

Planejamento de Ordem Parcial

(ou planejamento como
busca no espaço de planos)

Revisão baseada no livro AIMA

Introdução

- O que é planejamento?
- Artigo recomendado: *A Review of AI Planning Techniques*. Austin Tate, James Hendler e Mark Drummond ==> mostra as questões ainda não resolvidas no início dos anos 90.
- No final dos anos 90 houve uma grande evolução da área. Este curso pretende esclarecer as questões do artigo e apresentar as técnicas mais avançadas.

Suposições de planejamento clássico

- tempo atômico
- existência de um único agente
- determinismo
- onisciência
- causa de mudança única
- função objetivo: número de passos do plano

Modelo de Estados

O planejamento clássico pode ser caracterizado como um **Modelo de Estados**

- um espaço de estados S , finito e não-vazio
- um estado inicial $s_0 \in S$
- um conjunto de estados meta $S_G \subseteq S$
- um conjunto de ações aplicáveis $A(s) \subseteq A$, para $\forall s \in S$
- uma função de transição $s' = f(a, s)$, para $\forall s', s \in S$ e $\forall a \in A(s)$
- custos das ações $c(a, s) > 0$

A linguagem Strips

- **Strips** (*Stanford Research Institute Problem-Solving*) é a mais antiga, simples e usada linguagem de planejamento (Fikes e Nilsson, 1971)
- evolução: Strips \rightarrow ADL \rightarrow PDDL2.1 \rightarrow PDDL+
- Um **problema** em Strips é uma tupla $\langle A, O, I, G \rangle$
 - A é o conjunto de todos os átomos (variáveis booleanas como descritores de estados),
 - O é um conjunto de todos os operadores (ações proposicionais), e
 - $I \subseteq A$ representa a situação inicial
 - $G \subseteq A$ representa as situações meta

A linguagem Strips

- Os operadores $o \in O$ são representados por três listas:
 - lista **Pre**, $Pre \subseteq A$
 - lista **Add**, $Add(o) \subseteq A$
 - lista **Del**, $Del \subseteq A$

Intuitivamente: $Pre(o)$ especifica os átomos que devem ser verdadeiros para o ser aplicável, $Add(o)$ especifica os átomos que passam a ser verdadeiros após a execução de o e $Del(o)$ especifica os átomos que passam a ser falsos após a execução de o

Strips: da linguagem aos modelos

Um problema Strips, $P = \langle A, O, I, G \rangle$ determina um modelo de estados $S(P)$ em que:

- os estados $s \in S$ são coleções de átomos;
- o estado inicial s_0 é I
- os estados meta $s \in S_G$ são aqueles em que $G \subseteq s$
- as ações $\alpha \in A(s)$ são operadores $o \in O$ tal que $Prec(o) \subseteq s$
- a função de transição f mapeia estados s em estados $s' = f(s, \alpha)$, tal que $s' = s - Del(\alpha) + Add(\alpha)$ para $\alpha \in A(s)$
- os custos das ações são iguais a 1
- a solução (ótima) de um problema de planejamento é a solução (ótima) do Modelo de Estados $S(P)$

Modelo de ações Strips

Nome da ação: $Go(there)$

Pré-condição: $At(there), Path(there,there)$ *conjunção de literais positivos*

Efeito: $At(there), \neg At(there)$ *conjunção de literais positivos ou negativos
(a versão original de Strips dividia os efeitos em **add list** e **delete list**)*

Notação gráfica:

