A NONMONOTONIC THEORY OF PLAN SYNTHESIS

Salvinder S. Hundal and Frank M. Brown

Department of Computer Science
The University of Kansas
Lawrence, Kansas, 66045

ABSTRACT

In this paper, a theory of plan synthesis is proposed that reasons about actions to solve the frame problem. Also, it reasons about the plan synthesis to detect possible or impossible orderings of the actions. This theory uses the frame axiom and the modal quantificational logic Z to propagate the facts from the current situation to the next situation. The explicit results of an action are provided only, no delete list is needed. The facts are automatically added and deleted from one situation to the next by the nonmonotonic reasoning as the actions are performed. A plan synthesis algorithm is also developed. This algorithm builds a plan incrementally by inserting new actions into the plan or by modifying the plan using nonmonotonic reasoning system of the modal quantificational logic Z. In this algorithm the reasoning about plan synthesis, helps to detect impossible orderings of actions that reduces the search space. This theory of plan synthesis reasons about actions to solve the frame problem and also, it reasons about the plan synthesis to detect possible or impossible orderings of the actions.

2. THE FRAME PROBLEM IN PLANNING

In this section we describes how the modal quantificational logic Z can be used to propagate the facts from the current situation to the next situation after an action is executed. To solve the frame problem default reasoning is expressed by reflective expression in the quantificational modal logic [6]. For example, The default "If p holds and q can be assumed in situation K then q holds" can be expressed in Z as follows:

The symbolism \( <K>q \) means that K is possible with q.

\( K = ((p \land <K>q) \rightarrow q) \).

To solve this we do case analysis, first split:

\( <K>q \land (K = (p \rightarrow q)) \lor (\neg<K>q \land (K = T)) \),

substitute: \( (p \rightarrow s) q \land (K = (p \rightarrow q)) \lor (\neg<T>q \land (K = T)) \),

by axiom A5 of Z: \( (K = (p \rightarrow q)) \), which is the solution of reflective equation (1).

If \( K_0 \) is the initial situation and K is the resulting situation, then the symbolism \( [K_0]p \) means that \( K_0 \) entails p and \( <K>p \) means that K is possible with p. Thus using reflective reasoning a simple frame default law may be written as:

\( \forall p (((K_0)p) \land (<K>p)) \rightarrow p) \). (2)

Since the initial situation \( K_0 \) is known, it remains to define the resulting situation K. K is essentially the conjunction of the obvious results of an action that were
applied, the physical laws which are true of all situations, and the frame default law. Thus in symbols the new situation $K$ is:

$$K = df \left( (K_0)(\text{pre-conds action}) \rightarrow (\text{results action}) \wedge \text{physical-laws} \wedge \forall p ((K_0 \land \langle K > p \rangle \rightarrow p) \right) (3')$$

The deduction techniques of the quantificational modal logic $Z$ described in this section have been implemented in the Schemata Deduction language [1,2]. The reflective reasoning in $Z$ have been used to propagate properties automatically from an initial situation $K_i$ into a succeeding situation $K_{i+1}$ as actions are performed by simplifying the definiens in: $K_{i+1} = df (\exists K_0 (\forall K((\text{results actions}) \wedge \text{physical-laws} \wedge \text{frame-default-law} (K_i, K))))$ when the results of the actions, the physical laws, and the particular frame default laws are defined [7].

3. Plan Synthesis

There are two assumptions on which this theory of plan synthesis is based. The first is that the state of the world does not change except as the results of an action, which means that if a condition is false at one situation and become true at a later situation, then there must be an action that causes the condition to become true in both the two situations. The second assumption is that a plan for achieving a set of goals is finite. This assumption is justified by the fact that we can only perform a finite number of actions in a finite amount of time. These two assumptions comprise the principle of plan synthesis: if some fact is not true in a situation, to make that fact true in a later situation, there must be an action or sequence of actions that make the fact true.

Using the plan synthesis principle, we build a plan incrementally, adding actions to the partial plan as needed to satisfy the goals and the preconditions of actions previously defined. Our theory select the actions purely on the basis of the changes required to be made in the world to achieve the goals. The actions that are not pertinent to construct a valid plan are not even considered. Further, we insert the actions in the plan in such a way so that a particular condition must remain true up to a certain point. This protection of facts helps us to detect impossible orderings of actions, which reduces the search space.

