agent Systems”, *Fundamenta Informaticae*, 63(2–3): 159–183.
- Alur, R., Henzinger, T., and Kupferman, O., 1998, “Alternating-time Temporal Logic”, in W.-P. de Roever, H. Langmaack and A. Pnueli (eds.), *Compositionality: The Significant Difference, Proceedings COMPOS-97* (Lecture Notes in Computer Science 1536), Berlin: Springer, pp. 23–60.
- Anderson, A. R., 1962, “Logic, norms, and roles”, *Ratio*, 4: 36–49.
- Audi, R. (ed.), 1999, *The Cambridge Dictionary of Philosophy*, Cambridge: Cambridge University Press.
- Austin, J. L., 1957, *How to Do Things with Words*, Oxford: Oxford University Press.
- Bakker, J. W. de, 1980, *Mathematical Theory of Program Correctness*, Englewood Cliffs, NJ: Prentice-Hall International.
- Balbani, P., Baltag, A., van Ditmarsch, H., Herzig, A., Hosi, T. and de Lima, S., 2008, “‘Knowable’ as ‘Known After an Announcement’”, *The Review of Symbolic Logic*, 1: 305–334
- Baltag, A., 1999, “A Logic of Epistemic Actions”, in W. van der Hoek, J.-J. Meyer, and C. Witteveen (eds.), *Foundations and Applications of Collective Agent-Based Systems* (Proceedings of the Workshop at the 11th European Summer School in Logic, Language, and

- Computation, Utrecht, 1999).
- Baltag, A. and Moss, L. S., 2004, “Logics for Epistemic Programs”, *Synthese*, 139: 165–224.
- Belnap, N., Perloff, M., and Xu, M., 2001, *Facing the future*, Oxford: Oxford University Press.
- Bradfield, J. and Stirling, C., 2007, “Modal μ -calculi”, in P. Blackburn, J. F. A. K. van Benthem, and F. Wolter (eds.), *Handbook of Modal Logic*, Amsterdam: Elsevier, pp. 721–756.
- Bratman, M. E., 1987, *Intentions, Plans, and Practical Reason*, Cambridge, Massachusetts: Harvard University Press.
- Broersen, J.M., 2004, “Action Negation and Alternative Reductions for Dynamic Deontic Logics”, *Journal of Applied Logic*, 2(1): 153–168.
- , 2009, “A Complete STIT Logic for Knowledge and Action, and Some of Its Applications”, in M. Baldoni, T. Cao Son, M.B. van Riemsdijk and M. Winikoff (eds.), *Declarative Agent Languages and Technologies VI*, 6th International Workshop, DALI 2008, Estoril, Portugal, May 12, 2008, Revised Selected and Invited Papers: Springer, Berlin, pp. 47–59.
- , 2010, “CTL.STIT: enhancing ATL to express important multi-agent system verification properties”, in *Proceedings 9th International Conference on Autonomous Agents and Multiagent Systems*. New York: ACM Press, pp. 683–690.
- Broersen, J.M., Herzig, A. and Troquard, N., 2006a, “From Coalition Logic to STIT”, *Electronic Notes in Theoretical Computer Science*, 157(4): 23–35.
- , 2006b, “Embedding Alternating-time Temporal Logic in strategic STIT logic of agency”, *Journal of logic and computation*, 16: 559–578.
- Chellas, B. F., 1969, *The Logical Form of Imperatives*, Stanford, CA: Perry Lane Press.
- , 1980, *Modal Logic: An Introduction*, Cambridge and London: Cambridge University Press.
- Clark, Herbert H., 1996, *Using Language*, Cambridge University Press.
- Cohen, P. R. and Levesque, H. J., 1990, “Intention is Choice with Commitment”, *Artificial Intelligence*, 42(3): 213–261.
- Cohen, P. and Levesque, H., 1991, “Teamwork”, *Noûs*, 24(4): 487–512.
- Cousot, P., 1990, “Methods and Logic for Proving Programs”, in J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, Amsterdam: Elsevier, pp. 841–993.
- Dastani, M.M., Hindriks, K.V. and Meyer, J.-J. Ch. (eds.), 2010, *Specification and Verification of Multi-Agent Systems*, New York/Dordrecht/Heidelberg/London: Springer.
- Davidson, D., 1967, “The Logical Form of Action Sentences”, in N. Rescher (ed.), *The Logic of Decision and Action*, Pittsburgh: University of Pittsburgh Press, pp. 81–120.
- Dennett, D. C., 1971, “Intentional Systems”, *Journal of Philosophy*, 68(4): 87–106.
- , 1984, “Cognitive Wheels: The Frame Problem of AI”, in C. Hookway (ed.), *Minds, Machines, and Evolution: Philosophical Studies*, Cambridge: Cambridge University Press.
- Dijkstra, E. W., 1976, *A Discipline of Programming*, Englewood Cliffs, NJ: Prentice-Hall.
- Dunin-Kępicz, B. and Verbrugge, R., 2002, “Collective Intentions”, *Fundamenta Informaticae*, 51(3): 271–295.
- Emerson, E. A., 1990, “Temporal and Modal Logic”, in J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, Amsterdam: Elsevier, pp. 995–1072.
- Faller, M., Kaufmann, S., and Pauly, M. (eds.), 2000, *Formalizing the Dynamics of Information*, Stanford, CA: CSLI Publications.
- Fischer, M. J. and Ladner, R. E., 1977, “Propositional Modal Logic of Programs”, in *Proceedings of the 9th ACM Annual Symposium on*