$At(there), Path(there,there)$

$Go(there)$

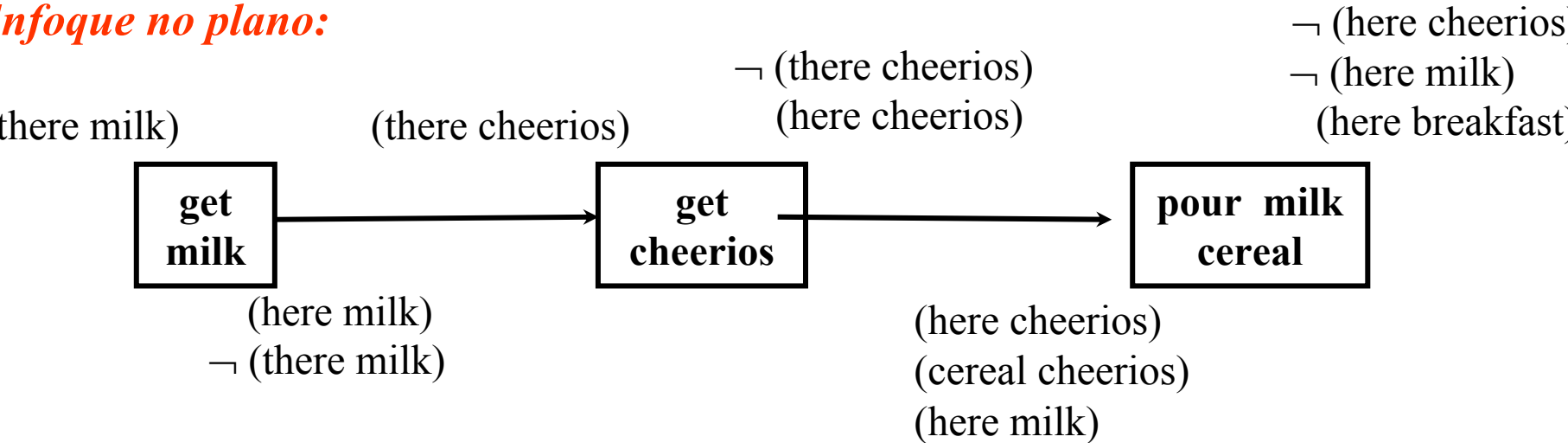
$At(there), \neg At(there)$

Linguagem do tipo Strips

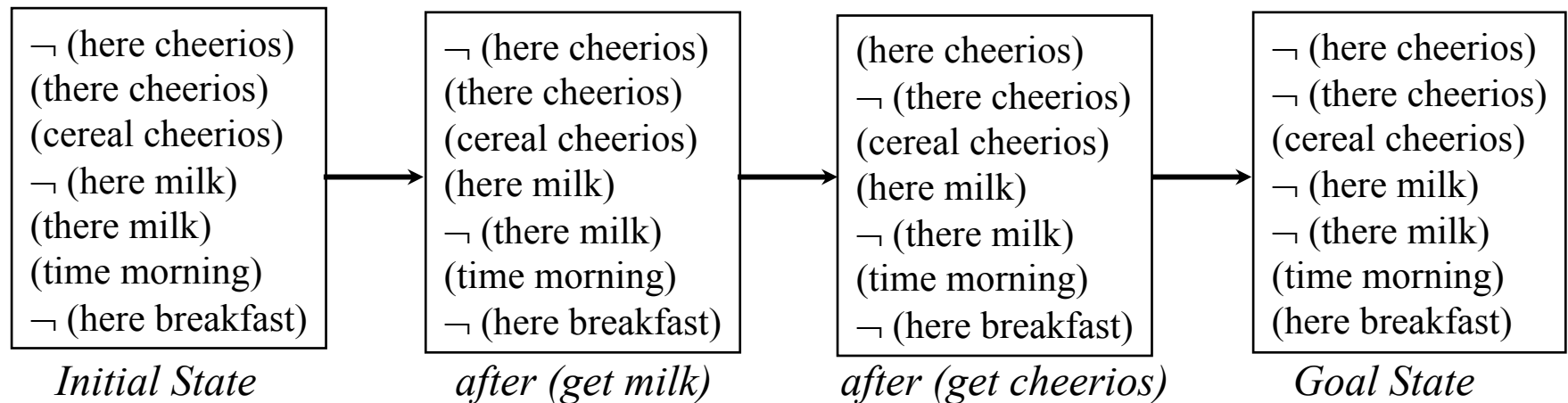
- Não existe variável explícita do estado do mundo (como por exemplo, tempo ou situação)
 - Tudo o que for declarado na pré-condição se refere à situação imediatamente antes da ação ser executada
 - Tudo o que for declarado no efeito se refere à situação imediatamente após a ação ser executada

Raciocínio sobre ações STRIPS

ênfoque no plano:

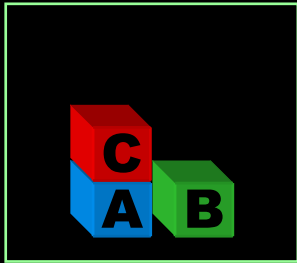


ênfoque no estado do mundo:

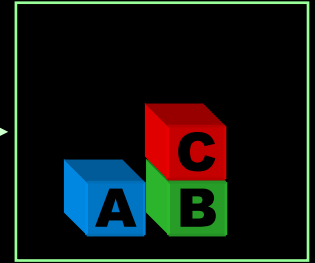


STRIPS: sintaxe e semântica

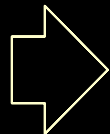
```
oper(act: move(c,a,b),  
pre: { clear(c), clear(b), on(c,a) },  
add: { clear(a), on(c,b) },  
del: { clear(b), on(c,a) })
```



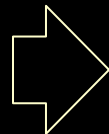
..... *move(c,a,b)*



clear(b)
clear(c)
ontable(a)
ontable(b)
on(c,a)



clear(b)
clear(c)
ontable(a)
ontable(b)
on(c,a)



clear(b)
clear(c)
ontable(a)
ontable(b)
on(c,a)
clear(a)
on(c,b)



~~*clear(b)*~~
clear(c)
ontable(a)
ontable(b)
~~*on(c,a)*~~
clear(a)
on(c,b)

Operador *schema*

- Operador *schema*: representação de um conjunto de ações (contém variáveis)
- Corresponde a uma família de ações
 - Assumi-se que variáveis são universalmente quantificadas
 - O algoritmo de planejamento deve garantir que cada variável tenha um valor no plano final. Somente ações totalmente instanciadas podem ser executadas.

