

SATPLAN

Leliane Nunes de Barros

SATPLAN

História: Kautz and Selman, 1992

- Inspiração nos avanços dos algoritmos de satisfazibilidade

Idéia

- Codificar o problema de planejamento como uma grande fórmula lógica do tipo:

Initial-state \wedge ***all-possible-actions*** \wedge ***meta***

- e encontrar uma valoração que satisfaça a fórmula que resulte no plano

SATPLAN

- Codificação baseada no Cálculo de Situações
- Tempo discreto. Para cada codificação de problema deve-se assumir um tamanho (intervalo) para o plano solução.
- Estado Inicial: **Lixo** \wedge **MaosLimpas** \wedge **Silencio**
- Meta: **Jantar** \wedge **Presente** \wedge \neg **Lixo**

SATPLAN: Jantar Surpresa

- Símbolos Proposicionais (*schemas* de fluentes)

lixo(I)

maos_limpas(I)

silencio(I)

jantar(I)

presente(I)

Schemas de ações

cozinhar(I)

embrulhar(I)

carregar_lixo(I)

triturar_lixo(I)

SATPLAN: Jantar Surpresa

- Símbolos Proposicionais

1 lixo(1)

2 maos_limpas(1)

3 silencio(1)

4 jantar(1)

5 presente(1)

6 lixo(2)

7 maos_limpas(2)

8 silencio(2)

9 jantar(2)

10 presente(2)

11 lixo(3)

...

32

TOTAL: conf 22.142

SATPLAN: Axiomas de Efeito

- Ações em i implicam seus efeitos em $i+1$
 - *Schemas* de Axiomas de efeitos
 - cozinhar(I) \rightarrow jantar($I + 1$)
 - embrulhar(I) \rightarrow presente($I + 1$)
 - Efeitos conjuntivos devem ser descritos por uma cláusula para cada efeito
 - carregar_lixo(I) \rightarrow \sim lixo($I + 1$)
 - carregar_lixo(I) \rightarrow \sim maos_limpas($I + 1$)
 - triturar_lixo(I) \rightarrow \sim lixo($I + 1$)
 - triturar_lixo(I) \rightarrow \sim silencio($I + 1$)

SATPLAN: Axiomas de Precondições

- Ações em i implicam em suas pré-condições também em i
 - *Schemas* de Axiomas de pré-condições
 - cozinhar(I) \rightarrow maos_limpas (I)
 - embrulhar(I) \rightarrow silencio(I)

SATPLAN: Axiomas de Persistência

Ações implicam em fluentes que não são afetados por ela

– aparecem sentenças do tipo:

$$A(I) \wedge \text{Fluente}(I) \rightarrow \text{Fluente}(I+1)$$

– que correspondem à cláusula:

$$\sim A(I) \vee \sim \text{Fluente}(I) \vee \text{Fluente}(I+1)$$

SATPLAN: Axiomas de Persistência

$[\text{lixo}(I) \wedge \text{cozinhar}(I)] \rightarrow \text{lixo}(I + 1)$
 $[\sim \text{lixo}(I) \wedge \text{cozinhar}(I)] \rightarrow \sim \text{lixo}(I + 1)$
 $[\text{lixo}(I) \wedge \text{embrulhar}(I)] \rightarrow \text{lixo}(I + 1)$
 $[\sim \text{lixo}(I) \wedge \text{embrulhar}(I)] \rightarrow \sim \text{lixo}(I + 1)$
 $[\text{maos_limpas}(I) \wedge \text{cozinhar}(I)] \rightarrow \text{maos_limpas}(I + 1)$
 $[\sim \text{maos_limpas}(I) \wedge \text{cozinhar}(I)] \rightarrow \sim \text{maos_limpas}(I + 1)$
 $[\text{maos_limpas}(I) \wedge \text{embrulhar}(I)] \rightarrow \text{maos_limpas}(I + 1)$
 $[\sim \text{maos_limpas}(I) \wedge \text{embrulhar}(I)] \rightarrow \sim \text{maos_limpas}(I + 1)$
 $[\text{maos_limpas}(I) \wedge \text{triturar_lixo}(I)] \rightarrow \text{maos_limpas}(I + 1)$
 $[\sim \text{maos_limpas}(I) \wedge \text{triturar_lixo}(I)] \rightarrow \sim \text{maos_limpas}(I + 1)$
 $[\text{silencio}(I) \wedge \text{cozinhar}(I)] \rightarrow \text{silencio}(I + 1)$
 $[\sim \text{silencio}(I) \wedge \text{cozinhar}(I)] \rightarrow \sim \text{silencio}(I + 1)$
 $[\text{silencio}(I) \wedge \text{embrulhar}(I)] \rightarrow \text{silencio}(I + 1)$
 $[\sim \text{silencio}(I) \wedge \text{embrulhar}(I)] \rightarrow \sim \text{silencio}(I + 1)$
 $[\text{silencio}(I) \wedge \text{carregar_lixo}(I)] \rightarrow \text{silencio}(I + 1)$
 $[\sim \text{silencio}(I) \wedge \text{carregar_lixo}(I)] \rightarrow \sim \text{silencio}(I + 1)$

SATPLAN: Axiomas de Persistência

[jantar(I) ^ embrulhar(I)] -> jantar(I + 1)
[~jantar(I) ^ embrulhar(I)] -> ~jantar(I + 1)
[jantar(I) ^ carregar_lixo(I)] -> jantar(I + 1)
[~jantar(I) ^ carregar_lixo(I)] -> ~jantar(I + 1)
[jantar(I) ^ triturar_lixo(I)] -> jantar(I + 1)
[~jantar(I) ^ triturar_lixo(I)] -> ~jantar(I + 1)
[presente(I) ^ cozinhar(I)] -> presente(I + 1)
[~presente(I) ^ cozinhar(I)] -> ~presente(I + 1)
[presente(I) ^ carregar_lixo(I)] -> presente(I + 1)
[~presente(I) ^ carregar_lixo(I)] -> ~presente(I + 1)
[presente(I) ^ triturar_lixo(I)] -> presente(I + 1)
[~presente(I) ^ triturar_lixo(I)] -> ~presente(I + 1)

SATPLAN: continuidade no plano

- Deve ser escolhida uma ação em cada instante
 - Schemas de axiomas de continuidade no plano
- cozinhar(I) v embrulhar(I) v carregar_lixo(I) v triturar_lixo(I)

SATPLAN: plano totalmente ordenado

- Apenas uma ação em cada instante
 - Schemas de axiomas de não paralelismo de ações
 - cozinhar(I) -> ~embrulhar(I)
 - cozinhar(I) -> ~carregar_lixo(I)
 - cozinhar(I) -> ~triturar_lixo(I)
 - embrulhar(I) -> ~cozinhar(I)
 - embrulhar(I) -> ~carregar_lixo(I)
 - embrulhar(I) -> ~triturar_lixo(I)
 - carregar_lixo(I) -> ~cozinhar(I)
 - carregar_lixo(I) -> ~embrulhar(I)
 - carregar_lixo(I) -> ~triturar_lixo(I)
 - triturar_lixo(I) -> ~cozinhar(I)
 - triturar_lixo(I) -> ~embrulhar(I)
 - triturar_lixo(I) -> ~carregar_lixo(I)

Blackbox

Graphplan: usa um algoritmo melhor para instanciar a estrutura proposicional

SATplan: usa algoritmos mais poderosos para busca (extração)

Blackbox: resposta ao desafio lançado no IJCAI'97 Subarao Khambhampati: *Como unificar as 2 abordagens?*