Planning as SAT

Tu Phan Cuong Pham

February 22, 2004

1 Planning as SAT

1.1 SAT problem in Propositional Logic

Given a set of propositional sentences, called *theory*, decide if there is some assignment to the propositions in the theory such that every propositional sentences are true. If such an assignment exists, it is called a *model* of the theory. A mechanism for finding a model for a propositional theory is called *SAT Solver*.

1.2 Planning as SAT

In this approach, a planning problem is encoded as a propositional satisfiability problem (SAT) and a SAT solver is then used to solve the resulting SAT problem. The output of the SAT solver, which is a model, is then converted back into plans.

Figure 1: Planning as SAT

What we are interested here is that how to encode planning problems into a corresponding propositional logic theory. Basically, we need to encode the intial state, the domain description and the goal. The most common way to do this is to associate each fluent or each action in the planning problem with a time instant to be a proposition in the destination theory. The initial state is refered to as time instant 1 and the goal is refered to as time instant n where n is the maximum length of the plan we want to find.

The next part describes the most straight-forward way to encode a planning problem as a SAT problem.

1.3 Straight-forward Encodings

Given a planning problem $P =< D, I, G >$ in a STRIPS-like language and an action a, let Pre_a, Add_a and Del_a denote the precondition, add-effect and delete-effect of action a respectively. The general procedure for translating P into the corresponding propositional theory T is sketched out as follows:

- Each fluent $f \in P$ corresponds to the set of propositions $\{f^i\}$ in T, where $i = 1 \dots n$.
- Each action $a \in P$ corresponds to the set of propositions $\{a^i\}$, where $i = 1 \dots n - 1$.

- The initial state is encoded as:

$$\bigwedge_{f \in I} f^1 \tag{1}$$

- The goal is encoded as:

$$\bigwedge_{g \in G} g^n \tag{2}$$

- Encoding the domain description
  (i) For every action $a \in D$ and integer i, $1 \leq i \leq n-1$,

$$a^i \Rightarrow \bigwedge_{l \in Pre_a} l^i$$

  (ii) For every fluent f and integer i, $1 \leq i \leq n-1$,

$$f^{i+1} \leftrightarrow (f^i \wedge (\bigwedge_{a \in D, f \in Del_a} \neg a^i)) \vee (\bigvee_{a \in D, f \in Add_a} a^i) \tag{3}$$

$$\neg f^{i+1} \leftrightarrow (\neg f^i \wedge (\bigwedge_{a \in D, f \in Add_a} \neg a^i)) \vee (\bigvee_{a \in D, f \in Del_a} a^i) \tag{4}$$

  Axioms of forms (3) and (4) are known as classical frame axioms. The first one is for positive literals, stating that a fluent is true at time moment $i+1$ if and only if it is true at time moment i and no action changes its value or it is an add-effect of some action that occurs at time moment i. The second one is frame axioms for negative literals.

For example, consider the Air Transportation domain that consists of two aiports SFO and JFK, two planes P_1 and P_2. Only one kind of actions allowed in the domain is Fly. Initially, plane P_1 is at SFO and plane P_2 is at JFK. Our goal is to have plane P_1 at JFK and P_2 at SFO. The STRIPS-like description for the domain is follows:

Initial State
  $At(P_1, SFO) \wedge At(P_2, JFK)$
Action Schemata
  $Action(Fly(plane, from, to),$
    PRECOND: $At(plane, from)$
    EFFECT: $\neg At(plane, from) \wedge At(plane, to))$
Goal
  $At(P_1, JFK) \wedge At(P_2, SFO)$

If we are going to find a plan of length up to at most 3, i.e., $n = 3$, the encoding will contain the following propositions:

$$\{At(P_1, SFO)^i, At(P_1, JFK)^i, At(P_2, SFO)^i, At(P_2, JFK)^i,$$

$$Fly(P_1, SFO, JFK)^i, Fly(P_1, JFK, SFO)^i, Fly(P_2, SFO, JFK)^i, Fly(P_2, JFK, SFO)^i\}$$

for $i = 1 \ldots 3$.