Ação: *Go(there, there)*

Pré-condição: *At(there), Path(there, there)*

Efeito:

Lista de Adição: *At(there)*

Lista de Eliminação: *At(there)*

Operador *instance*

- Operador *instance* ou proposicional: representa uma única ação sobre objetos específicos do domínio (constantes).
- Alguns planejadores raciocinam diretamente sobre ações proposicionais. É preciso usar um *parser* para gerar operadores proposicionais a partir de um operador *schema*

Ação: $Go(BlocoA, BlocoB)$

Pré-condição: $At(BlocoA), Path(BlocoA, BlocoB)$

Efeito:

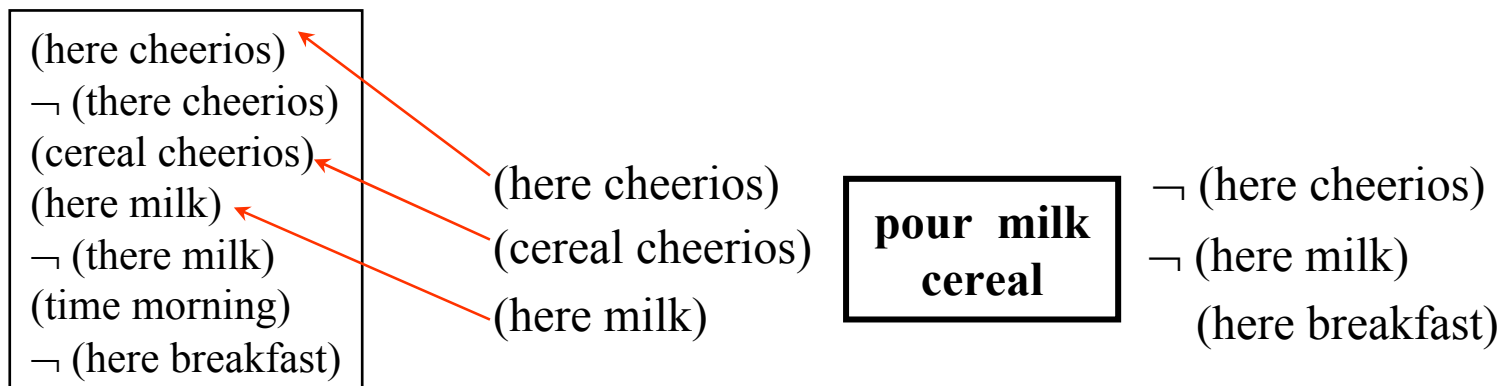
Lista de Adição: $At(BlocoB)$

Lista de Eliminação: $At(BlocoA)$

Aplicabilidade de um *schema*

- Um operador *o* é *aplicável* em um estado (ou situação) *s* se existe uma maneira de se **unificar** as variáveis em *o* de forma que todos os literais da pré-condição de *o* são verdadeiros, ou seja:

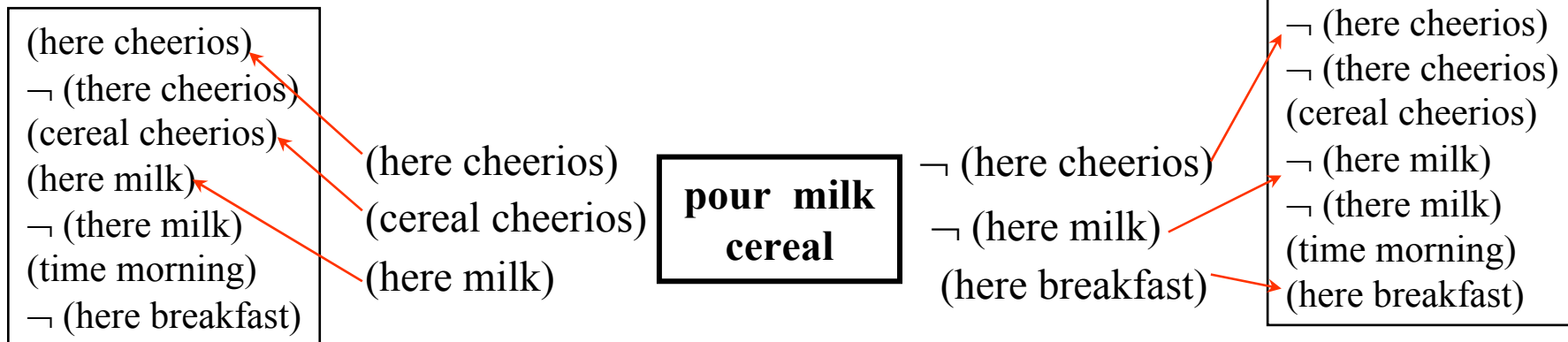
$$\text{Pré-condição}(o) \subset s$$



Efeitos de um *schema*

Estado resultante da aplicação de um operador no estado *s* :

- valem os literais positivos de *Efeito(o)* com as **unificações** feitas nas variáveis da pré-condição
- e aqueles em *s* exceto os pertencentes à lista de eliminação (*Delete List*), ou seja, os literais negativos de *Efeito(o)*



Planejamento usando a linguagem STRIPS

- Planos podem ser representados:
 - totalmente ordenado
 - parcialmente ordenado
 - grafo de planejamento
 - caminho no espaço de busca heurística (local ou global)
 - etc ...

Planejamento usando a linguagem STRIPS

- O raciocínio pode ser :
 - busca não-informada (planejamento progressivo ou regressivo)
 - busca num espaço de estados = planejadores de ordem-total (*total-order planners*) ~~*linear planners*~~
 - busca num espaço de planos = planejadores de ordem-parcial (*partial-order planners*) ~~*non-linear planners*~~
 - busca num grafo de planejamento
 - busca informada
 - busca heurística
 - uso de conhecimento sobre interação entre ações
 - etc ...