3.2. Plan Modification

In order to synthesize a plan, we begin with an empty plan and keep adding more actions until a valid plan is constructed. In each step of the plan synthesis, we have some current plan that is analyzed to identify those goals and preconditions not yet satisfied and to determine what additional actions are needed to bring them about. The appropriate action is then inserted, producing a modified current plan. The cycle of analysis and modification continues until all goals and their preconditions have been satisfied. In a situation where there are multiple ways of achieving a goal or a precondition, one of the alternatives is selected and the other alternatives are saved, so that if the selected one does not succeed, the other alternatives can be explored.

To solve a goal, the plan modification is done in two ways. First, if the goal is true in the current situation then we try to prevent it from becoming false. Second, if the goal is false in the current situation, then we insert an action that causes the goal to become true, and protect the goal from becoming false until a desired situation is reached. But this is not sufficient to build a plan for every problem. For example, there might be a goal that cannot be achieved as such by an action. However, the action may cause that goal to become true if additional enabling constraints are posted on the action. Since the plan is build incrementally, the action that causes a goal to become true in the final plan may already appear in the current plan. Therefore, another way of achieving a goal would be to establish the appropriate enabling condition that allows an existing action in the plan to cause the goal to become true. The following example illustrates this point.

Suppose that we have a world consisting of a cart, a television and a refrigerator, each of which may be placed in one of the two locations in the home, the living room or the kitchen. Actions are available for loading or unloading the television and the refrigerator from the cart, as well as for moving the cart between the living room and the kitchen. Initially, the television, the refrigerator and the cart are in the living room, and only the television is loaded on the cart as shown in figure 1. The goal is to have the cart and the refrigerator in the kitchen and the television in the living room.

![Figure 1. Initial State of the Cart-TV-Refrigerator.](image)

We begin the plan synthesis with an empty plan. Let us first consider the goal of having the cart in the kitchen. Since this goal is not true initially, we must find an action in our final plan that causes the goal to become true. As the current plan is empty, the only option is to
insert the action that causes the cart to be moved to the kitchen. The new situation is shown in figure 2.

Figure 2. State after Moving the Cart into the Kitchen.

Next, we consider the goal of having the refrigerator in the kitchen, which is not true in the current situation. However, if we were to load the refrigerator onto the cart before moving the cart into the kitchen, as shown in figure 3, the refrigerator would be moved to the kitchen as a side effect.

Figure 3. Loading the Refrigerator on the Cart.

In this case, the action that causes the cart to move to the kitchen already appears in the plan and an additional action is inserted into the current plan to impose the additional condition (i.e., the refrigerator must be on the cart before moving the cart), on the action of moving the cart into the kitchen. The additional constraints posted on an action are called the frame preconditions. At this point, the current plan has two actions: to load the refrigerator onto the cart and to move the cart into the kitchen. The resulting situation is shown in figure 4.

Figure 4. Cart, TV and Refrigerator in the Kitchen.

Finally, we are left with only one more goal to solve, which is to have the television in the living room. Since, the television was initially in the living room, the current plan caused it to be brought into the kitchen as a side effect. However, if we were to unload the television from the cart before moving the cart to the kitchen as shown in figure 5, we would have prevented the television from changing its location. The goal would then be achieved by virtue of the fact that it would never become false.

In the final plan, we load the refrigerator onto the cart, unload the television from the cart and move the cart to the kitchen. The final situation is shown in figure 6.

Figure 5. Unloading the TV from the Cart.

Figure 6. Final Desired State.

This example illustrates that a goal can be achieved in one of the three ways: by inserting a new action that causes the goal to become true, by forcing an existing action to cause the goal to become true, or by preventing the goal from becoming false. The latter two ways need the frame preconditions of an action, which are discussed in next section.

3.3. Frame Preconditions

When a goal is achieved by inserting a new action that causes the goal to become true, there are certain predefined facts that must be true before the action can be executed. These predefined facts are called preconditions of the action. However, when a goal is achieved by forcing an existing action to cause the goal to become true or by preventing the goal from becoming false, some additional conditions are imposed on the action. These additional conditions are added to the preconditions of the action. Since these additional conditions force an action to carry or not carry certain facts into the next situation, we call them frame preconditions. The Z solution to the frame problem is used to find the frame preconditions. The additional conditions that prevent the action from making a goal false (i.e., to carry a fact from current situation to the next situation) are called holding preconditions and the additional conditions intended to force an action to cause a goal to become true are called inducing preconditions.

3.3.1. Holding Preconditions

In the previous example, the physical laws state that what ever is on the cart is in the same room as the cart. If the television is on the cart, the cart is in the living room. If we move the cart into the kitchen,
television will be in the kitchen because of the physical laws. Suppose, we want the fact that the television is in the living room to hold and it is possible that the cart is in the kitchen, while the television is in the living room. The television being on the cart interacts with the physical laws (i.e., what ever is on the cart is in the same room as the cart) and the results of the action of moving the cart to the kitchen, which makes the television be in the kitchen. Thus, the fact that the television is in the living room does not hold after moving the cart; had the television not been on the cart, it would not have changed its location. Therefore a holding precondition for this fact is that the television must not be on the cart.

