

SAT-Based Planning

Sven Koenig, USC

Russell and Norvig, 3rd Edition, Section 10.4.1

These slides are new and can contain mistakes and typos.
Please report them to Sven (skoenig@usc.edu).

SAT-Based Planning

- Planning problem (specified in STRIPS)
↓
- SATisfiability problem (propositional sentence)
↓
- Solution to SATisfiability problem (interpretation that makes the propositional sentence true)
↓
- Solution to planning problem (shortest operator sequence that transforms the start state to the goal state)

STRIPS Example

Start state:

At(Home)
 Unequal(Home, SM) Unequal(Home, HW)
 Unequal(SM, Home) Unequal(SM, HW)
 Unequal(HW, Home) Unequal(HW, SM)
 Sells(SM, Milk) Sells(SM, Bananas)
 Sells(HSW, Drill)

Goal state:

At(Home)
 Have(Milk)
 Have(Bananas)
 Have(Drill)

Operator schemata:

Go(x,y)

PRECOND: At(x), Unequal(x,y)

EFFECT: At(y), NOT At(x)

Buy(t,x)

PRECOND: Sells(x,t), At(x) ← To keep it simple, we do not model money.

EFFECT: Have(t)

SAT-Based Planning

The grey ground(ed) predicates are in all states that can be reached from the start state by operator applications.

Start state:

At(Home)
 Unequal(Home, SM) Unequal(Home, HW)
 Unequal(SM, Home) Unequal(SM, HW)
 Unequal(HW, Home) Unequal(HW, SM)
 Sells(SM, Milk) Sells(SM, Bananas)
 Sells(HSW, Drill)

Goal state:

At(Home)
 Have(Milk)
 Have(Bananas)
 Have(Drill)

Operator schemata:

Go(x,y)

PRECOND: At(x), Unequal(x,y)

EFFECT: At(y), NOT At(x)

Buy(t,x)

PRECOND: Sells(x,t), At(x)

EFFECT: Have(t)

SAT-Based Planning

We do this to keep the resulting SAT problem small, which makes it easier to solve it.

- Instantiate all predicates and eliminate those that are in all states reachable from the start state by operator applications.

Remaining ground(ed) predicates (= propositions):

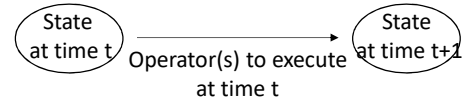
- At(Home), At(SM), At(HWS),
Have(Milk), Have(Bananas), Have(Drill)

- Instantiate all operator schemata and eliminate those operators whose preconditions are not satisfied in any state reachable from the start state.
- Remaining operators:

- Go(Home,SM), Go(Home,HWS), Go(SM,Home), Go(SM,HSW), Go(HSW,Home),
Go(HSW,SM), Buy(Milk,SM), Buy(Bananas,SM), Buy(Drill,HWS)

- Convert the remaining ground(ed) predicates and operators for each time slice 1...T (for a given time horizon T) into propositional symbols

SAT-Based Planning



These are just propositional symbols. We could call them P, Q, R, and so on.

	Time 0	Time 1	Time 2 = T
State	At(Home,0)	At(Home,1)	At(Home,2)
	At(SM,0)	At(SM,1)	At(SM,2)
	At(HWS,0)	At(HWS,1)	At(HWS,2)
	Have(Milk,0)	Have(Milk,1)	Have(Milk,2)
	Have(Bananas,0)	Have(Bananas,1)	Have(Bananas,2)
	Have(Drill,0)	Have(Drill,1)	Have(Drill,2)
Operators to execute	Go(Home,SM,0)	Go(Home,SM,1)	
	Go(Home,HWS,0)	Go(Home,HWS,1)	
	Go(SM,Home,0)	Go(SM,Home,1)	
	Go(SM,HWS,0)	Go(SM,HWS,1)	
	Go(HWS,Home,0)	Go(HWS,Home,1)	
	Go(HWS,SM,0)	Go(HWS,SM,1)	
	Buy(Milk,SM,0)	Buy(Milk,SM,1)	
	Buy(Bananas,SM,0)	Buy(Bananas,SM,1)	
	Buy(Drill,HWS,0)	Buy(Drill,HWS,1)	

SAT-Based Planning

1. Set $T := 0$.
2. Use a SAT solver to determine truth values for all propositional symbols so that they make the conjunct of the constraints on the following slides true, where the constraints use T as a parameter and are expressed as propositional sentences that need to be true.
3. If the SAT solver found no solution, then set $T := T+1$ and go to 2.
4. A plan is obtained by executing the operators whose propositional symbols are true at the corresponding time steps. For example, if $\text{Go}(\text{Home}, \text{SM}, 0) (\equiv \text{true})$ then operator $\text{Go}(\text{Home}, \text{SM})$ should be executed at time 0. (If no operators are executed during a time step, it can be skipped.)

