

STRIPS PLANNING [Nilsson]

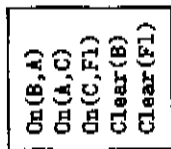
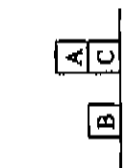


Figure 22.1

A State Description



Precondition:
On(B,A)
Clear(B)
Clear(F1)

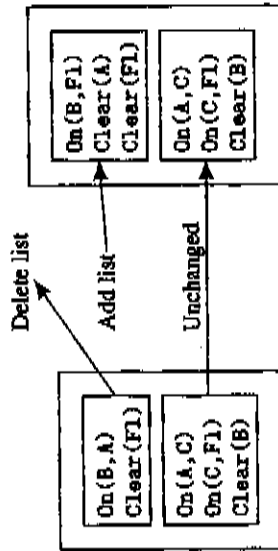


Figure 22.2

A STRIPS Operator



Figure 22.4

The Sussman Anomaly

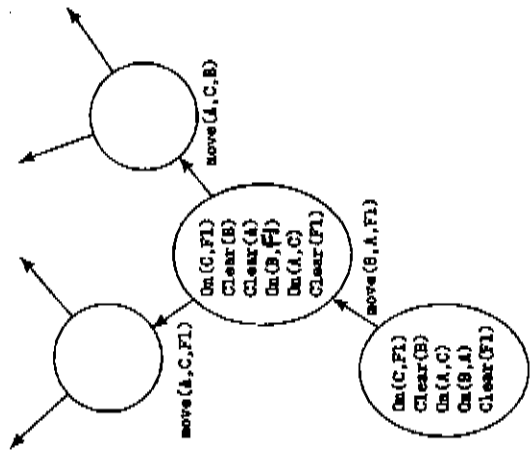
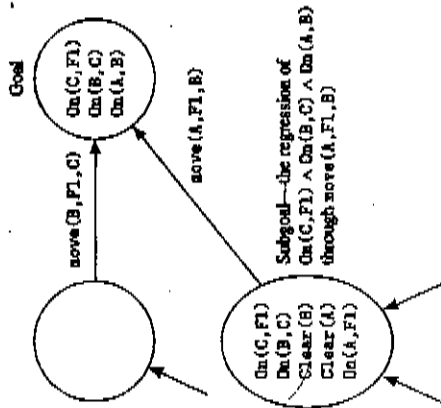


Figure 22.3

Forward Search



Continue until a subgoal is produced that is satisfied by current world state

Figure 22.5

Regressing a Conjunction through a STRIPS Operator

11.4 The monkey-and-bananas problem is faced by a monkey in a laboratory with some bananas hanging out of reach from the ceiling. A box is available that will enable the monkey to reach the bananas if he climbs on it. Initially, the monkey is at *A*, the bananas at *B*, and the box at *C*. The monkey and box have height *Low*, but if the monkey climbs onto the box he will have height *High*, the same as the bananas. The actions available to the monkey include *Go* from one place to another, *Push* an object from one place to another, *ClimbUp* onto or *ClimbDown* from an object, and *Grasp* or *Ungrasp* an object. Grasping results in holding the object if the monkey and object are in the same place at the same height.

- Write down the initial state description.
- Write down STRIPS-style definitions of the six actions.
- Suppose the monkey wants to fool the scientists, who are off to tea, by grabbing the bananas, but leaving the box in its original place. Write this as a general goal (i.e., not assuming that the box is necessarily at *C*) in the language of situation calculus. Can this goal be solved by a STRIPS-style system?
- Your axiom for pushing is probably incorrect, because if the object is too heavy, its position will remain the same when the *Push* operator is applied. Is this an example of the ramification problem or the qualification problem? Fix your problem description to account for heavy objects.

```

Init(Have(Cake))
Goal(Have(Cake) ∧ Eaten(Cake))
Action(Eat(Cake))
  PRECOND: Have(Cake)
  EFFECT: ¬ Have(Cake) ∧ Eaten(Cake)
Action(Bake(Cake))
  PRECOND: ¬ Have(Cake)
  EFFECT: Have(Cake)
    
```

Figure 11.11 The "have cake and eat cake too" problem.

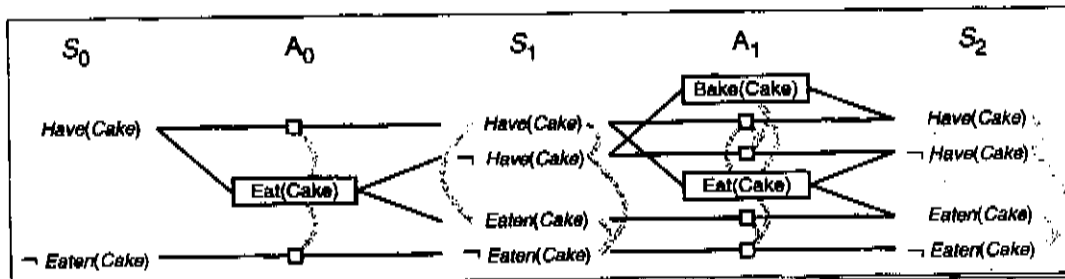


Figure 11.12 The planning graph for the "have cake and eat cake too" problem up to level S_2 . Rectangles indicate actions (small squares indicate persistence actions) and straight lines indicate preconditions and effects. Mutex links are shown as curved gray lines.