Let $\Theta$ be the physical laws, $\alpha$ be the results of an action, and $\phi$ be the facts that must hold after the action is applied to the current situation. If the preconditions of an action were true in the initial situation $K_i$, the resulting situation $K_{i+1}$ is the conjunction of the results of the action, the physical laws that are true of all situations, and the frame default law that propagates all the facts from the initial situation $K_i$ to the new situation $K_{i+1}$ that do not cause any contradiction in the new situation. Thus in symbols the new situation $K_{i+1}$ is:

$$K_{i+1} = \Theta \land \alpha \land (\forall p \ (K_{i} \land \phi \land <K_{i+1} \Gamma>) \implies p) \quad (4)$$

If a fact $\phi$ holds in the current situation and $\phi$ is possible with respect to the physical laws and the results of an action (i.e., $<\Theta \land \alpha \land \phi$ is true), it is possible for $\phi$ to hold in the new situation after the action is performed. But if $\phi$ does not hold in the new situation, although it held in the previous situation, and $\phi$ is possible with respect to the physical laws and the results of an action (i.e., $<\Theta \land \alpha \land \phi$ is true), then there must be some fact $\gamma$ (the "culprit") in the previous situation, which propagated into the new situation and interacted with the physical laws and the results of the action, that make it impossible for $\phi$ to hold in the new situation (i.e., $\neg<\Theta \land \alpha \land \gamma \land \phi$ is true). The negation of such culprits ($\gamma$) are the holding preconditions for $\phi$ with respect to $\alpha$.

To find the holding preconditions for the fact $\phi$, let us consider the state space diagram in figure 7. The oval $K_i$ represents a situation before performing an action and the oval $K_{i+1}$ represents a situation after performing the action.

Since the physical laws $\Theta$ are true in both situations, it is shown in the intersection of the two situations. All the facts that are propagated from situation $K_i$ to situation $K_{i+1}$ are common to both situations, therefore, they must be present in the intersection part of the two ovals. The results $\alpha$ of an action, which are entailed in $K_{i+1}$, are shown in oval $K_{i+1}$. The fact $\phi$, which is true in $K_i$, is shown in oval $K_i$. The "culprit" $\gamma$, which is propagated from the previous situation to the resulting situation, interacts with the physical laws in the presence of the results $\alpha$ of an action, and causes a contradiction with $\phi$. Thus, the fact $\phi$ is not propagated into the next situation $K_{i+1}$. As shown in figure 7, there are three conditions that must be satisfied by $\gamma$ to be a "culprit".

(a) $\Theta$, $\phi$, and $\gamma$ together are consistent ($<\Theta \land \phi \land \gamma$ must hold if it is possible for $\gamma$ to be in $K_i$ along with $\phi$ and $\Theta$)

(b) $\Theta$, $\alpha$, and $\gamma$ together are consistent ($<\Theta \land \alpha \land \gamma$ must hold if it is possible for the culprit to propagate into the next situation along with the results of the action).

(c) $\Theta$, $\alpha$, $\phi$, and $\gamma$ together are not consistent ($\neg<\Theta \land \alpha \land \gamma \land \phi$) which says that $\gamma$ is the culprit).

If $\gamma$ satisfies all the three conditions, $\gamma$ is a real culprit, and the conjunction of the negation of all such propositions $\gamma$ will be the holding preconditions for the fact $\phi$. The holding preconditions can be expressed in the modal quantificational logic $Z$ as follows:

$$H(\Theta, \alpha, \phi) = \forall \gamma \ (\neg<\Theta \land \alpha \land \gamma \land \phi \land \gamma) \land \forall \gamma \ (\Theta \land \alpha \land \gamma \land \phi \land \gamma) \implies \neg \gamma$$

which is:

$$H(\Theta, \alpha, \phi) = \forall \gamma \ (\neg<\Theta \land \alpha \land \gamma \land \phi \land \gamma) \land \forall \gamma \ (\Theta \land \alpha \land \gamma \land \phi \land \gamma) \implies \neg \gamma$$

