

*Planejamento
em Inteligência Artificial*

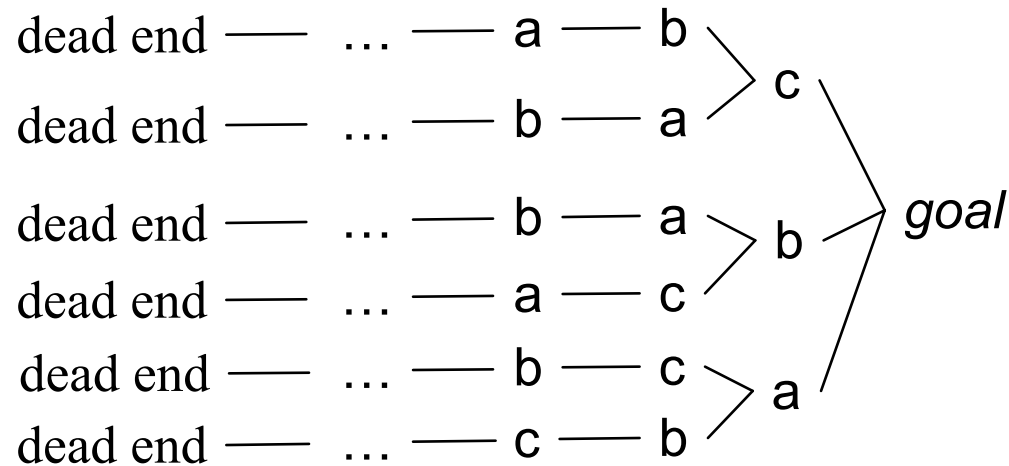
**Capítulo 5
Planejamento no
espaço de planos**

Leliane Nunes de Barros

MAC5788
IME-USP 2005

Motivação

- Problema com a busca de um caminho no grafo Σ de um sistema de transição de estados (planejamento no espaço de estados)
 - ◆ Em alguns casos nós podemos tentar diferentes ordenações de um mesmo conjunto de ações antes de perceber que não existe solução



- *Estratégia de compromisso mínimo (least-commitment strategy)*: não fazer compromissos com ordenações, instanciações, etc., até que seja necessário

Tópicos dessa aula

- Planejamento no espaço de planos: idéia básica
- Metas não satisfeitas (*open goals*)
- Ameaças (*threats*)
- O algoritmo PSP
- Exemplo
- Comentários

Planejamento no espaço de planos: idéia básica

- Busca dirigida por metas e sub-metas (pré-condições de operadores no plano)
- Cada nó do espaço de busca é um *plano parcialmente especificado* com
 - » um conjunto de ações parcialmente instanciadas
 - » um conjunto de restrições
- Arcos são operações de refinamento de planos
 - ◆ e.g.: estabelecer metas ou remover uma inconsistência
- Inicia com um nó que corresponde a um plano vazio
- Planejar: fazer refinamentos sucessivos, até obtermos um plano solução, ou completamente especificado que satisfaz as metas, segundo o ***princípio de compromisso mínimo***

Planejamento no espaço de planos: idéia básica

- Tipos de restrições:

- ◆ *restrição de precedência*:
a deve preceder *b*

- ◆ *restrições de unificação*:

- » restrições de desigualdade, e.g., $v_1 \neq v_2$ ou $v \neq c$

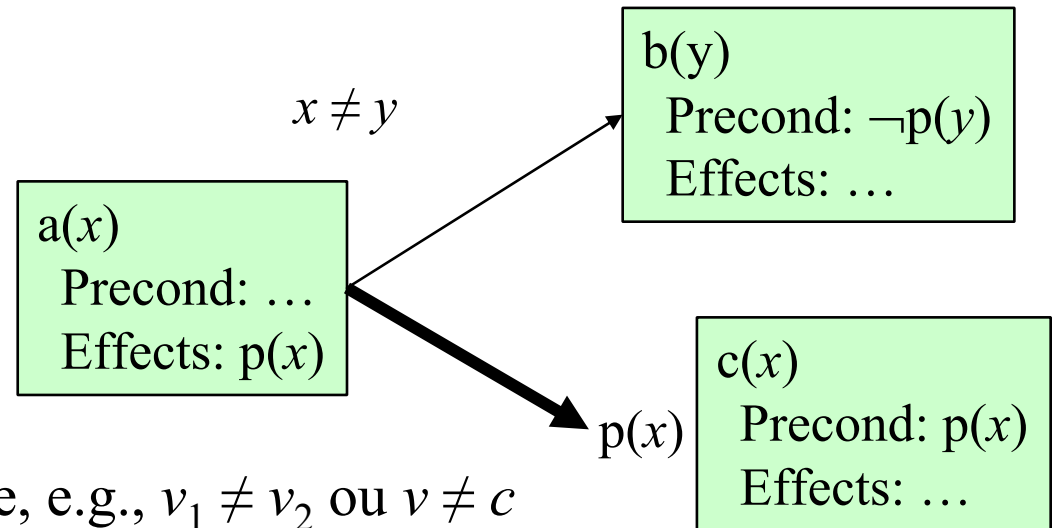
- » restrições de igualdade (e.g., $v_1 = v_2$ or $v = c$) ou substituições

- ◆ *vínculo causal (causal link)*:

- » use ação *a* para estabelecer a pré-condição (ou sub-meta) *p* necessária pela ação *b*

- » a ação *a* é chamada de fornecedor ou ação que adiciona a sub-meta *p*

- » a ação *b* é chamada de consumidor ou ação que necessita da sub-meta *p*



Ordem topológica

- As seqüências de ações são ordens topológicas da relação de ordem parcial \prec
- Pode existir um número exponencial de seqüências de ações
- Uma ordem parcial define um grafo dirigido
- Uma ordem parcial é consistente se grafo é não-cíclico

Plano parcial

- Plano parcial $\Pi = (A, \prec, B, L)$ tal que:
 - ◆ $A = \{a_1, \dots, a_k\}$ é um conjunto de operadores de planejamento parcialmente instanciados (incluindo 2 ações *dumies*: a_0 e a_∞)
 - ◆ \prec é um conjunto de restrições de ordem sobre A da forma $(a_i \prec a_j)$
 - ◆ B é um conjunto de restrições de unificação entre as variáveis de ações em A da forma $x = y$, $x \neq y$ ou $x \in D_x$
 - ◆ L é um conjunto de vínculos causais da forma $(a_i \xrightarrow{p} a_j)$ tal que a_i e a_j são ações em A ; a restrição $(a_i \prec a_j)$ pertence a \prec ; a proposição p é um efeito de a_i e uma precondição de a_j ; as restrições de unificação das variáveis em a_i e a_j pertencem a B .