- Theory of Computing*, New York: Association for Computing Machinery, pp. 286–294.
- Fitch, F. B., 1963, “A logical analysis of some value concepts”, *Journal of Symbolic Logic*, 28: 135–142.
- Floyd, R. W., 1967, “Assigning Meanings to Programs”, in J. T. Swartz (ed.), *Proceedings Symposium in Applied Mathematics*, American Mathematical Society, pp. 19–32.
- Goldstein, H. H. and Neumann J. Von, 1963, “Planning and Coding Problems for an Electronic Computer Instrument”, in A. M. Taub (ed.), *John von Neumann Collected Works* (Vol. 5), Oxford: Pergamon Press, pp. 80–235.
- Groenendijk, J. and Stokhof, M., 1991, “Dynamic Predicate Logic”, *Linguistics and Philosophy*, 14: 39–100.
- Gärdenfors, P., 1988, *Knowledge in Flux*, Cambridge, Massachusetts: MIT Press.
- Gerbrandy, J. D., 1998, *Bisimulations on Planet Kripke*, Ph.D. Thesis, Amsterdam: University of Amsterdam.
- Harel, D., 1979, *First-Order Dynamic Logic*, Berlin: Springer-Verlag.
- , 1984, “Dynamic Logic”, in D. Gabbay and F. Guentner (eds.), *Handbook of Philosophical Logic* (Vol. II), Dordrecht and Boston: Reidel, pp. 497–604.
- Harel, D., Kozen, D., and Tiuryn, J., 2000, *Dynamic Logic*, Cambridge, Massachusetts: MIT Press.
- Heim, I., 1983, “On the projection problem for presuppositions”, in M. Barlow, D. Flickinger and D. Westcoat (eds.), *Proceedings of the 2nd West Coast Conference on Formal Linguistics*, Stanford, CA: Stanford University, pp. 114–126.
- Hendricks, V. and J. Symons, J., 2006, “Epistemic Logic”, in *The Stanford Encyclopedia of Philosophy* (Spring 2009 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/spr2009/entries/logic-epistemic/>.

