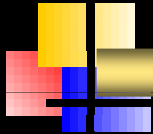




IME-USP

Planejamento Abduativo no Cálculo de Eventos



Dissertação apresentada ao IME-USP para
obtenção do grau de Mestre em Computação

Candidato: Silvio do Lago Pereira

Orientadora: Prof^a. Dr^a. Leliane Nunes de Barros

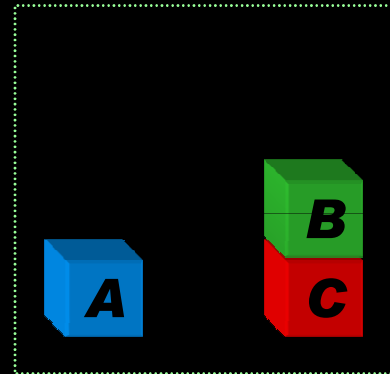
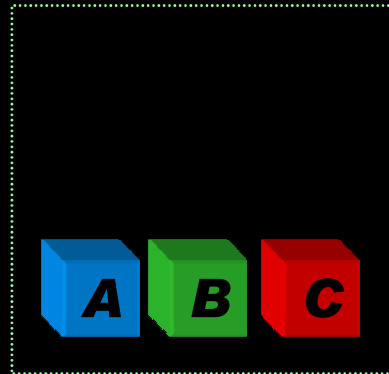
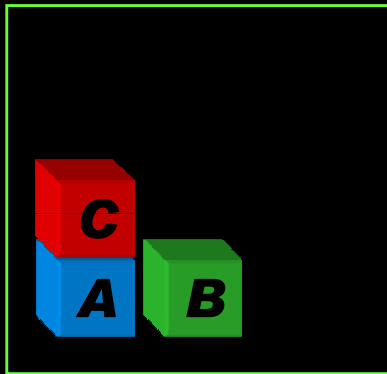


IME-USP

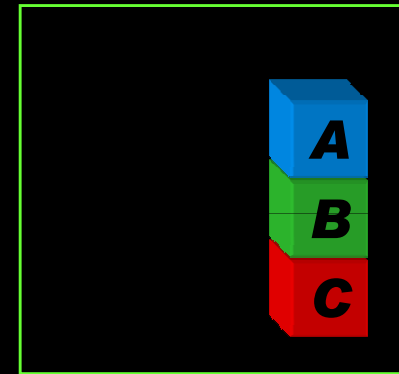


A tarefa de planejamento

Estado inicial



Estado final



Plano

desempilha c de a

empilha b sobre c

empilha a sobre b



IME-USP



Nossa conjectura

- (1) o comportamento inteligente é resultado de um raciocínio correto sobre uma representação correta;
- (2) a melhor maneira de se explicar os conceitos de representação correta e raciocínio correto é usar lógica formal.



IME-USP



Nosso objetivo

Mostrar que:

- raciocínio abdutivo no cálculo de eventos e planejamento de ordem parcial são isomorfos;
- um planejador abdutivo pode implementar métodos de planejamento sistemático e redundante;
- a eficiência de um planejador não depende apenas do método por ele implementado mas, também, das características específicas do domínio considerado.



IME-USP



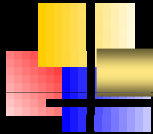
Organização

- Planejamento dedutivo
- Planejamento algorítmico
- Planejamento abdutivo
- Resultados experimentais
- Conclusão



IME-USP

Planejamento dedutivo



no cálculo de situações



IME-USP

Cálculo de situações

- **Ontologia:** situações, fluentes e ações
- **Linguagem**
 - s_0
 - $do(\alpha, \sigma)$
 - $poss(\alpha, \sigma)$
 - $holds(\phi, \sigma)$



IME-USP

O domínio do mundo dos blocos

- **fluentes:**

clear(X)

ontable(X)

on(X, Y)

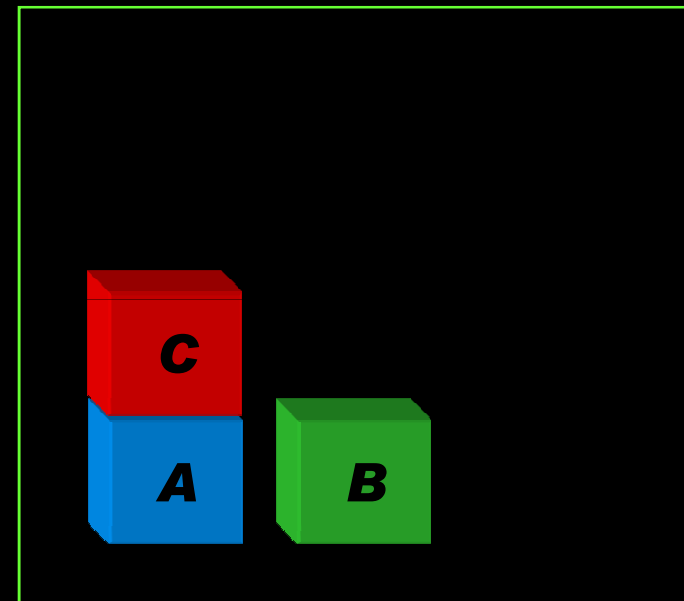
- **ações:**

stack(X, Y)

unstack(X, Y)

move(X, Y, Z)

S_0





IME-USP

Especificação lógica do domínio

- axiomas de observação:

$holds(clear(c), s_0)$

$holds(on(c, a), s_0)$

...

- axiomas de efeito:

$holds(clear(Y), do(move(X, Y, Z), S))$

$holds(on(X, Z), do(move(X, Y, Z), S))$

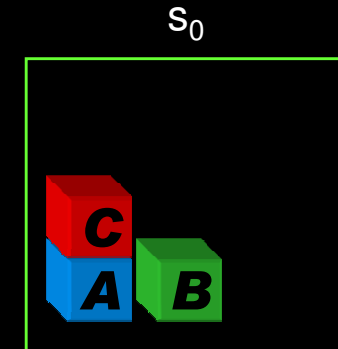
...

- axiomas de precondições:

$poss(move(X, Y, Z), S) \leftarrow$

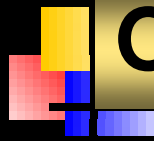
$holds(clear(X), S) \wedge holds(clear(Z), S) \wedge holds(on(X, Y), S)$

...