If we replace $\neg \gamma$ with $\psi$, the above expression becomes:

$$H(\Theta, \alpha, \phi) = \forall \psi \ (\neg<\Theta \land \alpha \land \gamma \land \phi \land \gamma) \land \forall \gamma \ (\Theta \land \alpha \land \gamma \land \phi \land \gamma) \implies \psi$$

and it further becomes:

$$H(\Theta, \alpha, \phi) = \forall \psi \ (\neg<\Theta \land \alpha \land \gamma \land \phi \land \gamma) \land \forall \gamma \ (\Theta \land \alpha \land \gamma \land \phi \land \gamma) \implies \psi \quad (5)$$

This expression says that the holding preconditions of $\phi$ with respect to the physical laws and the results of an
action are the facts entailed by the conjunction of physical laws, the results of the action and $\phi$, but not entailed by the conjunction of physical laws and the results of the action, and not entailed by the conjunction of physical laws and $\phi$.

We can find the holding preconditions of the cart and TV example formally by using expression (5), if the physical laws are defined as follows:

$$\text{Phy-laws} = \text{df } (\forall x, y, z ((\text{IN} x y) \land (\text{IN} x z)) \rightarrow (\exists [I] y z))$$

$$\land ((\forall u, y ((\text{OnCart} u) \land (\text{IN Cart} y)) \rightarrow (\text{IN} u y)).$$

Where the domain of $u$ is TV and Fr (refrigerator), the domain of $x$ is cart, TV and Fr, and the domains of both $y$ and $z$ are LivingRoom and Kitchen.

Suppose, $\phi$ is (IN TV LivingRoom) and the result ($\alpha$) of the action (MoveC Kitchen) is (IN Cart Kitchen). After plugging these values into expression (5), we get the holding preconditions of the fact (IN TV LivingRoom) with respect to the results of the action (MoveC Kitchen) as follows:

$$\forall \psi ((\text{Phy-laws} \land \text{IN Cart Kitchen}) \land \text{IN TV LivingRoom}) =$$

$$(\text{IN Cart Kitchen}) \land \text{IN TV LivingRoom} \land \neg(\text{OnCart TV})$$

$$\land \neg(\text{IN Cart LivingRoom}) \land \neg(\text{IN TV Kitchen})$$

$$\land (\neg(\text{OnCart Fr}) \lor (\text{IN Fr Kitchen}))$$

$$\land (\neg(\text{IN Fr LivingRoom}) \lor (\text{IN Fr Kitchen})),$$

$$(\text{Phy-laws} \land (\text{IN Cart Kitchen})) =$$

$$(\text{IN Cart Kitchen}) \land \neg(\text{IN Cart LivingRoom})$$

$$\land (\neg(\text{OnCart Fr}) \lor (\text{IN Fr Kitchen}))$$

$$\land (\neg(\text{IN Fr LivingRoom}) \lor (\text{IN Fr Kitchen}))$$

$$\land (\neg(\text{OnCart TV}) \lor (\text{IN TV Kitchen})),$$

and

$$(\text{Phy-laws} \land (\text{IN TV LivingRoom}) =$$

$$(\text{IN TV LivingRoom}) \land \neg(\text{IN TV Kitchen})$$

$$\land (\neg(\text{OnCart TV}) \lor (\text{IN Cart Kitchen}))$$

$$\land (\neg(\text{OnCart Fr}) \lor (\text{IN Cart LivingRoom}))$$

$$\land (\neg(\text{OnCart Fr}) \lor (\text{IN Fr Kitchen}))$$

$$\land (\neg(\text{OnCart Fr}) \lor (\text{IN Cart LivingRoom})).$$

Suppose, $\phi$ is the physical laws, $\alpha$ be the results of an action, and $\phi$ be the fact that is to be forced as a side effect when the action is applied to the current situation. Suppose, $\phi$ does not hold in the current situation and it is possible for $\phi$ to hold in the new situation after the action is performed. If $\phi$ does not hold in the new situation, the results of the action have not caused the side effect to force $\phi$ to hold in the new situation. If there were a fact $\gamma$ (the "inducer") in the current situation that propagated into the new situation, and interacted with the physical laws and the results of the action to force $\phi$ to hold in the new situation then $\gamma$ should be the inducing precondition for $\phi$ with respect to $\alpha$.

To find the inducing preconditions for the fact $\phi$, let us consider the state space diagram in figure 8. The oval $K_i$ represents a situation before performing an action and the oval $K_{i+1}$ represents a situation after performing the action.

