

Planejamento em IA

Blackbox

Graphplan + SATplan

Introdução

- **SATplan**

- métodos para compilar problemas de planejamento em fórmulas proposicionais (CNF) resolvidas por algoritmos SAT sistemáticos e estocásticos.

- **Graphplan**

- planejador de propósito geral baseado nas idéias usadas em algoritmos de grafos. Dado um problema de planejamento, Graphplan constrói e anota um grafo de planejamento, no qual um plano é um tipo de “fluxo” de valores verdade através do grafo.

Essas duas abordagens envolvem técnicas avançadas de busca e satisfação de restrições que revolucionaram a área em meados dos anos 90.

SATplan

- Mostrou que um provador de teoremas geral pode ser competitivo com alguns dos melhores planejadores da época
 - uso de lógica proposicional
 - convenção de representação de tempo e ações
 - uso de novos algoritmos de raciocínio geral (SAT-solvers)

Graphplan e Satplan

- apresentam desempenho comparável para problemas do Mundo dos Blocos
- SATplan é mais rápido em domínios complexos de logística
- Graphplan é mais rápido em vários outros domínios
- [Khambhampati, 97] [Weld, 1998]

Diferenças e semelhanças

Graphplan: recebe um conjunto de operadores Strips e usa algoritmos específicos para encontrar um plano solução

SATplan: recebe como entrada um conjunto de esquemas de axiomas e usa algoritmos gerais para encontrar um plano solução

Kambhampati: define a classe de *planejadores disjuntivos*

Kautz & Selman: mostram como traduzir um grafo de planejamento em uma CNF \implies BlackBox

Semelhanças

Ambas as abordagens trabalham em 2 fases:

- (1) criação de uma estrutura proposicional (CNF ou grafo de planejamento)
- (2) realizam uma busca (sobre atribuição de variáveis ou em caminhos do grafo) restrita pela estrutura proposicional

A estrutura proposicional corresponde a um tamanho fixo de plano e a busca informa se um plano daquele tamanho existe, caso contrário, a estrutura é expandida

Hipótese (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?*

SATplan: estrutura proposicional

SATplan: usa um simplificador lógico geral
(algoritmo de inferência limitada de tempo
polinomial)

Exemplos de regras de inferência de simplificação:

- *Propagação da Cláusula Unitária (Unit Propagation)*
- *Purificação (Purification)*

SATplan: estrutura proposicional

Simplificação

Propagação da Cláusula Unitária

- Se existir uma cláusula *unitária*, existe somente uma atribuição possível para a variável correspondente (restrição unária)
- Exemplo: $(x_1 \vee \sim x_2 \vee x_3)(x_2)(x_1 \vee x_5)(x_2 \vee x_4)$
- Faça $(x_2 = 1)$:
 - elimine a cláusula em que a variável x_2 for positiva
Ex: ~~$(x_2)(x_2 \vee x_4)$~~
 - elimine a variável x_2 quando for negada.
Ex: ~~$(x_1 \vee \sim x_2 \vee x_3)$~~

SATplan: estrutura proposicional

Simplificação

Purificação

- Uma variável *pura* sempre aparece na forma positiva (x_i) ou negativa ($\sim x_i$)
- Faça $x_i = 1$ no caso positivo e elimine todas as cláusulas em que ela aparece
- Faça $x_i = 0$ no caso negativo e elimine essa variável de todas as cláusulas em que ela aparece

Graphplan: estrutura proposicional

Propagação de Mutex

- A estrutura é gerada enquanto um algoritmo de simplificação é aplicado, baseado na computação de mutex (determina que par de ações ou de proposições são mutex)
- Um mutex pode ser representado como uma cláusula “negativa binária”. Isso permite uma “propagação negativa binária”, por exemplo:
dado: $\{\sim p \vee \sim q\}, \{p \vee \sim r\}$
infere: $\{\sim q \vee \sim r\}$

Graphplan: estrutura proposicional

Propagação de Mutex

- Cada aplicação da regra permite que se deduza uma cláusula de negação binária (um mutex)
- O algoritmo mutex não é completo (nem todos os mutex que podem ser inferidos logicamente são descobertos) e termina em tempo polinomial