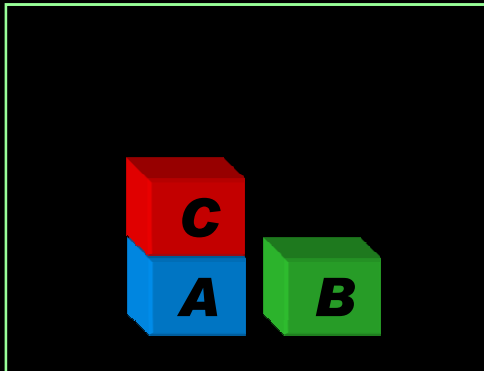


IME-USP



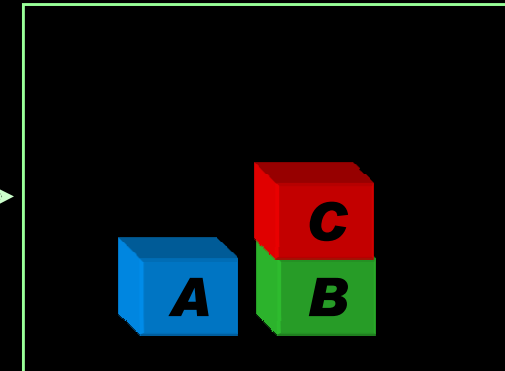
O problema da persistência

s_0



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

$s_1 := do(move(c,a,b), s_0)$



$holds(clear(b), s_0)$
 $holds(clear(c), s_0)$
 $holds(ontable(a), s_0)$
 $holds(ontable(b), s_0)$
 $holds(on(c,a), s_0)$

axiomas de efeito

$holds(clear(a), s_1)$
 $holds(on(c,b), s_1)$

$holds(clear(c), s_1)$
 $holds(ontable(a), s_1)$
 $holds(ontable(b), s_1)$

axiomas de persistência



IME-USP



Planejamento dedutivo

Dados:

\mathcal{A} : axiomatização do domínio

I : situação inicial

\mathcal{G} : meta de planejamento

O planejamento consiste em provar que

$$\mathcal{A} \wedge I \models \exists S[\text{exec}(S) \wedge \mathcal{G}(S)],$$

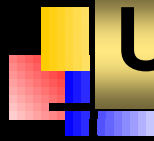
sendo executabilidade definida indutivamente por:

$$\text{exec}(s_0)$$

$$\text{exec}(\text{do}(A, S)) \leftarrow \text{poss}(A, S) \wedge \text{exec}(S)$$



IME-USP



Um planejador em PROLOG

holds(clear(b),s0).

holds(clear(c),s0).

holds(ontable(a),s0).

holds(ontable(b),s0).

holds(on(c,a),s0).

holds(on(X,Y),do(stack(X,Y),S)).

holds(clear(Y),do(unstack(X,Y),S)).

holds(ontable(X),do(unstack(X,Y),S)).

poss(stack(X,Y),S) :- holds(ontable(X),S), holds(clear(X),S), holds(clear(Y),S), X\=Y.

poss(unstack(X,Y),S) :- holds(clear(X),S), holds(on(X,Y),S).

holds(F,do(A,S)) :- poss(A,S), holds(F,S), not affects(A,F).

affects(stack(X,Y),clear(Y)).

affects(stack(X,Y),ontable(X)).

affects(unstack(X,Y),on(X,Y)).

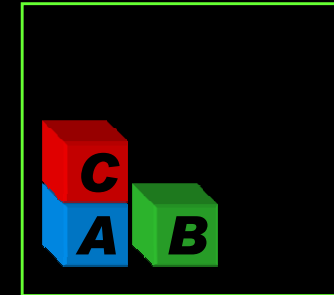
exec(s0).

exec(do(A,S)) :- poss(A,S), exec(S).

plan(s0).

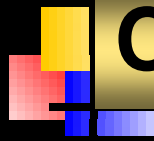
plan(do(A,S)) :- plan(S).

s₀





IME-USP



Consultando o planejador

?- plan(S), exec(S), holds(on(a,c),S).

S = do(stack(a,c),do(unstack(c,a),s0))

yes

?- plan(S), exec(S), holds(on(a,b),S), holds(on(b,c),S).

S = do(stack(a,b),do(stack(b,c),do(unstack(c,a),s0)))

yes

?- holds(F, do(stack(a,b),do(stack(b,c),do(unstack(c,a),s0)))).

F = on(a,b) ;

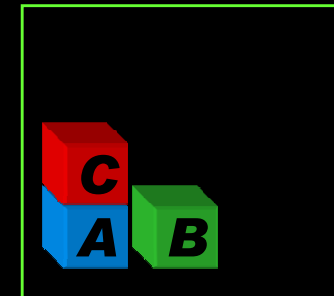
F = on(b,c) ;

F = clear(a) ;

F = ontable(c) ;

no

s₀





IME-USP

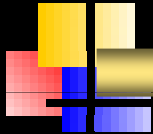
Considerações

- usando o cálculo de situações podemos descrever formalmente um domínio;
- usando prova automática de teoremas, essa descrição corresponde a uma implementação de um sistema de planejamento para o domínio;
- embora esse sistema possa ser facilmente entendido e modificado, ele não é muito eficiente.



IME-USP

Planejamento algorítmico



baseado em STRIPS



IME-USP

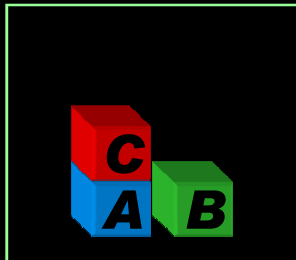


STRIPS: sintaxe e semântica

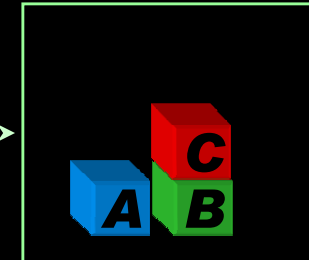
```

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) })

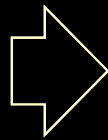
```



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)

