Artificial Intelligence

Planning as Satisfiability

E. Giunchiglia, F. Leofante, A. Tacchella

Computer Engineering
Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi

Last update: April 18, 2024

Agenda

1. What and Why
2. Planning as propositional satisfiability
   - Intuition
   - From a planning problem Π to a propositional wff Π_n
   - Encoding based on the Classical Frame Axioms
   - Encoding based on the Explanatory Frame Axioms
   - Remarks on the encodings
   - Encoding the Initial and Goal states
   - Constructing the propositional wff Π_n
3. Some references

What

We have already seen how to represent classical planning problems:

- representation of operators based on FOL notation
- states: sets of fluents (propositional variables)
- actions: ground operators (propositional variables) that change truth values of fluents

Now, how do we solve them, i.e., how can we search for a valid plan?

A solution: By reduction to the propositional satisfiability problem, i.e., by encoding the planning problem as a propositional formula whose models correspond to valid plans.

Why?

- **Dramatic speed-up** of decision procedures for propositional logic (SAT solvers) in the last decade
- Propositional satisfiability problems with **thousands variables** are now solved routinely in seconds by state-of-the-art SAT solvers
- Many success stories, e.g.,
  - circuit and program verification
  - ...
  - **planning**!

... but planning is PSPACE-complete and SAT is NP-complete and

$$\text{NP} \subseteq \text{PSPACE!}$$

Agenda

1 What and Why

2 Planning as propositional satisfiability
- Intuition
- From a planning problem Π to a propositional wff Π_n
- Encoding based on the Classical Frame Axioms
- Encoding based on the Explanatory Frame Axioms
- Remarks on the encodings
- Encoding the Initial and Goal states
- Constructing the propositional wff Π_n

3 Some references

Intuition

Formulate a **planning problem** Π as a **SAT problem** by:

1. Fixing a bound n on the "length" of the solution we are looking for
2. Building a formula Π_n that encodes the planning problem with bound n
3. Calling a satisfiability decision procedure to determine if Π_n is satisfiable
4. If Π_n is satisfiable, a plan is extracted from the satisfying assignment of Π_n
5. If Π_n is unsatisfiable, n can be incremented.

Properties of the encoding Π_n

all possible solution -> 2^n

Given	a **planning problem** $\Pi = \langle I, \mathcal{A}, G \rangle$ and a bound n
build	a **propositional formula** Π_n
such that	**any model** of Π_n corresponds to a **valid plan** of Π, and
such that	**any valid plan with bound** n of Π corresponds to a **model** of Π_n.

Agenda

1. What and Why

2. Planning as propositional satisfiability
   - Intuition
   - From a planning problem Π to a propositional wff Π_n
   - Encoding based on the Classical Frame Axioms
   - Encoding based on the Explanatory Frame Axioms
   - Remarks on the encodings
   - Encoding the Initial and Goal states
   - Constructing the propositional wff Π_n

3. Some references

From a planning problem Π to a propositional wff Π_n

Running example: TSP (very simplified)

- Objects:
  - locations x, y
- Predicates: $At(?loc)$
- Operators: $move(?loc_1, ?loc_2)$
  - $pre := At(?loc_1)$
  - $eff := At(?loc_2) \land \neg At(?loc_1)$
- Initial state: robot starts at x
- Goal state: robot must visit y

From a planning problem Π to a propositional wff Π_n

Exercise: write a First Order, STRIPS-style formalization of the TSP problem introduced in the previous slide.

From Π to a propositional STRIPS representation

We want to go from a FOL description of Π with operators characterized by

- (finitely many) **predicate** symbols
- applied to (finitely many) **objects**

to a propositional representation.

Solution: **grounding**, i.e., translating a task from a **FOL** to a **grounded/propositional** representation by

- computing all valid **instantiations** that assign objects to the variables of predicates and operators, and
- using **one Boolean variable** to capture each grounded predicate

From Π to a propositional STRIPS representation

Exercise: consider our running example. Compute all instantiations of the predicate $At(?loc)$.