Plano solução

- Plano parcial $\Pi = (A, \prec, B, L)$ é solução de um problema de planejamento P se:
 - ◆ suas restrições de ordem \prec e restrições de unificação B são consistentes
 - ◆ toda seqüência de ações totalmente instanciada e ordenada satisfazendo \prec e B é uma seqüência que define um caminho no sistema de transição de estados Σ e do estado inicial s_0 , que corresponde aos efeitos da ação a_0 a um estado contendo todas as proposições meta em g dadas pelas pré-condições de a_∞
- Essa definição não fornece teste computável para verificar planos: não é possível verificar todas as seqüências instanciadas de A .
- Como saber se temos uma solução?
 - ◆ eliminando todas as **falhas** no plano

Metas não-estabelecidas (*open goals*)

- Falha I: uma ação a tem uma pre-condição p que ainda não decidimos como estabelecer (satisfazer)

- Resolvendo a falha:

- ◆ encontrar uma ação b

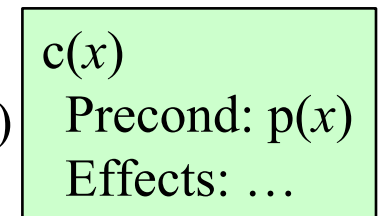
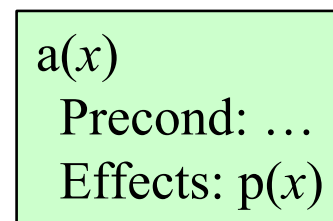
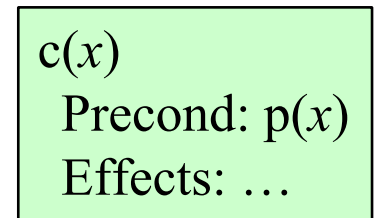
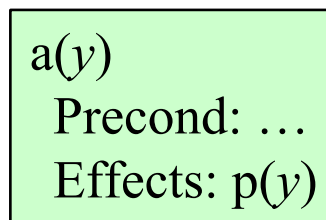
- (que já pertença ao plano ou uma nova ação que será inserida)

- ◆ que pode ser usada para estabelecer p

- pode preceder a e produzir p

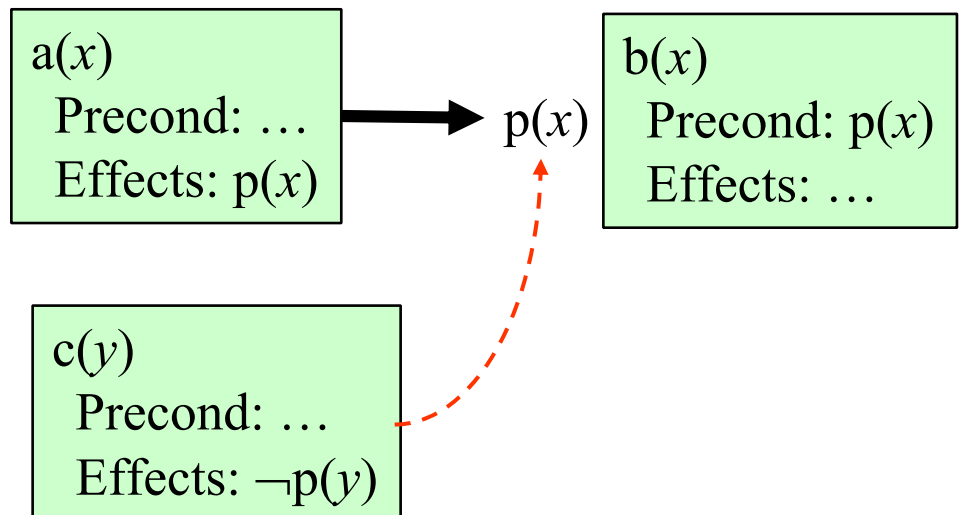
- ◆ instanciar variáveis

- ◆ criar um *vínculo causal*



Ameaças (threats)

- Falha II: uma interação de eliminação de condição
 - ◆ Ação a estabelece uma condição (por exemplo, $p(x)$) para ação b
 - ◆ Outra ação c é capaz de eliminar a condição $p(x)$
- Resolvendo a falha:
 - ◆ impor uma restrição para prevenir que c elimine $p(x)$
- Três possibilidades:
 - ◆ fazer com que b preceda c
 - ◆ fazer com que c preceda a
 - ◆ restringir variáveis para prevenir que c elimine $p(x)$



Comparação

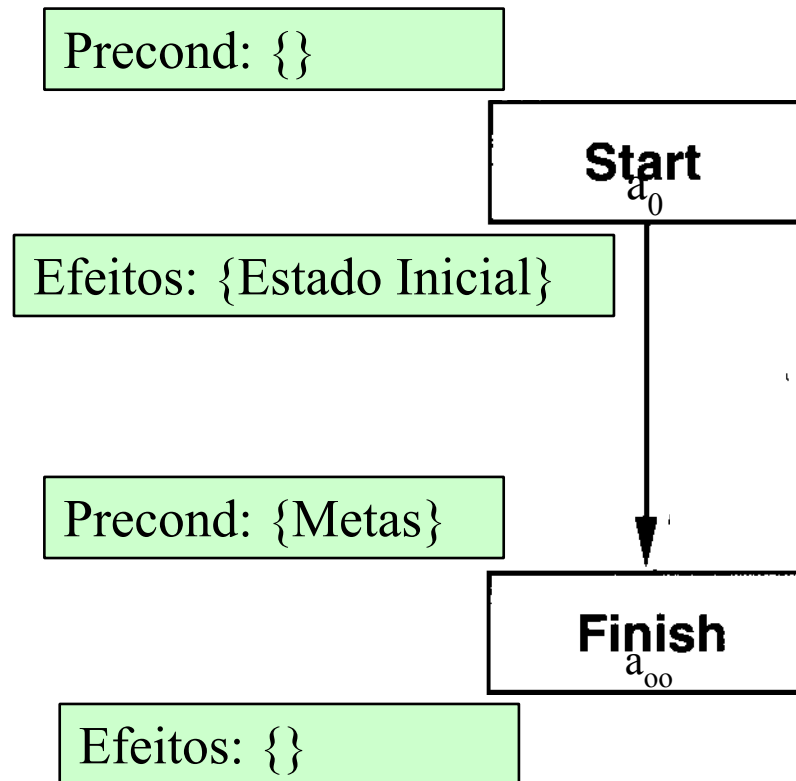
planejar no espaço de estados

- no espaço de busca: nós são estados e arestas são ações
- estrutura de plano: uma sequência de ações
- refinamento de plano: adição de ações (no fim ou início do plano)
- definição de plano solução: se há estados s_0, s_1, \dots, s_n tal que
 - » $\gamma(s_0, a_1) = s_1$
 - » $\gamma(s_1, a_2) = s_2$
 - » ...
 - » $\gamma(s_{n-1}, a_n) = s_n$
 - » s_n *satisfies* g

planejar no espaço de planos

- no espaço de busca: nós são planos parciais e arestas são operações de refinamento do plano
- estrutura de plano: usa uma estrutura mais geral do que uma sequência de ações, incluindo:
 - » ações não totalmente instanciadas
 - » restrições de ordem
 - » restrições de unificação
 - » vínculos causais
- refinamento de plano: adição de ações, adição de restrições de ordem e de *binding*; adição de vínculos causais
- **não possui uma representação explícita de estado do mundo !!!**
- definição de um plano solução: quando não existirem mais falhas no plano, sem inconsistências.