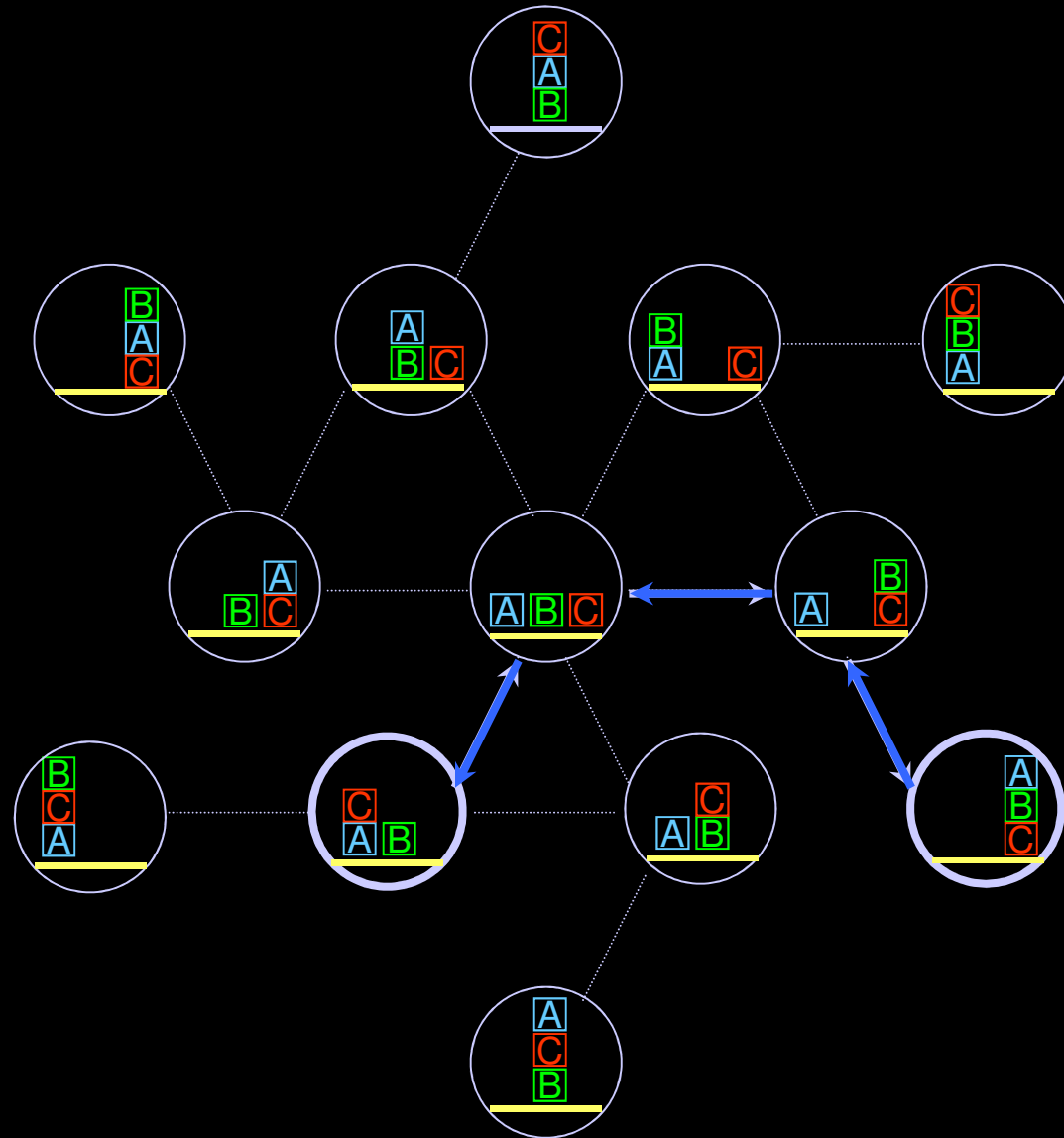


~~*clear(b)*~~
clear(c)
ontable(a)
ontable(b)
~~*on(c,a)*~~
clear(a)
on(c,b)