- Hennessey, M. and Milner, R., 1980, “On Observing Nondeterminism and Concurrency”, in *Proceedings ICALP '80* (Lecture Notes in Computer Science: 85), Berlin: Springer, pp. 295–309.
- Henry, D. P., 1967, *The Logic of St. Anselm*, Oxford: Oxford University Press.
- Hitchcock, P. and Park, D., 1972, “Induction Rules and Termination Proofs”, in M. Nivat (ed.), *Proceedings First International Colloquium on Automata, Languages and Programming*, Amsterdam: North-Holland, pp. 225–251.
- Hoare, C. A. R., 1969, “An Axiomatic Basis for Computer Programming”, *Communications of the ACM*, 12: 576–580.
- Horty, J. F., 2001, *Agency and Deontic Logic*, Oxford: Oxford University Press.
- Kamp, H., 1981, “A theory of truth and semantic representation”, in J. Groenendijk (ed.), *Formal Methods in the Study of Language*, Amsterdam: Mathematisch Centrum.
- Kanger, S., 1957, “New foundations for ethical theory”, Technical Report, Stockholm University; reprinted in R. Hilpinen (ed.), *Deontic Logic: Introductory and Systematic Readings*, D. Reidel: Dordrecht, 1971, pp. 36–58.
- , 1972, “Law and logic”, *Theoria*, 38: 105–132
- Kooi, B., 2003, *Knowledge, Chance, and Change*, Ph.D. Thesis, Groningen: University of Groningen.
- Kowalski, R. and Sergot, M., 1986, “A Logic-Based Calculus of Events”, *New Generation Computing*, 4: 67–95.
- Kozen, D., 1983, “Results on the Propositional μ -calculus”, *Theoretical Computer Science*, 27: 333–354.
- Kozen, D. and Tiuryn, J., 1990, “Logics of Programs”, in J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, Amsterdam: Elsevier, pp. 789–840.
- Kracht, M., 1993, “Logic and Control: How They Determine the

- Behaviour of Presuppositions”, in J. van Eijck and A. Visser, (eds.), *Logic and Information Flow*, Cambridge, Massachusetts: MIT Press, pp. 89–111.
- Krifka, M., 1986, *Nominalreferenz und Zeitkonstitution. Zur Semantik von Massentermen*, Ph.D. Thesis, München, Universität München.
- , 1992, “Thematic Relations as Links between Nominal Reference and Temporal Constitution”, in I. Sag and A. Szabolcsi (eds.), *Lexical Matters*, Stanford: Stanford University Press, pp. 29–53.
- Kvarnström, J. and Doherty, P., 2000, “TALplanner: A Temporal Logic-Based Forward-Chaining Planner”, *Annals of Mathematics and Artificial Intelligence*, 30: 119–169.
- Lampert, L., 1977, “Proving the Correctness of Multiprocess Programs”, *IEEE Transactions on Software Engineering*, SE-3(2): 125–143.
- Levesque, H. J., Cohen, P. R., and Nunes, J. H. T., 1990, “On Acting Together”, in *Proceedings AAAI ‘90*, pp. 94–99.
- Lindahl, L., 1977, *Position and Change: A Study in Law and Logic*, Dordrecht and Boston: Reidel.
- Lindström, S. and Segerberg, K., 2006, “Modal Logic and Philosophy”, in P. Blackburn, J. van Benthem, and F. Wolter (eds.), *Handbook of Modal Logic* (Studies in Logic and Practical Reasoning: 3), Amsterdam: Elsevier, pp. 1153–1218.
- McCarthy, J., 1963a, “Towards a Mathematical Science of Computation”, in C. M. Popplewell (ed.), *Proceedings of IFIP Congress ‘62*, Amsterdam: North-Holland, pp. 21–28.
- , 1963b, “Situations, Actions, and Causal Laws”, Technical Report Memo 2, Stanford University Artificial Intelligence Project, Stanford University; reprinted in M. Minsky (ed.), *Semantic Information Processing*, Cambridge, MA: MIT Press, 1968, pp. 410–417.
- , 1986, “Programs with Common Sense”, in M. L. Minsky (ed.), *Semantic Information Processing*, Cambridge, Massachusetts: MIT Press, pp. 403–418.