Plano Inicial



The PSP Procedure

```
PSP( $\pi$ )
   $flaws \leftarrow \text{OpenGoals}(\pi) \cup \text{Threats}(\pi)$ 
  if  $flaws = \emptyset$  then return( $\pi$ )
  select any flaw  $\phi \in flaws$ 
   $resolvers \leftarrow \text{Resolve}(\phi, \pi)$ 
  if  $resolvers = \emptyset$  then return(failure)
  nondeterministically choose a resolver  $\rho \in resolvers$ 
   $\pi' \leftarrow \text{Refine}(\rho, \pi)$ 
  return(PSP( $\pi'$ ))
end
```

- PSP é correto e completo

PSP

- *OpenGoals(π)*: encontra todas as sub-metas do plano π não estabelecidas por um vínculo causal (*agenda de metas*). Para toda ação adicionada ao plano, suas precondições vão para a *agenda* e a sub-meta para qual a ação faz um vínculo causal é excluída da agenda
- *Threats(π)*: encontra todas as ações que ameaçam algum vínculo causal. Isso é feito testando todas as triplas de ações em $\pi \implies O(n^3)$, onde n é o número atual de ações em π . Um processo incremental é mais eficiente. Para cada novo vínculo causal, todas as outras ações podem ser analisadas em $O(n)$

PSP

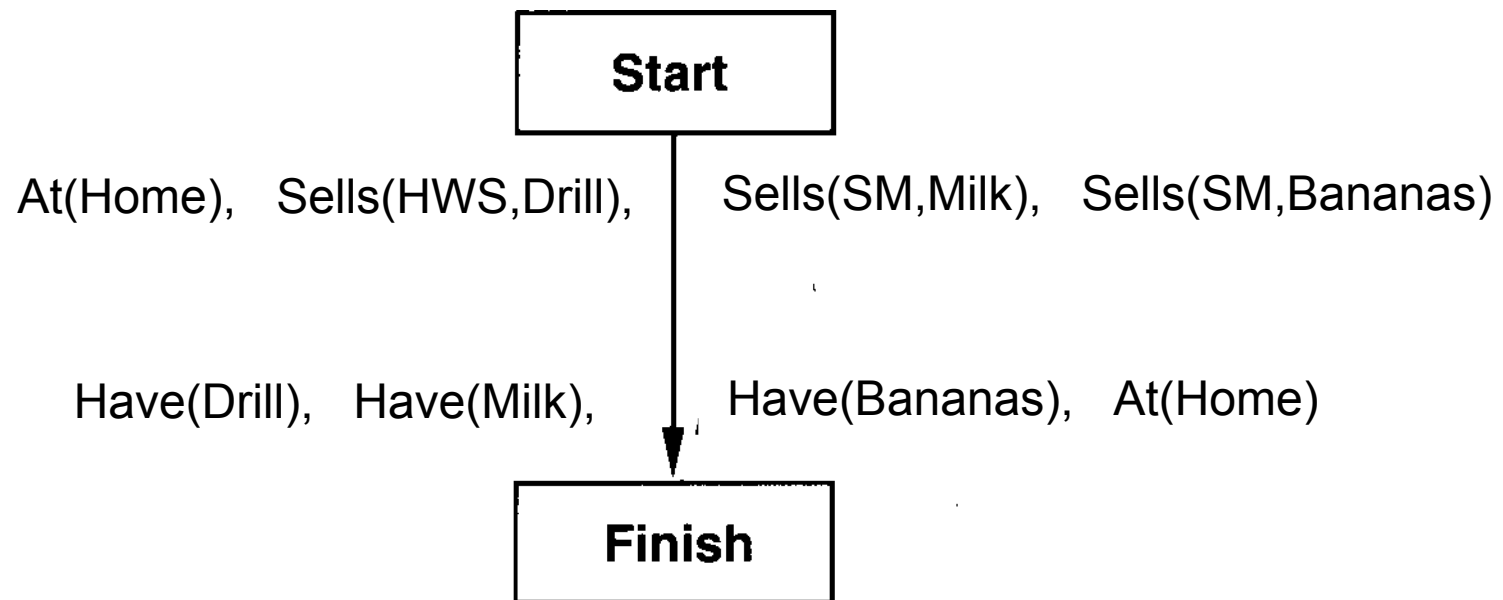
- $\text{Resolve}(\phi, \pi)$: encontra uma maneira de resolver uma falha ϕ .
 - ◆ Se ϕ é uma sub-meta para uma pré-condição p de uma ação a_j então sua solução é :
 - › criar um vínculo causal com uma ação existente no plano
 - › adicionar uma nova ação que pode fornecer p
 - ◆ Se ϕ é uma ameaça a um vínculo causal $(a_i \xrightarrow{p} a_j)$ de uma ação a_i que tem um efeito $\neg q$ que pode ser unificado com p , então sua solução é :
 - › adicionar a restrição de ordem $(a_i \prec a_j)$, se for consistente com \prec ou
 - › adicionar a restrição de ordem $(a_j \prec a_i)$, se for consistente com \prec
 - › adicionar uma restrição de unificação que faz com p e q não unifiquem

Exemplo

- Similar (mas não idêntico) a um exemplo em Russell and Norvig's *Artificial Intelligence: A Modern Approach* (1st edition)
- Operadores:
 - ◆ **Start**
 - Precond: none
 - Effects: At(Home), sells(HWS,Drill), Sells(SM,Milk), Sells(SM,Banana)
 - ◆ **Finish**
 - Precond: Have(Drill), Have(Milk), Have(Banana), At(Home)
 - Effects: none
 - ◆ **Go(*l,m*)**
 - Precond: At(*l*)
 - Effects: At(*m*), \neg At(*l*)
 - ◆ **Buy(*p,s*)**
 - Precond: At(*s*), Sells(*s,p*)
 - Effects: Have(*p*)

