4 Planning in Situation Calculus

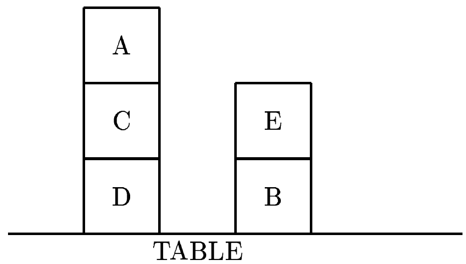

Figure 2: A Block World

We present a slight different situation calculus theory for the block world in the above picture. It consists of:

1. Constants: A, B, C, D, E, T, two unary (domain) predicate $Block$ and $Location$, where $Block = \{A, B, C, D, E\}$ and $Location = \{A, B, C, D, E, T\}$;
2. Actions: $Pickup(x, y)$, $Put(x, y)$, and $Put(x, T)$ (Instead of $Pickup(x)$ we use $Pickup(x, y)$ – to say that the agent pickup the block x from a location y – which can be a block or the table);
3. Fluents: $On(x, y)$, $On(x, T)$, $Holding(x)$, $Clear(x)$.
4. The set of precondition axioms:

$$Poss(Pickup(x, y), s) \leftrightarrow x \neq T \wedge Location(y) \wedge \forall z(Holds(\neg Holding(z), s)) \wedge Holds(Clear(x), s) \tag{34}$$

$$Poss(Put(x, y), s) \leftrightarrow Block(x) \wedge Block(y) \wedge x \neq y \wedge Holds(Holding(x), s) \wedge Holds(Clear(y), s) \tag{35}$$

$$Poss(Put(x, T), s) \leftrightarrow Block(x) \wedge Holds(Holding(x), s) \tag{36}$$

5. The set of effect axioms:

$$Poss(Pickup(x, y), s) \rightarrow Holds(Holding(x), Do(Pickup(x, y), s)) \tag{37}$$

$$Poss(Pickup(x, y), s) \rightarrow Holds(\neg On(x, y), Do(Pickup(x, y), s)) \tag{38}$$

$$Poss(Pickup(x, y), s) \rightarrow [y \neq T \rightarrow Holds(Clear(y), Do(Pickup(x, y), s))] \tag{39}$$

$$Poss(Put(x, y), s) \rightarrow Holds(On(x, y), Do(Put(x, y), s)) \tag{40}$$

$$Poss(Put(x, y), s) \rightarrow Holds(\neg Holding(x), Do(Put(x, y), s)) \tag{41}$$

$$Poss(Put(x, y), s) \rightarrow Holds(\neg Clear(y), Do(Put(x, y), s)) \tag{42}$$

$$Poss(Put(x, T), s) \rightarrow Holds(\neg Holding(x), Do(Put(x, T), s)) \tag{43}$$

$$Poss(Put(x, T), s) \rightarrow Holds(On(x, T), Do(Put(x, T), s)) \tag{44}$$

6. The set of successor state axioms:

$$Holds(On(x,y),Do(a,s)) \leftrightarrow (a = Put(x,y)) \wedge Poss(a,s) \\ \vee(Holds(On(x,y),s) \wedge \neg(a = Pickup(x,y) \wedge Poss(a,s))) \tag{45}$$

$$Holds(On(x,T),Do(a,s)) \leftrightarrow (a = Put(x,T)) \wedge Poss(a,s) \\ \vee(Holds(On(x,T),s) \wedge \neg(a = Pickup(x,T) \wedge Poss(a,s))) \tag{46}$$

$$Holds(Holding(x),Do(a,s)) \leftrightarrow (a = Pickup(x,y)) \wedge Poss(a,s) \\ \vee(Holds(Holding(x),s) \wedge \\ \neg((a = Put(x,z) \vee a = Put(x,T)) \wedge Poss(a,s))) \tag{47}$$

$$Holds(Clear(x),Do(a,s)) \leftrightarrow (a = Pickup(y,x)) \wedge Poss(a,s) \\ \vee(Holds(Clear(x),s) \wedge \\ \neg(a = Put(y,x) \wedge Poss(a,s))) \tag{48}$$

What do I miss in the above formulas (45)-(48)?

7. Initial situation:

$$Holds(On(A,C),S_0) \tag{49}$$
$$Holds(On(C,D),S_0) \tag{50}$$
$$Holds(On(D,T),S_0) \tag{51}$$
$$Holds(On(E,B),S_0) \tag{52}$$
$$Holds(On(B,T),S_0) \tag{53}$$

Let us denote the theory by $\mathcal{T}$. Under the assumptions that

1. changes are caused by actions and
2. no thing happens before S_0

we can view each situation as a sequence of actions. Thus, we can prove what is true/false in each situation and hence we can formulate the planning task as follows.