- McCarthy J. and Hayes, P. J., 1969, “Some Philosophical Problems from the Standpoint of Artificial Intelligence”, in B. Meltzer, D. Michie and M. Swann (eds.), *Machine Intelligence 4*, Edinburgh: Edinburgh University Press, pp. 463–502.
- Merin, A., 1994, “Algebra of elementary social acts”, in S. L. Tsohatzidis (ed.), *Foundations of Speech Act Theory: Philosophical and Linguistic Perspectives*, London: Routledge, pp. 234–264.
- Meyer, J.-J. Ch., 1985, *Programming Calculi Based on Fixed Point Transformations: Semantics and Applications*, Ph.D. Thesis, Amsterdam: Vrije Universiteit Amsterdam.
- , 1988, “A Different Approach to Deontic Logic: Deontic Logic Viewed as a Variant of Dynamic Logic”, *Notre Dame Journal of Formal Logic*, 29(1): 109–136.
- , 2000, “Dynamic Logic for Reasoning about Actions and Agents”, in J. Minker, (ed.), *Logic-Based Artificial Intelligence*, Boston and Dordrecht: Kluwer, pp. 281–311.
- , 2006, “Reasoning about Emotional Agents”, *International Journal of Intelligent Systems*, 21(6): 601–619.
- Meyer, J.-J. Ch. and Van der Hoek, W., 1995, *Epistemic Logic for AI and Computer Science*, Cambridge: Cambridge University Press.
- Meyer, J.-J. Ch., Van der Hoek, W., and Van Linder, B., 1999, “A Logical Approach to the Dynamics of Commitments”, *Artificial Intelligence*, 113: 1–40.
- Meyer, J.-J. Ch. and Veltman, F., 2007, “Intelligent Agents and Common Sense Reasoning”, in P. Blackburn, J. van Benthem and F. Wolter (eds.), *Handbook of Modal Logic*, Amsterdam: Elsevier, pp. 991–1029.
- Moens, M. and Steedman, M., 1988, “Temporal ontology and temporal reference”, *Computational Linguistics*, 14: 15–28.
- Moore, R. C., 1977, “Reasoning about Knowledge and Action”, in *Proceedings of the 5th International Joint Conference on Artificial*

- Intelligence (IJCAI-77)*, Cambridge, Massachusetts: William Kaufmann, pp. 223–227.
- Muskens, R., van Benthem, J., and Visser, A., 1997, “Dynamics”, in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*, Amsterdam: Elsevier, pp. 587–648.
- Naumann, R., 2001, “Aspects of changes: a dynamic event semantics”, *Journal of Semantics*, 18: 27–81.
- Naur, P., 1966, “Proof of Algorithms by General Snapshots”, *BIT Numerical Mathematics*, 6: 310–316.
- Nilsson, N. J., 1998, *Artificial Intelligence: A New Synthesis*, San Francisco: Morgan Kaufmann.
- Park, D., 1976, “Finiteness is μ -ineffable”, *Theoretical Computer Science*, 3: 173–181.
- Parsons, T., 1990, *Events in the Semantics of English: A Study in Subatomic Semantics* (Current Studies in Linguistics: 19), Cambridge, Massachusetts: MIT Press.
- Pauly, M., 2001, *Logic for Social Software*, ILLC Dissertations Series, Amsterdam.
- Plaza, J. A., 1989, “Logics of Public Communication”, in M. L. Emlich, et al. (eds.), *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, Amsterdam: North-Holland Publishing, pp. 201–216.
- Pnueli, A., 1977, “The Temporal Logic of Programs”, in *Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science*, Piscataway, New Jersey: IEEE, pp. 46–57.
- Pörn, I., 1977, *Action Theory and Social Science*, Dordrecht: Reidel.
- Pratt, V. R., 1976, “Semantical Considerations on Floyd-Hoare Logic”, in *Proceedings of the 17th Annual IEEE Symposium on Foundations of Computer Science*, New York: ACM, pp. 109–121.
- , 1979a, “Dynamic Logic”, in J. W. de Bakker and J. van Leeuwen (eds.), *Proceedings Foundations of Computer Science III*





