The Knowledge Engineering Review, Vol. 8:2, 1993, 91–120

Logic-based reasoning about actions and plans in artificial intelligence

HUAMING LEE, JAMES TANNOCK AND JON SIMS WILLIAMS
Faculty of Engineering, University of Bristol, Bristol BS8 1TR, UK

Abstract

Reasoning about actions and plans is a vital aspect of the rational behaviour of intelligent agents, and hence represents a major research domain in artificial intelligence. Much work has been undertaken to develop logic-based formalisms and problem solving procedures for plan representation and plan synthesis. This paper consists of a survey of various paradigms for reasoning about actions and plans in artificial intelligence. Attention is focused on the logic-based theoretical frameworks which have built a formal foundation for the domain-independent approaches to the general principles of reasoning about actions and plans.

1 Introduction

It has been widely recognized that one of the most important problem solving behaviours for an intelligent agent is the ability to reason about actions and plans. Since the 1960s, much work has been done to create various formalisms and problem solving technologies for implementing this ability. As a result, *reasoning about plans* or *planning* has become a major research domain in artificial intelligence.

In this paper, the various paradigms of reasoning about actions and plans are discussed. An attempt has been made to briefly characterize the nature of each of the different formalisms. The focus of attention is on representations of *the world, world models, events, actions* and *plans.* To achieve a good understanding of the various paradigms, some consideration of fundamental theory is useful, and this is briefly reviewed. Widely-accepted formal notations are used throughout, so that many original concepts have been described in translated form.

The survey on which this article is based was carried out to provide a theoretical framework for automated planning in engineering applications. The authors have been primarily concerned with the implementation of automated planning in manufacturing, where important areas of interest include *robot planning, process planning* and *inspection planning.* Because of the potential consequences of failure, applications of this nature should be ideally based on provable logic-based formalisms to ensure that the automated planners function correctly.

1.2 Preliminary formalizations

A plan usually means *a detailed synthesis of a program of actions*; where actions represent the rational activities of agents aimed at problem solving. Reasoning about plans can be informally defined as a process in which a set of actions is inferred and formulated through an understanding of the world. The agent's objective will be achieved when this set of actions, which forms a plan, is performed.

The earliest proposal to formalize the problems of reasoning about plans was situational calculus, introduced by McCarthy and Hays (1969). As an extension of classical predicate logic, situational calculus not only maintained all the machinery of classical logic, but also elegantly

characterized the concept of world states and actions by introducing a temporal notion, i.e., *situation*. Green (1969a,b) was the first researcher to implement the deductive procedures in reasoning about actions and plans. By means of resolution theorem proving, Green illustrated how constructive proof procedures could be used for synthesizing plans of action for robots. Situational calculus and deductive techniques based on theorem proving procedures underlie most paradigms of reasoning about actions and plans. As a result, the fundamental formalisms and problem solving techniques developed later are intensively based on various logics.

A problem domain can be regarded as a world which is formalized as a set S of potential world models, each of which represents a state of the world at an instant of time. A set A of allowable actions can be defined in terms of the agent's problem solving activities. A sequence P of actions forms a plan if the agent's objective can be achieved by performing P. Reasoning about plans is essentially a problem of synthesizing a sequence of actions that transforms the world from an initial state model to one in which a given goal is fulfilled. The following definitions are presented here as a basis for discussion in the following sections.

Definition 1.2a A world model $s \in S$ is a logical description of a state of the world at a given instant of time. s is generally represented as a set of formulae of a first-order language L: $s = \{l_1, l_2, \ldots l_m\}$, in which l_i $(i = 1, 2, \ldots, m)$ must be *wffs* in L. A world model is also referred to as a state of the world.

Definition 1.2b An action $f \in A$ is a mapping from state to state. f can be defined as a binary relation R upon S: f: $\langle s_i, s_j \rangle \in R$, in which $s_i \in S$, $s_j \in S$. It is said that s_j is directly reachable to s_i, written as $s_i \rightarrow s_j$.

Definition 1.2c Let M be a sequence of states as follows: $M = [s_0, s_1, \ldots s_{n+1}]$, in which $s_i \in S$ $(0 \leqslant i \leqslant n + 1)$, then M is referred to as a world history iff for any pair of adjacent states $\langle s_i, s_{i+1} \rangle$, s_i is a history of s_{i+1}, i.e., there exists an action $f_i \in A$ such that f_i: $\langle s_i, s_{i+1} \rangle \in R$ $(0 \leqslant i \leqslant n)$.

Definition 1.2d Let c_I and c_G be an initial condition and a goal condition respectively, then an action sequence $P = [f_0, f_1, \ldots, f_n]$ forms a plan if P produces a world history $M = [s_0, s_1, \ldots, s_{n+1}]$ such that $c_I \supset s_0$ and $c_G \supset s_{n+1}$. It is said that s_{n+1} is reachable to s_0, written as $s_i \supset s_j$.

Definition 1.2e Let S be a set of potential world states, c_I an initial condition, and c_G a goal condition. A plan can be achieved by proving the following theorem:

$$(\forall s_0) s_0 \supset c_I \vdash (\exists s_n)((s_0 \Rightarrow s_n) \wedge (s_n \supset c_G))$$

which states that for an initial state $s_0 (s_0 \in S)$ in which c_I is logically true, there exists a state $s_n (s_n \in S)$ such that s_n is reachable to s_0 and c_G logically holds in s_n.

1.3 Classical problems

A problem of reasoning about plans is referred to as a classical problem if the problem domain is only concerned with a single agent as a potentially state-based model of the world. The classical problems of reasoning about plans have been generally considered as follows: at any instant of time, the world is regarded as being in one of a potentially infinite number of states. Various properties remain unchanged in each of these states. The world changes from one state to the next only by the occurrence of an action.

The classical problem framework has been used to express real-world problems that involve a single agent, such as a robot operating in an environment in which the world can be formalized as a sequence of state models and changes take place only as a result of the occurrence of actions. This framework underlies the development of many planners, such as *STRIPS* (Fikes & Nilsson, 1971), *ABSTRIPS* (Sacerdoti, 1973), *NOAH* (Sacerdoti, 1975), *NONLIN* (Tate, 1976), and others.

2 Three basic problems

Reasoning about plans gives rise to three basic problems: *the qualification problem*, *the ramification problem* and *the frame problem*. They result from attempting to formalize changes to the world, caused by the occurrence of events or actions. One of the major difficulties in reasoning about actions and plans has been identified as seeking solutions to these three problems, as it is impossible to discuss reasoning about plans without involving these problems. Historically, the various paradigms in the classical problem framework, result, to some extent, from the different solutions which have been proposed.

2.1 The qualification problem

The *qualification problem* was introduced by McCarthy (1977). It is also called *the interframe problem* (Shoham, 1986). The problem arises as a result of the fact that specifying the effects of actions is usually subject to qualification (Georgeff, 1987b). In other words, the qualification problem is concerned with specifying the preconditions under which actions can take place.

Suppose that α is an action, P_α denotes the precondition, and C_α the consequences, expressed as follows:

$$P_\alpha = p_1 \wedge p_2 \wedge \ldots \wedge p_m$$

$$C_\alpha = \{c_1, c_2, \ldots, c_n\}$$

where $p_i (1 \leqslant i \leqslant m)$ and $c_i (1 \leqslant i \leqslant n)$ are facts and relations regarding world states. The performance of action α can be defined as follows:

$$(\forall s_i)(s_i \supset P_\alpha) \vdash (\exists s_{i+1})((s_i \rightarrow s_{i+1}) \wedge (s_{i+1} \supset C_\alpha))$$

The difficulty is that it may not be epistemologically feasible to define P_α by explicitly listing all $p_i (1 \leqslant i \leqslant m)$ since there might be too many things that could possibly affect the qualification of α. As an example, consider all the factors which could prevent an aircraft from taking off.

2.2 The ramification problem

The ramification problem was named by Finger (1986). The problem is that the consequences of actions must be exhaustively recorded to represent changes of the world which are caused by the actions.

To record the consequences $C_\alpha = \{c_1 c_2 \ldots, c_n\}$ of α, all the $c_n (1 \leqslant i \leqslant n)$ have to be explicitly listed to represent intended changes of the world caused by α. However, for any given action α, there are potentially an infinite number of possible results that might occur, depending upon the details of the situation in which the action occurs (Georgeff, 1988). For instance, there are a great number of trivial changes which might result from moving a desk from one place to another in a study.

2.3 The frame problem

The frame problem was recognized by McCarthy and Hayes (1969). Essentially, it is the problem of describing and inferring all properties that remain unchanged when actions are performed. This problem has been regarded as a major difficulty for reasoning about actions and plans.

As an illustration for this and other issues, we can use a stacking example which illustrates planning problems in the block world. Suppose that in this world there are four blocks a_1, a_2, a_3, a_4, and the floor l which are characterized as *locations*. A state of the block world is described as follows:

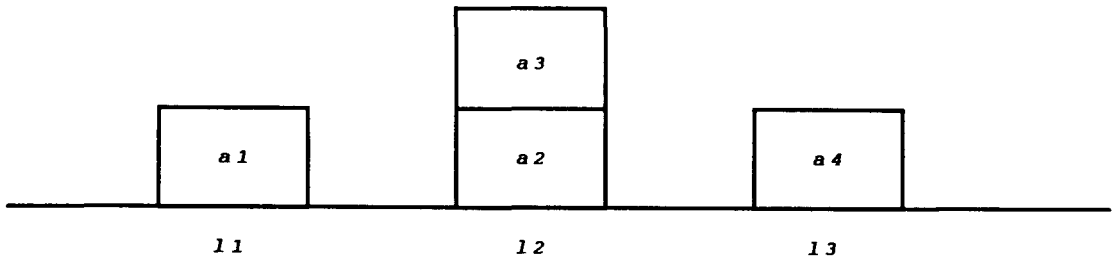

Figure 1

s_0: {

$on(a_1, l_1), clear(a_1),$

$on(a_2, l_2),$

$on(a_3, a_2), clear(a_3),$

$on(a_4, l_3), clear(a_4)$

},

which is illustrated in Figure 1. Consider two simple move actions in the block world. The two actions are defined as (1) *stack block x to the top of block y*, and (2) *unstack block x to location l*. They can be expressed as follows:

$$clear(x) \wedge clear(y) \supset on(x, y)_{d(stack(x, y))}$$

$$clear(x) \wedge clear(l) \supset at(x, l)_{do(unstack(x, l))}$$

These two descriptions are referred to as *the action descriptions* which represent intended changes of the world caused by the actions. However, to reason about the future of the world, describing what will remain unaffected is as important as describing what will be changed following an action. For instance, stacking $_{a1}$ onto a_4 leaves a_2 and a_3 untouched. Therefore, the following descriptions are also required to state all unchanged properties after action $stack(x, y)$:

$$on(z, z_1) \wedge \neg is(x, z) \supset on(z, z_1)_{do(stack(x, y))}$$

$$at(z, l) \wedge \neg is(x, z) \supset at(z, l)_{do(stack(x, y))}$$

$$clear(z) \wedge \neg is(z, y) \supset clear(z)_{do(stack(x, y))}$$

These descriptions are referred to as the *frame descriptions*. Similarly, the frame descriptions are also required to state all unchanged properties after action $unstack(x, l)$.

The difficulty is that a large number of such frame descriptions have to be explicitly provided for stating that many properties remain unaffected by actions. From an epistemological point of view, it is very tedious to list exhaustively all frame descriptions for every action and every property of interest. Generally, if there are m possible actions and n properties, then as many as $m \times n$ frame descriptions have to be provided for describing all properties which persist following actions (McCarthy & Hayes, 1969). As a result, most steps of a proof will focus on checking that many properties still remain true following every occurrence of event and action (McDermott, 1987).

2.4 Summary

The qualification problem, the ramification problem and the frame problem arise from attempts to formalize world changes caused by the occurrence of events or actions. To a great extent, the first two problems display a contradiction in attempting to draw a conclusion about the future with incomplete knowledge of the world. In practice, these two problems do not seem to be so difficult. This is because successful problem solving usually does not necessarily require a *complete knowledge of the world*, but only *knowledge related to the local problem*. For example, to find out what prevents an aircraft from taking off, the ground engineers do not have to worry about external

events like the runway is blocked or the weather is bad. That means that some appropriate assumptions regarding the world have to be made for effective problem solving. However, the frame problem reflects an intrinsic difficulty in formalizing changes of the world in any logic-based deductive process. As long as reasoning is dealing with *state*, which temporally represents the world, we will be confronted with the frame problem in one way or other.