1. The initial state is encoded as:

$$At(P_1, SFO)^1 \wedge At(P_2, JFK)^1 \wedge \neg At(P_1, JFK)^1 \wedge \neg At(P_2, SFO)^1 \tag{5}$$

2. The goal is encoded as:

$$At(P_1, JFK)^3 \wedge At(P_2, SFO)^3 \tag{6}$$

3. Encoding the domain description

(i) Actions imply their preconditions

$$Fly(P_1, SFO, JFK)^i \rightarrow At(P_1, SFO)^i \tag{7}$$
$$Fly(P_1, JFK, SFO)^i \rightarrow At(P_1, JFK)^i \tag{8}$$
$$Fly(P_2, SFO, JFK)^i \rightarrow At(P_2, SFO)^i \tag{9}$$
$$Fly(P_2, JFK, SFO)^i \rightarrow At(P_2, JFK)^i \tag{10}$$

(ii) Classical Frame Axioms

For positive fluent literals:

$$At(P_1, SFO)^{i+1} \leftrightarrow (At(P_1, SFO)^i \wedge \neg Fly(P_1, SFO, JFK)^i) \vee Fly(P_1, JFK, SFO)^i \tag{11}$$
$$At(P_1, JFK)^{i+1} \leftrightarrow (At(P_1, JFK)^i \wedge \neg Fly(P_1, JFK, SFO)^i) \vee Fly(P_1, SFO, JFK)^i \tag{12}$$
$$At(P_2, SFO)^{i+1} \leftrightarrow (At(P_2, SFO)^i \wedge \neg Fly(P_2, SFO, JFK)^i) \vee Fly(P_2, JFK, SFO)^i \tag{13}$$
$$At(P_2, JFK)^{i+1} \leftrightarrow (At(P_2, JFK)^i \wedge \neg Fly(P_2, JFK, SFO)^i) \vee Fly(P_2, SFO, JFK)^i \tag{14}$$

For negative fluent literals:

$$\neg At(P_1, SFO)^{i+1} \leftrightarrow (\neg At(P_1, SFO)^i \wedge \neg Fly(P_1, JFK, SFO)^i) \vee (Fly(P1, SFO, JFK)^i) \tag{15}$$
$$\neg At(P_1, JFK)^{i+1} \leftrightarrow (\neg At(P_1, JFK)^i \wedge \neg Fly(P_1, SFO, JFK)^i) \vee (Fly(P1, JFK, SFO)^i) \tag{16}$$
$$\neg At(P_2, SFO)^{i+1} \leftrightarrow (\neg At(P_2, SFO)^i \wedge \neg Fly(P_2, JFK, SFO)^i) \vee (Fly(P2, SFO, JFK)^i) \tag{17}$$
$$\neg At(P_2, JFK)^{i+1} \leftrightarrow (\neg At(P_2, JFK)^i \wedge \neg Fly(P_2, SFO, JFK)^i) \vee (Fly(P2, JFK, SFO)^i) \tag{18}$$

for $i = 1 \ldots 2$.

The corresponding propositional theory for the planning problem is therefore (5)-(18). It is easy to check that one possible model[1] for the theory is

$$\{At(P_1, SFO)^1, At(P_2, JFK)^1,$$
$$At(P_1, JFK)^3, At(P_2, SFO)^3,$$
$$Fly(P_1, SFO, JFK)^1, Fly(P_2, JFK, SFO)^2\}$$

As such, the following plan can be easily extracted from the model:

$$\{Fly(P_1, SFO, JFK); Fly(P_2, JFK, SFO)\}$$

[1]Note that we are assuming that everything that is not specified in the model is false (Close-World Assumption).

2 Other Encoding Methods

This section summarizes several encoding methods that are currently used in existing SAT-planners.