- Mathematical Centre Tracts 108, Amsterdam: Mathematisch Centrum, pp. 53–82.
- , 1979b, “Process Logic: Preliminary Report”, in *Proceedings of the 6th Symposium on Principles of Programming Languages*, New York: ACM, pp. 93–100.
- Rao, A. S. and Georgeff, M. P., 1991, “Modeling rational agents within a BDI-architecture”, in J. Allen, R. Fikes and E. Sandewall (eds.), *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR '91)*, San Francisco: Morgan Kaufmann, pp. 473–484.
- , 1998, “Decision Procedures for BDI Logics”, *Journal of Logic and Computation*, 8(3): 293–344.
- Reiter, R., 1980, “A Logic for Default Reasoning”, *Artificial Intelligence*, 13(1–2): 81–132.
- , 2001, *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, Cambridge, Massachusetts: MIT Press.
- Roever, W.-P. de, Boer, F. S. de, et al., 2001, *Concurrency Verification: Introduction to Compositional and Noncompositional Methods*, Cambridge: Cambridge University Press.
- Russell, S. and Norvig, P., 1995, *Artificial Intelligence: A Modern Approach*, Englewood Cliffs, NJ: Prentice-Hall.
- Salwicki, A., 1970, “Formalized Algorithmic Languages”, *Bulletin de l'Académie Polonaise des Sciences (Série des sciences mathématiques, astronomiques et physiques)*, 18(5): 227–232.
- Sandewall, E. and Shoham, Y., 1994, “Nonmonotonic Temporal Reasoning”, in D. M. Gabbay, C. J. Hogger and J. A. Robinson (eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4: Epistemic and Temporal Reasoning*, Oxford: Oxford University Press.
- Scott, D. S. and Bakker, J. W. de, 1969, *A Theory of Programs*, Vienna:

- IBM.
- Searle, J. R., 1969, *Speech Acts: An Essay in the Philosophy of Language*, Cambridge: Cambridge University Press.
- Searle, J. R. and Vanderveken D., 1985, *Foundations of Illocutionary Logic*, Cambridge: Cambridge University Press.
- Segeberg, K., 1992, “Getting started: beginnings in the logic of action”, *Studia Logica*, 51: 347–378.
- , 1995, “Belief Revision from the Point of View of Doxastic Logic”, *Bulletin of the IGPL*, 3: 535–553.
- Shanahan, M., 1990, “Representing continuous change in the event calculus”, in *Proceedings of ECAI '90*, pp. 598–603.
- , 1995, “A circumscriptive calculus of events”, *Artificial Intelligence*, 77: 249–284.
- Shoham, Y., 1993, “Agent-Oriented Programming”, *Artificial Intelligence*, 60(1): 51–92.
- Steunebrink, B., Dastani, M. and Meyer, J.-J. Ch., 2007, “A Logic of Emotions for Intelligent Agents”, in Holte, R. C. and Howe, A. E. (eds.), *Proceedings AAAI '07*, Vancouver: AAAI Press, pp. 142–147.
- Steunebrink, B.R., Dastani, M.M. and Meyer, J.-J. Ch., 2012, “A Formal Model of Emotion Triggers for BDI Agents with Achievement Goals”, *Synthese/KRA 185 (1)*: 83–129 (KRA: 413–459).
- Thielscher, M., 2005, *Reasoning Robots, The Art and Science of Programming Robotic Agents*, Dordrecht: Springer.
- Thomason, R., 2003, “Logic and Artificial Intelligence”, in *The Stanford Encyclopedia of Philosophy* (Spring 2009 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/spr2009/entries/logic-ai/>.
- Turing, A. M., 1949, “On Checking a Large Routine”, Report of a Conference on High-Speed Automatic Calculating Machines, Cambridge: University Mathematical Laboratory, pp. 67–69.
- van Benthem, J., 1989, “Semantic Parallels in Natural Language and