Planejamento de ordem total: busca em **espaço de estados** (situações)

Espaço de estados: o plano é o caminho a partir do estado inicial ao estado goal

- *planejador progressivo*: busca para frente (forward) a partir da situação inicial à situação goal no espaço de situações possíveis (fator de ramificação grande)

adiciona ações **aplicáveis** no fim do plano

- *planejador regressivo* : busca para trás (backward) a partir do estado goal para o estado inicial (goals tipicamente contém poucos literais (descrição parcial de estados) com poucos operadores aplicáveis)

adiciona ações **apropriadas** no início do plano

Um operador o é apropriado para um goal G se $G \subset \text{Efeito}(o)$

Planejamento de ordem total: busca em **espaço de estados** (situações)

- Problema: goals conjuntivos nem sempre são independentes
- Strips: *situation-regression planner* (**incompleto**)
- outras abordagens mais modernas: busca heurística (local ou global). Resolvem conflitos através do uso de heurísticas.
 - Hill Climbing e Hill Climbing Reforçado
 - WA*

Algoritmo geral de busca

função Busca-Geral(*problema*, *Queueing-FN*) devolve uma solução, ou falha

/* inicializa a árvore de busca com o estado inicial do *problema* */

nós ← CRIA-LISTA(Estado-Inicial[*problema*])

laço faça

se a lista de *nós* estiver vazia então devolva falha

nó ← Remove-Primeiro (*nós*)

se Testa-Meta(*nó*) então devolva *nó* (e o caminho da raiz até o nó)

nós ← *Queueing-FN* (*nós*, EXPANDA(*nó*, Ações[*problema*]))

fim

Estratégias em *Queueing-FN*:

inserção numa fila: busca em largura

inserção numa pilha: busca em profundidade

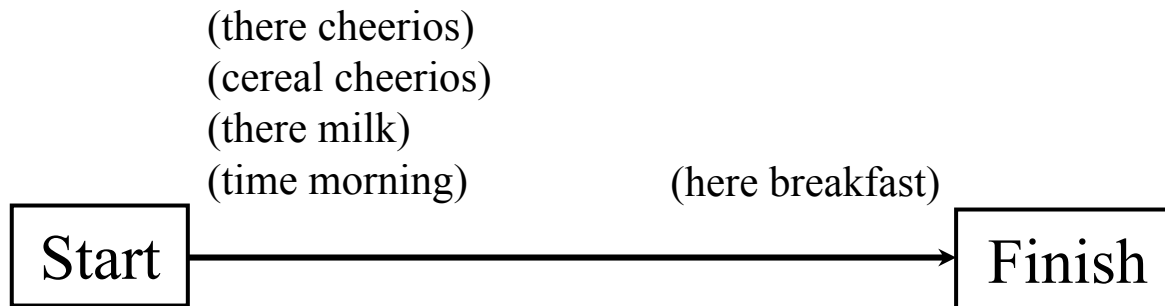
fila de prioridade: A* (ordenar a fila de acordo uma função de avaliação

$$f(n) = g(n) + h(n)$$

Busca no Espaço de Planos

Busca no Espaço de Planos

- Começa-se com um plano simples, incompleto: *plano parcial*
- Expande-se o plano parcial até gerar um plano solução que resolve o problema.



Operadores de refinamento do plano

Operações no plano podem ser de dois tipos:

refinamento e modificação

- adicionar um passo no plano
- adicionar um vínculo causal
- adicionar restrições de ordem
- adicionar restrições de valores das variáveis
- detectar inconsistências e interações
- corrigir interações (modificar ordem ou valores de variáveis)
- *backtracking*

Planejamento de compromisso fraco

Planejadores podem adotar uma estratégia de compromisso fraco (*least commitment planners*) com relação a:

- ordenação de passos de plano e
- instanciação de variáveis (*variable binding*)
- vínculos causais

Planejamento de ordem parcial X

Planejamento de ordem total

Estratégia de compromisso fraco com relação a ordem

- *Planejamento de ordem total*: representa planos onde todo passo do plano é ordenado com relação a todos os passos restantes .
- *Planejamento de ordem parcial*: representa planos onde somente alguns passos são ordenados.

Um plano totalmente ordenado, derivado de um plano parcialmente ordenado p é chamado *linearização de p*

Exemplo

- Objetivo:

LeftShoeOn, RightShoeOn

- Operadores:

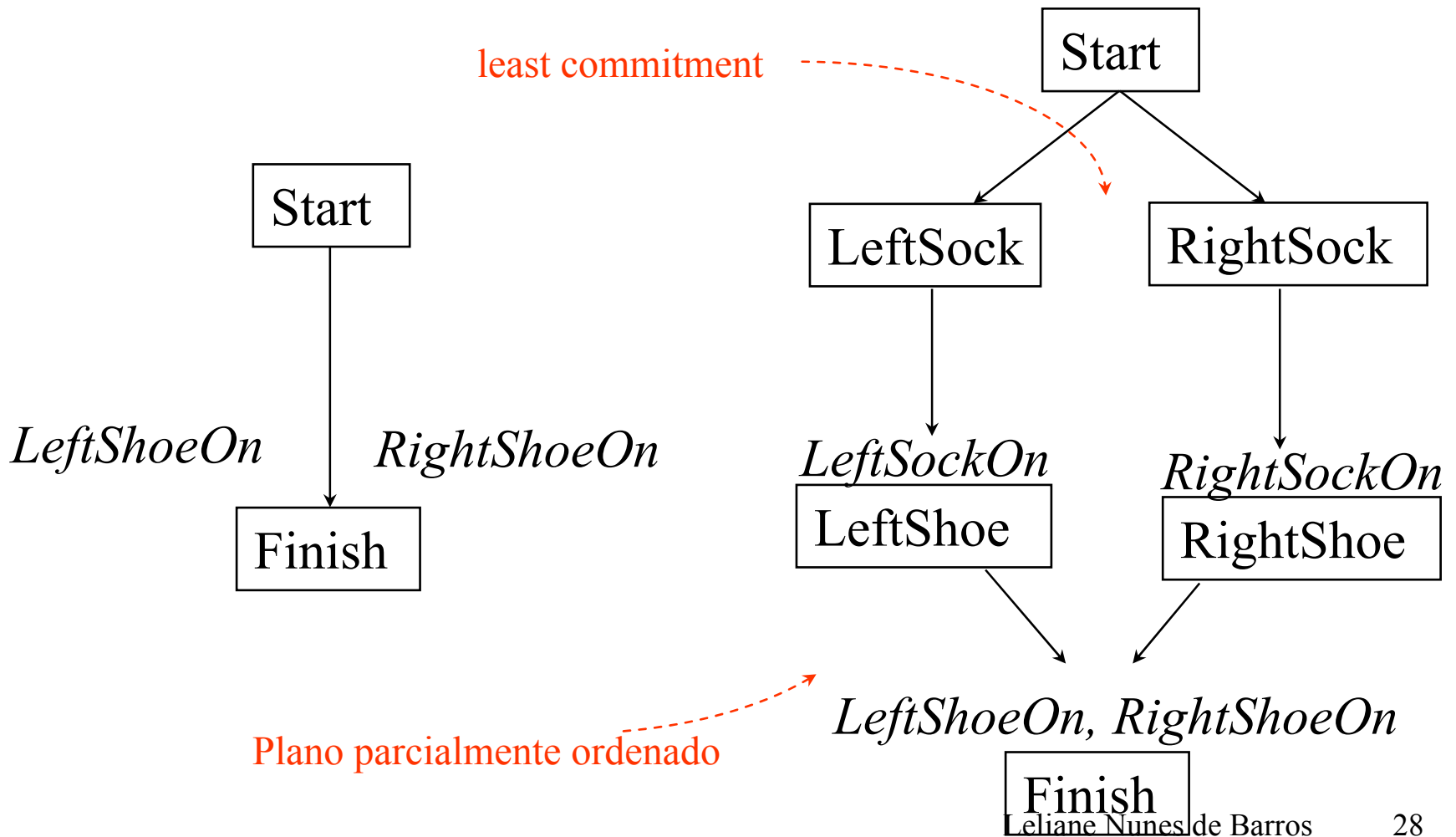
Op(Action:RightShoe,Precond:RightSockOn,Effect:RightShoeOn)

Op(Action:RightSock,Effect:RightSockOn)

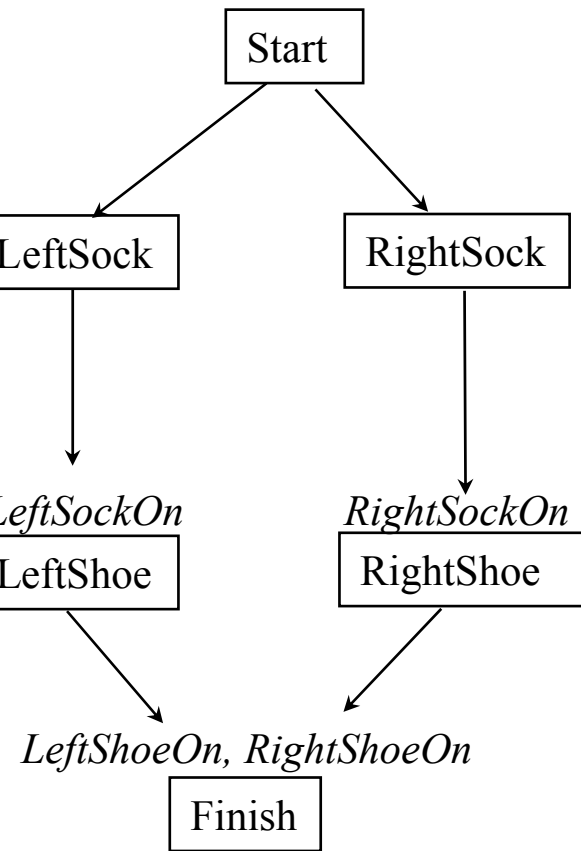
Op(Action:LeftShoe,Precond:LeftSockOn,Effect:LeftShoeOn)

Op(Action:LeftSock,Effect:LeftSockOn)