AT(x)
AT(y)

Exercise: consider our running example. Compute all instantiations of the action $move(?loc_1, ?loc_2)$.

M(x,x) M(y,y)
M(x,y)
M(y,x)

P:AT(y)
E:AT(y)
D:AT(y)

Encoding STRIPS states as propositional wffs

Exercise: Write a propositional formula that encodes the state $\{At(x)\}$ of our running example.

sono 2

AT(X) $\wedge$ $\neg$ AT(Y)

1- AT(X)=T AT(y)=T
2- AT(X)=T AT(y)=F

Question: does *At_x* completely define the state?

- ⇝ **closed-world assumption** of STRIPS representation of states:

  what is not in the state, is false.

- ⇝ must encode everything (both what is true and what is false) in the formula!

Single STRIPS transitions as propositional wffs

Given an action $A = \langle Pre(A), Eff(A), Del(A) \rangle$ and a state S,

1. A is **applicable** in S only if its precondition $Pre(A)$ are satisfied by S,
2. if A is executed, the effects of A hold in the resulting state:
   - positive effects $Eff(A)$ hold in $Res(A, S)$,
   - the negation of the fluents in $Del(A)$ hold in $Res(A, S)$,
   - the fluents not in $Eff(A) \cup Del(A)$ keep the value they had in S

   i.e., (assuming $Eff(A) \cap Del(A) = \emptyset$)

$$Res(A, S) = S \cup Eff(A) \setminus Del(A) = S \setminus Del(A) \cup Eff(A).$$

Single STRIPS transitions as propositional wffs

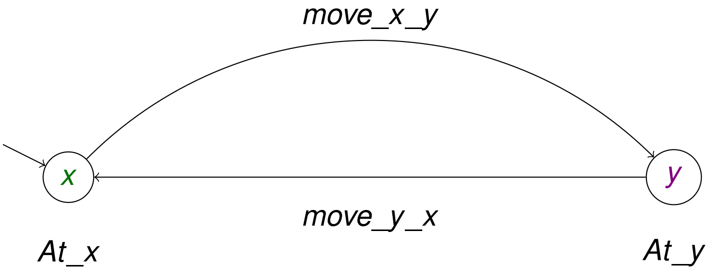
Thus,

1. the preconditions are a necessary condition for A being true in S,
2. A is a sufficient condition for the effects of A to be true in $Res(A, S)$,
3. A is a sufficient condition for the fluents not in $Eff(A) \cup Del(A)$ to keep the value they had in S.

M(x,y) -> AT(x)
M(x,y) -> AT(y) $\wedge$ $\neg$ AT(x)'

Single STRIPS transitions as propositional wffs

Question: specify a formula that encodes a valid execution of the action $move(?loc_1, ?loc_2)$



Single STRIPS transitions as propositional wffs

Question: does the formula

$$(move\_x\_y \Rightarrow At\_x) \land (move\_x\_y \Rightarrow \neg At\_x \land At\_y)$$

encode a valid transition?

Single STRIPS transitions as propositional wffs

What we actually need is a formula to assert that

- At_x holds **before** executing the action
- $\neg At\_x \land At\_y$ holds **after** executing the action

How?

1. create two copies of state variables
2. rename each set of variables with a time-index:
   - the first index refers to when the action is executed and the preconditions must hold.
   - the second index refers to when the effects must hold if the action is executed.

Example: $At\_x_i \land \neg At\_x_{i+1}$ models the fact that r is in location x at time i but not at time $i + 1$.