There have been many solutions to these problems, especially to the frame problem: *STRIPS* (Fikes & Nilsson, 1971), *modal logic* (Schwind, 1978; Brown, 1978), *truth maintenance system* (Doyle, 1979), *non-monotonic logic* (McDermot & Doyle, 1980), *default logic* (Reiter, 1980), *circumscription* (McCarthy, 1980, 1986), and *abnormality* (McCarthy, 1986). All solutions can be viewed as meta-theories regarding the making of appropriate assumptions about the given problem domain (Poole et al., 1986).

3 The monotonic paradigm

The monotonic approach to reasoning about plans is inherently based on classical logics which obey the property of monotonicity. In this approach, a state-based world model is adopted to formalize the problem domains. A state of the world is represented as a set of sentences, whereas an action is characterized as a transition from state to state. By means of the classical rules of inference, the resolution theorem proving procedure is applied to infer a state in which the goal axiom can be proved.

3.1 Monotonic logics

Monotonic logics are logics such that logical conclusions of any theory cannot be invalidated by adding new knowledge. This property of monotonicity can be defined as follows:

Definition 3.1a A logic is referred to as a monotonic logic is when a proper axiom is added to a theory Th_1 to create a new theory Th_2, then any theorem p of Th_1 is also a theorem of Th_2, i.e.,

$$(\forall p)(Th_1 \vdash p) \wedge (Th_1 \subseteq Th_2) \vdash (Th_2 \vdash p)$$

All classical logics, such as propositional logic and predicate logic, are monotonic.

3.2 Situational calculus

The earliest formalism to represent reasoning about plans is *situational calculus* (McCarthy & Hayes, 1969). Situational calculus retains the monotonic character of classical logics, and inherits the complete expressiveness of first-order predicate logic. Therefore, situational calculus provides a well-defined semantics which facilitate the proving of completeness and soundness. Most paradigms so far developed adopt the fundamental notions of situational calculus and its variants to formalize problems.

To cope with changes of the world, a temporal notion, advanced as a new argument called *situation* or *state*, was introduced into situational logic. A situation represents a state of the world at a given instant of time:

Definition 3.2a Situation s is a temporal argument associated with a predicate such that a statement that $p(\mathbf{x})$ logically holds in situation s is expressed as $p(\boldsymbol{x}, s)$, where $\boldsymbol{x}$ is a vector $(x_1, x_2 \ldots, x_m)$ of variables.

For example, in the block world which we have considered previously, $on(a_2, a_3, s_0)$ describes that block a_2 is on block a_3 in situation s_0. Furthermore, a_2 may be moved elsewhere in a subsequent situation s_1, causing formula $clear(a_3, s_1)$.

The concept of situation was also introduced into the description of actions in situational calculus. The transformation from one situation to another is formalized as an event term that represents a relation between action and situation.

Definition 3.2b A transformation from situation s_x to s_y is an event term that can be described as the following relation R between s_x and s_y: $s_y = R(\alpha, s_x)$, where α is an action which occurred in situation s_x.

Returning to the block world example, suppose that block a_2 is unstacked from the top of a_3 to location l on the floor in situation s_0. Thus, the result of action $unstack(a_2, l)$ can be stated as follows:

$$(\forall s_0)(on(a_2, a_3, s_0) \wedge clear(a_2, s_0) \wedge clear(l, s_0)) \supset (clear(a_3, s_1) \wedge at(a_2, l, s_1))$$

in which $s_1 = R(unstack(a_2, l), s_0)$. Situational calculus provides explicit axioms to represent the action descriptions and frame descriptions, as discussed in Section 2.3. Two sets of axioms are used to formalize reasoning: (1) *action axioms*, which are concerned with all facts and relations that will change following each action; and (2) *frame axioms*, which are concerned with all facts and relations that will not be affected by each action.

In the block world example discussed in Section 2.3, the action axiom about $stack(x, y)$ is as follows:

$$(\forall x \forall y)(on(x, y, s) \wedge clear(x, s) \wedge clear(y, s)) \supset clear(y, R(stack(x, y), s) \wedge on(x, y, R(stack(x, y), s))$$

In addition, the following frame axioms are needed to state what remains unaffected by the action $stack(x, y)$:

$$(\forall z \forall z_1)(on(z, z_1, s) \wedge \neg is(x, z, s)) \supset on(z, z_1, R(stack(x, y), s))$$
$$(\forall z \forall l) at(z, l, s) \wedge \neg is(x, z, s) \supset at(z, l, R(stack(x, y), s))$$
$$(\forall z) clear(z, s) \wedge \neg is(z, y, s) \supset clear(z, R(stack(x, y), s))$$

A major obstacle with paradigms based on situational calculus is the frame problem. Two difficulties arise in the frame problem: (1) the *epistemological difficulty*, where a large number of frame axioms must be explicitly provided, describing all properties that are not affected by each action; and (2) the *computational difficulty*, in which all properties and relations must be examined to make sure that they still hold following every action.

4 Non-monotonic paradigms

Classical logics follow the principle of monotonicity, i.e., the addition of new theorems to a theory does not invalidate the conclusions which have been reached previously. The position, however, is different in human commonsense reasoning, where a conclusion is often reached with incomplete knowledge about the world. This implies that a conclusion might be subsequently revised when new knowledge is added.

In reasoning about actions and plans, any form of problem formalization will lead to an incomplete description of the world. That means that the reasoning is essentially carried out with incomplete knowledge of the world. The three basic problems, to a great extent, result from the conflict between the requirement of sufficient knowledge for problem solving and the impossibility of completely describing the world.

Classical logics strictly follow the property of monotonicity such that they may lack mechanisms to deal with an incomplete description of the world, for instance, drawing a conclusion with insufficient knowledge, or revising a conclusion that has been made previously. The limitation of monotonic logics has prompted the development of non-monotonic logics in which the principle of monotonicity is no longer obeyed.

4.1 Non-monotonic logic

Non-monotonic logics are logics such that the logical conclusions of any theory may be invalidated by adding new knowledge. This non-monotonicity can be defined as follows:

Definition 4.1a A logic is referred to as a non-monotonic logic if when a proper axiom is added to a theory Th_1 to create a new theory Th_2, some theorems of Th_1 may be invalidated in Th_2, i.e.,

$$(\exists p)(Th_1 \vdash p) \wedge (Th_1 \subseteq Th_2) \vdash_{not} (Th_2 \vdash p)$$

where the connective $\vdash_{not}$ denotes *non-provability*.

There are many circumstances in which knowledge about the world is so incomplete that some assumptions may have to be made for drawing a possible conclusion, but this conclusion is thereafter subject to being revised if and when more knowledge becomes available.

For instance, suppose that x is an animal. In the absence of any other knowledge about x, an assumption may be made with respect to the normal abilities of animals. Thus, a conclusion is inferred that x cannot fly. However, when more knowledge about x, such as *that* x is a bird, is added, then the previous conclusion about x is violated according to knowledge about birds, and it may be revised as: x can fly. Furthermore, such revisions could happen again with the addition of new knowledge about x, such as: x is a penguin.

A conclusion is referred to as a *belief* if it is drawn on the basis of some assumptions, and may be invalidated with the addition of new knowledge about the world. Non-monotonic logics intend to model beliefs with incomplete knowledge such that they can be revised in the light of new knowledge of the world.

As a result of making assumptions about the world, a number of sets of new beliefs may be produced from available beliefs under the mon-monotonic inference. A set of new beliefs is called an *extension* which represents an internally consistent and coherent state of the world.

McDermott and Doyle (1980a,b) created a general framework of non-monotonic logics by analysing non-monotonic semantics and provability. They presented a formalization of non-monotonic reasoning which captures most of the important properties, especially with regard to the structure of models of non-monotonic theories and their behaviour upon extension by new beliefs. McDermott and Doyle established their non-monotonic logic based on classical logics by extending them with a modality **M**, standing for *consistency*:

Definition 4.1b Let p be a formula, if $\neg p$ can not be proved from all the beliefs available, then p is regarded as being consistent with everything believed.

Definition 4.1c Let p be a formula, **M** is a modal operator such that **M**p denotes that p is consistent with all the beliefs available.

Definition 4.1d A new rule of inference in the context of non-monotonic logic can be defined as: *if* $\neg p$ *is not provable, then* **M**p *can be inferred*, i.e., $\neg provable(\neg p) \vdash \mathbf{M}p$.

For example, consider the following theory which formalizes the belief about birds with respect to their ability to fly:

1 $(\forall x)(bird(x) \wedge \mathbf{M}ability\text{-}to\text{-}fly(x)) \supset fly(x)$
2 $bird(penguin)$
3 $fly(penguin)$

(1) is translated as: if x is bird and if x's ability to fly is consistent with general beliefs about birds, then that x can fly is inferred. According to (1) and (2), conclusion (3) can be drawn. However, (3) may become inconsistent with the beliefs when knowledge about penguins is added. Consequently, (3) may be revised according to the new set of beliefs.

Other important approaches to non-monotonic reasoning include *truth maintenance system* (Doyle, 1979), *default logic* (Reiter, 1980) and *circumscription* (McCarthy, 1980, 1986). In the context of reasoning about actions and plans, we will only discuss default logic and circumscription.

5 Default logic

Default logic was proposed by Reiter (1980). As one of the formalizations of non-monotonic reasoning, a noteworthy feature of default logic is the introduction of default rules which serve as

additional rules of inference. Default reasoning can be generally described as the process of drawing a conclusion in the absence of any information which is inconsistent to do so.

Definition 5a Let x, $y_i(1 \leqslant i \leqslant m)$ and z be sentences of first-order predicate logic, then a default is represented as follows:

$$\frac{x\text{: } \mathbf{M}y_1, \ldots, \mathbf{M}y_m}{z}$$

where x is called *the prerequisite*, $y_i\,(1 \leqslant i \leqslant m)$ *the assumptions*, and z *the consequence*. The syntax of default rule can be translated as: if x holds, and it is consistent to believe that $y_i(1 \leqslant i \leqslant m)$ hold, then z is believed to hold.

One important example of default reasoning is the so-called *closed world assumption*: only positive knowledge about the world need be explicitly represented. Negative knowledge is not so represented, but is inferred by default (Reiter, 1978).

Definition 5b Let p be a predicate, and $\boldsymbol{x}$ a vector $(x_1, x_2, \ldots, x_m)$ of variables, then the closed world assumption is formally expressed as follows:

$$\frac{:\mathbf{M}\neg p(\boldsymbol{x})}{\neg p(\boldsymbol{x})}$$

which states that, for any predicate p and vector $\boldsymbol{x}$, $\neg p(\boldsymbol{x})$ can be assumed whenever it is consistent to do so.

Unlike traditional monotonic inference, default rules are defeasible. For example, in Definition 5a, z can be inferred in the absence of knowledge which contradicts beliefs $y_i(1 \leqslant i \leqslant m)$; but once any $\neg y_i$ becomes provable, then the default rule is no longer applicable and belief z may be no longer true. Thus, default reasoning is non-monotonic.

For instance, if x is a bird, and there is no knowledge available suggesting that x should not fly, then the belief that x can fly is consistently inferred, this is expressed as follows:

$$\frac{bird(x)\text{: } \mathbf{M}fly(x)}{fly(x)}$$

But as soon as $\neg fly(x)$ is provable with the addition of knowledge, such as that x is a penguin becomes known, belief $fly(x)$ will no longer be true.

For a set D of default rules and a set W of beliefs about the world, default reasoning enables one to derive additional information which, together with D, is composed of a more complete extension of beliefs. Reiter formally defined the concept of extension in order to provide a proof theory of default logic.

Definition 5c A default theory T is defined as the following tuple: $T = \langle D, W\rangle$, where D is a set of default rules, and W is a set of sentences describing the facts that are believed to be true.

A default theory $T = \langle D, w\rangle$ may define none or more sets of sentences which are derived by applying D to W, and these sets are referred to as the extensions of default theory $T = \langle D, W\rangle$.