Set of Constraints 1

- The ground(ed) predicates at time 0 correspond to the start state.
 - $\text{At}(\text{Home}, 0)$ (that is, $\text{At}(\text{Home}, 0) \equiv \text{true}$)
 - $\text{NOT At}(\text{SM}, 0)$ (that is, $\text{At}(\text{SM}, 0) \equiv \text{false}$)
 - $\text{NOT At}(\text{HWS}, 0)$
 - $\text{NOT Have}(\text{Milk}, 0)$
 - $\text{NOT Have}(\text{Bananas}, 0)$
 - $\text{NOT Have}(\text{Drill}, 0)$

Set of Constraints 2

- The ground(ed) predicates at time T correspond to the goal state. If the goal state is only partially specified, then not all predicates at time T are assigned a truth value.
 - At(Home,T)
 - Have(Milk,T)
 - Have(Bananas,T)
 - Have(Drill,T)

Set of Constraints 3

- Operators imply their preconditions.
 - Go(Home,SM,0) IMPLIES At(Home,0)
 - Go(Home,HWS,0) IMPLIES At(Home,0)
 - Go(SM,Home,0) IMPLIES At(SM,0)
 - ...
 - Go(Home,SM,1) IMPLIES At(Home,1)
 - ...
 - Buy(Milk,SM,0) IMPLIES At(SM,0)
 - Buy(Bananas,SM,0) IMPLIES At(SM,0)
 - Buy(Drill,HWS,0) IMPLIES At(HWS,0)
 - Buy(Milk,SM,1) IMPLIES At(SM,1)
 - ...

Set of Constraints 4

If we allow this, then a plan is no longer an operator sequence.
Rather, the operators are only partially ordered.

- If at most one operator can be executed at any time...
 - One cannot execute two different operators at the same time.
 - NOT(Go(Home,SM,0) AND Go(Home,HWS,0))
 - NOT(Buy(Milk,SM,0) AND Buy(Bananas,SM,0))
 - ...
- Alternatively, if several operators can be executed at the same time provided that they can be executed in any order...
 - One cannot execute two different operators at the same time if at least one of them deletes at least one precondition of the other one or if one of them adds at least one ground(ed) predicate that the other one deletes.
 - NOT(Go(Home,SM,0) AND Go(Home,HWS,0))
 - NOT(Go(SM,Home,0) AND Buy(Milk,SM,0))
 - ...

Set of Constraints 5

- For each ground(ed) predicate that is true at time $t+1$, (at least one operator was executed at time t that added it) or (it was true at time t already and no operator was executed at time t that deleted it).
 - Have(Milk,1) IMPLIES Buy(Milk,SM,0) OR Have(Milk,0)
 - At(Home,2) IMPLIES Go(SM,Home,1) OR Go(HWS,Home,1) OR At(Home,1) AND NOT Go(Home,SM,1) AND NOT Go(Home,HWS,1)
 - ...

Set of Constraints 6

- For each ground(ed) predicate that is false at time $t+1$, (at least one operator was executed at time t that deleted it) or (it was false at time t already and no operator was executed at time t that added it).
 - NOT Have(Milk,1) IMPLIES NOT HAVE(Milk,0) AND NOT Buy(Milk,SM,0)
 - NOT At(Home,2) IMPLIES Go(Home,SM,1) OR Go(Home,HWS,1) OR NOT At(Home,1) AND NOT Go(SM,Home,1) AND NOT Go(HWS,Home,1)
 - ...

SAT-Based Planning

1. Set $T := 0$.
2. Use a SAT solver to determine truth values for all propositional symbols so that they make the conjunct of the constraints on the previous slides true, where the constraints use T as a parameter and are expressed as propositional sentences that need to be true.
3. If the SAT solver found no solution, then set $T := T+1$ and go to 2.
4. A plan is obtained by executing the operators whose propositional symbols are true at the corresponding time steps. For example, if $\text{Go(Home,SM,0)} (\equiv \text{true})$ then operator Go(Home,SM) should be executed at time 0. (If no operators are executed during a time step, it can be skipped.)

SAT-Based Planning

- The resulting planner works well in practice.
- Issues
 - If the SAT solver is not guaranteed to find a solution even if one exists (for example, if it is based on hillclimbing with a time out), then the procedure is not guaranteed to find a shortest operator sequence.
 - If no plan exists, then the procedure will not terminate.