IME-USP

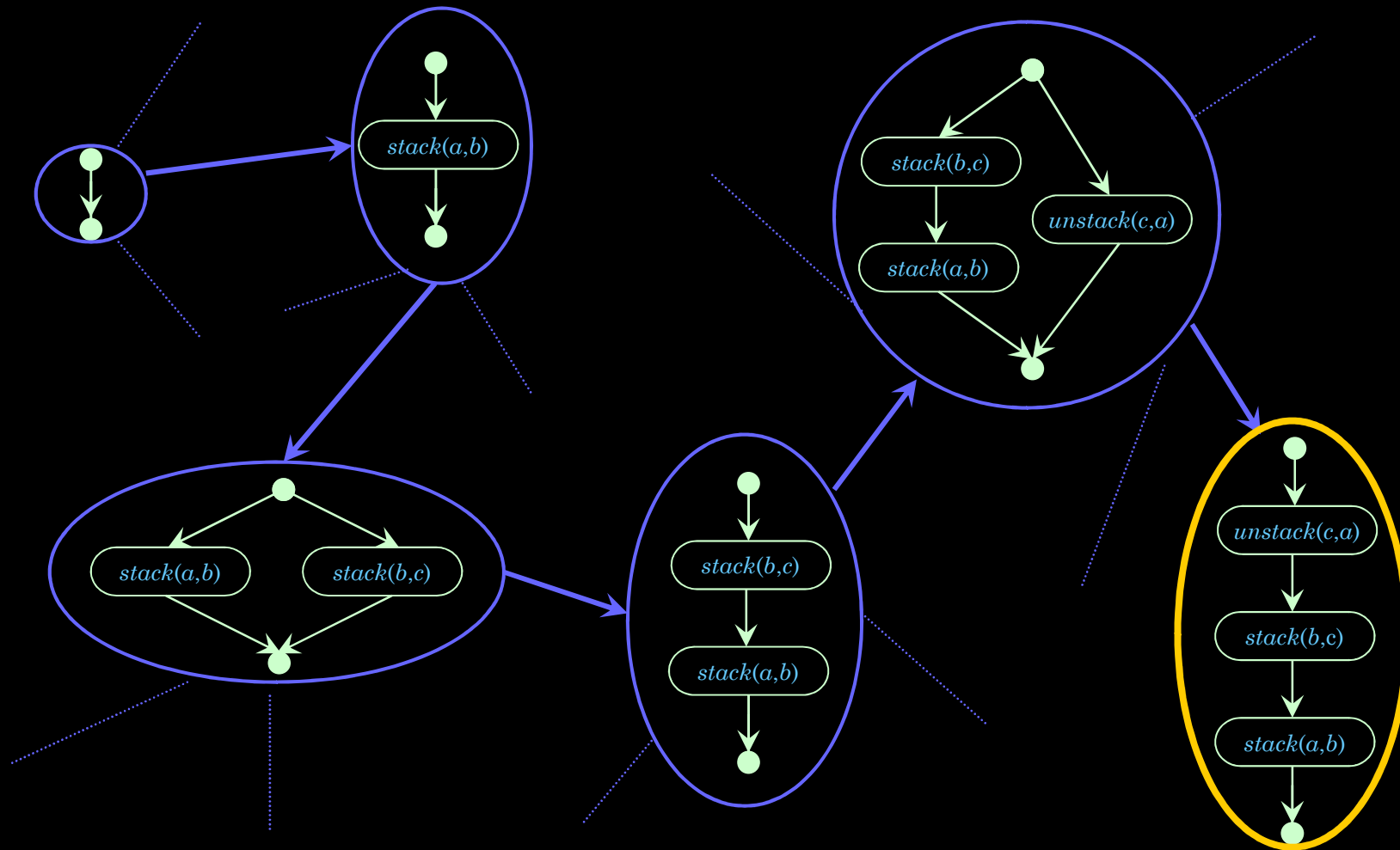
Busca no espaço de estados





IME-USP

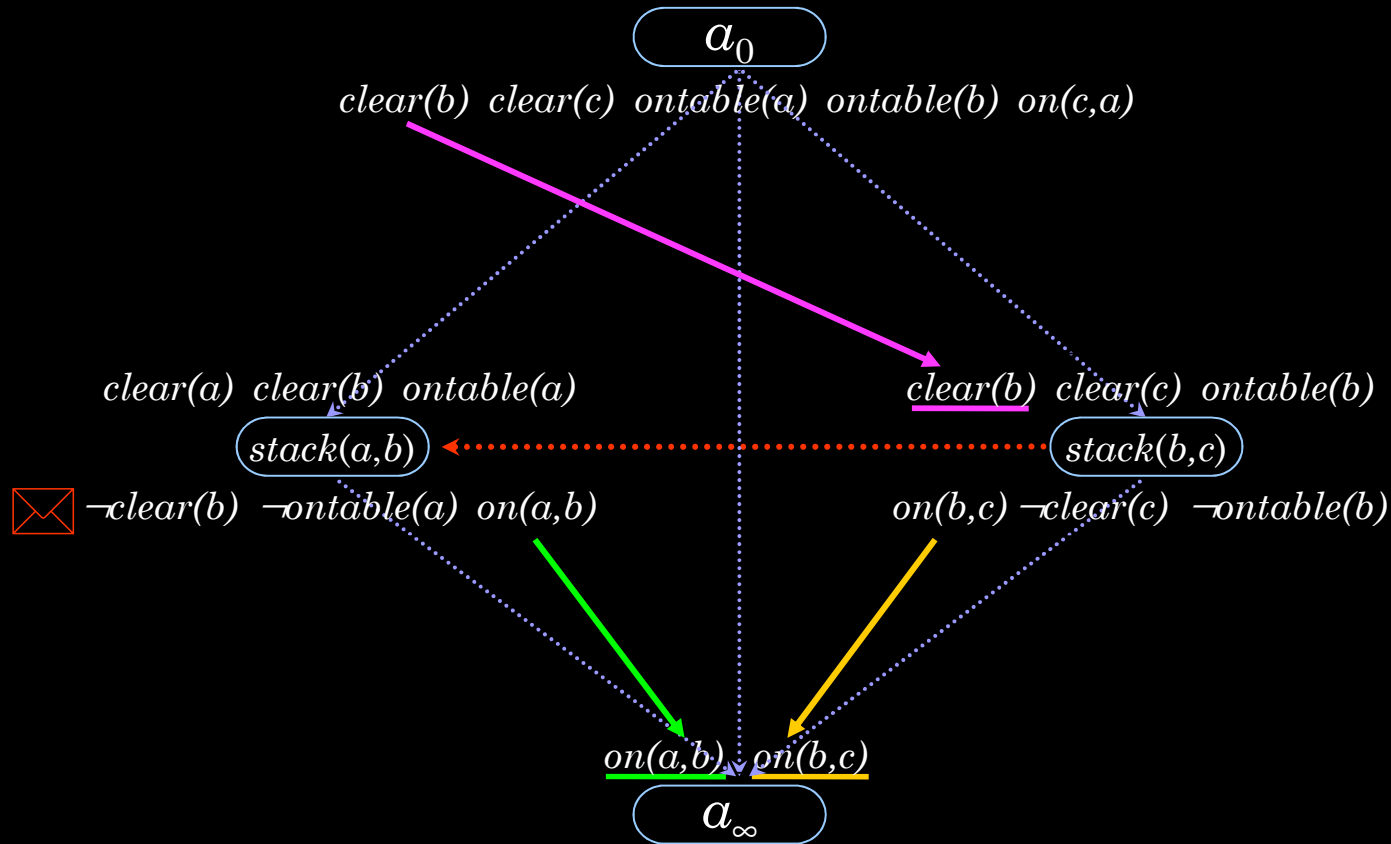
Busca no espaço de planos





IME-USP

Ordem parcial, vínculos e ameaças



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

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

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



IME-USP

Políticas de proteção de submetas

- **POP**
 - trata apenas ameaças negativas
 - refina apenas planos consistentes
- **SNLP**
 - trata ameaças positivas e negativas
 - refina apenas planos consistentes
 - nunca examina um mesmo plano mais de uma vez
- **TWEAK**
 - trata apenas parte das ameaças negativas
 - refina planos consistentes e inconsistentes
 - pode examinar um mesmo plano várias vezes



IME-USP

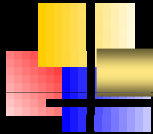
Considerações

- planejamento de ordem parcial é mais eficiente que planejamento de ordem total;
- planejamento de ordem parcial pode ser sistemático ou redundante (pode tirar vantagem das características do domínio de planejamento considerado);