Definition 5d $T = \langle D, w\rangle$ is a default theory. For any set S of closed first-order wffs, let $\Gamma(S)$ be the smallest set with the following properties:

1 $W \subset \Gamma(S)$;

2 $\Gamma(S)$ is a subset of closed *wffs* of L such that

$$\Gamma(S) = Th_L(\Gamma(S)) = \{s | s \in L \wedge \Gamma(S) \vdash s\};$$

3 for any default

$$\frac{x\text{: } \mathbf{M}y_1, \ldots, \mathbf{M}y_m}{z} \in D$$

if $x \in \Gamma(S)$ and $\neg y_i (1 \leqslant i \leqslant m) \notin \Gamma(S)$, then $z \in \Gamma(S)$. Any set E of closed first-order wffs is called an extension of $T = \langle D, W \rangle$ if $E = \Gamma(E)$. Note that a wff of L is said to be closed iff it contains no free variables.

Definition 5e Let $T = \langle D, W \rangle$ be a default theory. For any sentence p, if p is derivable from $T = \langle D, W \rangle$, i.e., $T = \langle D, W \rangle \vdash_D p$, then p must belong to one of extensions of $T = \langle D, W \rangle$. $\vdash_D$ denotes provability under the default inference.

Any extension of default theory $T = \langle D, W \rangle$ is essentially a possible set of beliefs derived from W under D. Extensions can be looked upon as internally consistent and coherent states of the world. Default reasoning is really an inferential process of selecting one extension of a default theory, and then reasoning about this extension until new knowledge establishes a further modification of beliefs, and hence results in selecting a new extension (Hanks & McDermott, 1987).

As was discussed in Section 2, the basic problem with the monotonic approach to reasoning about plans is that a great number of the frame axioms must be provided to state all properties which remain unchanged following every action. Default logic suggests a solution to the problem by using the default rules to infer unchanged properties. The basic idea is that a frame default is coupled with world models such that the frame axioms are neither explicitly represented nor explicitly used in reasoning.

Definition 5f Let $p(x)$ be a sentence of first-order language, α an action and s a situational argument, then a frame default is expressed as follows:

$$\frac{p(x, s): \mathbf{M}p(x, R(\alpha, s))}{p(x, R(\alpha, s))}$$

which states that if $p(x)$ holds in situation s, and it is consistent to believe that $p(x)$ still holds following action α, then $p(x)$ is believed to hold in situation $R(\alpha, s)$. According to this default rule, every action α is assumed to leave every $p(x)$ unaffected unless it is possible to prove $p(x)$ has been changed by action α.

By using default rules, the solutions to the qualification problem were also proposed by some researchers (McCarthy, 1980, 1986; Shoham, 1986; Ginsberg, 1988b; Lifschitz, 1987a; Goebel, 1987). Suppose that α is an action with the qualification P_α, the consequence C_α and the negative qualification N_α which disables the intended effect of α. A predicate *prevent*(x) is defined as: something prevents action x from being successfully performed. Thus, action α can be expressed as follows:

$$(hold(P_\alpha, s) \wedge \neg prevent(\alpha, s)) \supset hold(C_\alpha, R(\alpha, s))$$

Without computing the negative qualification, it is consistently believed to be false unless knowledge to the contrary can be proved. To achieve this, a default is defined as follows:

$$\frac{hold(P_\alpha, s): \mathbf{M}\neg hold(N_\alpha, s)}{\neg prevent(\alpha, s)}$$

which means that, for any action α, if P_α holds and $\neg N_\alpha$ is consistently believed to hold, then $\neg prevent(\alpha, s)$ can be inferred, i.e., there is nothing that prevents α from being successfully applied in situation s.

Although default logic suggests a promising solution to the problems of the monotonic paradigms, some problems still remain to be solved. For example, a computational difficulty stems from the satisfiability tests required in the proof procedure. Ginsberg et al. (1988a,b) discussed the potential complexity in computing the negative qualification of actions when a complicated domain was involved. Furthermore, a foundational difficulty was uncovered by Hanks and McDermott (1986). They presented the so-called *Yale shooting problem* to show that the default reasoning formalism was still inadequate for reasoning about plans. Various methods to implement default reasoning have been proposed to overcome those problems; Shoham (1986) developed a default

reasoning associated with temporal reasoning; Morris (1986) proposed an alternative form of default reasoning which was connected with *truth maintenance system*, etc.

6 Predicate circumscription

Circumscription was introduced by McCarthy (1980, 1986) as a general formalization of non-monotonic reasoning, extending classical first order logic. Being used along with the classical rules of inference in first order logic, circumscription acts as a rule of conjecture to arrive at conclusions about the world. The correctness of the conclusions depends on the available beliefs about the world that are taken into account when the circumscription is made. This rule of conjecture can be described as: *the objects that can be shown to have a certain property P by reasoning from certain facts A are all the objects that satisfy P* (McCarthy, 1980).

For instance, a car can be driven unless there is something wrong with it or there exists something which prevents the car from being driven. Circumscription may be applied as a rule to conjecture that the only factors which keep the car from being driven are those that can be confirmed from all known facts about the car. In other words, if no evidence, such as the presence of a wheel clamp or an engine malfunction, can be proved from the known facts, then the car should be concluded to be in normal condition for driving.

Definition 6a Let $A(P)$ be a sentence of first order language L, $P(\mathbf{x})$ a predicate in which $\boldsymbol{x}$ is a vector $(x_1, \ldots, x_m)$, and Φ a predicate expression, then the circumscription of predicate $P(\mathbf{x})$ in sentence $A(P)$ can be defined as the following sentence schema:

$$A(\Phi) \wedge (\forall \boldsymbol{x})(\Phi(\boldsymbol{x}) \supset P(\mathbf{x})) \supset (\forall \boldsymbol{x})(P(\mathbf{x}) \supset \Phi(\boldsymbol{x}))$$

In this sentence schema, $A(\Phi)$ assumes that Φ satisfies what $P(\mathbf{x})$ satisfies in $A(\Phi)$. $(\forall \boldsymbol{x})\Phi(\boldsymbol{x}) \supset P(\mathbf{x})$ assumes that the vectors $\boldsymbol{x}$ satisfying $\Phi(\boldsymbol{x})$ also satisfy $P(\mathbf{x})$. As a result, it is concluded, by asserting $(\forall \boldsymbol{x})P(\mathbf{x}) \supset \Phi(\boldsymbol{x})$, that the vectors satisfying $\Phi(\boldsymbol{x})$ are those that have to satisfy $P(\mathbf{x})$ in $A(P)$.

In Definition 6a, A can be regarded as some known facts, and P a certain property by reasoning from A. The circumscription of P formalizes a special rule of inference: for any property Φ, if A still holds by replacing P with Φ in A, and that object $\boldsymbol{x}$ satisfies Φ logically implies that object $\boldsymbol{x}$ satisfies P, then property Φ is not something else, but coincidentally P.

Definition 6b If sentence q can be proved by deduction under the circumscription of predicate $P(\boldsymbol{x})$ in $A(P)$, then sentence q is referred to as a theorem of $A(P)$. This is denoted as follows: $A(P) \vdash_c q$, in which $\vdash_c$ is called *circumscriptive inference*.

Consider the following illustration. Suppose that A is a sentence about our example block world:

$$(\forall x \forall y \forall s) on(y, x, s) \supset \neg clear(x, s)$$

This asserts that x is not clear if there exists a block y on x in situation s. Circumscribing predicate $\neg clear(x, s)$ syntactically formalizes the inference that the blocks that can be shown satisfying $\neg clear(x, s)$ by reasoning from facts A are coincidentally those blocks that have to satisfy $\neg clear(x, s)$, this can be expressed as follows:

$$(\forall x \forall y \forall s)(on(y, x, s) \supset \Phi(x, s)) \wedge (\Phi(x, s) \supset \neg clear(x, s)) \supset (\neg clear(x, s) \supset \Phi(x, s)) \quad (1)$$

Suppose that s is a situation regarding the block world consisting of five blocks and the floor:

s:{

$on(a_1, a_2)$, $clear(a_1)$

$on(a_3, a_4)$, $clear(a_3)$

$on(a_5, floor)$, $clear(a_5)$

}.

If we substitute $\Phi(x, s) \equiv (on(a_1, x) \vee on(a_3, x))$ into (1), then we have

$$(\forall x \forall s) \neg clear(x, s) \supset (on(a_1, x) \vee on(a_3, x)) \tag{2}$$

Consequently, we have all the theorems of A under the circumscription of $\neg clear(x, s)$:

$$A \vdash_c \neg clear(a_2, s), A \vdash_c \neg clear(a_4, s)$$

To represent knowledge about an entity, it is usually stated which property is normally true unless a certain abnormality is mentioned against the normal case of the entity. For instance, a bird can normally fly unless this bird is abnormal with respect to the ability to fly. To reason about the world with incomplete knowledge, the abnormal aspects of entities should be represented in such a way that normality may be assumed about other aspects of the entities. Circumscription can be used to minimize the abnormality of entities involved. To do this, McCarthy (1986) introduced a predicate *ab* to represent the abnormal aspects of entities.

Definition 6c *ab* is such a predicate that any one of abnormal aspects of entity z can be expressed as follows: $ab(aspect, z)$. *ab* is referred to as *abnormality predicate*.

For instance, the following sentence axiomatizes the fact that a bird can normally fly unless the bird is abnormal with respect to the ability to fly:

$$(\forall x)(bird(x) \wedge \neg ab(ability\text{-}to\text{-}fly, x)) \supset fly(x)$$

which states that x can fly if x is a bird and x is normal with respect to the ability to fly. Circumscribing $\neg ab$ in the sentence inferentially assumes the normality of x.

Definition 6d Any action α with the consequence C_α can be axiomatized by using the abnormality predicate *ab*

$$(\forall s) \neg ab(aspect, \alpha, s) \supset hold(C_\alpha, R(\alpha, s))$$

which states that action α in situation s will successfully achieve its intended effect C_α unless something is abnormal in s.

For example, action $stack(x, y)$ is described as follows:

$$(\forall x \forall y \forall s) \neg ab(aspect, stack(x, y), s) \supset on(x, y, R(stack(x, y), s))$$

and the abnormal aspects which prevent action $stack(x, y)$ from being successfully performed could be:

$$(\forall x \forall y \forall s) \neg clear(x, s) \supset ab(aspect1, stack(x, y), s)$$

$$(\forall x \forall y \forall s) \neg clear(y, s) \supset ab(aspect2, stack(x, y), s)$$

The predicate *ab* can be circumscribed in the conjunction of the two sentences above to minimize the abnormal aspects of action $stack(x, y)$:

$$(\forall x \forall y \forall s) \neg clear(x, s) \wedge \Phi(ab(aspect1, stack(x, y), s)) \wedge (\forall x \forall y \forall s) \neg clear(y, s)$$

$$\wedge \Phi(ab(aspect2, stack(x, y), s)) \wedge (\forall x \forall y \forall s \forall z) \Phi(z)$$

$$\supset ab(z, stack(x, y), s) \supset (\forall z) ab(z, stack(x, y), s) \supset \Phi(z)$$

As can be seen from Definition 6d, the precondition of action α is not explicitly listed. Instead, only the abnormal aspects of α are specified. The circumscription of *ab* minimizes the abnormality by stating that the only factors that can disable an action are as already specified. Thus, circumscription provides a solution to the qualification problem in reasoning about plans.

Circumscription also provides a solution to the frame problem. The basic idea is that, by assuming the normality of an action, any property that cannot be proved to change remains as it was, following the action. McCarthy used an axiom schema to state that property p remains unchanged in situation $R(\alpha, s)$ following action α in situation s unless a certain aspect of action α is abnormal. For instance, $on(x, y, s)$ remains unchanged following action α, if α consists in doing something other than moving block x away from y:

$$(\forall x \forall y \forall s) \neg ab(aspect1, \alpha, s) \wedge on(x, y, s) \supset on(x, y, R(\alpha, s))$$

Similarly, $at(x, l, s)$ and $clear(y, s)$ are not effected if action α neither intends to move x away from location l nor to move another block onto y:

$$(\forall x \forall l \forall s) \neg ab(aspect2, \alpha, s) \wedge at(x, l, s) \supset at(x, l, R(\alpha, s))$$

$$(\forall y \forall s) \neg ab(aspect3, \alpha, s) \wedge clear(y, s) \supset clear(y, R(\alpha, s))$$

By circumscribing predicate ab, the abnormality of an action will be minimized such that property p is inferred to remain unaffected by action α in any situation which does not make α abnormal.