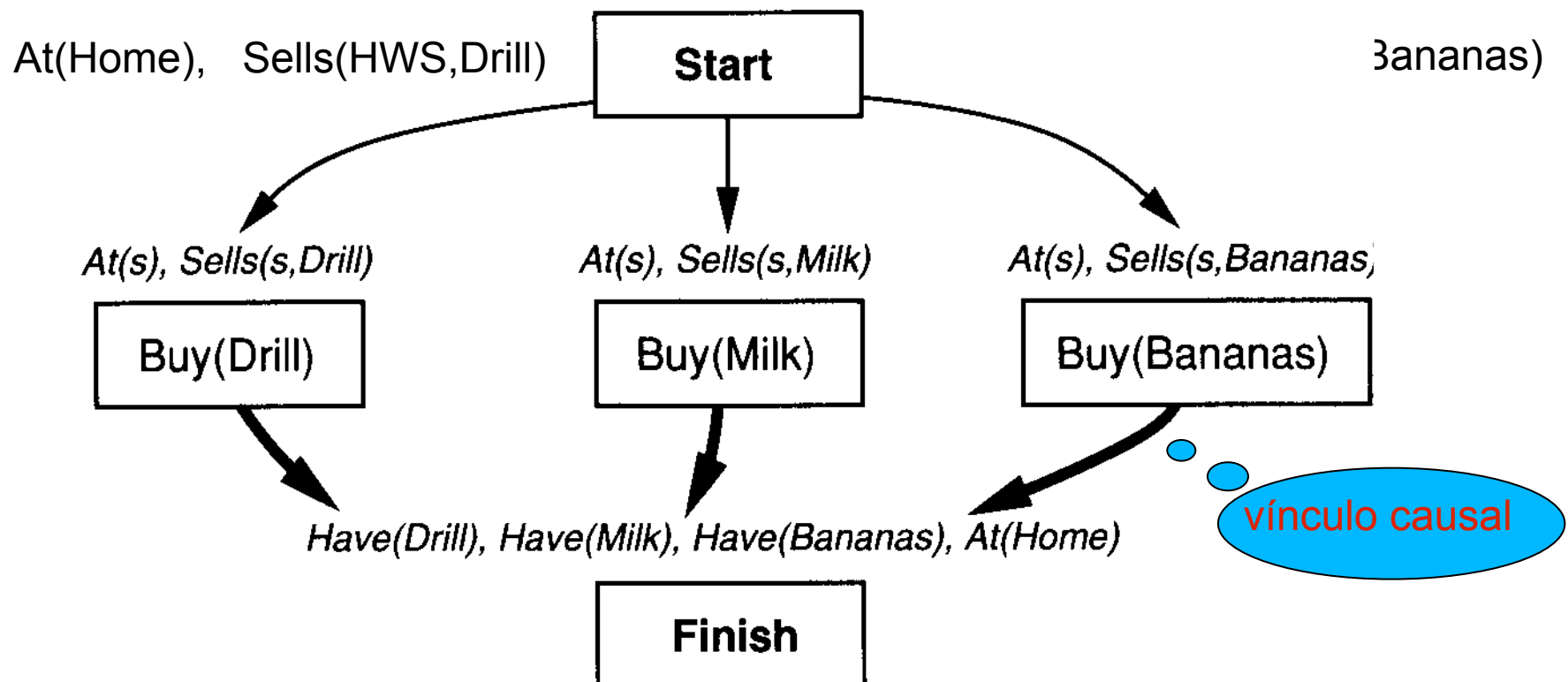
Exemplo (continuação)

- Plano inicial



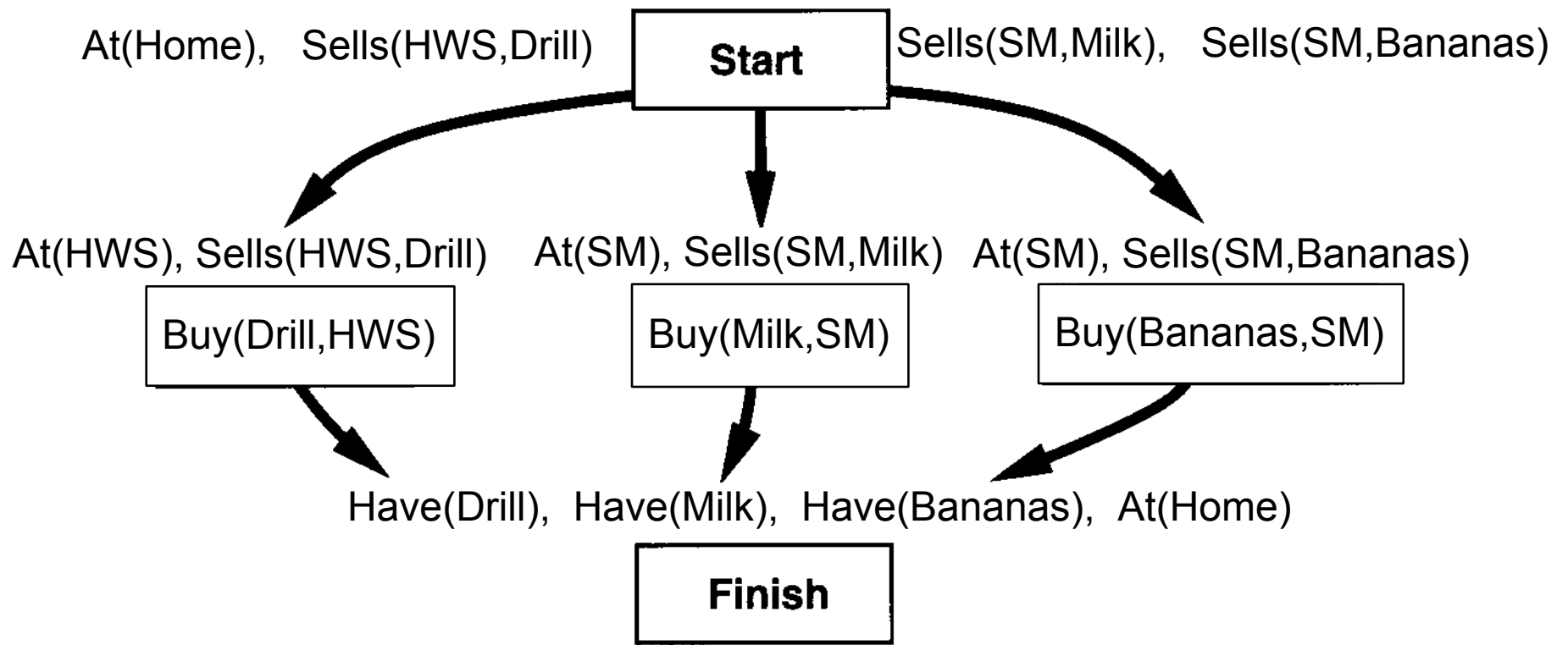
Exemplo (continuação)

- as únicas maneiras possíveis de **estabelecer** as pré-condições “Have”



Exemplo (continuação)