- o planejamento no cálculo de situações é de ordem total, já que um termo da forma $do(\alpha_n, do(\dots, do(\alpha_1, s_0)))$ denota um plano $\langle \alpha_1, \dots, \alpha_n \rangle$ completamente ordenado;
- usando um formalismo lógico que nos permita representar planos de ordem parcial poderíamos ter especificações não apenas executáveis, mas também eficientes.



IME-USP

Planejamento abduativo



no cálculo de eventos



IME-USP

Abdução

- **Princípio:**

conhecemos: $\alpha \rightarrow \beta$

observamos: β

inferimos: α

- **Definição:**

Δ : descrição de um domínio

Γ : descrição de uma observação nesse domínio

Abdução consiste em encontrar Π tal que:

(a) $\Delta \cup \Pi \models \Gamma$

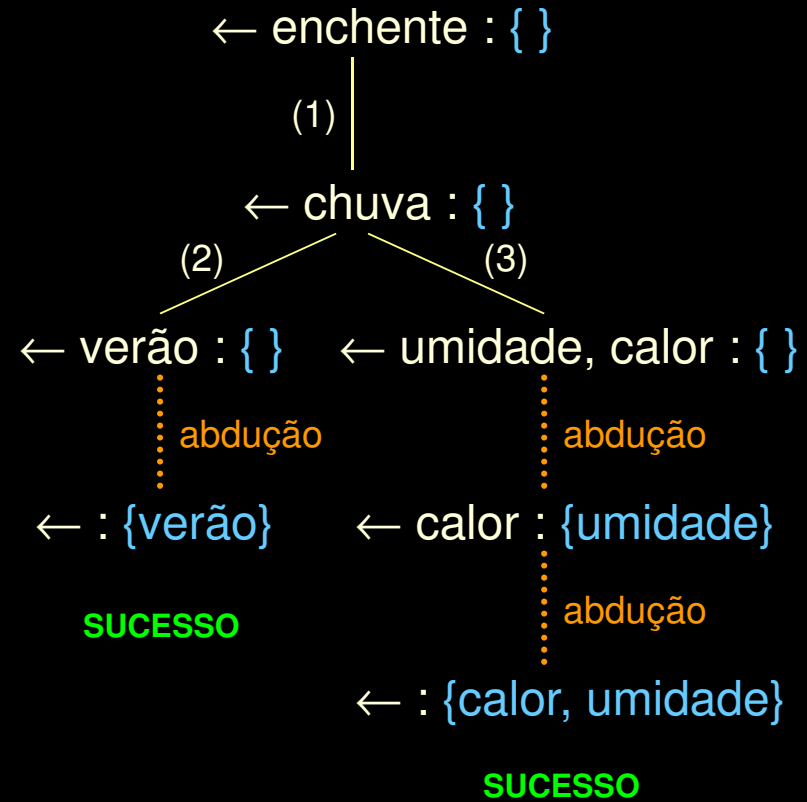
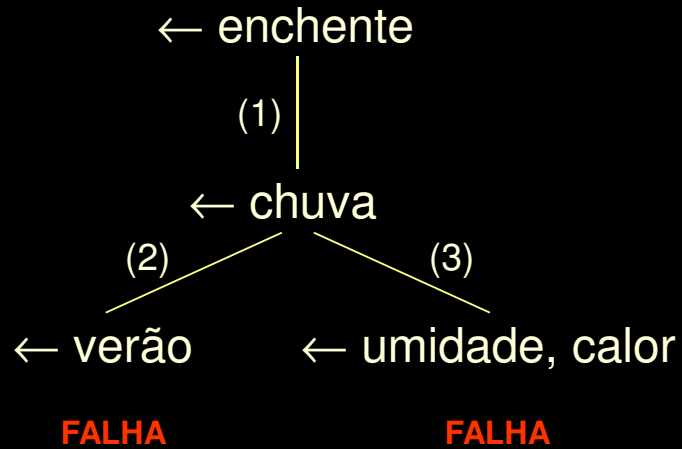
(b) $\Delta \cup \Pi$ é consistente



IME-USP

Mecanismo abdutivo em PL

- (1) enchente ← chuva
- (2) chuva ← verão
- (3) chuva ← umidade, calor





IME-USP

O meta-interpretador abductivo

```
axiom(enchente,[chuva]).  
axiom(chuva,[verão]).  
axiom(chuva,[umidade, calor]).
```

} programa-objeto no metanível

```
abducible(verão).  
abducible(umidade).  
abducible(calor).
```