![Figure 8. State Space Diagram for Inducing Conditions.](image)

Since the physical laws $\Theta$ are true in both situations, it is shown in the intersection of the two situations. All the facts that are propagated from situation $K_i$ to situation $K_{i+1}$ are common to both the situations, therefore, they must be present in the intersection part of the two ovals. The results $\alpha$ of an action that are entailed in $K_{i+1}$, and the fact $\phi$ that is not true in $K_i$ but is induced in $K_{i+1}$, are shown in oval $K_{i+1}$. The "inducer" $\gamma$, which is propagated from the previous situation to the resulting situation, intersects...
with the physical laws in the presence of the results α of an action and induces φ, as a side effect, in the next situation Ki+1.

As shown in figure 8, there are three conditions, which must be satisfied by γ to be an "inducer".

(a) Θ, −φ, and γ together are consistent in Ki

(b) Θ, α, and γ together are consistent (<Θ ∧ α>γ must hold if φ is not holding in Ki)

(c) Θ, α, −φ, and γ together are not consistent in Ki+1 (¬<Θ ∧ α>φγ) which says that γ is the inducer.

If γ satisfies these three conditions, it will be an inducing precondition for the fact φ with respect to the result α of an action. The inducing preconditions can be expressed in the modal quantificational logic Z as follows:

\[ I(\Theta, \alpha, \phi) = df \exists y (¬<\Theta ∧ \alpha>φγ ∧ ¬<\Theta ∧ \alpha>φ ∧ γ) \]

If we replace γ with −ψ, the above expression becomes:

\[ I(\Theta, \alpha, \phi) = df \exists y (¬<\Theta ∧ \alpha>φ∧ψ ∧ ¬<\Theta ∧ \alpha>φ∧¬ψ) \]

which can be simplified to

\[ I(\Theta, \alpha, \phi) = df \exists y (¬<\Theta ∧ \alpha>φψ ∧ ¬<\Theta ∧ \alpha>φ¬ψ) \]

and it further becomes

\[ I(\Theta, \alpha, \phi) = df \exists y (¬<\Theta ∧ \alpha>φψ ∧ ¬<\Theta ∧ \alpha>φ¬ψ) \]

This expression says that the inducing preconditions of φ with respect to the physical laws and the results of an action, are the negation of each proposition entailed by the conjunction of physical laws, the results of the action and −φ, but not entailed by the conjunction of physical laws and the results of the action, and not entailed by the conjunction of physical laws and −φ.

Theorem: I(Θ, α, φ) = ¬H(Θ, α, −φ).

Proof: I(Θ, α, φ) = df \exists y (¬<Θ ∧ α>φψ ∧ ¬<Θ ∧ α>φ¬ψ)

by pushing the negation through the quantifier we get: ¬<vψ ¬(¬<Θ ∧ α>φψ ∧ ¬<Θ ∧ α>φ¬ψ)>

which can be rewritten as:

¬<vψ ¬(¬<Θ ∧ α>φψ ∧ ¬<Θ ∧ α>φ¬ψ) v ψ)>

which can be rewritten as:

¬<vψ (¬<Θ ∧ α>φψ ∧ ¬<Θ ∧ α>φ¬ψ) v ψ)>

using expression (5) which is the definition of the holding conditions, we get ¬H(Θ, α, −φ).

Hence the proof.

We can find the inducing preconditions of the cart and TV example formally by using expression (6), if the physical laws are defined as follows:

\[ \text{Phy-laws } = df (\forall x,y, z ((\text{IN } x \ y) ∧ (\text{IN } x \ z)) \rightarrow (\lnot y \ z)) \]

Where the domain of u is TV and Fr (refrigerator), the domain of x is cart, TV and Fr, and the domains of both y and z are LivingRoom and Kitchen.

Now suppose, φ is the fact that Fr (refrigerator) is in the kitchen (IN Fr Kitchen) and the result α of the action (MoveC Kitchen) is (IN Cart Kitchen). After plugging these values into expression (6), we get the inducing preconditions of the fact (IN Fr Kitchen) with respect to the results of the action (MoveC Kitchen) as follows:

\[ \forall y ((\text{Phy-laws } ∧ (\text{IN Cart Kitchen}) ∧ ¬(\text{IN Fr Kitchen})) \rightarrow \neg y, \]

since:

\[ \text{Phy-laws } ∧ (\text{IN Cart Kitchen}) ∧ ¬(\text{IN Fr Kitchen}) = (\text{IN Cart Kitchen}) ∧ (\text{IN Fr LivingRoom}) ∧ ¬(\text{OnCart Fr}) \]

and

\[ (\text{IN TV LivingRoom}) v ¬(\text{IN TV Kitchen}) \]