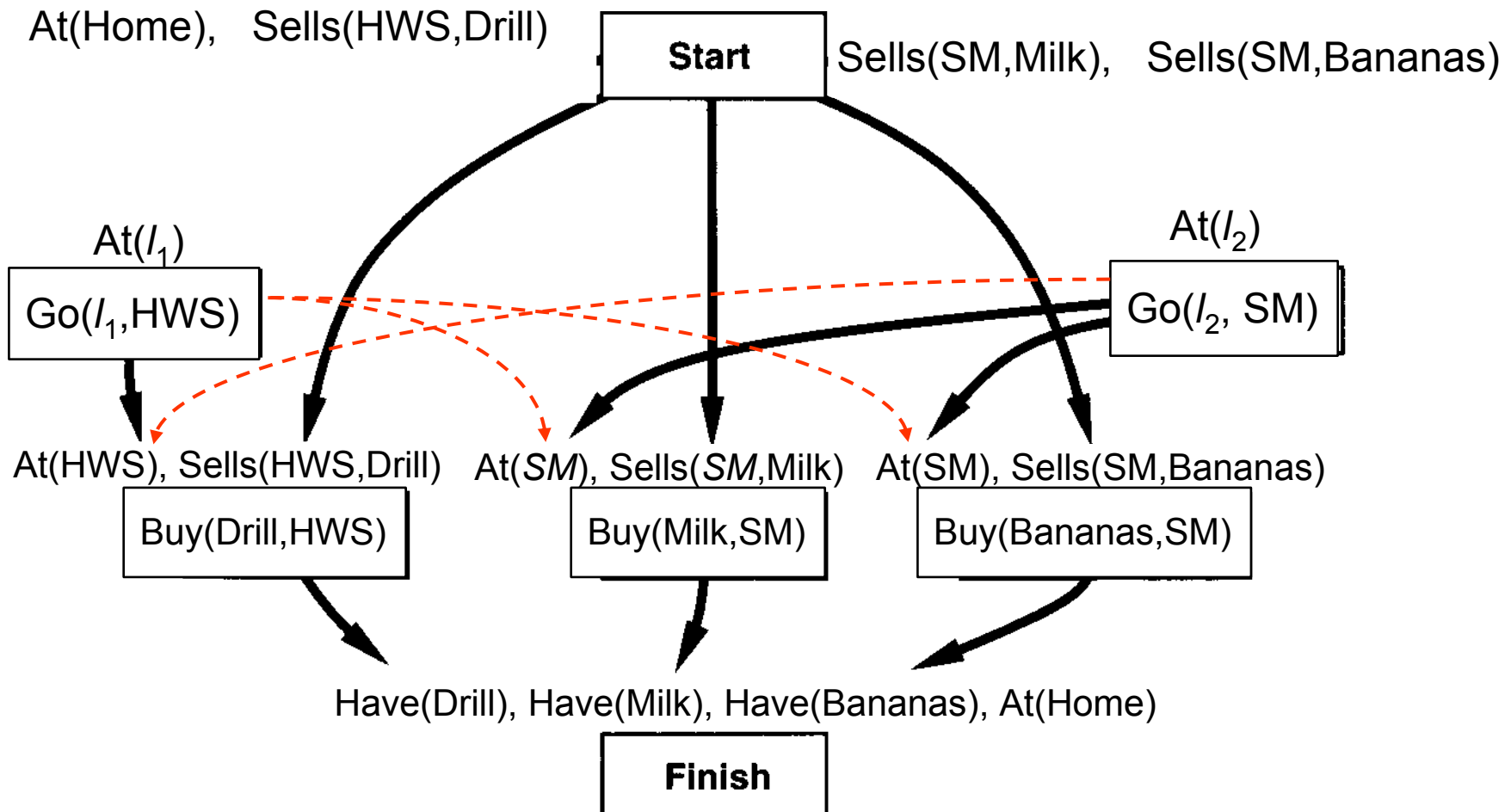
- As únicas maneiras possíveis de estabelecer pré-condições “Sells”



Exemplo (continuação)

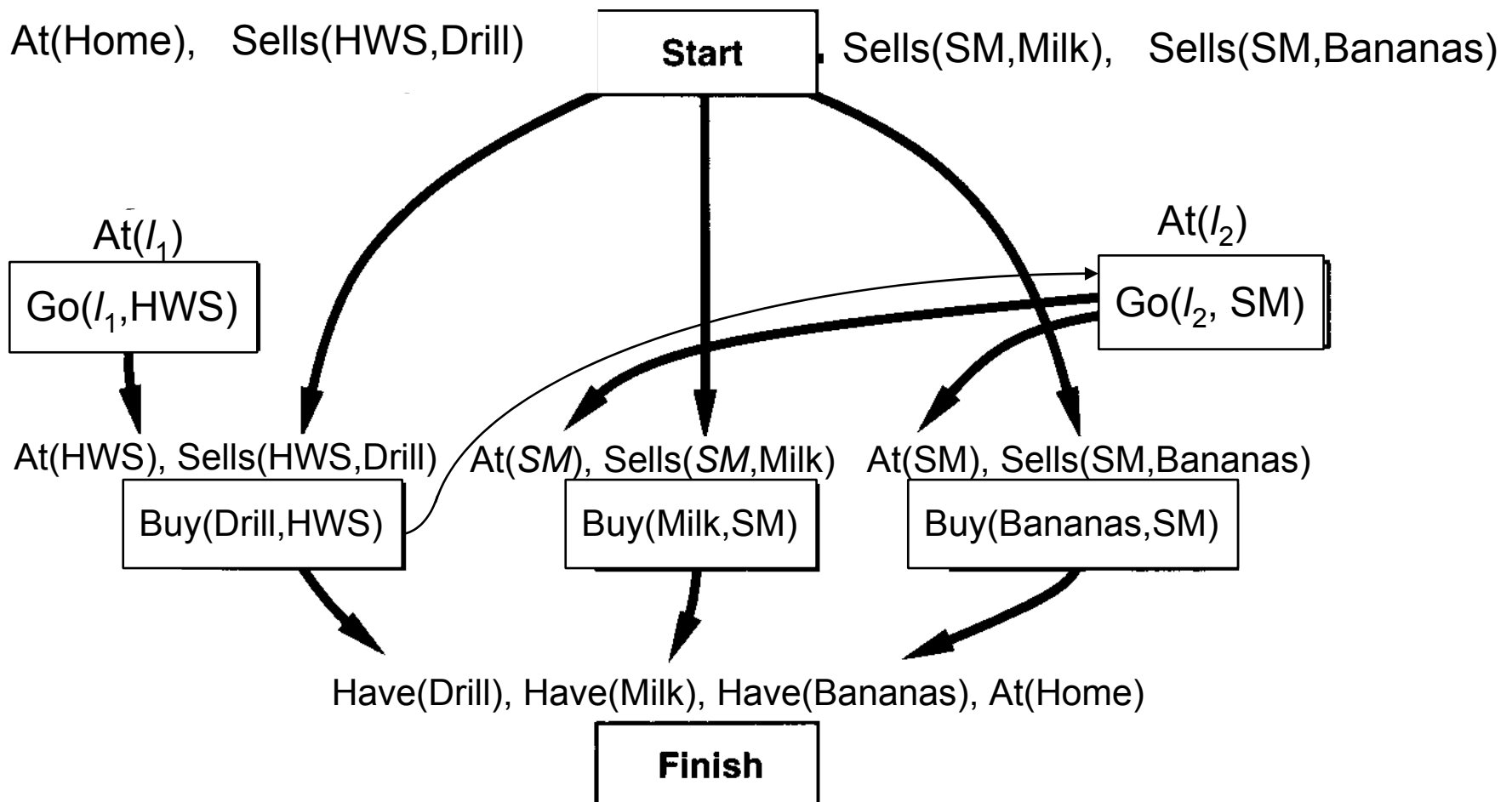
- A única maneira de estabelecer $At(HWS)$ e $At(SM)$