Circumscription provides a formalized rule of conjecture that can act as a special inference rule for non-monotonic reasoning. In reasoning about plans, causal connections between actions and changes can be axiomatized by using circumscription such that the frame problem and the qualification problem may be effectively solved. However, circumscription suffers the same problem as default logic. The result of circumscription is still inadequate for the successful formalization of reasoning about actions. McCarthy discussed an example to illustrate the difficulty. Details can be found in McCarthy (1986) and Lifschitz (1987b).

Default reasoning and circumscription can be applied to formalize reasoning about actions in the framework of situational calculus. An effective solution to the frame problem is provided in both methods. Compared with default logic, circumscription is more general because it is applicable to any predicate, although they produce the same results in many simple cases (McCarthy, 1986). Both McCarthy and Reiter suggested that there might be some interesting relationships between default logic and circumscription.

7 The *STRIPS* calculus

A different approach to reasoning about plans is STRIPS due to Fikes and Nilsson (1971). As one of the alternatives to situational calculus, the STRIPS formalism has been widely used in many planning systems, such as ABSTRIPS (Sacerdoti, 1973), NOAH (Sacerdoti, 1975, 1977), WARPLAN (Waldinger, 1977), NONPLAN (Tate, 1977) and SIPE (Wilkins, 1983).

The STRIPS formalism does not explicitly deal with *situation* or *state*. As a special case of a default schema, STRIPS maintains a single model of world state at any instant of time by assuming that every property that exists in the model must be true. A change of the world is formalized as a state transformation which, as a result of performing an action, is a process of deleting properties from and adding properties to the current state model of the world. From an initial state model of the world, a goal formula and a set of possible actions, STRIPS attempts to synthesize a sequence of actions that transforms the initial world model to the one in which the goal formula is satisfied.

Definition 7a A STRIPS system Σ can be formally defined as the following 4-tuple: $\Sigma = \langle L, P, C, OP \rangle$, where L is a first order language, $P \subset L$ is set of predicates, C is a set of formulae which contains all true axioms, and OP is a set of STRIP operator schemata.

In a particular STRIPS system, a world model is a set of sentences of L; predicates in $P \subset L$ describe the state attributes of objects in the world; C axiomatizes all the domain constraints; and OP characterizes all actions. As an example, let us consider a STRIPS system $\Sigma = \langle L, P, C, OP \rangle$ for the block world, in which

$L = \{$

$\neg, \vee, \wedge, \supset, \ldots$ /*connectives*/

$A, B, \ldots$ /*constants*/

$a, b, \ldots$ /*variables*/

$on, at, above, clear, \ldots$ /*predicates*/

$\}$.

$P = \{$

$on(x, y)$/*x is on y*/,

$at(x, l)$/*x is at l*/,

$above(x, y)$/*x is above y*/,

$clear(x, y)$/*the top of x is clear*/,

....

$\}$.

$C = \{$

$(\forall x \forall y) on(x, y) \supset \neg clear(y)$,

$(\forall x \forall l) at(x, l) \supset \neg clear(l)$,

$(\forall x \forall y) on(x, y) \supset above(x, y)$,

$(\forall x \forall y \forall z) on(x, y) \wedge y \neq z \supset \neg on(x, z)$,

.....

$\}$.

In this example, L contains a subset P of predicates which is defined to describe the spatial relationships between blocks. C states the domain constraints in the block world. Any sentence in C is true in any situation. We ignore the set OP of operators for the time being.

In STRIPS, actions are formalized by operator schemata that organize all possible actions into different families. To describe a family of actions, an operator schema is established by specifying the preconditions and consequences of the actions.

Definition 7b An operator schema $OP(x_1, \ldots, x_m)$ can be defined by an operator description consisting of a triple: $\langle OP_P, OP_D, OP_A \rangle$, in which OP_P is called *the preconditions* which is a conjunction of atomic formulae of L, OP_D and OP_A are sets of sentences of L, and are respectively called *the delete list* and *the add list*.

An instance operator $OP(X_1, \ldots, X_m)$ is obtained by substituting a specific constant vector $X = (X_1, \ldots, X_m)$ for the parameter vector $x = (x_1, \ldots, x_m)$ in $OP(x_1, \ldots, x_m)$, i.e., $OP(x)|_{x=X}$. During the problem-solving process, the parameter vector x will have been already replaced with the specific constant vector X whenever an operator becomes applicable to a world model.

For instance, the STRIPS system for the block world, as described previously, may have a set OP of operator schemata as follows:

$OP = \{$

$move_b(x, y, z)$

OP_P: $on(x, y)$, $clear(x)$, $clear(z)$

OP_D: $on(x, y)$, $clear(z)$

OP_A: $on(x, z)$, $clear(y)$

$move_t(x, l_1, l_2)$

OP_P: $at(x, l_1)$, $clear(x)$, $clear(l_2)$

OP_D: $at(x, l_1)$, $clear(l_2)$

OP_A: $at(x, l_2)$, $clear(l_1)$

$stack(x, l, y)$

OP_P: $at(x, l)$, $clear(x)$, $clear(y)$

OP_D: $at(x, l)$, $clear(y)$

OP_A: $on(x, y)$, $clear(l)$

$unstack(x, y, l)$

OP_P: $on(x, y)$, $clear(x)$, $clear(l)$

OP_D: $on(x, y)$, $clear(l)$

OP_A: $at(x, l)$, $clear(y)$

. . . . .

$\}$.

The satisfaction of OP_P will lead to the application of operator $OP(X)$ which is fully instantiated. As a result, the state model initiating $OP(X)$ is modified with two mechanisms: deleting OP_D from and adding OP_A to this model. Consequently, a new world model is produced.

Definition 7c Let s_i be a state model of the world, and $OP(X)$ an operator. If $s_i \supset OP_P$, then the application of $OP(X)$ will produce a new model s_{i+1} as follows:

$$s_{i+1} = (s_i - OP_D(X)) \cup OP_A(X)$$

where $s_i - OP_D(X) = \{x | x \in s_i \wedge x \notin OP_D(X)\}$.

A problem space with respect to a STRIPS system $\Sigma = \langle L, P, C, OP \rangle$ is defined as a triple $\langle S, s_1, c_G \rangle$, where S is a set of state models, $s_1 \in S$ is the initial world model and c_G is the goal statement which is a set of axioms. A plan $\Gamma = [\alpha_1, \ldots, \alpha_m]$ with respect to $\langle S, s_1, c_G \rangle$ is accepted by $\Sigma = \langle L, P, C, OP \rangle$ if Γ defines a sequence of world models $M = [s_0, \ldots, s_m]$ such that $s_i = (s_{i-1} - \alpha_D) \cup \alpha_A$, in which $\alpha_i (1 \leqslant i \leqslant m)$ are operators and $s_{i-1} \supset \alpha_P (1 \leqslant i \leqslant m)$. Note that $s_0 \supset s_I$ and c_G.

A theorem proving procedure is applied to synthesize plans. For a given problem space $\langle S, s_I, c_G \rangle$, the theorem prover starts to search for a plan by attempting to prove $s_0 \supset c_G$. If it succeeds, then the problem is trivially solved. Otherwise the unsuccessful proof is specified as the difference between s_0 and c_G. In order to reduce this difference, the GPS means-ends analysis strategy (Newell & Simon, 1963) is applied to search for the operators which may affect s_0 such that the unsuccessful proof can be continued. As a result, a number of subgoals are set up in terms of the preconditions of those relevant operators. The prover works on these subgoals with depth-first resolution, following the same procedures with which it attempted to prove $s_0 \supset c_G$, until it produces a world model s_m in which $s_m \supset c_G$.

By assuming that all the undeleted properties still remain true after an action, STRIPS proposes a very effective solution to the frame problem. As a result, the lack of frame axioms allows reasoning to concentrate on the search effort. However, the STRIPS formalism is not as expressive as situational calculus (Waldinger, 1977). The operator calculus behaves very awkwardly in many circumstances. For instance, an operator description requires the delete list to include explicitly all sentences that could be possibly affected by an action, although the truth value of some of them can be strictly deduced from other axioms. In addition, the operator formalism fails to maintain the consistency of world models. For instance, if q is a sentence, for a fully instantiated operator $OP(X) = \langle OP_P, OP_D, OP_A \rangle$, if $OP_P \supset \neg q$ and $OP_D \not\supset \neg q$, then there is no way to prevent $OP_A \supset q$, Thus, operator $OP(X)$ could result in a world model s in which $s \supset q$ and $s \supset \neg q$, Furthermore, Lifschitz (1986) has shown that there are certain problems with respect to the semantics of operator descriptions in STRIPS.

8 The possible worlds semantics and modal logic

Over the last decade, many non-classical logics have been introduced into artificial intelligence. A different theoretical framework based on the possible worlds semantics and modal logic has been gradually established to approach reasoning about plans. This approach inherits a view of the world as a set of state models and of actions as a binary relation between state models. Kripke structures (Kripke, 1963) were introduced to model the problem domains as a collection of possible worlds with some relations. By characterizing actions in terms of some kind of state modal logic, such as Z (Brown, 1986a,b, 1987a,b) and ZK (Schwind, 1978a,b, 1987), the relative possibilities between possible worlds can be semantically represented and inferred to allow reasoning about plans to be elegantly formalized.

8.1 The possible worlds semantics

The concept of possible worlds was first introduced to develop the semantics for the modal logics of necessity and possibility (Bradly & Swartz, 1979). As an abstraction of a real world or process, a possible world may be intuitively regarded as a set of states of affairs that might have been true in the real world. A possible world is not an independent entity, i.e., a possible world can be defined only when it is possibly related to other worlds (Kripke, 1963). Such a relation of one world being a possible alternative to another is referred to as *the accessibility relation*.

Definition 8.1a Let w_i and w_j be two worlds, and R be a binary relation, world w_j is regarded as a possible world with respect to world w_i only if w_i is possibly related to w_j by R, this can be expressed as: $\langle w_i, w_j \rangle \in R$, where R is called *the accessibility relation* that represents the possibility to relate two possible worlds.

It was shown by Kripke that the differences between the most important axiom systems of modal logic consist in some restrictions upon the accessibility relation of the possible-world models in these axiom systems (Kripke, 1971). In terms of the possible world analysis, Kripke proposed a structure to create a semantic framework for modal logics (Kripke, 1963).

Definition 8.1b Let $S = \{I_i: w_i \in W\}$ be a classical structure in which W is a non-empty set of possible worlds, I an interpretation upon W, and R a binary relation upon W. A modal structure is defined as a pair: $K = \langle S, R \rangle$. For $w_i \in W$ and $w_j \in W$, $w_i R w_j$ denotes that w_j is possible with respect to w_i or w_j is accessible from w_i. K is called *the Kripke structure*.

The Kripke structure constitutes the model theoretic semantics for modal logic. Furthermore, it can be used effectively to formalize reasoning about plans: a Kripke structure $K = \langle \{I_w: w \in W\}, R \rangle$ is constructed with a set W of possible worlds, each of which can be interpreted as the description of a state of affairs regarding the actual world at an instant of time. The accessibility relation R upon W characterizes relations between possible worlds. Actions can be modelled in terms of the properties

of R such that for any $w_i \in W$ abd $w_j \in W$, w_j may be accessible from w_i iff there is an action or a sequence of actions which transforms the world state described by w_i into the world state described by w_j.

8.2 Modal logic

Classical logics are concerned with indicative sentences which may be referred to as truth-functional, i.e., each atomic sentence can be interpreted by assigning a value of true or false. As an extension of classical logics, modal logics are not concerned only with indicative sentences, but also with modal sentences that represent states of affairs or possible worlds. Consider, for instance, the following sentences:

1 Block a_1 is on block a_2.
2 It is possible that block a_1 is on block a_2.

Sentence (1) is undoubtedly an indicative sentence because it can be interpreted unequivocally as true or false. Sentence (2) is a modal sentence. It is true if sentence (1) is true, but may be true or false even if (1) is false.

Any modal logic must include the following two modal operators: (1) **N**, which means that *it is necessary (or logical) that*; and (2) **P**, which means that *it is possible that*. **N** and **P** define necessary truth and possible truth respectively in modal logics.

Definition 8.2a Sentence s is necessarily true—expressed as **N***s*—in a world w, iff s is true in all worlds which are possible with respect to w. **N***s* is referred to as the necessary truth of sentence s.

Definition 8.2b Sentence s is possibly true—expressed as **P***s*—in a world w, iff s is true in at least one world which is possible with respect to w. **P***s* is referred to as the possible truth of sentence s.