} fatos abductíveis (causas primitivas)

```
dem([],R,R).
```

```
dem([G1|Gs1],R1,R2) :-  
    axiom(G1,Gs2),  
    append(Gs2,Gs1,Gs3),  
    dem(Gs3,R1,R2).
```

} meta-interpretador abductivo básico

```
dem([G1|Gs1],R1,R2) :-  
    abducible(G1),  
    dem(Gs1,[G1|R1],R2).
```



IME-USP

Cálculo de eventos

- **Ontologia:** eventos, fluentes e tempo
- **Linguagem:**
 - $initiates(\alpha, \phi, \tau)$, $terminates(\alpha, \phi, \tau)$ e $releases(\alpha, \phi, \tau)$
 - $initially_p(\phi)$ e $initially_n(\phi)$
 - $happens(\alpha, \tau_1, \tau_2)$
 - $holdsAt(\phi, \tau)$
 - $clipped(\tau_1, \phi, \tau_2)$ e $declipped(\tau_1, \phi, \tau_2)$



IME-USP



Axiomas do cálculo de eventos

$$\begin{aligned} \text{holdsAt}(F, T) \leftarrow & \quad \text{[EC1]} \\ & \text{initially}_p(F) \wedge \neg \text{clipped}(0, F, T) \end{aligned}$$

$$\begin{aligned} \text{holdsAt}(F, T) \leftarrow & \quad \text{[EC2]} \\ & \text{happens}(A, T_1, T_2) \wedge \text{initiates}(A, F, T_1) \wedge \\ & (T_2 < T) \wedge \neg \text{clipped}(T_1, F, T) \end{aligned}$$

$$\begin{aligned} \text{clipped}(T_1, F, T_2) \leftrightarrow & \quad \text{[EC5]} \\ & \exists A, T_3, T_4 [\text{happens}(A, T_3, T_4) \wedge (T_1 < T_3) \wedge (T_4 < T_2) \wedge \\ & (\text{terminates}(A, F, T_3) \vee \text{releases}(A, F, T_3))] \end{aligned}$$

$$\text{happens}(A, T_1, T_2) \rightarrow T_1 \leq T_2 \quad \text{[EC7]}$$



IME-USP



Especificação lógica do domínio

$initially_p(clear(c))$

$initially_p(on(c,a))$

...

$initiates(move(X,Y,Z), clear(Y), T) \leftarrow$

$holdsAt(clear(X), T) \wedge$

$holdsAt(clear(Z), T) \wedge$

$holdsAt(on(X,Y), T) \wedge$

$X \neq Z$

...

$terminates(move(X,Y,Z), clear(Z), T) \leftarrow$

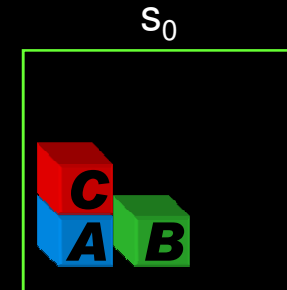
$holdsAt(clear(X), T) \wedge$

$holdsAt(clear(Z), T) \wedge$

$holdsAt(on(X,Y), T) \wedge$

$X \neq Z$

...





IME-USP

Planejamento abduutivo

Sejam

- Δ um domínio (*initiates*, *terminates* e *releases*)
- Σ uma situação inicial (*initially_n* e *initially_p*)
- Γ uma meta de planejamento (*holdsAt*)
- EC** a conjunção dos axiomas do cálculo de eventos

O planejamento abduutivo consiste em encontrar um conjunto de fatos Π (*happens* e *before*) tal que:

- (a) $\text{CIRC}[\Delta; \textit{initiates}, \textit{terminates}, \textit{releases}] \wedge \text{CIRC}[\Sigma \wedge \Pi; \textit{happens}] \wedge \text{EC}$ é consistente
- (b) $\text{CIRC}[\Delta; \textit{initiates}, \textit{terminates}, \textit{releases}] \wedge \text{CIRC}[\Sigma \wedge \Pi; \textit{happens}] \wedge \text{EC} \models \Gamma$



IME-USP

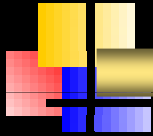
Considerações

- usando o cálculo de eventos podemos descrever formalmente um domínio;
- usando prova automática de teoremas estendida com abdução, essa descrição corresponde a uma implementação de um sistema de planejamento de ordem parcial para o domínio;
- esse sistema pode ser mais facilmente entendido e modificado, resta investigar se é tão eficiente quanto os sistemas de planejamento baseados em STRIPS.



IME-USP

Resultados experimentais



abordagem lógica vs. algorítmica



IME-USP

Sistemas implementados

- **Planejadores algorítmicos:** POP, SNLP e TWEAK
- **Planejadores abduativos:** ABP, SABP e RABP
- **Cuidados de Implementação:**
 - empregar as mesmas estruturas de dados e os mesmos recursos de programação, sempre que possível;
 - manter a equivalência de acesso à representação de ações no metanível;
 - implementar do cálculo de eventos apenas aspectos relevantes para o planejamento clássico.