- ◆ Observe as ameaças



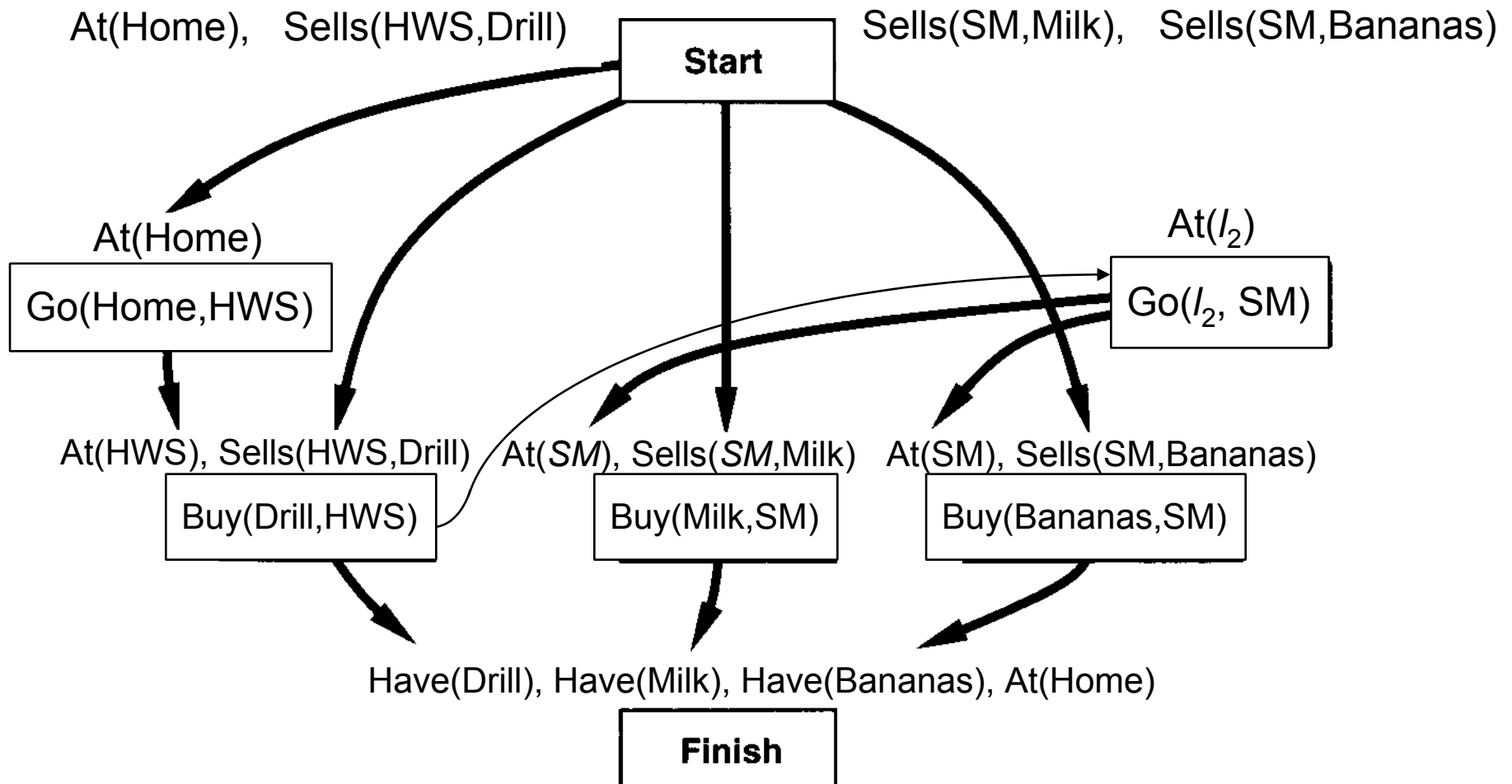
Exemplo (continuação)

- Para resolver a terceira ameaça $\langle \text{Go}(I_2, \text{SM}) \text{--AT}(\text{HWS}) \text{--} \rightarrow \text{Go}(I_1, \text{HWS}) \rangle$, fazer $\text{Buy}(\text{Drill})$ preceder $\text{Go}(\text{SM})$. Isso resolve as três ameaças.



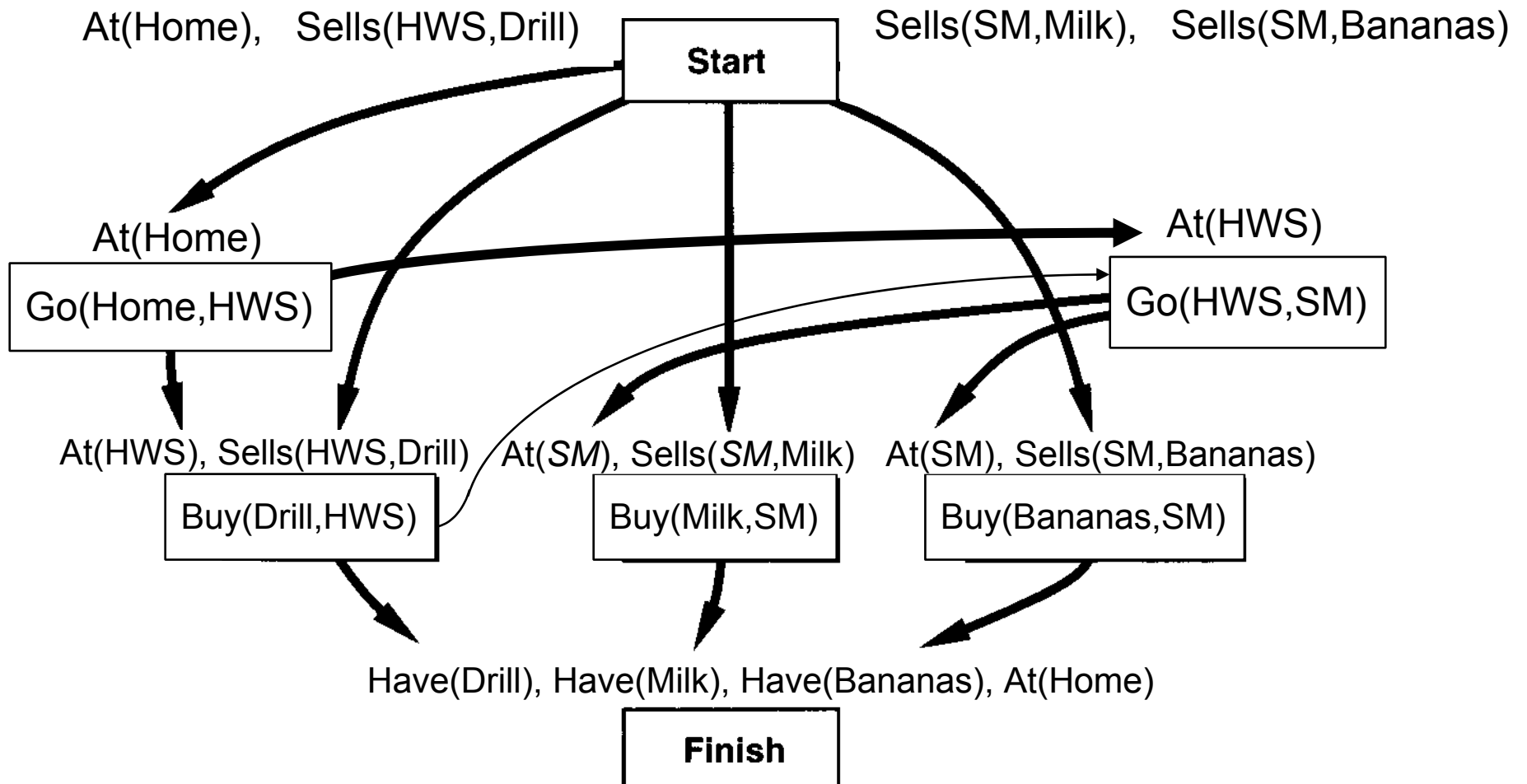
Exemplo (continuação)