Next, suppose, φ is the fact that Fr (refrigerator) is in the kitchen (IN Fr Kitchen) and the result α of the action (MoveC Kitchen) is (IN Cart Kitchen). After plugging these values into expression (6), we get the inducing preconditions of the fact (IN Fr Kitchen) with respect to the results of the action (MoveC Kitchen) as follows:

\[ \forall y ((\text{Phy-laws } ∧ (\text{IN Cart Kitchen}) ∧ ¬(\text{IN Fr Kitchen})) \rightarrow \neg y, \]

since:

\[ \text{Phy-laws } ∧ (\text{IN Cart Kitchen}) ∧ ¬(\text{IN Fr Kitchen}) = (\text{IN Cart Kitchen}) ∧ (\text{IN Fr LivingRoom}) ∧ ¬(\text{OnCart Fr}) \]

and

\[ (\text{IN TV LivingRoom}) v ¬(\text{IN TV Kitchen}) \]

only if the fact, (OnCart Fr) satisfies expression (6); therefore, it is the inducing precondition for the fact (IN Fr Kitchen) with respect to the results of the action (MoveC Kitchen).
to be defined. As proposed by the block stacking example in the beginning of this paper, a plan is represented as a totally ordered directed graph with an initial node representing the initial situation, a goal node representing the goal situation, and intermediate nodes representing action nodes as shown in figure 9. The arcs of the plan graph are directed and define the order of the nodes.

Each node has four slots: the first slot (Action-slot) holds an action; the second slot (K-slot) holds the next situation after the action is performed; the third slot (agenda-slot) holds the goals that must be achieved before this node, and the fourth slot (P-Facts-slot) holds the facts that must be preserved from the previous node up to this node until the action is executed. The action slot of the initial node contains a literal "START" to indicate that this is the starting point of a plan. The K-slot of the initial node contains an initial situation ($K_0$). The value $ST$ in any slot indicates that this slot is empty. The action slot of the goal node contains a literal "STOP" to indicate that this is the end point of a plan. The agenda-slot of the goal node contains the goals to be achieved by a plan.

3.5. Algorithm for Plan Synthesis

A plan graph is initialized having an initial node and a goal node. The K-slot of the initial node is filled with the initial situation, and the agenda-slot of the goal node is filled with the goals to be achieved. We apply the following rules to obtain a valid plan, if one exists. Update the K-slot of a node as needed using expression (4).

A. Find a node that is closest to the initial node whose agenda is not empty (i.e., if there are some goals posted on the agenda), and designate this node as a current node. If no such node is found, a valid plan has been obtained and the plan synthesis process returns the plan graph.

B. The agenda of the current node is tested to identify a goal that is not yet satisfied. If no goal is found, that is, the goals left on the agenda are satisfied, the remaining goals are moved from the agenda to the preserved facts (P-facts slot) of the current node, and the goals that are propagated by the frame laws through the previous nodes are added to the preserved facts of all the previous nodes. Then, return to step A. If an unsatisfied goal is found on the agenda, the plan graph is modified either by moving the goal toward the initial node over the intermediate nodes or by adding a new node if moving the goal to the previous node is not possible.

C. The goal is moved toward the previous node if the action of the previous node can either induce or preserve the goal, and the previous node is not the initial node.

(i) Find the inducing preconditions of the goal with respect to the action of the previous node using expression (6). If the inducing preconditions contradict the physical laws, the agenda and the preserved facts of the node, go to step (ii). Otherwise, add the inducing preconditions to the agenda and move the goal from the agenda to the P-Facts-slot of the current node. Return to step A.

(ii) Find the holding preconditions of the goal with respect to the action of the node using expression (5). If the holding preconditions contradict the goal, the physical laws, the agenda and the preserved facts of the current node, go to step D. Otherwise, the goal and the holding preconditions are added to the agenda of the current node. Then, move the goal from the agenda to the P-Facts-slot of the current node. Return to step A. The purpose of this step is to move a goal toward the initial node while moving it we may find an existing action that can induce the goal. Further, this technique helps remove cycles in a plan so that no redundant actions are included in the plan.

D. Find an action that can cause the goal to be true, and create a new node for the action. Insert the new node into the plan graph immediately before the current node using the following rules.

(i) Find all the actions from the actions list such that the goal is entailed by the results of each action. If no action is found go to step (ii). Otherwise, pick one of the actions for the new node and mark the other actions as backtracking points of the plan synthesis process if it fails at some point. Create a new node. Initialize the action-slot with the action, the agenda-slot with the preconditions of the action, and the P-Facts-slot with the preserved facts (P-Facts-slot) of the current node, then
insert the new node into the plan graph just before the current node. Now move the goal from the agenda to the P-Facts-slot of the current node and go to step A.