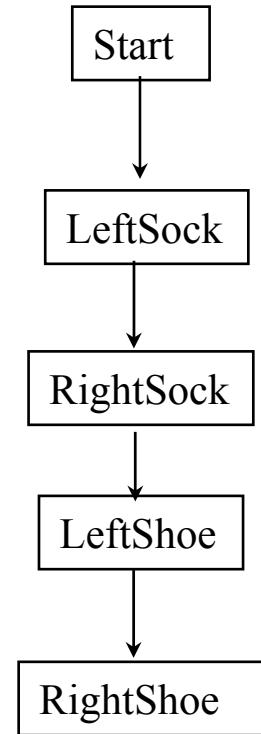
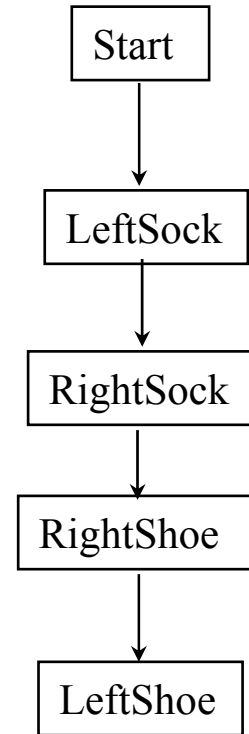
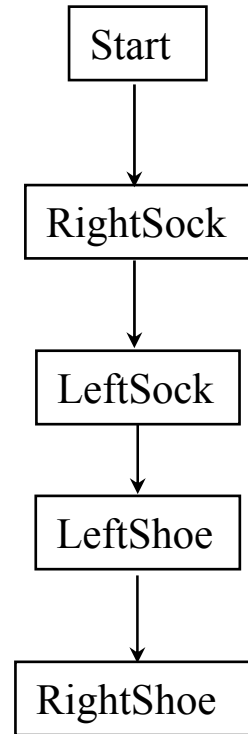
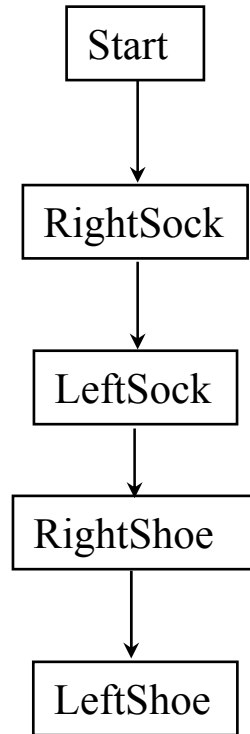
Representações de planos



Linearizações de p



Partial Order Plan



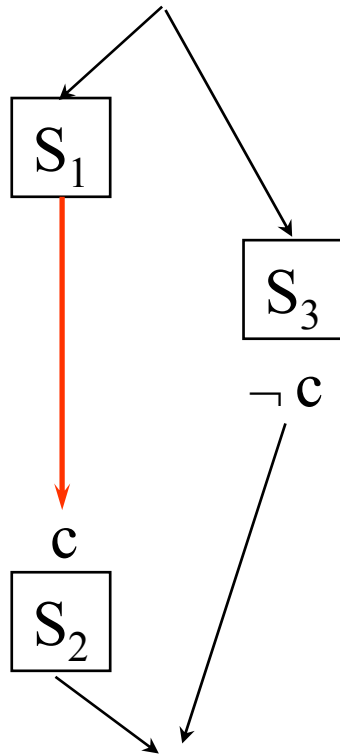
Total Order Plans

Representação de Plano

estrutura de dados contendo 4 componentes:

- ★ Um conjunto de **passos de plano** + um mapeamento entre passos e operadores com suas unificações
- ★ Um conjunto de **restrições de ordenação** sobre os passos do plano :
 - $S_i < S_j$, que deve ser lido “ S_i ocorre antes de S_j “, que significa: S_i deve ocorrer antes de S_j mas não necessariamente imediatamente antes.
- ★ Um conjunto de restrições de instanciações do tipo $v = x$ ou $v = \neg x$ ou $v \neq x$, sendo x uma constante ou uma outra variável (codesignação ou não-codesignação).