IME-USP

Experimento I

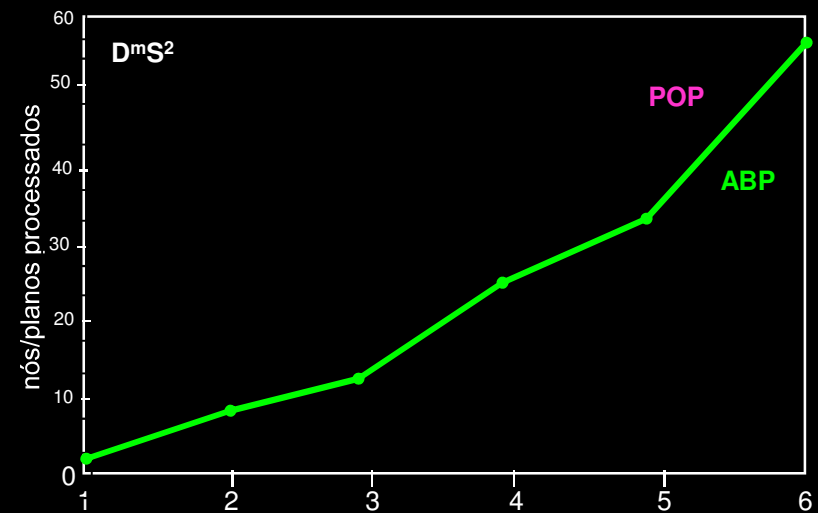
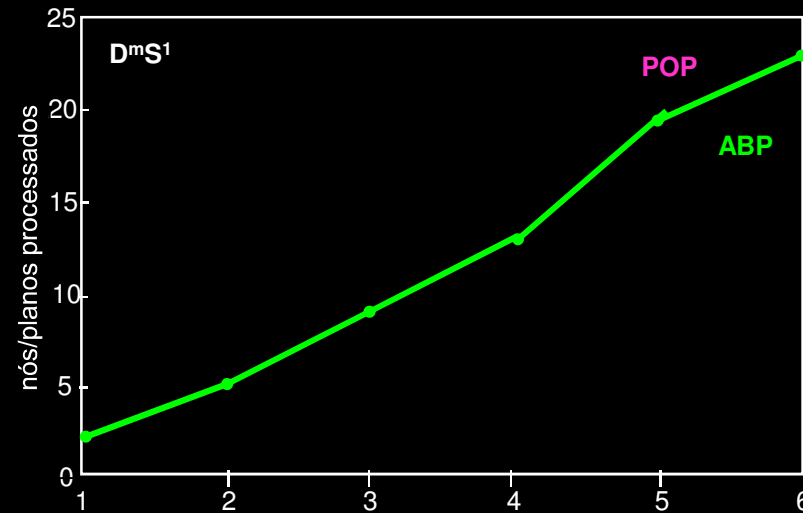
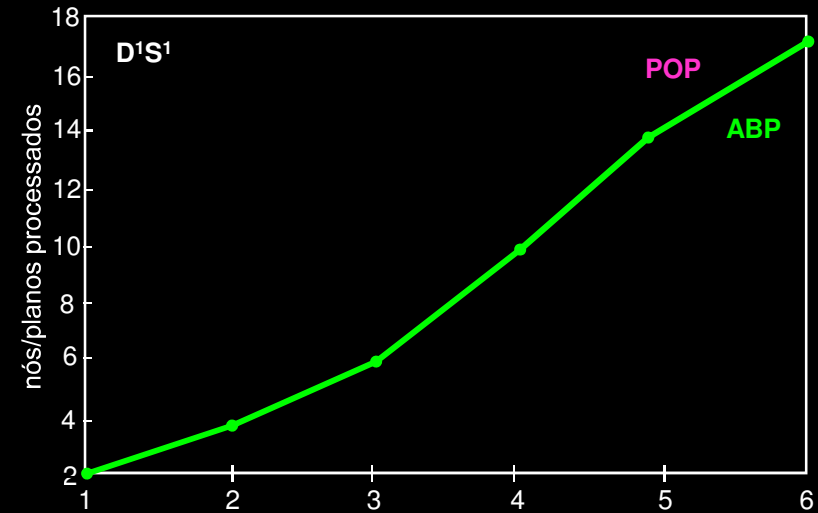
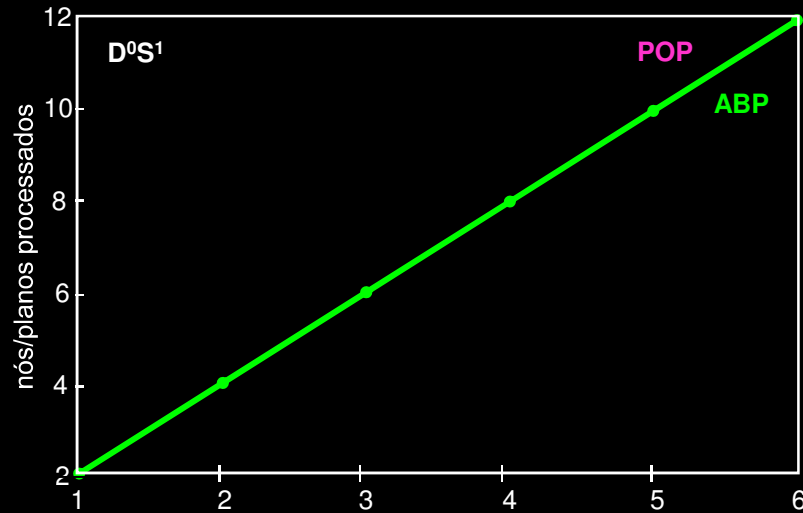
- **Objetivo:**
 - mostrar que raciocínio abduativo no cálculo de eventos é isomorfo a planejamento de ordem parcial
- **Sistemas comparados:**
 - POP \times ABP
- **Domínios de teste (*Barret & Weld*):**
 - independentes: D^0S^1
 - serializáveis: D^1S^1 e D^mS^1
 - não-serializáveis: D^mS^2
- **Dados levantados para cada sistema e domínio:**
 - o espaço de busca explorado
 - o tempo médio de CPU consumido



IME-USP



Equivalência entre os métodos





IME-USP



Equivalência qualitativa

```
plan([step(42, unstack(c,a)),  
      step(40, stack(b,c)),  
      step(18, stack(a,b))],  
     [42<40, ...], [link(i,on(c,a),42), ...])
```

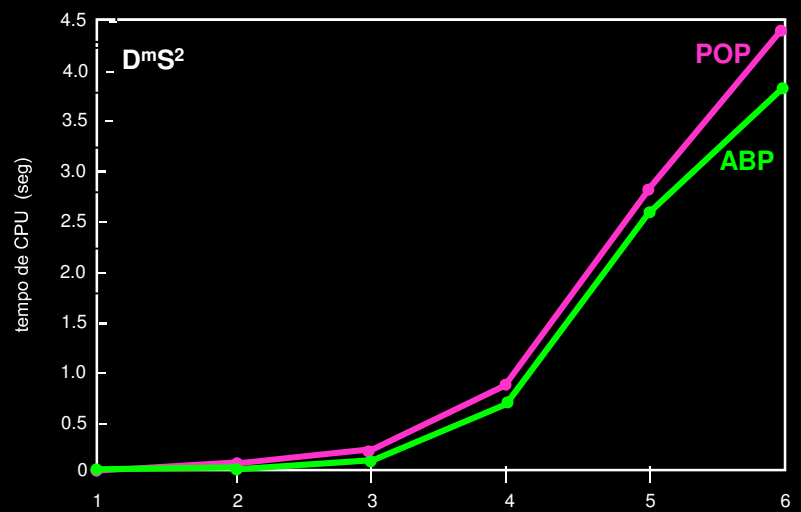
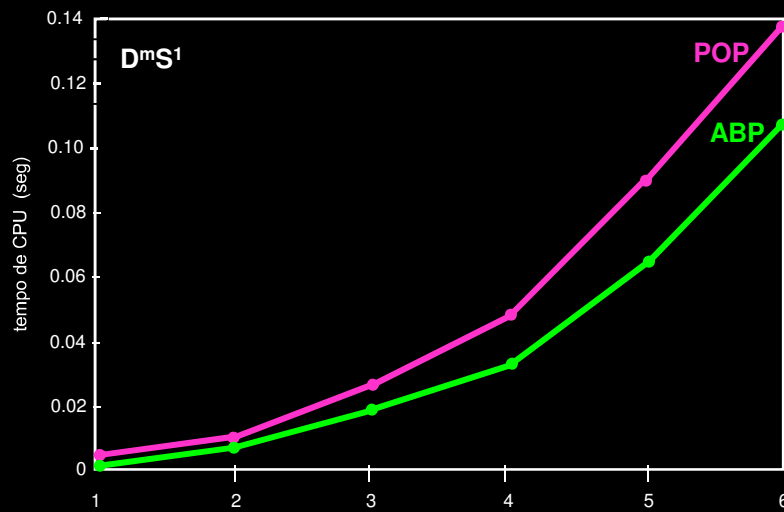
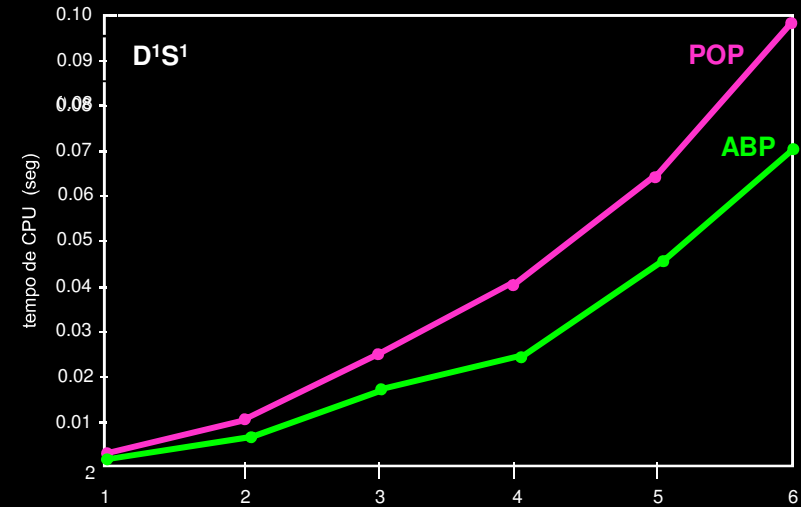
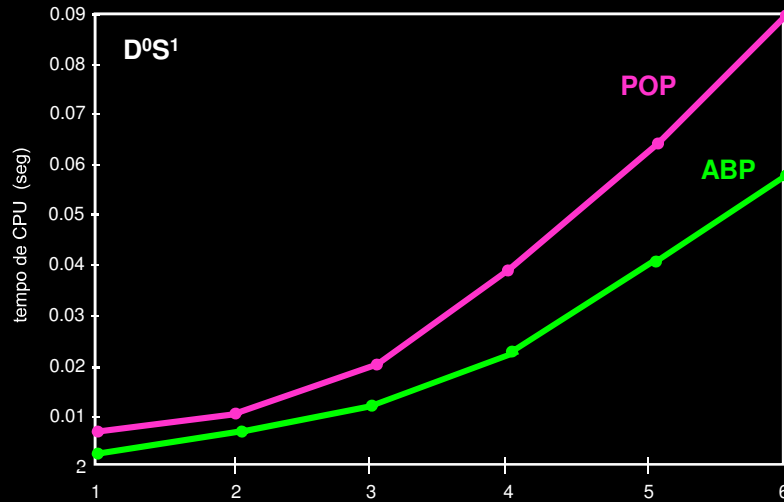
```
res([happens(unstack(c,a), 42),  
     happens(stack(b,c), 40),  
     happens(stack(a,b), 18)],  
    [before(42,40), ...], [clipped(0,on(c,a),42), ...])
```