- Computation”, in H. D. Ebbinghaus, *et al.* (eds.), *Logic Colloquium '87*, Amsterdam: North-Holland, pp. 331–375.
- , 1994, “Logic and the Flow of Information”, in D. Prawitz, *et al.* (eds.), *Proceedings of the 9th International Congress of Logic, Methodology and Philosophy of Science*, Amsterdam: Elsevier.
- van der Hoek, W., van Linder, B., and Meyer, J.-J. Ch., 1998, “An Integrated Modal Approach to Rational Agents”, in M. Wooldridge and A. Rao (eds.), *Foundations of Rational Agency* (Applied Logic Series: 14), Dordrecht: Kluwer, pp. 133–168.
- van der Hoek, W. and Pauly, M., 2007, “Modal Logic for Games and Information”, in P. Blackburn, J. van Benthem, and F. Wolter (eds.), *Handbook of Modal Logic*, Amsterdam: Elsevier, pp. 1077–1148.
- van der Hoek, W., and Wooldridge, M. J., 2003, “Cooperation, Knowledge, and Time: Alternating-Time Temporal Epistemic Logic and Its Applications”, *Studia Logica*, 75(1): 125–157.
- van der Sandt, R. A., 1991, “Denial”, in *Papers from the 27th Regional Meetings of the Chicago Linguistic Society Meetings* (Volume 2: Parasession on Negation), Chicago Linguistics Society, pp. 331–344.
- van Ditmarsch, H. P., 2000, *Knowledge Games*, Ph.D. Thesis, Groningen: University of Groningen.
- van Ditmarsch, H., Van der Hoek, W., and Kooi, B., 2007, *Dynamic Epistemic Logic*, Dordrecht: Springer.
- van Eijck, J., 1994, “Presupposition failure: a comedy of errors”, *Formal Aspects of Computing*, 3: 766–787.
- van Lambalgen, M. and Hamm, F., 2005, *The Proper Treatment of Events*, Oxford: Blackwell.
- van Linder, B., Van der Hoek, W., and Meyer, J.-J. Ch., 1995, “Actions That Make You Change Your Mind”, in A. Laux and H. Wansing (eds.), *Knowledge and Belief in Philosophy and Artificial Intelligence*, Berlin: Akademie Verlag, pp. 103–146.
- Vanderveken, D., 1990, *Meaning and Speech Acts, Volume 1: Principles of*

- Language Use*, Cambridge: Cambridge University Press.
- , 1991, *Meaning and Speech Acts, Volume 2: Formal Semantics of Success and Satisfaction*, Cambridge: Cambridge University Press.
- Veltman, F., 1985, *Logics for Conditionals*, Ph.D. Thesis, Amsterdam: University of Amsterdam.
- , 1996, “Defaults in Update Semantics”, *Journal of Philosophical Logic*, 25: 221–261.
- Vendler, Z., 1957, “Verbs and times”, *Philosophical Review*, 66: 143–160.
- Verkuyl, H., 1993, *A Theory of Aspectuality: The Interaction Between Temporal and Atemporal Structure*, Cambridge: Cambridge University Press.
- Walton, D., 1976, “St. Anselm and the logical syntax of agency”, *Franciscan Studies*, 36: 298–312.
- Wooldridge, M. J., 2000, *Reasoning about Rational Agents*, Cambridge, Massachusetts: MIT Press.
- , 2002, *An Introduction to MultiAgent Systems*, Chichester: Wiley.
- Wright, G. H. von, 1963, *Norm and Action: A Logical Inquiry*, London: Routledge & Kegan Paul.
- Xu, M., 1998, “Axioms for deliberative stit”, *Journal of Philosophical Logic*, 27: 505–552.

Academic Tools

-  How to cite this entry.
-  Preview the PDF version of this entry at the Friends of the SEP Society.
-  Look up this entry topic at the Indiana Philosophy Ontology Project (InPhO).
-  Enhanced bibliography for this entry at PhilPapers, with links to its database.

Other Internet Resources

[Please contact the author with suggestions.]

Related Entries

events | frame problem | logic: dynamic epistemic | logic: non-monotonic | logic: propositional dynamic | logic: temporal | semantics: dynamic | situations: in natural language semantics | speech acts

Notes to The Logic of Action

1. The similarity between program logic and modal logic was suggested to Pratt by R. Moore; cf. Harel *et al.* 2000, p. 187.
2. To be fair, we must also mention that independently similar ideas were developed by Salwicki (1970) with *Algorithmic Logic*.

Copyright © 2016 by the authors

Krister Segerberg, John-Jules Meyer, and Marcus Kracht