Planning problem: Given a situation calculus $\mathcal{T}$ and a fluent formula φ. A *plan achieving* φ is a sequence of actions $a_1, \ldots, a_n$ such that

$$\mathcal{T} \models Holds(\varphi, Do(a_n, \ldots, Do(a_1, S_0)))$$

Example 4.1 *[Planning in Block World]*

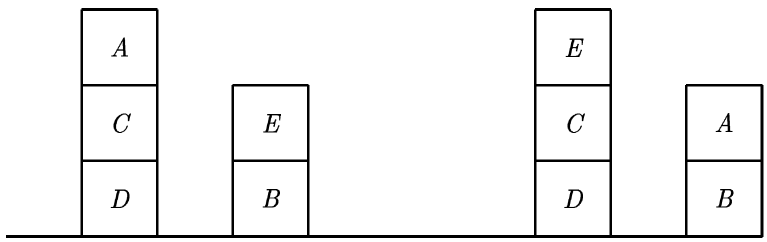

Figure 3: A Planning Problem in Block World

Solving a planning problem means to finding a plan! How can we find a plan?

Generate and Test: The most simple way to find a plan. We can generate a sequence of actions, say α, and then test for $\mathcal{T} \models Holds(\varphi, Do(\alpha, S_0))$. So, we have the following algorithm (pseudo code for implementation):

Algorithm 4.1 (First Idea) *Given are $\mathcal{T}$ and φ*

main

while true *do*
  α = *generate_sequence_action*
  if testOK(α) *return* α *endwhile*

function

generate_sequence_action
  return a sequence of actions

function

testOK(α)
  if $\mathcal{T} \models Holds(\varphi, Do(a_n, \ldots, Do(a_1, S_0)))$ *return* true
  else return false

Implementation problem: Checking if φ holds in $Do(\alpha, S_0)$, i.e., checking whether

$$(*) \quad \mathcal{T} \models Holds(\varphi, Do(\alpha, S_0))$$

holds or not.

Another problem is how to represent the goal?

Answer: We can use a theorem prover, that tells us what is true in $Do(\alpha, S_0)$? to check (*). We can simplify our life (for now) by representing the goal by a set of fluents. Thus, the goal of the planning problem in Example 4.1 can be represented by the set

$$\{On(A,B), On(C,D), On(D,T), On(E,C), On(B,T)\}.$$

Running the algorithm:

The set of actions: $\{Put(x,y), Put(x,T), Pickup(x,y)$

Iteration No. 1

Possible action in S_0: $Pickup(A,C), Pickup(E,B)$

We then have two situation $S_1 = Pickup(A,C)$ and $S_2 = Pickup(E,B)$. In S_1, the following fluents are true: $On(C,D), On(D,T), On(E,B), On(B,T), Holding(A)$ whereas $On(A,C), On(C,D), On(D,T), On(B,T), Holding(E)$ are true in S_2. (Notice that we have not listed ALL the fluents that are true in each situation, i.e., this list is incomplete.)

The following picture illustrates the different possible situations

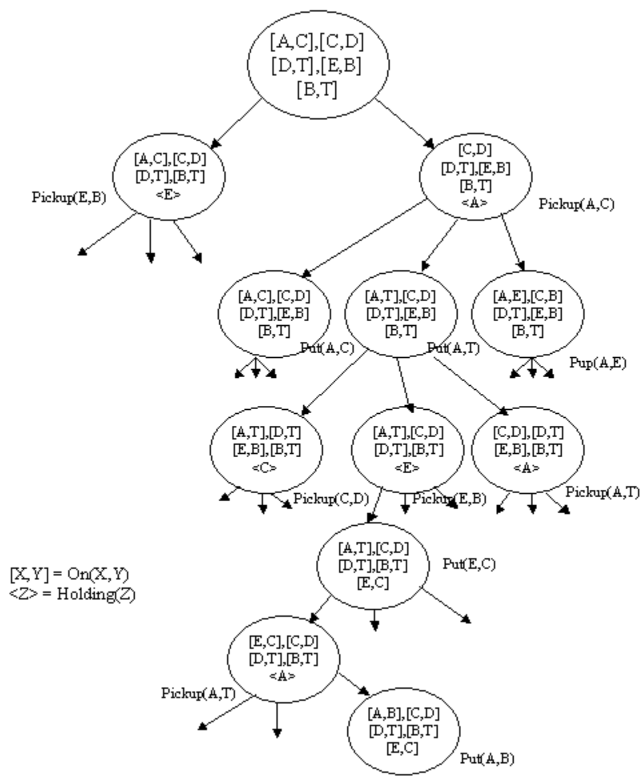

Figure 4: A part of the situation tree of the block world