- Estabelecer $At(l_1)$ com $l_1=Home$



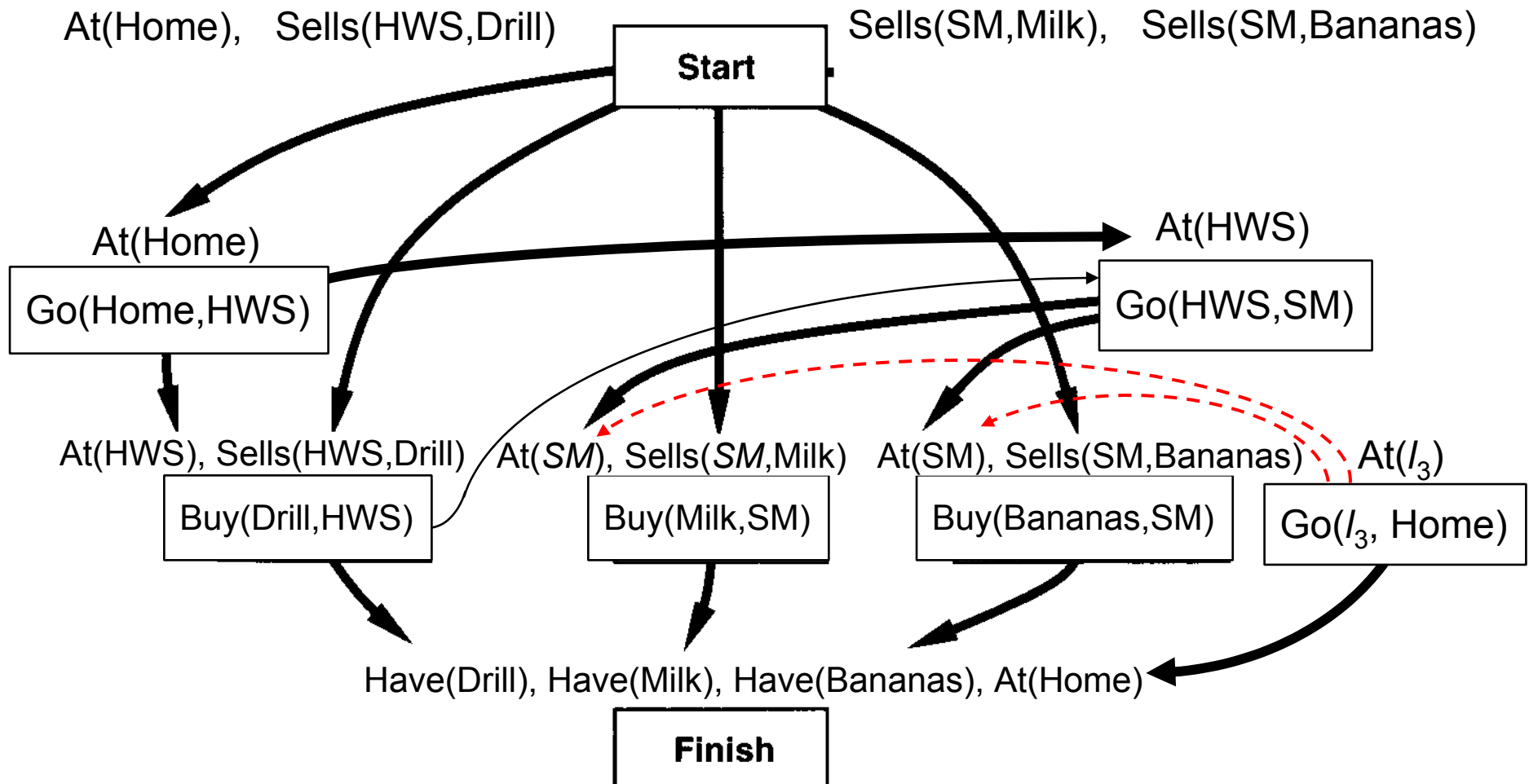
Exemplo (continuação)

- Estabelecer $At(l_2)$ com $l_2=HWS$



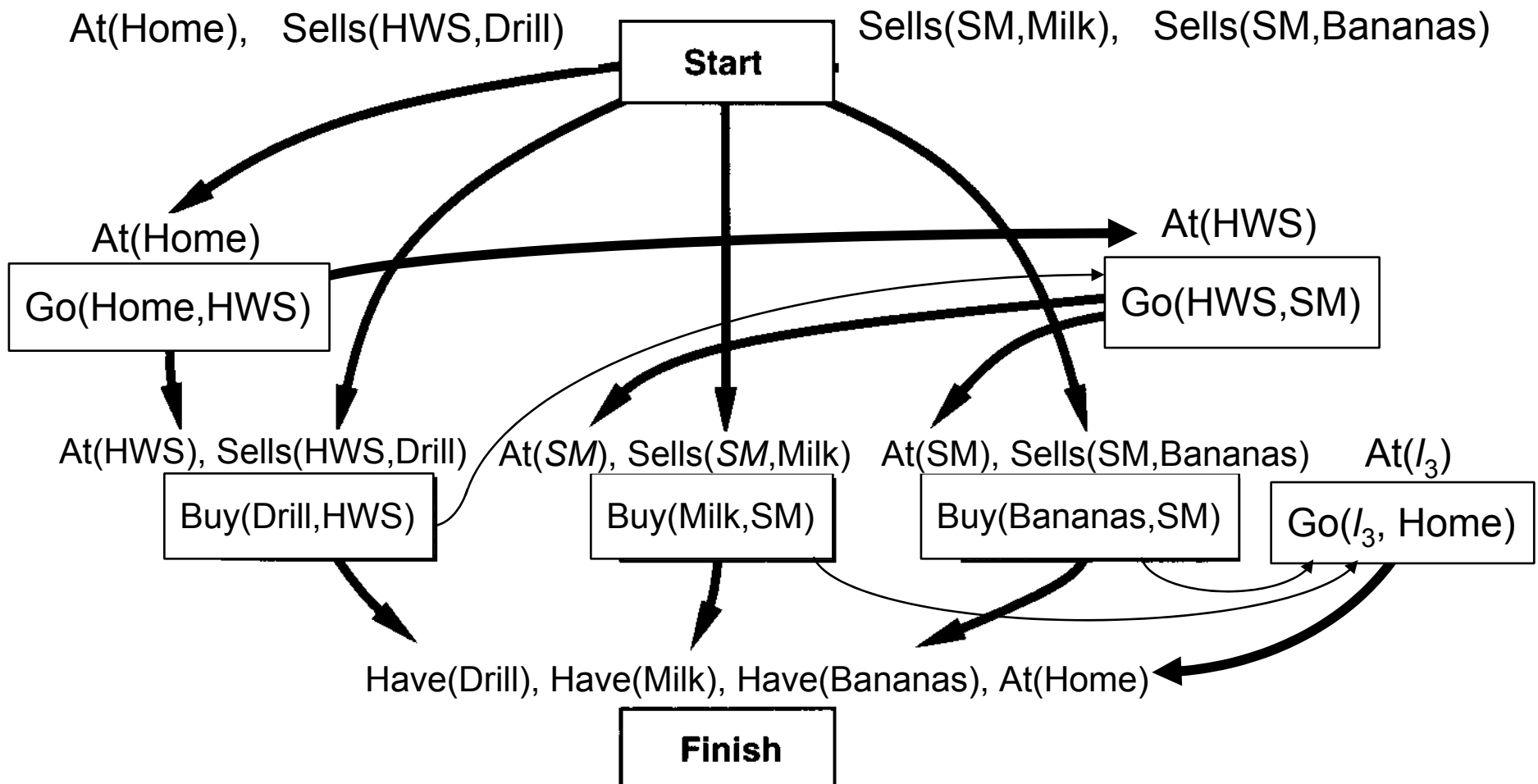
Exemplo (continuação)