2.1 Graphplan based Encodings

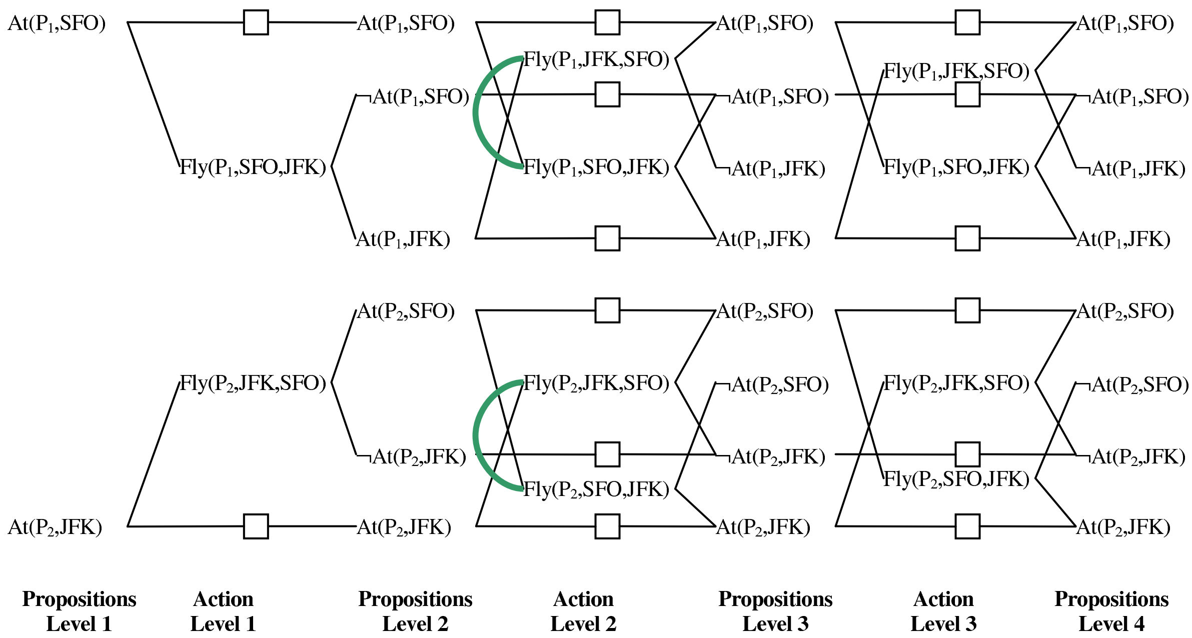

Figure 2: Planning Graph for the Air Transportation Domain (Only some mutexes are shown)

(1) **Each fact at level i implies the disjunction of all the operators at level $i-1$ that have it as an add-effect, e.g,**

$$At(P_2, SFO)^2 \Rightarrow Fly(P_2, SFO, JFK)^1 \tag{19}$$
$$At(P_1, SFO)^3 \Rightarrow (Fly(P_1, JFK, SFO)^2 \vee no\text{-}op(At(P_1, SFO))^2) \tag{20}$$

(2) **Operators imply their preconditions, e.g.,**

$$Fly(P_1, SFO, JFK)^1 \Rightarrow At(P_1, SFO)^1 \tag{21}$$
$$Fly(P_2, SFO, JFK)^1 \Rightarrow At(P_2, JFK)^1 \tag{22}$$

(3) **Conflicting actions are mutually exclusive, e.g.,**

$$\neg(Fly(P_1, JFK, SFO)^2 \wedge Fly(P_1, SFO, JFK)^2) \tag{23}$$
$$\neg(Fly(P_2, JFK, SFO)^2 \wedge Fly(P_1, SFO, JFK)^2) \tag{24}$$

One possible model for this encoding is

$$\{At(P_1, SFO)^1, \neg At(P_1, JFK)^1, \neg At(P_2, SFO)^1, At(P_2, JFK)^1,$$
$$Fly(P_1, SFO, JFK)^1, \neg Fly(P_1, JFK, SFO)^1, \neg Fly(P_2, SFO, JFK)^1, \neg Fly(P_2, JFK, SFO)^1,$$
$$\neg At(P_1, SFO)^2, At(P_1, JFK)^2, \neg At(P_2, SFO)^2, At(P_2, JFK)^2,$$
$$\neg Fly(P_1, SFO, JFK)^2, \neg Fly(P_1, JFK, SFO)^1, \neg Fly(P_2, SFO, JFK)^2, Fly(P_2, JFK, SFO)^2,$$

$$\neg At(P_1, SFO)^3, At(P_1, JFK)^3, At(P_2, SFO)^3, \neg At(P_2, JFK)^3,$$
$$\neg Fly(P_1, SFO, JFK)^3, \neg Fly(P_1, JFK, SFO)^3, \neg Fly(P_2, SFO, JFK)^3, \neg Fly(P_2, JFK, SFO)^3,$$

From this model, we can extract the plan $Fly(P_1, SFO, JFK), Fly(P_2, JFK, SFO)$.