Modal operator **P** can be defined in terms of modal operator **N**: for any sentence s, $\mathbf{P}s \Leftrightarrow \neg\mathbf{N}\neg s$. This can be translated as: sentence s is possibly true iff it is not necessarily untrue.

As a simple extension of classical logics, a basic modal logic includes: (1) all the machinery of classical logics, (2) modal operator **N** and **P**, (3) the law $\mathbf{P}s \Leftrightarrow \neg\mathbf{N}\neg s$, and (4) the additional rules of inference. For instance, most modal logics take the so-called Godel rule as an additional rule of inference (Godel, 1933):

$$(Th \vdash s) \supset (Th \vdash \mathbf{N}s)$$

In addition, a basic modal logic can be extended to meet the requirements of different deductions by adding additional axioms. For example, accessibility relations can be axiomatized in terms of their various properties. Thus, by including these additional axioms, a more comprehensive and appropriate modal logic can be constructed to support the Kripke structures. The following axioms are related to some of the main properties of the accessibility relation used to construct appropriate Kripke structures for different applications. Other properties can be axiomatized in the same way:

1 *Reflexivity*: $\mathbf{N}s \vdash s$. If sentence s is necessarily true in a world w, then s is true in w.
2 *Transitivity*: $\mathbf{N}s \vdash \mathbf{NN}s$. If sentence **N***s* is necessarily true in a set W_1 of all worlds which are possible with respect to w, then s is true in the set W_2 of all worlds which are possible with respect to W_1.
3 *Symmetry*: $s \vdash \mathbf{NP}s$. If sentence s is true in a world w, then **P***s* is necessarily true in all worlds which are possible with respect to w.

For example, for a Kripke structure K with an accessibility relation R that is both reflexive and symmetric, then K should be constructed on the basis of a basic modal logic plus axioms (1) and (3).

8.3 The modal logic paradigm

The theoretical framework of reasoning about plans based on modal logics and the possible worlds semantics are due to contributions from a number of researchers. Schwind (1978a,b, 1983, 1986,

1987) proposed an action theory in terms of a modal logic, ZK, which was semantically characterized by a Kripke-type structure with two accessibility relations. Moor (1980) described a formal theory of knowledge and action, based on the possible-world semantics and modal logic, to represent agents' knowledge about the world and actions in planning. Brown (1986a,b) constructed the modal logic Z for non-monotonic reasoning. Furthermore, Brown and Park (Brown, 1986a,b, 1987a,b; Park, 1987) defined an action model by using the modal logic Z as a formal basis. Kautz (1982) described a model for plan generation in terms of a dynamic logic which was essentially an extended modal logic. Basically, the various approaches all concentrate on the construction of action models to formalize reasoning about actions. In the following, we emphasize Schwind's work since it displays a complete framework of the modal logic approach.

Schwind's action theory is developed on the basis of a modal logic ZK, which is an extension of classical predicate logic plus: (1) modal operator **N** and **P**; (2) the law $\mathbf{P}s \Leftrightarrow \neg\mathbf{N}\neg s$; (3) the temporal operators $+$, which *means for all immediately following worlds and* $-$, *for all immediately preceding worlds*; (4) additional rules of inference and axioms (Schwind, 1987).

ZK is characterized semantically by a Kripke-type structure with two accessibility relations formalized by the temporal operators $+$ and $-$, and modal operators **N** and **P**.

Definition 8.3a The model theoretic semantics of ZK is a Kripke-type structure which is defined as a triple: $\langle S, R_1, R_2\rangle$, in which $S = \{I_w: w \in W\}$, R_1 is called *the state relation*, and R_2 *the modal relation* which is reflective and equivalent. R_1 characterizes state operator $+$ and $-$, and R_2 modal operator **N**.

In terms of ZK structure, Schwind's action theory is developed by introducing the notion of action model. An action model consists of: (1) a ZK structure, which constructs a problem domain, (2) some general laws, which represent all the domain constraints, (3) a set of actions, and (4) an initial world description. Let $T(w, s)$ be a function representing the truth value of sentence s in $w \in W$, then the formal definition of ZK action model is as follows:

Definition 8.3b A ZK action model is the following 5-tuple:

$$AM = \langle M, g, AC, w_0, d_0\rangle$$

1 $M = \langle\{I_w: w \in W\}, R_1, R_2\rangle$ is a ZK structure;
2 g is a formula of first-order language, and represents the general laws of AM. Hence, $T(w, g) = true$, $w \in W$;
3 $AC = \{\langle P_i, C_i\rangle | 1 \leqslant i \leqslant n\}$ is a set of actions. Formula P_i and C_i are, respectively, precondition and effect;
4 $w_0 \in W$ is an initial world;
5 d_0 is a formula which completely describes the initial world w_0, and $T(w_0, d_0) = true$.

In a ZK action model $AM = \langle M, g, AC, w_0, d_0\rangle$, for any action $\alpha \in AC$, a non-logical axiom is established to describe: if action α is applicable to a world w_i in which a property p is true, and p and C_α are consistent with the general laws g, then there must be an immediately following world w_{i+1} in which p and C_α are still true:

$$P_\alpha \wedge p \wedge \mathbf{P}(p \wedge C_\alpha \wedge g) \supset \neg + \neg(p \wedge C_\alpha \wedge g)$$

This non-logical axiom is called the *execution axiom* or the *action law*, and added to the action model to support deduction in the following two aspects:

1 *persistence of property*. p persists if it is unaffected by action α and is consistent with C_α and g.
2 *detection of inconsistency*. Unchanged property p following action α will be no longer true if it is inconsistent with C_α or g. Thus, an effective solution to the frame problem can be achieved by establishing such execution axioms.

In the ZK action theory, the execution of any action is characterized as a process of deduction. As an example, for instance, the execution of action *stack*(x, y) is axiomatized as a proof that the following sentence is a theorem of the action model AM

$$g \wedge clear(x) \wedge clear(y) \vdash \neg + \neg(g \wedge on(x, y))$$

which means that if $clear(x)$ and $clear(y)$ are true, and they are consistent with general law g, then $on(x, y)$, which is consistent with g, holds in the following state resulting from $stack(x, y)$. Furthermore, state relation R_1 is logically related to actions. For $w_1 \in W$ and $w_2 \in W$, w_2 is possible with respect to w_1. i.e., $w_1 R_1 w_2$, iff there is an action to transform w_1 to w_2 such that

$$AM \vdash ((g \wedge d_1) \vdash \neg + \neg(g \wedge d_2))$$

where $T(w_1, d_1) = true$ and $T(w_2, d_2) = true$. Accordingly, for a goal sentence c_G specifying a particular problem, reasoning attempts to search for a path $[w_1, w_2, \ldots, w_n]$ such that

$$AM \vdash ((g \wedge d_i) \vdash \neg + \neg(g \wedge d_{i+1})), \quad (1 \leqslant i \leqslant n - 1)$$

in which $T(w_i, d_i) = true$ $(1 \leqslant i \leqslant n)$ and $d_n \supset c_G$.

In summary, the ZK action theory provides a complete framework of the modal logic approaches to reasoning about actions. An effective solution to the frame problem is achieved by using execution axioms or action laws. The same mechanism also works for consistency maintenance. Time is represented by means of two simple operators, + and −, stating *directly following* and *directly preceding*. Hence, a world can be considered with respect to its history and future.

8.4 The possible worlds approach

The possible worlds approach was presented by Ginsberg and Smith (1988a,b). This approach is not actually based on modal logic, but is similar to the modal logic approach in modelling the problems. For a given world w_1, a possible world w_2 with respect to w_1 is constructed in terms of the consequences of an action and the domain constraints. Instead of a modal logic, the proof theoretic notion of consistency is applied to reason about possible worlds. Here we only discuss some of the important concepts in this approach.

Without concerning logical possibility between worlds explicitly, a possible world w_2 with respect to w_1 is constructed by consistently adding new facts into w_1 such that w_2 is the nearest world in which the new facts hold. Let S_1 be a set of formulae that describes a world w_1, and G be a set of sentences that describes the domain constraints, a possible world w_2 with respect to w_1 is defined as follows:

Definition 8.4a For a set C of formulae, a possible world w_2 for C with respect to w_1 is described as any set of formulae $S_2 \subseteq S_1 \cup C$ such that

1 $S_2 \supseteq C$;
2 $S_2 \supseteq G \cap S_1$;
3 S_2 is consistent;
4 If there is another S satisfying (1), (2) and (3), then $S \supseteq S_2$.

The collection of all possible words for C with respect to w_1 is denoted by $W_1(C, S_1)$.

The above definition states: for a world w_1 described by S_1, a possible world w_2 with respect to w_1 can be described as a consistent subset S_2 of S_1 and a set C of new knowledge. Because the addition of new knowledge C results from actions, this definition illustrates how actions are inferred by consistently constructing the results of actions in terms of the current world, domain constraints and consequences of actions. Consequently, the frame problem is reduced to the persistence proof by maintaining the consistency between worlds.

To maintain the consistency, Ginsberg and Smith developed an algorithm to construct possible worlds. The basic idea is: for a world w_1 with descriptions S_1 and a set C of formulae, the construction of S_2 that describes a possible world w_2 with respect to w_1 is implemented by proving the negation $\neg q$ for every sentence $q \in C$. The removal of all these negations from S_1 will guarantee that $S_2 \subseteq S_1 \cup C$ is consistent. Note that $S_2 \supseteq C \cup (G \cap S_1)$.

Suppose that the proof of $\neg q_i$ ($q_i \in C$) is generally described as $\langle P_i: \neg q_i \rangle$ where P_i is a set of premises validating the proof of $\neg q_i$. Consider the following n proofs: $\{\langle P_i: \neg q_i \rangle | 1 \leqslant i \leqslant n\}$.

Apparently, the validation of $\neg q_i$ is subject to $P_i (1 \leqslant i \leqslant n)$. Therefore, the removal of $\neg q_i$ from S_1 can be implemented by destroying any premise $p \in P_i (1 \leqslant i \leqslant n)$. Thus any possible world $w \in W_1(C, S_1)$ with respect to w_1 is as follows:

$$w = S_1 \cup C - \{p_{ik} | 1 \leqslant k \leqslant r\}$$

where $p_{ik} \in P_i (1 \leqslant k \leqslant r)$.

In a similar manner to STRIPS, an action is formalized in terms of the precondition and consequence in the possible world approach. Let $\alpha = \langle P_\alpha, C_\alpha \rangle$ be an action, for a given world w_1 with the description S_1, α will potentially produce a set $W_1(C_\alpha, S_1)$ of possible worlds with respect to w_1, by adding C_α into S_1. Accordingly, the result of α is defined as follows:

Definition 8.4b The result $R(\alpha, w_1)$ of action α in world w_1 with description S_1 can be defined as follows:

$$R(\alpha, w_1) = \begin{cases} S_1, & \text{if } W_1(C_\alpha, S_1) = \phi; \\ w_{21} \cap \ldots \cap w_{2m}(w_{2i} \in W_1(C_\alpha, S_1)), & \text{otherwise.} \end{cases}$$

That is, the result of action α is the intersection of all the possible worlds for C_α.

Fundamentally, the computational mechanism of actions in the possible world approach is similar to that in STRIPS. The result of an action is characterized as a modification to the initiating world. Unlike STRIPS, however, the results of actions do not require an explicit complete description, but are inferentially constructed in terms of the consequence of the action and the domain constraints.

8.5 Summary

The possible world analysis provides well-understood model theoretic semantics, such that it may be nicely applied to formalize the problem domains in reasoning about plans. In fact, the basic motivation to use modal logic for reasoning about actions comes from the observation that the Kripke structure, which establishes the model theoretic semantics for modal logics, suggests a proper formalization for reasoning about plans (Hughes & Cresswell, 1968). This approach has a well-founded theoretical basis and is facilitated by a very rich language from modal logics. However, most of modal logics can not directly use the resolution proof procedures. Instead, their theorem provers are generally based on other non-uniform procedures, such as *tableau calculi*, which complicate problem solving. Most of the work in this area was motivated by the theoretic nature of the possible world semantics and modal logics, rather than the requirements of practical problem solving.