Multiple STRIPS transitions as propositional formulas

If $\mathcal{A}$ and $\mathcal{F}$ are the set of actions and fluents in Π, by

- making n copies $\mathcal{A}_0, \ldots, \mathcal{A}_{n-1}$ of the set of actions $\mathcal{A}$,
- making $n+1$ copies $\mathcal{F}_0, \ldots, \mathcal{F}_n$ of the set of fluents $\mathcal{F}$

and then

- encoding each possible transition from step i to step $i+1$ ($i = 0, \ldots, n-1$), as a propositional wff

$$TR(\mathcal{F}_i, \mathcal{A}_i, \mathcal{F}_{i+1}),$$

we can encode n transitions with the formula

I(Io) and $$\bigwedge_{i=0}^{n-1} TR(\mathcal{F}_i, \mathcal{A}_i, \mathcal{F}_{i+1}).$$ and G(Fn)

Exercise

Exercise: specify a formula that encodes a valid execution of the action resulting from $move(?loc_1, ?loc_2)$ with two locations x, y

The Frame Problem

In the previous exercise we we have seen that imposing that

- the preconditions are a necessary condition for executing an action, and
- the execution of an action is a sufficient condition for its effects to hold in the resulting state

defines the valid transitions, assuming an action is executed.

Is it always enough?

- specifying only which conditions are changed by the actions does not entail that all other conditions are not changed
- need to make sure fluents do not change values arbitrarily!

This is the **frame problem**, historical problem in AI!

Exercise TSP (simplified)

```
(define (domain tsp)
(:predicates
(at ?x)
(visited ?x)
(connected ?x ?y)
)

(:action move
:parameters (?x ?y)
:precondition (and (at ?x) (connected ?x ?y))
:effect (and (at ?y) (visited ?y) (not (at ?x)))
)
)
```

Exercise TSP (simplified)

```
(define (problem tsp-2)
(:domain tsp)
(:objects P1 P2 P3)
(:init
(at P1)
(connected P1 P2)
(connected P2 P3)
)
(:goal
(and
(visited P2)
(visited P3))
)
)
```

...asserting that if an action is executed at time i then its preconditions and effects must hold at time i and $i+1$ respectively, is not enough!

Objective

Given a set of actions $\mathcal{A}$ and a set of fluents $\mathcal{F}$, our goal is to build a propositional formula $TR(\mathcal{F}_i, \mathcal{A}_i, \mathcal{F}_{i+1})$ in the propositional variables $\mathcal{F}_i \cup \mathcal{A}_i \cup \mathcal{F}_{i+1}$ such that

1. for each transition from a state $S_i \subseteq \mathcal{F}$ to a state $S_{i+1} \subseteq \mathcal{F}$ because of the execution of an action $A \in \mathcal{A}$, there is a corresponding model of $TR(\mathcal{F}_i, \mathcal{A}_i, \mathcal{F}_{i+1})$, and
2. for each model of $TR(\mathcal{F}_i, \mathcal{A}_i, \mathcal{F}_{i+1})$ there is a corresponding transition from a state $S_i \subseteq \mathcal{F}$ to a state $S_{i+1} \subseteq \mathcal{F}$ because of the execution of an action $A \in \mathcal{A}$.

Two solutions:

1. Encoding based on Classical Frame Axioms, and
2. Encoding based on Explanatory Frame Axioms.

Agenda

1. What and Why

2. Planning as propositional satisfiability
   - Intuition
   - From a planning problem Π to a propositional wff Π_n
   - Encoding based on the Classical Frame Axioms
   - Encoding based on the Explanatory Frame Axioms
   - Remarks on the encodings
   - Encoding the Initial and Goal states
   - Constructing the propositional wff Π_n

3. Some references

Encoding based on the Classical Frame Axioms

In the encoding based on the classical frame axioms, $TR(\mathcal{F}_i, \mathcal{A}_i, \mathcal{F}_{i+1})$ consists of:

1. **Universal Axioms**: for each action $A \in \mathcal{A}$

$$
\begin{array}{rcl}
A_i & \rightarrow & \bigwedge_{F \in Pre(A)} F_i \\
A_i & \rightarrow & \bigwedge_{F \in Eff(A)} F_{i+1} \\
A_i & \rightarrow & \bigwedge_{F \in Del(A)} \neg F_{i+1}
\end{array}
$$

   Cardinality? number of fluents mentioned in an operator (usually a small number).