2.2 Linear Encodings

(1) **An action implies both its preconditions and effects, e.g.,**

$$Fly(P_1, SFO, JFK)^2 \Rightarrow At(P_1, SFO)^2 \tag{25}$$
$$Fly(P_1, SFO, JFK)^2 \Rightarrow At(P_1, JFK)^3 \tag{26}$$

(2) **Exactly one action occurs at each time instant, e.g.,**

$$Fly(P_1, SFO, JFK)^2 \vee Fly(P_1, JFK, SFO)^2 \vee Fly(P_2, SFO, JFK)^2 \vee Fly(P_2, JFK, SFO)^2 \tag{27}$$
$$Fly(P_1, SFO, JFK)^2 \Rightarrow \neg Fly(P_1, JFK, SFO)^2 \wedge \neg Fly(P_2, SFO, JFK)^2 \wedge \neg Fly(P_2, JFK, SFO)^2 \tag{28}$$
$$Fly(P_1, JFK, SFO)^2 \Rightarrow \neg Fly(P_1, SFO, JFK)^2 \wedge \neg Fly(P_2, SFO, JFK)^2 \wedge \neg Fly(P_2, JFK, SFO)^2 \tag{29}$$
$$\ldots$$

(3) **The initial state is completely specified, e.g.,**

$$At(P_1, SFO)^0 \wedge At(P_2, JFK)^0 \wedge \neg At(P_1, JFK)^0 \wedge \neg At(P_2, SFO)^0 \tag{30}$$

(4) **Classical frame conditions for all actions asserting facts about propositions that do not change value when an action occurs, e.g.,**

$$\begin{aligned} At(P_1, JFK)^2 \Leftrightarrow (At(P_1, JFK)^1 \wedge \neg(Fly(P_1, JFK, SFO)^1) \wedge At(P_1, JFK)^1) \\ \vee(Fly(P_1, SFO, JFK)^1 \vee At(P_1, SFO)^1) \end{aligned} \tag{31}$$

2.3 State-Based Encodings

The idea is based on the observation that only a relatively small number of axioms are needed to describe state transitions. This kind of representation emphasizes the use of axioms that force a state to be internally consistent rather than focuses on axioms describing about actions.

The encoding includes (1) axioms describing the initial state and goals, (2) axioms asserting that actions entail both their preconditions and effects, (3) axioms asserting that conflicting actions are mutually exclusive, (4) axioms ensuring that each state is *internally consistent*, and (5) axioms describing what it means for a fact to change its truth value.

The first three types of axioms are similar to those of Linear Encodings. The fourth type describes what it means for an individual state to be valid. For example, for the domain above, a state is valid (internally consistent) if it satisfies the following properties:

(i) Plane P_1 and P_2 must be at either SFO or JFK

$$(At(P_1, SFO)^i \vee At(P_1, JFK)^i) \tag{32}$$
$$(At(P_2, SFO)^i \vee At(P_2, JFK)^i) \tag{33}$$

(ii) One plane cannot be at two locations at the same time

$$\neg(At(P_1, SFO)^i \wedge At(P_1, JFK)^i) \tag{34}$$
$$\neg(At(P_2, SFO)^i \wedge At(P_2, JFK)^i) \tag{35}$$

The last type specifies all the possible actions that make a fluent to change its value, e.g.,

$$(\neg At(P_1, JFK)^1 \land At(P_1, JFK)^2) \Rightarrow Fly(P_1, SFO, JFK)^1 \tag{36}$$

This constraint says that if plane P_1 is at JFK at time 2 but it was not there at time 1 then there must be action $Fly(P_1, SFO, JFK)$ occuring at time 1 (assuming that the domain has only 2 airports SFO and JFK).