9 Multiagent problems

In the classical problem framework, the world is considered to be in a particular state at any given instant of time, i.e., any world state represents an instantaneous model of the world without time duration. The only way that the passage of time can be observed is through change of state caused by actions. In terms of such an assumption, a world or a continuous process is constructed as a succession of static models. Whereas actions are characterized by the transition from state to state which logically does not take time at all. As a result, every action is instantaneously performed without any interference from external events. As described previously, this framework nicely formalizes reasoning about actions and plans in single-agent problem domains. However, many real-world domains are populated with more than one agent. This means that there exist concurrent activities in these domains which are beyond the control of a single agent.

Major work in domain-independent planning so far has been concerned itself with the classical problems. Nevertheless, the achievements have substantially exposed the nature of problems and the major difficulties. This certainly has facilitated research into multiagent planning. During the past decade, more work has been done to develop such an effective world model that simultaneous events and actions may be formalized to reason about concurrent plans:

A problem of reasoning about plans is referred to as a multiagent problem if the problem domain is potentially involved with simultaneous activities that are concurrently carried out by different agents operating in a dynamic process.

The classical theoretical framework unquestionably builds a solid foundation for approaches to reasoning about plans in multiagent problem domains. However, many concepts that are based on the assumptions of the classical problems are no longer appropriate and need to be modified. For instance, it is not possible to describe every action as a transition from state to state when some actions are concurrently performed and interfere with each other during their execution. Therefore, new formalisms and problem solving techniques are required for representing dynamically changing worlds and reasoning about simultaneous actions. The following discussion mainly concentrates on the development of new formalisms for modelling multiagent problem domains.

9.1 Extended state-based formalism

As an extension of the classical problem framework, various state-based formalisms are also extended to represent reasoning about plans in the multiagent problem domains. The basic idea is that by assuming some degree of interactions amongst actions, certain multiagent problems may be formulated in such a way that they may be solved in the classical problem framework.

Georgeff (1983, 1984) suggested a theory of action which was based on a device called a process model. To show the effects of actions as well as how the actions may be performed, a process model is a formalization of the observable behaviour of an agent in performing an action. Every action is semantically characterized by a process model whose execution defines a set of behaviours or acts with respect to this action. Furthermore, concurrent actions can be represented by an interleaving model that is a parallel composition of process models with respect to these actions. This is substantially different from the previous concept of actions. Georgeff (1987) proposed another theory of actions which allowed for formalizing simultaneous execution of actions. In this theory, an extended situational calculus was used to describe the effects of actions in multiagent domains. By introducing the notion of a correctness condition, the interference between actions can be checked in order to determine whether or not actions can be performed simultaneously.

Pednault (1986) proposed that some simultaneous actions could be sequentially formalized if these actions repectively affect independent parts of the world without any interaction amongst them, i.e., the effect of performing these actions simultaneously is the same as that of performing them sequentially. In this case, some multiagent problems may be solved with techniques originally used for the classical problems. The resulting sequential plans are transformed into parallel plans for coordinating simultaneous executions.

All these approaches aim at solving multiagent problems within the classical framework by assuming some degree of independence amongst events and actions. They are only applicable to some problems and not adequate enough to model general multiagent domains.

9.2 Temporal formalisms

In the classical framework, a world state represents an instantaneous model of the world without time duration. The only way that the passage of time can be observed is through the change of state caused by the actions. Apparently, time is not particularly important in this world model. In contrast, when simultaneous actions are involved in multiagent domains, time becomes very important because: (1) actions take time; (2) more than one action may occur at the same time; (3)

complex plans of actions may involve complex temporal ordering constraints; and (4) actions may interact with external events that are beyond the control of agents and only occur at particular time (Allen et al., 1991). All the aspects are suggesting the significance of time in multiagent domains.

McDermott (1982) proposed a temporal logic for formalizing time, world model, continuous change, actions and plans. This formalism provides some fundamental representations to model simultaneous events or actions in multiagent domains. McDermott still reserved the traditional notion of world state in this formalism, however, time was taken into account to establish temporal relations between states. The following concepts were introduced to model time:

1 *temporal ordering*. For state s_i and s_j, the following temporal relations are defined: (i) $s_i < s_j$ (or $s_j > s_i$): s_i *comes before* s_j; and (ii) $s_i = s_j$: s_i *is identical* s_j. All states are partially ordered by the above relations.
2 *time of occurrence*. Any state s has a time of occurrence *occurs*(s) that is a real number called *date*. For instance, if $s_i < s_j$, then $occur(s_i) < occur(s_j)$.
3 *chronicle*. A chronicle is a complete possible history of the world that can be characterized as a totally ordered set of states extending infinitely in time.

Definition 9.2a Let *state*(x) be a predicate representing that x is a state, then a chronicle is defined as a set *ch* such that:

1 $(\forall s)(s \in ch) \supset state(s)$
2 $(\forall s_i \forall s_j)((s_i \in ch) \wedge (s_j \in ch)) \supset ((s_i < s_j) \vee (s_i > s_j) \vee (s_i = s_j))$
3 $(\forall t \exists s)(s \in ch) \wedge occur(s) = t$

(1), (2) and (3) can be semantically translated as: chronicles consist only of states; any two states are totally ordered by the temporal relations $<$ (or $>$) or $=$; and any date corresponds to a state in a chronicle. Intuitively, a state is a set of facts representing the world at a given moment. Facts change in truth value over time such that states may change over chronicles, resulting from the occurrence of events and actions. Thus, chronicles and states provide a space for events and actions to act. Following Definition 9.2a, another interesting property of chronicles is that every chronicle is convex:

Definition 9.2b Let s_i and s_j be any two states in chronicle *ch* and $s_i < s_j$, then there must be a state s which is located between s_i and s_j in *ch*, i.e.,

$$(\forall s_i \forall s_j)(s_i \in ch \wedge s_j \in ch \wedge s_i < s_j) \supset (\exists s)(s \in ch \wedge s_i < s \wedge s < s_j)$$

The convexity of chronicles means that there are infinite number of states between any two different states in a chronicle. The significance consists in the fact that this makes it possible to represent (1) how long an action takes when it occurs, and (2) many states may occur between the initiating state and the termination state during every action. This representation obviously would facilitate modelling continuous process in a dynamic world.

In terms of the above concepts, the world may be constructed as a chronicle tree which branches into the future over time. Every branch corresponds to a chronicle that not only builds a future from a given state, but also records a complete history before this state.

McDermott defined events and actions in terms of the notion of *interval*. This concept was also proposed by Allen (1983) to represent a time model in reasoning about plans and became the basis of another temporal formalism, *the interval calculus*. We will discuss it later.

Definition 9.2c An interval is a totally ordered and convex set of states, $itv = \{s_i | 1 \leqslant i \leqslant m\}$, as follows:

1 $(\forall s_i \forall s_j)(s_i \in itv \wedge s_j \in itv) \supset (s_i < s_j \vee s_i = s_j \vee s_i > s_j)$
2 $(\forall s_i \forall s_j)(s_i \in itv \wedge s_j \in itv) \wedge (s_i < s_j) \supset (\exists s)(s \in itv \wedge s_i < s \wedge s < s_j)$

An interval $itv = \{s_i | 1 \leqslant i \leqslant m\}$ is also written as $itv = [s_1, s_m]$. Intuitively, *itv* models continuous change from state s_1 to s_m in a chronicle.

Apparently, events or actions occur over intervals. Therefore, events or actions can be defined in terms of intervals over which they take place. Generally, event e happening during interval $itv = [s_1, s_m]$ is represented as $([s_1, s_m], e)$. This interval-based model provides a representation to formalize interactions between events or actions such that simultaneous activities can be represented.

For example, suppose that $itv = [s_1, s_m]$ is an interval in which $s_1 < s_m$, then some simultaneous events may be formalized as follows:

1 events e_1 and e_2 concurrently take place during interval $[s_1, s_m]$: $([s_1, s_m], e_1)$, $([s_1, s_m], e_2)$.
2 event e_2 takes place during $[s_1, s_j]$ that is contained in $[s_1, s_m]$ over which event e_1 takes place: $([s_1, s_m], e_1)$, $([s_i, s_j, e_2)$, in which $s_1 < s_i < s_j < s_m$.
3 event e_2 takes place during $[s_i, s_n]$ that partly overlaps $[s_1, s_m]$ over which event e_1 takes place: $([s_1, s_m], e_1)$, $([s_i, s_n], e_2)$, where $s_1 < s_i < s_m < s_n$.

In this formalism, perhaps the most significant progress is that continuous change over time can be properly described by a temporal world model, facilitating the formalization of simultaneous events or actions. However, since McDermott was not primarily concerned with multiagent planning in his work, the underlying model of events and actions is still only suitable for the classical problem domains.

Another important temporal formalism was *the interval calculus* due to Allen and Pelavin (Allen, 1983, 1984; Pelavin & Allen, 1986, 1987). The basic time model is again referred to as *intervals* in this formalism. Intuitively, intervals are periods of time that are intimately associated with facts, events and actions that may occur simultaneously. The various relationships between intervals can be defined to obtain an appropriate time model.

Definition 9.2d Let x and y be two intervals, then x may be related to y by the following primitive relationships:

1 *before*(x, y). x is before y.
2 *meet*(x, y). x meets y.
3 *overlap*(x, y). x overlaps y.
4 *during* (x, y). x is during y.
5 *start* (x, y). x starts y.
6 *finish*(x, y). x finishes y.

Figure 2 is a pictorial explanation to the relationships above. More complex relationships upon intervals can be constructed in terms of the above primitives. For instance, a *contain* relation can be defined as follows:

$$contain(x, y) \supset during(x, y) \wedge start(x, y) \wedge finish(x, y)$$

In terms of this time model, a temporal predicate logic is established by introducing a time interval argument into every predicate. Let $p(x)$ be a predicate, and t an interval, then $p(x, t)$ states that $p(x)$ holds over the time interval t. Consider an example about the block world. For an interval t_1, the facts which hold until the time interval t_1 is finished can be expressed as follows:

{

$clear(a_1, t_1)$/*a_1 is clear over t_1*/

$clear(a_2, t_2)$/*a_2 is clear over t_2*/

$on(a_3, a_4, t_3)$/*a_3 is on a_4 over t_3*/

$clear(a_3, t_4)$/*a_3 is clear over t_4*/

}

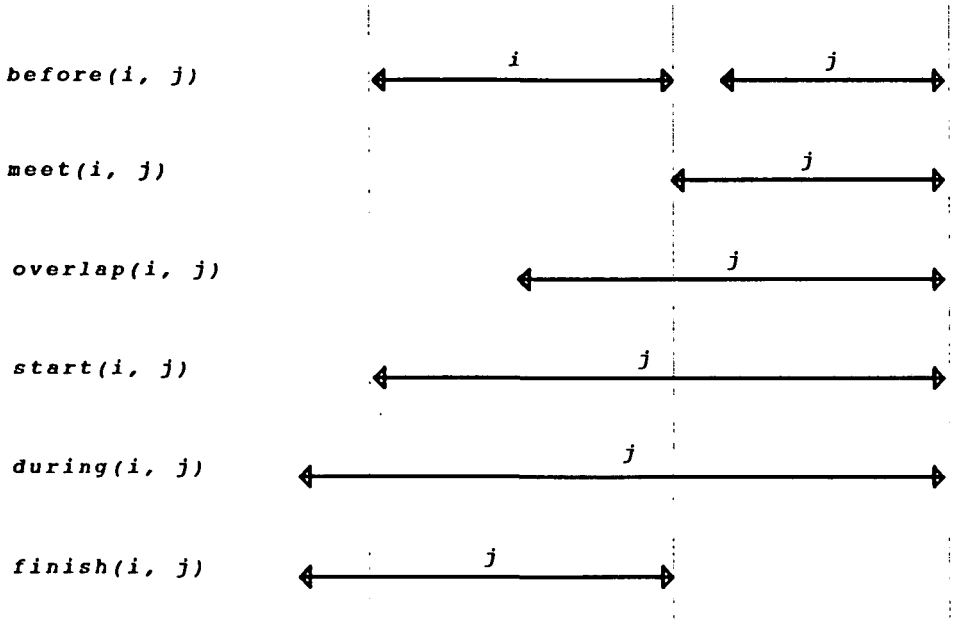

Figure 2

in which we have the following time constraint: *during* $(t_I, t_i) \wedge$ *finish*(t_I, t_i), in which $i = 1, 2, 3, 4$. Suppose that t_I is the time interval for the initial state consisting of all the above facts, and t_G for the goal state, then we have another time constraint *before*(t_I, t_G).