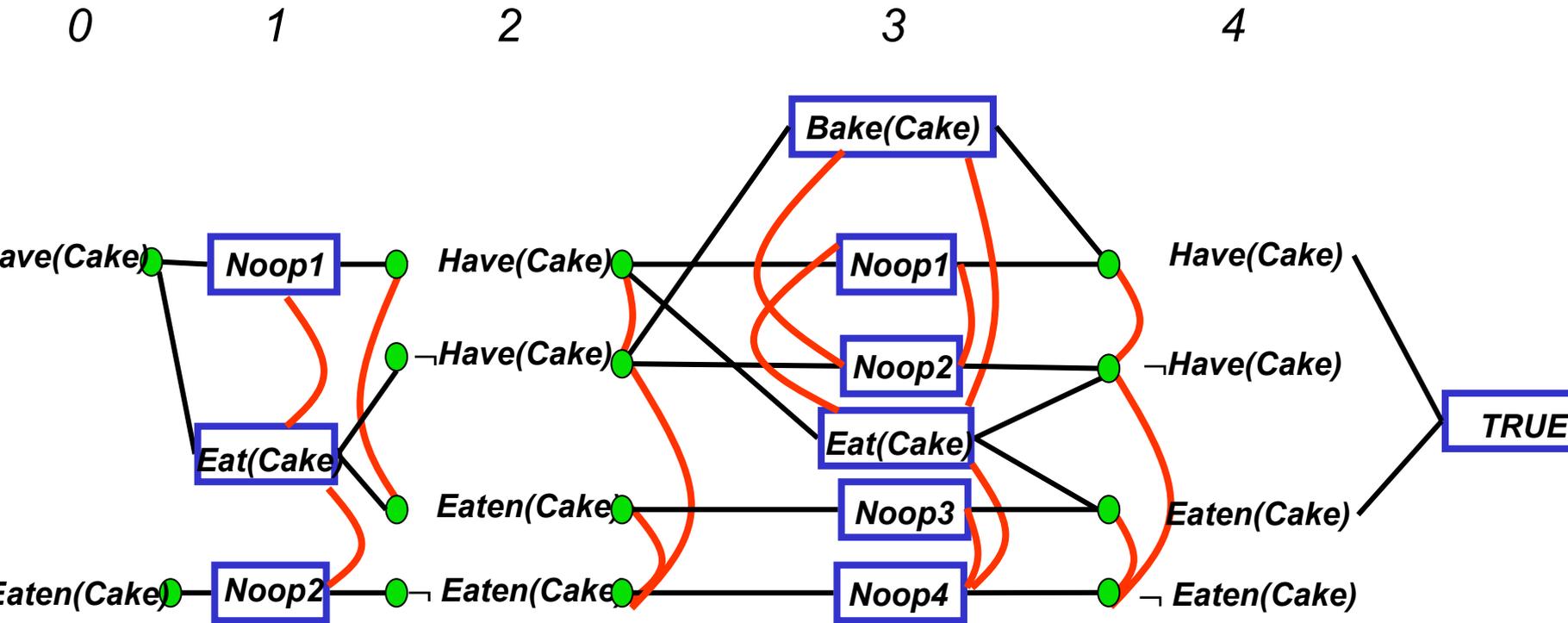
Blackbox

- Combina o melhor das 2 abordagens
- Trabalha em várias fases:
 - (1) é gerado um grafo de planejamento de comprimento k e os mutex são computados
 - (2) O grafo de planejamento é convertido para uma CNF (usando codificação binária negativa)
 - (3) a CNF é simplificada por um algoritmo geral
 - (4) a CNF é resolvida por qualquer SAT-solver
 - (5) se o modelo da CNF é encontrado então ele é convertido para o plano correspondente, caso contrário, k é incrementado e o processo se repete

Blackbox

- Tradução do grafo de planejamento para uma CNF:
 - as relações de mutex são traduzidas diretamente em cláusulas binárias negativas \implies gerando fórmulas mais fáceis de serem resolvidas por um SAT-solver.
 - a construção da CNF é feita a partir do último nível

Grafo de Planejamento



Blackbox

- Inicialmente, criamos proposições para as sub-metas do problema
- Sub-metas do problema são vistas como precondições da ação fictícia TRUE
- Para cada sub-meta devemos selecionar a lista de ações que a adiciona.
- No exemplo:
 - Sub-metas de TRUE são *Eaten_cake(4)* e *Have_cake(4)*

Sub-meta *Eaten_cake(4)*

- Lista de ações que a adiciona:

Eat_cake(3) e *Noop_3(3)*

- para essas ações são criados símbolos proposicionais para o SAT-solver

- **CLÁUSULA tipo1:**

- *se a ação TRUE for executada, devo escolher uma ação que satisfaça cada uma de suas precondições*

