PLANNING AS SATISFIABILITY

HENRY KAUTZ & BART SELMAN

Presented by Shane Racey

INTRO OF SATISFIABILITY

- Satisfiability (SAT): determining whether a given logical formula can be assigned truth values to make the entire formula true
- Recent success with these problems
  - Expressed as sets of propositional clauses
  - More flexible and accurate than deductive approach

Wumpus in CNF

Knowledge Base:

$(\neg F \lor PA2 \lor PB1) \land$
$(F \lor \neg PA2) \land$
$(F \lor \neg PB1) \land$
$(\neg BB1 \lor PB2 \lor PC1) \land$
$(BB1 \lor \neg PB2) \land$
$(BB1 \lor \neg PC1)$

Variables:

$BA1 = F$
$BB1 = ?$
$PA2 = ?$
$PB1 = ?$
$PB2 = ?$
$PC1 = ?$

Each clause is a disjunction. Only one variable in it needs to be true. As soon as one variable in the clause is T, the whole clause is T.

Wumpus in CNF

Knowledge Base:

$T \wedge$
$T \wedge$
$T \wedge$
$T \wedge$
$T \wedge$
T

Variables:

$BA1 = F$
$BB1 = F$
$PA2 = F$
$PB1 = F$
$PB2 = F$
$PC1 = F$

This model satisfies the expression. The Wumpus World rules allow at least one possible world.

PLANNING AS DEDUCTION

- First-order logic – using predicates
- Initial conditions + sequence of actions = goal conditions

- Blocks world two step example

$$
\begin{aligned}
&\exists x_1, y_1, z_1, x_2, y_2, z_2.\\
&\qquad on(A, B, 1) \wedge on(B, Table, 1) \wedge clear(A, 1) \wedge\\
&\qquad move(x_1, y_1, z_1, 1) \wedge move(x_2, y_2, z_2, 2) \supset\\
&\qquad\qquad on(B, A, 3)
\end{aligned}
$$

ANOMALOUS MODELS

- Model interpretations that satisfy the SAT formulation
- Does not correspond to valid solutions in the original planning problem
- When encoding a planning problem as a SAT instance, each time step and action is represented using boolean variables

WORLD CHANGES BUT NO ACTIONS OCCUR

$$\left\{ \begin{array}{l} on(A, B, 1), on(B, Table, 1), clear(A, 1), \\ on(B, A, 2), on(B, A, 3), clear(Table, 1), \\ clear(Table, 2), clear(Table, 3) \end{array} \right\}$$

FALSE PRECONDITIONS BUT ACTION STILL PERFORMED

$$\left\{ \begin{array}{l} on(A, B, 1), on(B, Table, 1), clear(A, 1), \\ move(B, Table, A, 1), on(B, A, 2), on(B, A, 3) \end{array} \right\}$$

PLANNING AS SATISFIABILITY

Any model of the axioms corresponds to a valid plan

$$on(A, B, 1) \wedge on(B, Table, 1) \wedge clear(A, 1) \wedge on(B, A, 3)$$

Block A is on top of block B at time 1

Block B is on the table at time 1

There is nothing on top of block A at time 1

At time 3 we want block B to be on top of block A

FALSE PRECONDITIONS AXIOM

Rule out that an action executes despite that its preconditions are false

$$\forall x, y, z, i.\ move(x, y, z, i) \supset (clear(x, i) \wedge clear(z, i) \wedge on(x, y, i))$$

ONE ACTION AT A TIME AXIOM

So that multiple actions don't happen simutaneously

$$\forall x, x', y, y', z, z', i.\ (x \neq x' \vee y \neq y' \vee z \neq z') \supset \neg move(x, y, z, i) \vee \neg move(x', y', z', i)$$

ACTION OCCURS AT EVERY TIME AXIOM

Can also introduce a "do nothing" action if wanted

$$\forall i < N.\ \exists x, y, z.\ move(x, y, z, i)$$

HOW CAN A PLANNING PROBLEM BE TRANSLATED INTO A SAT PROBLEM?

INITIAL SETUP USING PREDICATES

- Wumpus starts in room A
- Wumpus wants the cake in room C
- Actions
  - Move(A, B)
  - Move(B, C)
  - Move(B, A)
  - Move(C, B)
- Constraints
  - Move one room at a time
  - Valid preconditions
  - Has 2 time steps

PROPOSITIONAL ENCODING

- Variable states
  - W(A, t)
  - W(B, t)
  - W(C, t)
- Action states
  - Move(A, B, t)
  - Move(B, C, t)

ENCODE CONSTRAINTS INTO BOOLEAN CLAUSES

- Wumpus starts in room A
- Wumpus wants the cake in room C
- Actions
  - Move(A, B)
  - Move(B, C)
  - Move(B, A)
  - Move(C, B)

- W(A, 0)
- W(C, 2)
- ¬(W(A, t)∧W(B, t)) ∧ ¬(W(A, t)∧W(C, t)) ∧ ¬(W(B, t)∧W(C, t))
- Move(A, B, t) -> (W(A, t) ∧ W(B, t+1))
- Move(B, C, t) -> (W(B, t) ∧ W(C, t+1))
- (W(A, t) ∧ ¬Move(A, B, t)) -> W(A, t+1)

SOLVE WITH A SAT SOLVER

- Could use GSAT
- Possible solution could be
  - Move(A, B, 0) = True
  - Move(B, C, 1) = True

COMPRESSING PREDICATES

Predicates that take 3 or more args are replaced by several that only take 2 args

problem	using *move*			using *object*, *source*, *dest*		
	# props	# clauses	size	# props	# clauses	size
anomaly	127	2,364	5,529	94	375	933
reversal	429	22,418	51,753	215	993	2,533
medium	641	68,533	155,729	244	1,185	3,025
hanoi	1,005	63,049	137,106	288	1,554	3,798
huge	>7,000	>3,500,000	> 8,000,000	996	5,945	15,521

Table 1: Comparison of size of propositional theories using one ternary predicate versus three binary predicates.

EFFICIENCY

- Two algorithms used for solving SAT problems
- Davis-Putnam vs. GSAT
- Additional axioms helped GSAT

problem	vars	size	GSAT	DP
random A	100	1,290	6 sec	2.8 min
random B	140	1,806	14 sec	4.7 hours
random C	500	6,450	1.6 hours	—
coloring A	2,125	168,419	8 hours	—
coloring B	2,250	180,576	5 hours	—
anomaly	94	933	26 sec	0.1 sec
reversal	215	2,533	—	4 sec
medium	244	3,025	—	1.2 sec
hanoi	288	3,798	—	13 hours
huge	996	15,521	—	—

Table 2: Comparison of speed of GSAT versus DP on solving sample coloring, random, and planning satisfiability problems.

problem	original		expanded	
	size	time	size	time
anomaly	933	26 sec	1,325	1.9 sec
reversal	2,533	—	3,889	1.2 min
medium	3,025	—	5,235	1.2 min

Table 3: Improvement in performance of GSAT by adding additional axioms to rule out impossible states.

CONCLUSION

- Formal model based on satisfiability rather than deduction
- Deductive axioms must be strengthened to prevent anomalous models
- How a planning problem is translated into a SAT problem
- Efficiency of SAT algorithm models

THANK YOU