Events and actions can also be temporally represented by this time model. For instance, action *stack*(a_1, a_2, t_{stack}), which describes that a being stacked on b takes place over the time interval t_{stack}, is formalized as follows:

$$\{$$
$$clear(a_1, t)$$
$$clear(a_2, t_p)$$
$$on(a_1, a_2, t_c)$$
$$\}$$

in which the time constraints are

$$\{$$
$$during(t_{stack}, t),$$
$$finish(t_{stack}, t_p),$$
$$meet(t_{stack}, t_c)$$
$$\}$$

The syntax can be translated as: *clear*(a_1) is true during the action; *clear*(a_2) is terminated at the end of the action; and *on*(a_1, a_2) holds after the action. Furthermore, simultaneous events or actions also can be modelled in terms of this time model. Consider for example concurrently performing actions *stack*(a_1, a_2, t_{stack}) and *unstack*$(a_3, a_4, t_{unstack})$. The formalization is as follows:

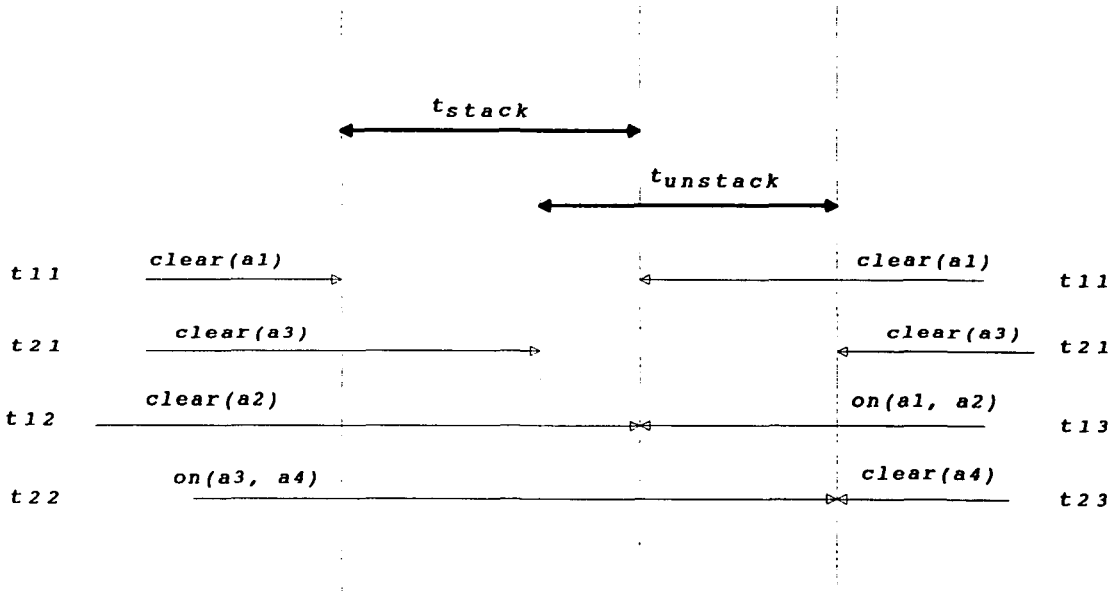

Figure 3

$$
\begin{aligned}
&\{\\
&clear(a_1, t_{11})\\
&clear(a_2, t_{12})\\
&on(a_1, a_2, t_{13})\\
&clear(a_3, t_{21})\\
&on(a_3\ a_4, t_{22})\\
&clear(a_4, t_{23})\\
&\}
\end{aligned}
$$

where the time constraint might be

$$
\begin{aligned}
&\{\\
&overlap(t_{stack}, t_{unstack})\\
&during(t_{stack}, t_{11})\\
&during(t_{unstack}, t_{21})\\
&finish(t_{stack}, t_{12})\\
&finish(t_{unstack,t22})\\
&meet(t_{stack}, t_{13})\\
&meet\ (t_{unstack}, t_{23})\\
&\}
\end{aligned}
$$

Because these two actions are totally independent of each other, t_{stack} and $t_{unstack}$ could be related by *overlap* or *during* relationships. The above time constraints are shown in Figure 3. More complex simultaneous actions can also be formalized in this way. Some examples can be found in Allen and Koomen (1983).

An inference procedure was established to reason about intervals in terms of constraint propagation (Allen, 1981). For example, the following are some rules of inference derivable from the transitivity behaviour of the relations

$$before(x, y) \wedge before(y, z) \vdash before(x, z)$$

$$during(x, y) \wedge meet(y, z) \vdash before(x, z)$$

A working system was also developed in terms of this inference procedure. In summary, the most significant work in this formalism is to reason about temporal durations. The fact that actions take time can be characterized so that the durations of actions may be considered. This knowledge facilitates formalizing simultaneous actions, an important step in the development of multiagent planning.

9.3 Event-based formalism

The event-based formalism was introduced by Lansky (1986). One important feature in this approach is that events are regarded as the primary elements to construct world models, while states are defined derivatively in terms of events, i.e., the state of the world at any instant of time is merely considered as a record of past activities resulting from events that have occurred. Thus, time is naturally taken into account when a world model is constructed: all properties only temporally hold in certain periods of time over which the related events take place. A state at a particular moment is essentially a possible world history that not only records what holds, but also what held in the past.

The event-based world model is structured on two representation levels: (1) *the plan level*, which is based on partially ordered sets of events, each of which is referred to as a world plan; and (2) *the execution level*, which is composed of the sets of possible executions associated with world plans. The two levels of structure essentially represent the different views of the world.

Definition 9.3a Let e_1 and e_2 be any two events, then the following three binary relations can be defined: (1) *the causal relation* $R_{\rightarrow}$. $\langle e_1, e_2\rangle \in R_{\rightarrow}$ means that e_1 causes e_2; (2) *the temporal relation* $R_{\Rightarrow}$. $\langle e_1, e_2\rangle \in R_{\Rightarrow}$ means that e_1 must occur before e_2; and (3) *the simultaneous relation* $R_{\rightleftarrows}$. $\langle e_1, e_2\rangle \in R_{\rightleftarrows}$ means that e_1 and e_2 must occur simultaneously.

If e_1 and e_2 are related by none of the three relations, they may potentially occur in any order or simultaneously. To model a domain which is affected by locality, such as, only sequential actions are allowed within a certain region, all events in a world plan are organized into different relational clusters, which are called *locations*, such that the events in the cluster must obey the same activity constraint. Two types of locations are defined: (1) *elements*: all events in an element location must be totally ordered with the temporal relation $R_{\Rightarrow}$, i.e., only sequential activities are allowed in an element location; (2) *groups*: a group location consists of more than one element. Events can be related to locations by a relation R_{ϵ}. Let e be an event and l a location (element or group), then $\langle e, l\rangle \in R_{\epsilon}$ means that event e occurs in l. Similarly, elements can be also related to groups by R_{ϵ}.

Definition 9.3b An event-based domain model can be formally defined as a 4-tuple: $W = \langle E, EL, G, R\rangle$, in which E is a set of events, EL a set of elements, G a set of groups, and R a set of relations, $R = \{R_{\rightarrow}, R_{\Rightarrow}, R_{\rightleftarrows}, R_{\epsilon}\}$.

A particular domain is formalized by a domain model that allows a set of world plans and the executions conforming to those world plans. A domain model consists of the location descriptions, stating the types of events that may occur at these locations, and the constraints on the possible relationships amongst the events. For instance, consider a world plan as follows:

- *Event set*: $E = \{e_1, e_2, e_3, e_4\}$ such that $\langle e_1, e_2\rangle \in R_{\Rightarrow}$, $\langle e_1, e_3\rangle \in R_{\Rightarrow}$, $\langle e_2, e_4\rangle \in R_{\Rightarrow}$, $\langle e_{3,e4}\rangle \in R_{\Rightarrow}$
- *Element set*: $EL = \{el_1, el_2\}$, in which $\langle e_1, el_1\rangle \in R_{\epsilon}$, $\langle e_2, el_1\rangle \in R_{\epsilon}$, $\langle e_3, el_2\rangle \in R_{\epsilon}$, $\langle e_4, el_2\rangle \in R_{\epsilon}$

- *Group set*: $G = \{g_1\}$, in which $\langle el_1, g_1\rangle \in R_\in$, $\langle el_2, g_1\rangle \in R_\in$

This world plan actually defines the following three possible executions:

execution 1: $[e_1, e_2, e_3, e_4]$

execution 2: $[e_1, e_3, e_2, e_4]$

execution 3: $[e_1, (e_2, e_3), e_4]$

Since e_2 and e_3 are related by none of relations, they can occur either sequentially, such as *execution* 1 and *execution* 2, or simultaneously, as *execution* 3.

A temporal logic is applied to establish the semantics of constraints upon the possible relations amongst events. For a particular problem, an event-based model is established to define an initial world plan that reflects the initial and goal states in the problem. This world plan is elaborated by adding events during a process of incremental constraint satisfaction.

The event-based formalism provides a new approach to reasoning about plans in both single- and multiple-agent domains. A significant aspect of this model is that *event* is considered as the primary element of the world model instead of *state*. The explicit representation of event locations characterizes *a domain locality* that not only models the structural aspect and constraints in a domain, but also imposes temporal and causal constraints upon events. By the use of a temporal logic, complex synchronization properties can be formalized for modelling causality, temporal ordering, and simultaneity. However, it is not obvious how effective this formalism is for solving more complex problems in multiagent domains.

9.4 Summary

Multiagent domains require such a world model that simultaneous events and actions may be formalized in order to reason about concurrent plans. The state-based model may be extended to cope with multiagent problems in some domains, but they are certainly not applicable to the general formalization of multiagent domains. Temporal formalisms seem to provide a promising way to characterize multiagent domains by introducing the various time models. Finally, the event-based formalism, which may be regarded as a temporal formalism, seems to capture the semantics of concurrent activity in a more natural way for multiagent domains. None of them, however, has shown sufficient evidence of being able to solve complex multiagent problems.

Although much recent work has concentrated on multiagent planning, it is still far from forming a well-founded theoretical framework. This is the reason why relatively few planning systems have been fully implemented for solving multiagent problems to date. The difficulties still to be overcome are potentially challenging.

10 Conclusions

As one of the major research and application domains in artificial intelligence, reasoning about actions and plans plays a particular role in implementing rational behaviour of intelligent agents. As discussed previously, much work has been done to develop various theoretical frameworks and problem solving techniques, and a variety of planning systems have been developed. Tate (1985) provided a good review of some well-known planning systems and various plan synthesis techniques.

The three basic problems in reasoning about actions and plans, the *qualification problem*, the *ramification problem* and the *frame problem*, result from the intrinsic difficulty of reasoning about actions with any logic-based formalisms. Solutions to these problems have been built by introducing various formalizations, which make appropriate assumptions about the world. Non-monotonic logics provide a new approach to these problems by formalizing commonsense reasoning.

The classical problem framework effectively formalizes reasoning about plans where only a single agent is concerned, but is not suitable for modelling the domains which are involved with concurrent activity carried out by multiple agents operating in a continuous process or dynamic world. The achievements in single-agent domains have, however, exposed the nature of the problems and the major difficulties. This certainly facilitates research into multiagent planning.

A central view is that reasoning about actions and plans must be rigorously based on formal theories that support an in-depth understanding of the world, world model, events, actions and plans. More importantly, the formal theories should provide strictly-verified proving procedures which can logically ensure that a problem solving system functions correctly. This is essential if intelligent planners are to be eventually produced for effectively coping with realistic problems. The fundamental framework of domain-independent approaches is intensively based on logics. Consequently, a problem space is typically constructed by a set of axioms and a set of rules of inference. By means of theorem proving techniques, such as, the *tableau procedure* or the *resolution theorem procedure*, plan synthesis is actually characterized as searching for a constructive proof that there exists a state in which a given goal is logically true. This framework nicely supports a compact and elegant formalization of problem domains.

Logic-based formalisms and theorem proving techniques, however, are effective when (1) problem domains rigorously conform to various logical constraints, and (2) the problem domains can be logically formalized, i.e., various domain properties can be axiomatized. For instance, the block world is such a problem domain. Unfortunately, many real-world problems, such as some complex engineering applications, do not show these characteristics. These domains resist embodiment in axioms, or axiomatization may require enormous effort (Winston, 1981). Consequently, most of the logic-based approaches are still wandering over stacking blocks, instead of being able to provide a sufficient utility for the development of intelligent planners.