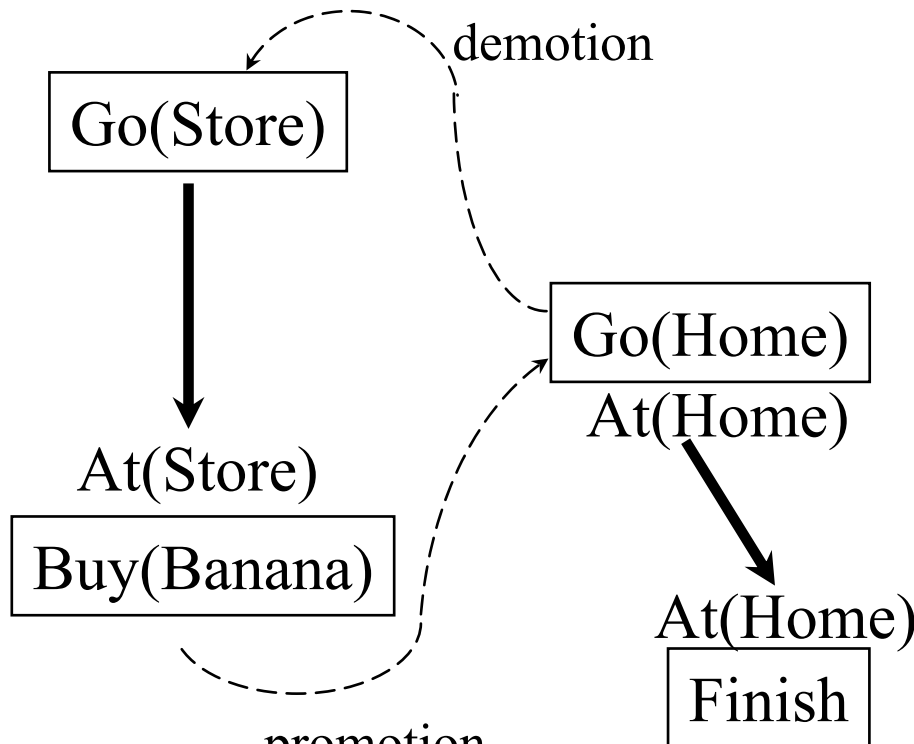
Verificando a satisfação de goals



Uma precondição c de um passo S_2 é satisfeita sse ela é efeito de um passo S_1 e não existir um passo S_3 , que ocorra possivelmente no intervalo do *vínculo causal* $S_1 \xrightarrow{c} S_2$, “ameaçando” a satisfação de c .

Modificando o plano

- Um passo ameaça (*clobberer* ou *threaten*) é um passo do plano que destrói a precondição realizada por um vínculo causal



Demotion: coloque antes de Go(Store)
Promotion: coloque depois de Buy(Banana)

Representação de Plano

* Um conjunto de *vínculos causais (causal links)*:

$$S_i \xrightarrow{c} S_j$$

que deve ser lido

“O passo contribuidor S_i , realiza c para o passo que necessita S_j “

Vínculos causais servem para registrar o propósito dos passos no plano, onde o propósito de S_i é realizar a pré-condição c de S_j .

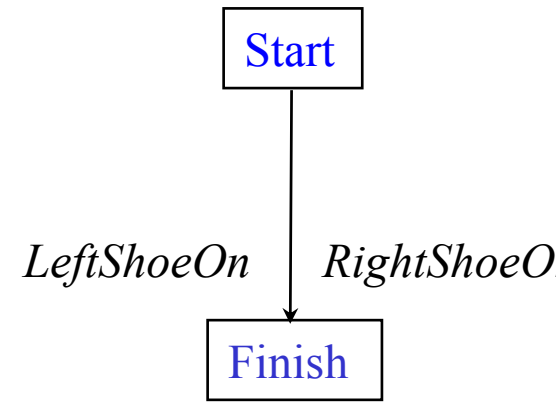
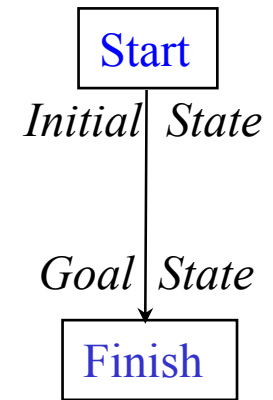
Também chamados por alguns autores de *intervalos de proteção (protection intervals)*

Plano inicial

- Corresponde à própria descrição do problema
- Dois passos: *Start* e *Finish*
 - onde, $Start < Finish$
 - estão associados a ações nulas

Start não tem pré-condição e seu efeito é o de adicionar as proposições verdadeiras no estado inicial

Finish tem como pré-condição o goal e nenhum efeito



Plano inicial

Plan(Steps: { S_1 : *Op*(Action: *Start*),
 S_2 : *Op*(Acion: *Finish*,
 Precond: *RightShoeOn* ^ *LeftShoeOn*)},
Ordering: { $S_1 < S_2$ },
Bindings: { },
Causal Links: { })

Plano solução

- Um plano que o agente pode executar e que garante a realização do objetivo
 - plano totalmente instanciado e totalmente ordenado ou
 - plano totalmente instanciado e parcialmente ordenado (corresponde a um conjunto de planos totalmente ordenados). Cabe ao agente escolher uma linearização ou o plano pode ser executado por multi-agentes.

Plano solução (def.)

- Plano completo (teste de fim de planejamento):
 - toda condição é realizada por algum passo do plano.
 - Uma condição é realizada sse ela é efeito de um passo e nenhum passo intermediário a desfaz (teoria sobre interação de ações, por exemplo: MTC, axiomas de frame do SC ou axiomas do EC)
- Plano consistente:
 - não existem contradições nas restrições de ordenação ou *bindings*

Algoritmo POP

function POP(*initial*, *goal*, *operators*) **returns** *plan*

plan \leftarrow Make-Minimal-Plan(*initial*, *goal*)

loop do

if Solution? (*plan*) **then return** *plan*

S_{need} , *c* \leftarrow Select-Subgoal(*plan*)

Choose-Operator(*plan*, *operators*, S_{need} , *c*)

Resolve-Threats(*plan*)

end

function Select-Subgoal(*plan*) **returns** S_{need} , *c*

pick a plan step S_{need} from Steps(*plan*)

with a precondition *c* that has not been achieved

return S_{need} , *c*

POP algorithm (cont.)