By "Cardinality" here we mean the number of clauses that are produced.

Encoding based on the Classical Frame Axioms

Exercise: write the universal axioms for the TSP problem.

Encoding based on the Classical Frame Axioms

2. **Classical Frame Axioms**: for each action $A \in \mathcal{A}$
   - $A_i \rightarrow \bigwedge_{F \notin Eff \cup Del} F_{i+1} \leftrightarrow F_i$
   - **Cardinality?**
3. **At-least-one Axioms**:
   - $\bigvee_{a \in \mathcal{A}} A_i$
   - **Cardinality?** =1

Encoding based on the Classical Frame Axioms

Questions:

1. Do we always need CFA?

2. Do we always need ALO axiom?

3. Do we need AMO axioms?

Encoding based on the Classical Frame Axioms

Exercise: write classical frame axioms for the TSP.

Agenda

1. What and Why
2. Planning as propositional satisfiability
   - Intuition
   - From a planning problem Π to a propositional wff Π_n
   - Encoding based on the Classical Frame Axioms
   - Encoding based on the Explanatory Frame Axioms
   - Remarks on the encodings
   - Encoding the Initial and Goal states
   - Constructing the propositional wff Π_n
3. Some references

Encoding based on the Explanatory Frame Axioms

In the encoding based on the classical frame axioms, $TR(\mathcal{F}_i, \mathcal{A}_i, \mathcal{F}_{i+1})$ consists of:

1. **Universal Axioms**: for each action $A \in \mathcal{A}$

$$
\begin{array}{lcl}
A_i & \rightarrow & \bigwedge_{F \in Pre(A)} F_i \\
A_i & \rightarrow & \bigwedge_{F \in Eff(A)} F_{i+1} \\
A_i & \rightarrow & \bigwedge_{F \in Del(A)} \neg F_{i+1}
\end{array}
$$

Cardinality? number of fluents mentioned in an operator (usually a small number).

||+||+||

Encoding based on the Explanatory Frame Axioms

2. **Explanatory Frame Axioms:** for all fluents $F \in \mathcal{F}$,
   - $(F_i \wedge \neg F_{i+1}) \rightarrow \bigvee_{A \in \mathcal{A}: F \in Del(A)} A_i$
   - $(\neg F_i \wedge F_{i+1}) \rightarrow \bigvee_{A \in \mathcal{A}: F \in Eff(A)} A_i$
   - **Cardinality?**

2*|F|

Encoding based on the Explanatory Frame Axioms

3. **Conflict Exclusion Axioms**: for all $A, A' \in \mathcal{A}$ such that
   - $A \neq A'$ and
   - $Pre(A) \cap Del(A') \neq \emptyset$ or
   - $Pre(A') \cap Del(A) \neq \emptyset$

   we enforce
   - $\neg(A_i \wedge A'_i)$
   - **Cardinality?**

MAB: P: ATA
E: ATB
D: ATA
MAC: P: ATA
E: ATC
D: ATA

foto 24/04

Encoding based on the Explanatory Frame Axioms

Questions:

- Do we always need EFA axioms?
- Do we always need CEA axioms?

Encoding based on the Explanatory Frame Axioms

Exercise: write explanatory frame axioms for the TSP.

Encoding based on the Explanatory Frame Axioms

Exercise: write mutex axioms for the TSP.