(ii) Identify all the actions from the actions list such that the results of each action can force the goal with appropriate inducing preconditions, which do not contradict the preconditions of the action and the physical laws. If no action is found, the plan synthesis process immediately fails and must therefore backtrack to a previously marked backtracking point. Otherwise, pick one of the actions for the new node and mark the remaining actions for backtracking of the plan synthesis process if it fails at some point. Create a new node; initialize the action-slot with the action, the agenda-slot with the conjunction of the preconditions of the action and the inducing preconditions of the goal with respect to the action, and the P-Facts-slot with the preserved facts (P-Facts-slot) of the current node. Then insert the new node into the plan graph just before the current node. Now move the goal from the agenda to the P-facts-slot of the current node and go to step A.

If a valid plan is constructed, this algorithm returns a plan graph. If all the backtracking points are explored, and no valid plan is possible, the plan synthesis process fails, and returns a nil plan. But if valid plans are possible for some other alternatives and all the backtracking points are explored, this algorithm can yield all the alternative plans. A computer program based on this algorithm is implemented in Schemata [2] and the nonmonotonic modal logic Z system.

To demonstrate the plan synthesis algorithm, let us consider the Cart-TV-Refrigerator example again. The initial situation is defined by $K_0$ and represented by figure 1.

$$K_0 = \text{df } (\text{IN Cart LivingRoom} \land \text{IN Fr LivingRoom}) \land (\text{IN Cart Kitchen} \land \text{IN Fr Kitchen}) \land (\text{IN TV LivingRoom} \land \text{IN Fr Kitchen})$$

Goals =df $\text{IN Cart Kitchen} \land \text{IN TV LivingRoom} \land \text{IN Fr Kitchen}$

Phys-laws =df $\forall x,y,z ((\text{IN x y}) \land (\text{IN x z})) \rightarrow (x = y)$

$(\text{Results (MoveC x)}) = \text{df } (\text{IN Cart x})$

$(\text{Results (LoadC x)}) = \text{df } (\text{ON Cart x})$

$(\text{Results (UnloadC x)}) = \text{df } \neg (\text{ON Cart x})$

$(\text{Pre-conds (MoveC x)}) = \text{df } \text{ST}$

The plan graph is initialized as shown in figure 10.

$$\text{Action-slot} = (\text{MoveC Kitchen})$$

$K$-slot = undefined;

Agenda-slot = (pre-conds (MoveC Kitchen)), that is, ST means no particular precondition for this action, and

P-Facts-slot = ST; that is, the preserved facts of the goal node.

After inserting this node into the plan graph, the goal (IN Cart Kitchen) is moved to the P-Facts-slots of the goal node, and we again go to step A. Now the goal node is the only node whose agenda contains some goals. First, we update the $K$-slot of Node-1. The new situation ($K$-slot) of Node-1 after the action (MoveC Kitchen) is executed, is given below and shown in figure 2:

$$K_1 = \text{df } \neg (\text{IN Cart LivingRoom}) \land \neg (\text{IN TV LivingRoom}) \land \neg (\text{IN Fr Kitchen})$$

Now (in step B), we pick the goal (IN TV LivingRoom). In step C(i), using expression (6), we find that the inducing precondition for the goal (IN TV LivingRoom) with respect to the action of the previous node is false which means that if the TV is not in the living room, moving the cart to the kitchen cannot force
the TV to be in the living room. Therefore, we go to step C(ii). In this step, using expression (5), we find that \( \neg(\text{OnCart TV}) \) is the holding precondition. This holding precondition does not contradict the conditions outlined in step C(ii); thus, the goal and the holding preconditions are added to the agenda of Node-1. Now the agenda of Node-1 is \( (\text{IN TV LivingRoom}) \land \neg(\text{OnCart TV}) \). After moving the goal (IN TV LivingRoom) from the agenda to the P-facts of the goal node as shown in figure 12, we jump to step A.

**Figure 12. Plan Graph after Adding Holding Conditions to Node-1.**

This time Node-1 is closest to the initial node, which has some goals on its agenda. We pick the unsolved goal \( \neg(\text{OnCart TV}) \) (step B), and since the previous node is the initial node, go to step D (skip step C). Now we find that the action (UnloadCTV) accomplishes the goal. Therefore, a new node (Node-2) is added to the plan graph after initializing its slots as follows:

- **Action-slot** = (UnloadCTV) that is unload TV from the cart.
- **K1** = undefined.
- **Agenda-slot** = (pre-conds (UnloadCTV)) that is $T$
- **P-Facts-slot** = $\neg T$ that is the preserved facts of Node-1.