procedure Chose-Operator($plan, operators, S_{need}, c$)

choose a step S_{add} from $operators$ or $Steps(plan)$ that has c as an effect

if there is no such step **then fail**

add the causal link $S_{add} \xrightarrow{c} S_{need}$ to $Links(plan)$

add the ordering constraint $S_{add} < S_{need}$ to $Orderings(plan)$

if S_{add} is a newly added step from $operators$ **then**

add S_{add} to $Steps(plan)$

add $Start < S_{add} < Finish$ to $Orderings(plan)$

POP algorithm (cont.)

procedure Resolve-Threats(*plan*)

for each S_{threat} that threatens a link $S_i \xrightarrow{c} S_j$ in Links(*plan*) **do**

choose either

Demotion: Add $S_{threat} < S_i$ to Orderings(*plan*)

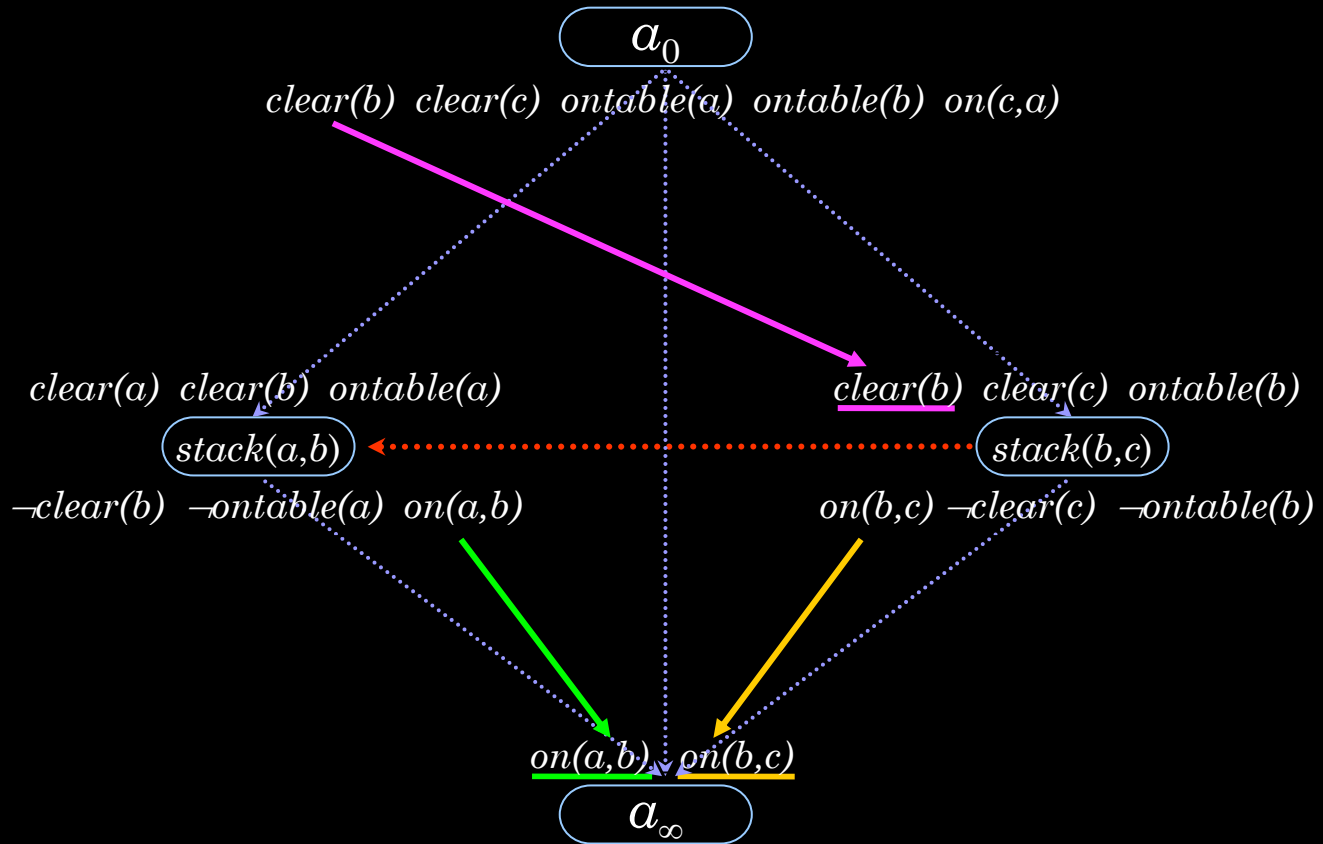
Promotion: Add $S_i < S_{threat}$ to Orderings(*plan*)

if not consistent(*plan*) **then fail**

end

POP é correto, completo e sistemático (busca sem repetição)

Plano encontrado pelo POP



$S = \{stack(b,c), stack(a,b), a_0, a_\infty\}$

$O = \{stack(b,c) < stack(a,b), a_0 < stack(b,c) < a_\infty, a_0 < stack(a,b) < a_\infty, a_0 < a_\infty\}$

$\mathcal{L} = \{a_0 \rightarrow clear(b)@stack(b,c), stack(b,c) \rightarrow on(b,c)@a_\infty, stack(a,b) \rightarrow on(a,b)@a_\infty\}$