$\text{TRUE} \rightarrow \text{Eat_cake}(3) \vee \text{Noop_3}(3)$

- **CLÁUSULA tipo 2:**

- *somente uma das ações precisa ser executada*

$\sim \text{Eat_cake}(3) \vee \sim \text{Noop_3}(3)$

Sub-meta *Have_cake(4)*

- Lista de ações que a adiciona:

Bake_cake(3) e Noop_1(3)

- para essas ações são criados símbolos proposicionais para o SAT-solver

- **CLÁUSULA tipo1:**

- *se a ação TRUE for executada, devo escolher uma ação que satisfaça cada uma de suas precondições*

$\text{TRUE} \rightarrow \text{Bake_cake}(3) \vee \text{Noop_1}(3)$

- **CLÁUSULA tipo 2:**

- *somente uma das ações precisa ser executada*

$\sim \text{Bake_cake}(3) \vee \sim \text{Noop_1}(3)$

Mutex

- **CLÁUSULA tipo 3:**

- *todas as relações de MUTEX das ações que possivelmente adicionam as pre-condições de TRUE, devem ser declaradas*

$$\sim\text{Bake_cake}(3) \vee \sim\text{Eat_cake}(3)$$

$$\sim\text{Noop_1}(3) \vee \sim\text{Eat_cake}(3)$$

Blackbox

- Seleccionamos todas as sub-metas que são precondições das ações possíveis no nível 3:

Bake_cake(3)

Eat_cake(3)

Noop_1(3)

Noop_3(3)

Exemplo

- A lista de precondições de $Eat_cake(3)$ é
 $\{Have_cake(2)\}$
- Criamos proposições para cada elemento dessa lista
- **CLÁUSULA tipo 1:**
 - *se a ação TRUE for executada, devo escolher uma ação que satisfaça cada uma de suas precondições*

$$Eat_cake(3) \rightarrow Noop_1(1)$$

- **CLÁUSULA tipo 2:**
 - *somente uma das ações precisa ser executada*
 - *...neste caso somente $Noop_1(1)$ adiciona $Have_cake(2)$*

Blackbox

- Inclue:
 - mecanismo para geração do grafo de planejamento
 - 1 SAT-solver de busca local: Walksat
 - 2 SAT-solvers de busca sistemática: satz e rel-sat
 - satz: baseado em *forward-checking*
 - rel-sat: emprega *dependency-directed backtracking*

Blackbox

- Meta-controle:
 - executa o Graphplan por 30 s
 - executa o Walksat por 2 min
 - executa o satz por 5 min
- Blackbox introduziu uma nova técnica de SAT: métodos completos de busca com aleatoriedade

Resultados

- Comparação de desempenho entre Graphplan, SATplan e Blackbox [1]

Resultados

- Re-início aleatório melhora muito o desempenho de algoritmos SAT sistemáticos para problemas de planejamento
 - exemplo: domínio de logística. Backbox: corte de 20 a 200 backtrackings (redução de 1/2 do tempo). SATplan: 16 backtrackings (redução de 1.8 hs para 1.6 min)

Resultados

- Codificação SAT se torna muito grande em domínios sequenciais com muitos operadores
 - ex: mundo dos blocos com um braço (grafo de planejamento deve conter tantas camadas quanto o no. de passos do plano)
 - para n blocos $O(n^2)$ ações e $O(n^4)$ mutex por nível.
 - A tradução de um problema de 15 blocos e de 28 passos de plano solução contém cerca de 2.5 milhões de cláusulas
 - É preciso fazer uma compressão da CNF: adicionar mais conhecimento sobre a estrutura do problema na CNF)

Resultados

- Para problemas muito restritivos em que a extração do plano é o gargalo computacional do Graphplan, a tradução do grafo de planejamento em SAT e uso de um SAT-solver apresenta um desempenho excepcionalmente maior

Propostas de extensão

- Uso de conhecimento específico do domínio como restrições de estado
- Uso de novas técnicas polinomiais de simplificação

Bibliografia

- [1] Unifying SAT-based and Graph-based Planning. H. Kautz & Bart Selman. Workshop on Logic-Based Artificial Intelligence, 1999
- [2] BLACKBOX: a New Approach to the Application of Theorem Proving to Problem Solving. H. Kautz & Bart Selman. Workshop notes of the Workshop Planning as Combinatorial Search. AIPS, 1998.