Because of the reasons described above, the utilisation of logic-based reasoning about actions and plans is still in its infancy in engineering applications. For instance, automated process planning in the manufacturing area has received considerable attention, but has as yet attained little progress in developing intelligent planners based on logical frameworks. Process planning is perceived to be a problem associated with substantial amounts of knowledge and processing. Intensively logic-based formalisms seem to be inadequate to represent this complex knowledge and to cope effectively with the domain-specific reasoning. Therefore, many researchers have tended to rely on knowledge-based approaches to achieve process planning. Some attempts have been made to combine the classical approaches with knowledge-based representations and processing mechanisms in engineering applications (see Miller et al., 1988; Kambhampati et al., 1991; Lee et al., 1992).

Many researchers have analysed the weaknesses of the classical approaches to reasoning about actions and plans such as Wilkins (1988, 1989), Levitt (1990), etc. The logic-based formalisms and procedures need to be enhanced by other representations and problem solving techniques in the development of intelligent planners for realistic problems. Over the past decade, various knowledge-based representations and processing mechanisms have been developed. They may suggest an effective way to improve the logic-based frameworks in dealing with large volumes of complex knowledge. The potential for using provably-correct intelligent planners in real-world problems should be realized within the next decade.

References

Allen, JF, 1981, "An interval-based representation of temporal knowledge" In: *Proceedings IJCAI.*
Allen, JF, 1982, "Maintaining knowledge about temporal intervals" *Communications of the ACM* **26** 832–843.
Allen, JF and Koomen, JA, 1983, "Planning using a temporal world model" In: *Proceedings IJCAI.*
Allen, JF, 1984, "Towards a general theory of action and time" *Artificial Intelligence* **23** 123–154.
Bradly, R. and Swartz, N. 1979, *Possible worlds: An introduction to logic and its philosophy,* Basil Blackwell.
Brown, FM, 1986a, "A commonsense theory of non-monotonicity" In: *Proceedings 4th Army Conference on Applied Mathematics and Computer Science.*

Brown, FM, 1986b, "A commonsense theory of non-monotonic reasoning" In: *Proceedings 8th International Conference on Computer Science.*

Brown, FM, 1987a, "The modal logic Z" Technical report (TR-87-1).

Brown, FM, 1987b, "A modal logic for the representation of knowledge" In: *Proceedings Workshop on the Frame Problem in Artificial Intelligence.*

Castaneda, HN, 1965, "The logic of change, actions and norms" *Journal Philosophy* **62** 333–334.

Davidson, D, 1967, "The logical form of action sentences" In: Rescher, N, ed., *The Logic of Decision and Action*, Pittsburgh University Press.

Doyle, J, 1979, "A truth maintenance system" *Artificial Intelligence* **12**(3) 231–272.

Fikes, RE and Nilsson, 1971, "STRIPS: A new approach the application of theorem proving to problem solving" *Artificial Intelligence* **2** 189–208.

Finger, JJ, 1986, "Exploiting Constraints in design synthesis" PhD thesis, Stanford University.

Georbel, RG and Goodwin, SD, 1987, "Applying theory formation to the planning problem" In: *Proceedings of Workshop on the Frame Problem in Artificial Intelligence* 207–232.

Georgeff, MP, 1983, "Communication and interaction in multiagent planning" In: *Proceedings AAAI.*

Georgeff, MP, 1983, "A theory of action Communication and interaction in multiagent planning" *Proceedings AAAI.*

Georgeff, MP, 1986, "Actions, Processes and causality" In: *Proceedings Workshop on Reasoning about Actions and Plans* 99–122.

Georgeff, MP, 1987a, "Many agents are better than one" In: *Proceedings Workshop on the Frame Problem in Artificial Intelligence* 59–75.

Georgeff, MP, 1987b, "Planning" *Annual Review of Computer Science* **2** 359–400.

Ginsberg, ML and Smith, DE, 1988a, "Reasoning about action I: A possible worlds approach" *Artificial Intelligence* **35** 165–195.

Ginsberg, ML and Smith, DE, 1988b, "Reasoning about action II: The qualification problem" *Artificial Intelligence* **35** 311–342.

Godel, K, 1933, "Eine interpretation des intuitionistischen Aussagenkalkuls" In: *Ergebnisse eines Mathematicschen Kolloquiums 4* 34–50 [translated by Hintikka 1969].

Green, C, 1969a, "Application of theorem proving to problem solving" In: *Proceedings IJCAI* 219–239.

Green, C, 1969b, "Theorem proving by resolution as a basis for question answering systems" *Machine Intelligence* **4** 183–205.

Hanks, S and McDermott, D, 1985, "Temporal reasoning and default logics" Technical report, YALEU/CSD/RR, 430, Yale University.

Hanks, S and McDermott, D, 1986, "Default reasoning, non-monotonic logics and the frame problem" In: *Proceedings AAAI.*

Hughes, GE and Cresswell, MJ, 1968, *An Introduction to Modal Logics*, Methuen.

Kambhampati, S, Cutkosky, M, Tenenbaum, M and Lee, SH, 1991, "Combining specialized reasoners and general purpose planners" In: *Proceedings AAAI.*

Kautz, H, 1982, "A first-order dynamic logic for planning" Technical report (TR-CSRG-144), Department of Computer Science, University of Toronto.

Kowalski, R, 1979, *Logic for Problem Solving*, North Holland.

Kripke, SA, 1963, "Semantical consideration on modal logic" *Acta Philosophica Fennica* **16** 83–94.

Kripke, SA, 1971, "Semantical considerations on modal logics" In: Linsky, L, ed., *Reference and Modality*, Oxford University Press.

Lansky, AL, 1985, "Behavioral specifications and planning for multiagent domains" Technical report, 360, SRI international.

Lansky, AL, 1986, "A representation of parallel activity based on events, structure and causality" *Proceedings Workshop on Reasoning about Actions and Plans* 123–159.

Lansky, AL, 1987, "Localized Event-based reasoning for multiagent domains" Technical report, 423, SRI International.

Lee, H, Williams, JHS and Tannock, JDT, 1992, "Knowledge-based inspection planning" *Journal of Artificial Intelligence in Engineering Design, Analysis and Manufacturing* **6**(3) 149–162.

Levitt, RE, 1990, "Knowledge-based planning systems: An engineering perspective" In: *Proceedings IEE on Expert Planning System.*

Lifschitz, V, 1986, "On the semantics of STRIPS" In: *Proceedings Workshop on Reasoning about Actions and Plans.*

Lifschitz, V, 1987a, "Formal theories of actions" In: *Proceedings Workshop on the Frame Problem in Artificial Intelligence* 35–57.

Lifschitz, V, 1987b, "Computing circumscriptions" In: *Proceedings IJCAI* 121–127.

McCarthy, J, 1963, "Situations, actions and causal laws" Stanford Artificial Intelligence Project: memo-2.

McCarthy, J and Hays, PJ, 1969, "Some philosophical problems from the standpoints of artificial Intelligence" *Machine Intelligence* **4** 463–502.

McCarthy, J, 1980, "Circumscription: A form of non-monotonic reasoning" *Artificial Intelligence* **13** 27–39.

McCarthy, J, 1986, "Applications of circumscription to formalizing commonsense knowledge" *Artificial Intelligence* **28** 89–118.

McDermott, D and Doyle, J, 1980a, "Non-monotonic logic I" *Artificial Intelligence* **13**.

McDermott, D and Doyle, J, 1980b, "Non-monotonic logic II" *Artificial Intelligence* **13**.

McDermott, D, 1982, "A temporal logic for reasoning about process and plans" *Cognitive Sciences* **6**(2) 101–155.

McDermott, D, 1987, "AI, logic and the frame problem" In: *Proceedings Workshop on the Frame Problem in Artificial Intelligence* 105–118.

Miller, SA and Schubert, LK, 1988, "Using specialists to accelerate general reasoning" In: *Proceedings AAAI*.

Moore, RC, 1980a, "Reasoning about knowledge and actions" Artificial Intelligence Centre Technical Report, 191, SRI International.

Moore, RC, 1980b, "A formal theory of knowledge and actions" In: Allen, J, Hendler, J and Tate, A, eds, *Readings in Planning* 480–519.

Morris, P, 1987, "A truth maintenance based approach to the frame problem" In: *Proceedings Workshop on the Frame Problem in Artificial Intelligence* 297–307.

Newell, A and Simon, HA, 1963, "GPS: A program that simulates human thought" In: Feigenbaum, EA and Feldman, J, eds., *Computers and Thought* 279–293.

Nilsson, NJ, 1982, *Principles of Artificial Intelligence*, Springer-Verlag.

Nilsson, NJ, 1990, "Forward" In: Allen, J, Hendler, J and Tate, A, eds., *Readings in Planning*.

Park, SS, 1987, "Doubting Thomas: action and belief revision" In: *Proceedings Workshop on the Frame Problem in Artificial Intelligence*.

Pednault, EPD, 1986, "Formulating multiagent, dynamic world problems in the classical planning framework" In: *Proceedings Workshop on Reasoning about Actions and Plans* 47–82.

Pool, DL, Goebel, RD and Aleliunas, R, 1968, *Theorist: A logical reasoning system for defaults and diagnosis*, Springer-Verlag.

Reiter, R, 1978, "On closed world data base" In: Gaillaire, H and Minker, J, eds., *Logic and data bases*, Plenum Press.

Reiter, R, 1980, "A logic for default reasoning" *Artificial Intelligence* **13** 81–130.

Rescher, N, 1967, "Aspects of action" In: Rescher, N. ed., *The Logic of Decision and Action*, Pittsburgh University Press.

Robinson, J, 1965, "A machine-oriented logic based on the resolution principle" *Journal of ACM* **12**(1) 23–41.

Sacerdoti, ED, 1973, "Planning in a hierarchy of abstraction spaces" In: *Proceedings IJCAI* 412–422.

Sacerdoti, ED, 1975, "The nonlinear nature of plans" In: *Proceedings IJCAI* 412–422.

Schwind, CB, 1978a, "Representing actions by state logic" In: *Proceedings AISB/GI Conference on Artificial Intelligence*.

Schwind, CB, 1978b, "The theory of actions" Research Report TUM-INFO 7807, Technische Universitat, Munchen.

Schwind, CB, 1983, "A completeness proof for a logic of action" Research report, Laboratoire D'Informatique pour les science de l'homme, Marseille.

Schwind, CB, 1986, "Action logic" Research report GRTC/137.

Schwind, CB, 1987, "Action theory and the frame problem" In: *Proceedings Workshop on the Frame Problem in Artificial Intelligence*.

Shoham, Y, 1986, "Chronological ignorance: Time, non-monotonicity, necessity and causal theories" In: *Proceedings AAAE* 389–393.

Simon, HA, 1965, "The logic of rational decision" *Journal of Philosophy of Science* **16** 169–186.

Simon, HA, 1966, "On reasoning about actions" Research report, complex information processing, 16, Carnegie Institute of Technology.

Stefik, M, 1981a, "Planning and meta-planning (MOLGEN: Part 1)" *Artificial Intelligence* **16** 111–140.

Stefik, M, 1981b, "Planning and meta-planning (MOLGEN: Part 2)" *Artificial Intelligence* **16** 141–170.

Tate, A, 1976, "Project planning using a hierarchic nonlinear planner" Research report, Department of Artificial Intelligence Research, University of Edinburgh, UK.

Tate, A, 1985, "A review of AI planning techniques" *Knowledge Engineering Review* **1** 4–17.

Waldinger, A, 1977, "Achieving several goals simultaneously" *Machine Intelligence* **8** 94–136.

Winston, PH, 1984, *Artificial Intelligence*, Addison-Wesley.

Wilkins, DE and Robinson, AE, 1981, "An interactive planning system" SRI Technical Note, 245.

Wilkins, DE, 1983a, "Representation in a domain-independent planner" In: *Proceedings IJCAI* 733–740.
Wilkins, DE, 1983b, *Practical Planning*, Morgan Kaufmann.
Wilkins, DE, 1984, "Domain-independent planning: Representation and plan generation" *Artificial Intelligence* **22**(3) 269–301.
Wright, CH, 1963, *Norm and action: A logic enquiry,* Routledge.
Wright, CH, 1967, "The logic of action: A sketch" In: Rescher N, ed., *The Logic of Decision and Action*, Pittsburgh University Press.