The goal \( \neg(\text{OnCart TV}) \) is moved to the P-Facts-slot of Node-1. The modified plan graph is given in figure 13.

**Figure 13. Plan Graph after Inserting Node-2.**

In step A, we again find that the Node-1 is closest to the initial node. But at this time in step B we find that the agenda of Node-1 contains a goal that is already satisfied. The goal is (IN TV LivingRoom) that is moved from the agenda to the P-facts of Node-1. Now the K-slots of Node-2 and Node-1 are updated as follows:

- **K1** = ((IN Cart LivingRoom) \land \neg(IN Cart Kitchen)
- **K2** = ((IN Cart Kitchen) \land \neg(IN Cart LivingRoom)
- **K3** = ((IN TV LivingRoom) \land \neg(IN TV Kitchen)
- **K4** = ((IN Cart TV) \land \neg(IN Fr LivingRoom)
- **K5** = ((IN Fr Kitchen) \land \neg(IN Cart Fr)

Now we pick the only remaining goal (IN Fr Kitchen) on the agenda of the goal node. In step C(i), using expression (6), we find that the inducing precondition for the goal (IN Fr Kitchen) with respect to the action of Node-1 is (OnCart Fr), which means putting the refrigerator on the cart will force the goal to become true. The inducing precondition (IN Fr Kitchen) is consistent with the conditions of step C(i); therefore, (OnCart Fr) is added to the agenda of node-1 and the goal is moved from the agenda to the P-Facts-slot of Node-1. Return to step A.

This time Node-1 is closest to the initial node, which has some goals on its agenda. We pick the unsolved goal (OnCart Fr) (step B), and in step C we find that this goal cannot be induced by Node-2 but, it can preserve the goal without any additional holding preconditions, so we add the goal to the P-Facts-slot of Node-1, and shift the goal (OnCart Fr) to the agenda of Node-2. Since the node previous to Node-2 is the initial node, we cannot shift the goal any farther, so we go to step D. Now we find that the action (LoadCFr) accomplishes the goal. Therefore, a new node (Node-3) is added to the plan graph after initializing its slots as follows:

- **Action-slot** = (LoadCFr), that is load refrigerator on the cart.
- **K-slot** = undefined.
- **Agenda-slot** = (pre-conds (LoadCFr)) that is (IN Cart LivingRoom).
- **P-Facts-slot** = $T$ that is the preserved facts of Node-2.

The goal (OnCart Fr) is moved from the agenda to the P-Facts-slot of Node-1. The modified plan graph is given in figure 14.

**Figure 14. Plan Graph after Inserting Node-3.**
In step A, we again find that the Node-3 is closest to the initial node. But at this time in step B, we find that the agenda of Node-3 contains a goal that is already satisfied. The goal (IN Cart LivingRoom) is moved from the agenda to the P-facts of Node-1. Now the K-slots of Node-3, Node-2 and Node-1 are updated as follows:

\[ K_1 = (\text{IN Cart LivingRoom}) \land \neg (\text{IN Cart Kitchen}) \land (\text{IN TV LivingRoom}) \land \neg (\text{IN Fr Kitchen}) \land (\text{OnCart Fr}) \]

\[ K_2 = (\text{IN Cart LivingRoom}) \land \neg (\text{IN Cart Kitchen}) \land (\text{IN TV LivingRoom}) \land (\text{IN Fr LivingRoom}) \land (\text{OnCart Fr}) \]

\[ K_3 = (\text{IN Cart Kitchen}) \land \neg (\text{IN Cart LivingRoom}) \land (\text{IN TV LivingRoom}) \land (\text{IN Fr Kitchen}) \land (\text{OnCart Fr}) \]

Finally the agendas of all node are empty so the plan graph is returned, which gives us a valid plan for the Cart-TV-refrigerator problem.

4. CONCLUSION

A theory of plan synthesis that uses nonmonotonic reasoning based on the modal quantificational logic Z is developed. This theory reasons about actions to solve the frame problem and also, it reasons about the plan synthesis to detect possible or impossible orderings of the actions. In this theory, only the explicit results of an action are provided and no delete list is needed. The facts from the current situation to the next situation are carried out by the frame axiom, and the facts are automatically added and deleted from one situation to the next by the nonmonotonic reasoning as the actions are performed, which makes the action's description simpler. A plan synthesis algorithm is also developed. This algorithm builds a plan incrementally by inserting new actions into the plan or by modifying the plan using the nonmonotonic reasoning system of the modal quantificational logic Z.

5. ACKNOWLEDGEMENTS

We thank Dr. Sung Park and Dr. Carlos Araya for their help in this research.

6. REFERENCES