Agenda

1 What and Why

2 Planning as propositional satisfiability
- Intuition
- From a planning problem Π to a propositional wff Π_n
- Encoding based on the Classical Frame Axioms
- Encoding based on the Explanatory Frame Axioms
- Remarks on the encodings
- Encoding the Initial and Goal states
- Constructing the propositional wff Π_n

3 Some references

Remarks on the encodings

- The classical encoding allows only for **sequential** execution of actions;
  - Exercise: explain why
- The explanatory encoding allows for the **parallel** execution of non-conflicting actions;
  - Exercise: explain why
- The explanatory encoding leads to far **more efficient** planning systems: a plan with n actions requires fixing the bound to
  - n in the classical encoding, and
  - $\leq n$ (but often it is $\ll n$) in the explanatory encoding.

Agenda

1. What and Why

2. Planning as propositional satisfiability
   - Intuition
   - From a planning problem Π to a propositional wff Π_n
   - Encoding based on the Classical Frame Axioms
   - Encoding based on the Explanatory Frame Axioms
   - Remarks on the encodings
   - Encoding the Initial and Goal states
   - Constructing the propositional wff Π_n

3. Some references

Encoding Initial and Goal states

Consider a planning problem Π and an horizon n.

- **Initial State Axioms**: The initial state $I \subseteq \mathcal{F}$ is encoded as
  - $I(\mathcal{F}_0) = \bigwedge_{F \in I} F_0 \wedge \bigwedge_{F \notin I} \neg F_0$;
  - **Cardinality?**
- **Goal State Axioms** : The formula encoding the goal state G is
  - $G(\mathcal{F}_n) = \bigwedge_{F \in G} F_n$;
  - **Cardinality?** small.

Agenda

1 What and Why

2 Planning as propositional satisfiability

- Intuition
- From a planning problem Π to a propositional wff Π_n
- Encoding based on the Classical Frame Axioms
- Encoding based on the Explanatory Frame Axioms
- Remarks on the encodings
- Encoding the Initial and Goal states
- Constructing the propositional wff Π_n

3 Some references

Putting everything together

Let $TR(\mathcal{F}_i, \mathcal{A}_i, \mathcal{F}_{i+1})$ be defined using either the classical or the explanatory frame axioms.

$$\Pi_n = I(\mathcal{F}_0) \wedge \bigwedge_{i=0}^{n-1} TR(\mathcal{F}_i, \mathcal{A}_i, \mathcal{F}_{i+1}) \wedge G(\mathcal{F}_n)$$

Properties: For any planning problem Π and bound n,

1. **any model** of Π_n corresponds to a **valid plan** of Π, and
2. **any valid plan with bound** n of Π corresponds to a **model** of Π_n.

Recap

Formulate a **planning problem** Π as a **SAT problem** by:

1. Fixing a bound n on the "length" of the solution we are looking for
2. Building a formula Π_n that encodes the planning problem with bound n
3. Calling a satisfiability decision procedure to determine if Π_n is satisfiable
4. If Π_n is satisfiable, a plan is extracted from the satisfying assignment of Π_n
5. If Π_n is unsatisfiable, n can be incremented.

But ...

…what happens if Π is not solvable?

…or, how can we fix an upper bound on n?

Reachability diameter

How large could n be?

- we would like it to be big enough to visit all reachable states but...
- we would like to be able to solve the resulting formula!

We call **reachability diameter** of Π the minimal number of steps required for reaching all reachable states.

Question: consider a planning problem Π whose states are defined by a set of Boolean variables F. How many steps do we need to visit all states in the worst case?

Reachability diameter

Clearly, setting $n = 2^{|\mathcal{F}|}$ is not an option:

$\rightsquigarrow$ solver would choke and die for any non-trivial problem

Solution:

- sacrifice completeness for practical viability
- start with $n = 0$ and increase until either goal is reached or upper bound is reached
- if Π_n is UNSAT within n, a plan may still exist for longer horizons.

References

Want to know more about Planning as SAT?

Have a look at **The Handbook of Satisfiability**, there's a full chapter on Planning as SAT!

Additionally, have a look at the following link: it provides useful resources for Planning as SAT (and more)

`https://users.aalto.fi/~rintanj1/satplan.html`