- Estabelecer $At(Home)$ para Finish



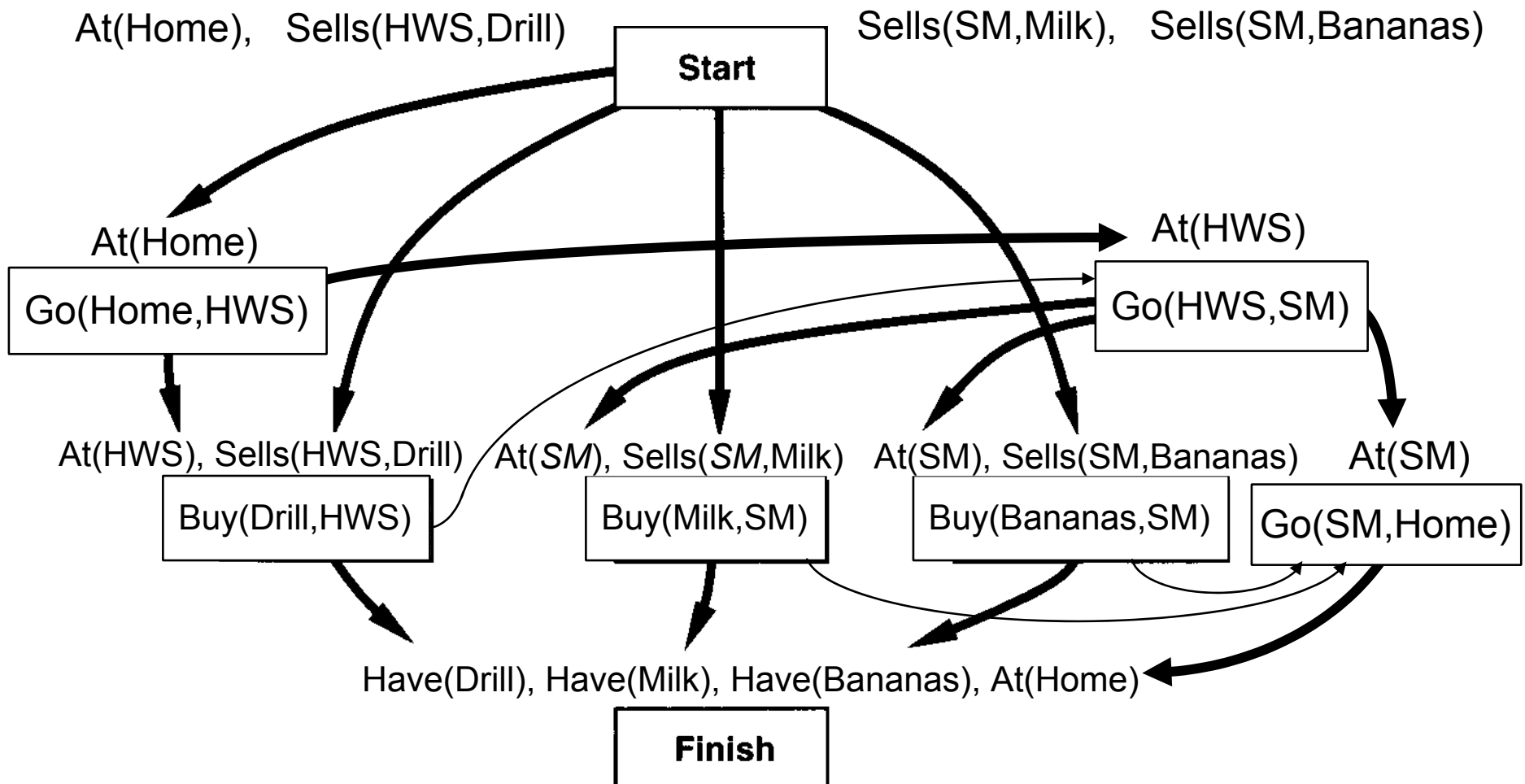
Exemplo (continuação)

- Restringir Go(Home) para remover ameaça a At(SM)



Plano Final

- Estabelecer $At(l_3)$ com $l_3=SM$



Comentários

- PSP não faz compromisso com ordem e instanciações até que seja necessário

- ◆ Evita gerar árvores de busca como essa:

Problema:

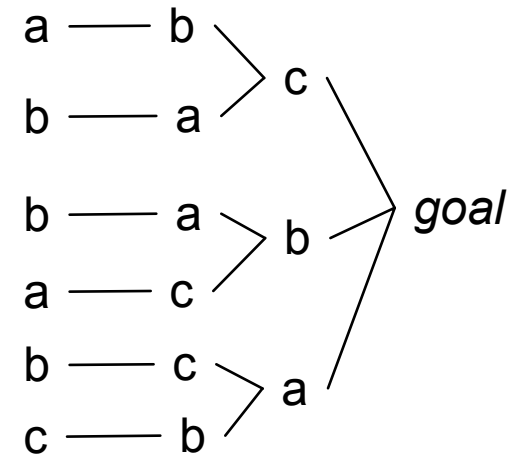
como podar caminhos infinitamente grandes?

- ◆ Detecção de laço baseada em reconhecimento de estados repetidos

- ◆ Em um plano parcialmente ordenado, nós não conhecemos os estados

... — go(b,a) — go(a,b) — go(b,a) — at(a)

Como podemos fazer a poda se vemos a mesma ação mais do que uma vez?



Não podemos. Algumas vezes é preciso usar a mesma ação várias vezes em diferentes estados do mundo (próximo slide)

POP – Partial Order Planner

- (Figura – capítulo 5)

Algoritmo POP

- instância de PSP
- Diferenças:
 - ◆ PSP processa os 2 tipos de falhas de forma similar: a cada recursão PSP seleciona (heurísticamente !) uma falha de um dos tipos para refinar
 - ◆ POP possui um controle diferente para sub-metas e ameaças
 - » toda recursão inicia com o refinamento de uma sub-meta e em seguida resolve todas as ameaças que foram possivelmente introduzidas. Assim, POP tem 2 pontos de escolha não determinísticas: (a) a escolha de uma ação relevante para satisfazer uma sub-meta e (b) a escolha de uma solução para ameaças
- Backtrackings para esses dois pontos de escolha são feitos cronologicamente

POP (extensões)

- operadores condicionais
 - ◆ os antecedentes dos efeitos condicionais são inseridos na agenda
 - ◆ ameaça de efeitos condicionais:
- Precondições disjuntivas
- Efeitos condicionais com quantificadores
- UCPOP