IME-USP

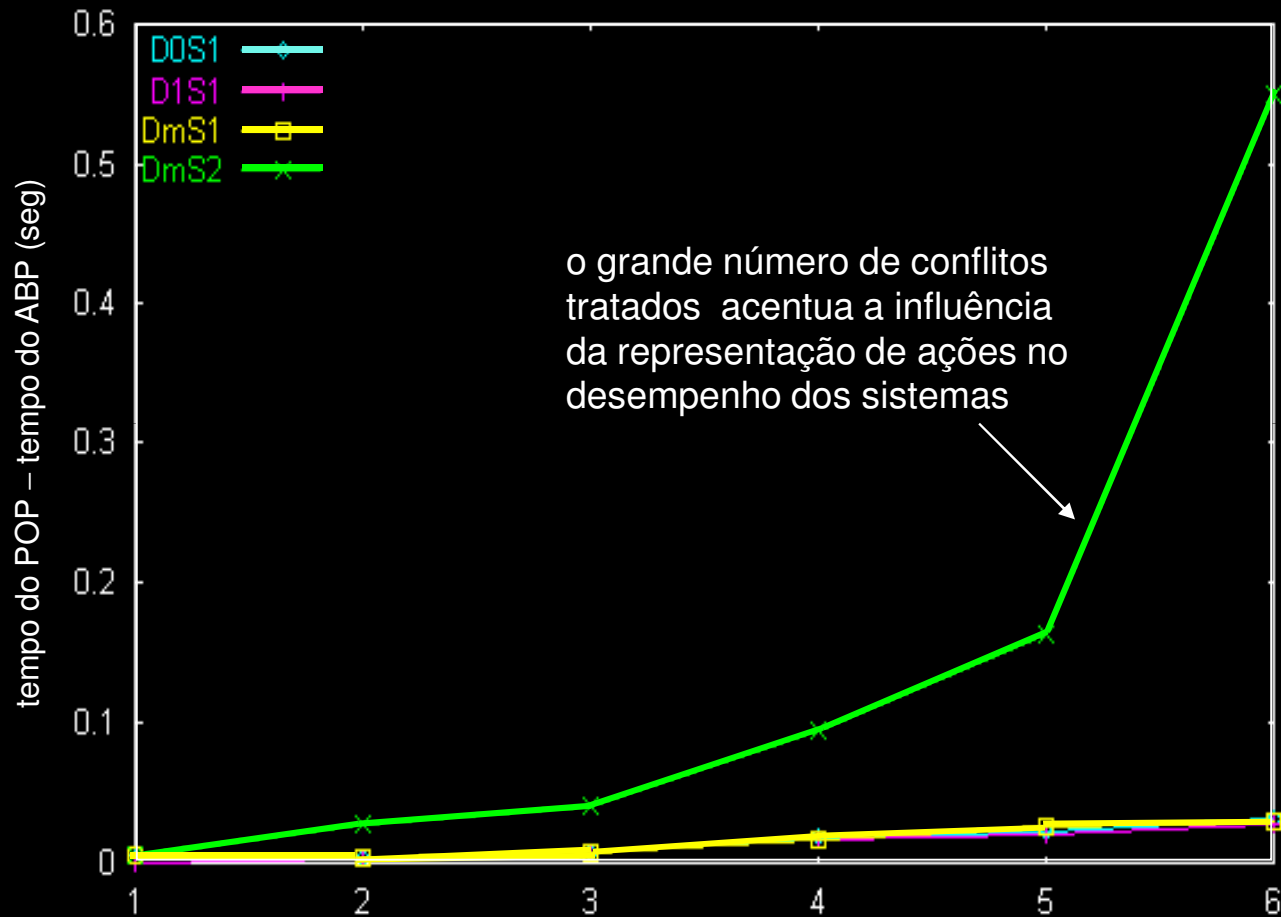


Desempenho dos planejadores





Influência da representação





IME-USP

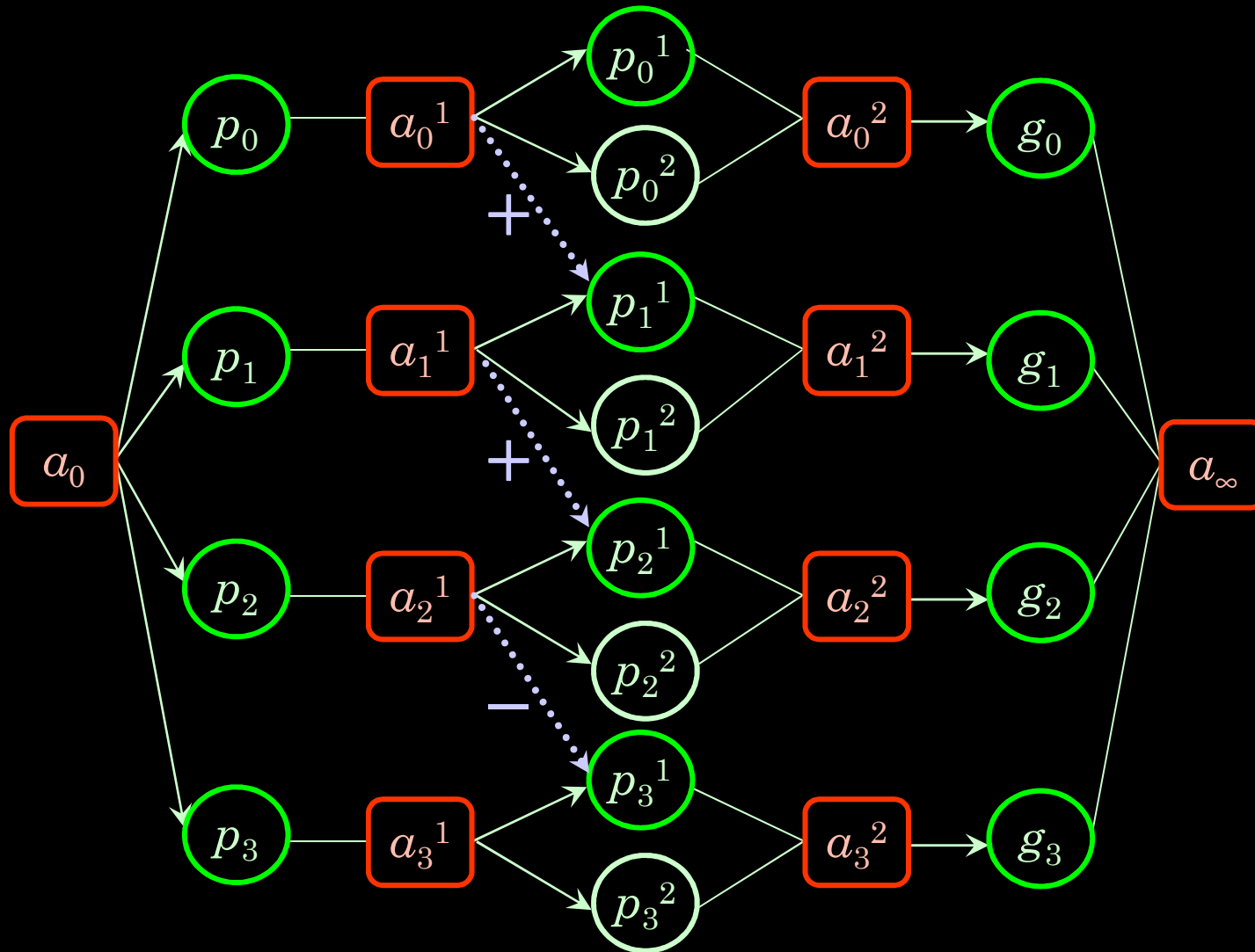
Experimento II

- **Objetivo:**
 - mostrar que a eficiência de planejamento depende não apenas do método (sistemático ou redundante), mas também das características do domínio considerado.
- **Sistemas comparados:**
 - POP × ABP, SNLP × SABP e TWEAK × RABP
- **Domínio de teste (baseado em *Knoblock & Yang*):**
 - complexidade variável: $A \times D^y S^2$
- **Dados levantados para cada sistema:**
 - o espaço de busca explorado
 - o tempo médio de CPU consumido



IME-USP

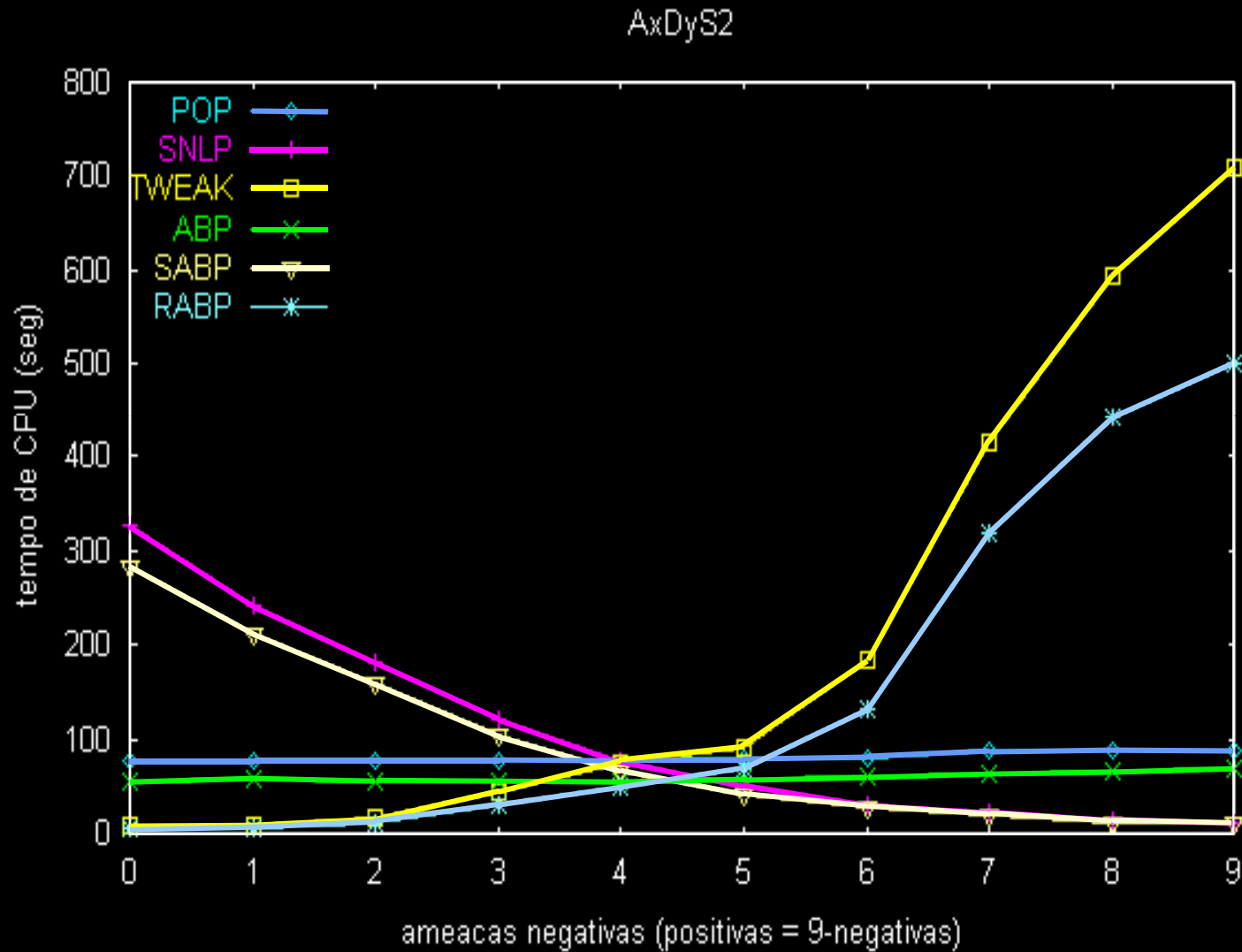
A família de domínios $A^x D^y S^z$





IME-USP

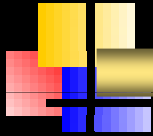
Sistematicidade vs. redundância





IME-USP

Conclusão





IME-USP

Principais resultados

- raciocínio abdutivo no cálculo de eventos é isomorfo a planejamento de ordem parcial;
- um planejador abdutivo pode implementar métodos de planejamento sistemático ou redundante;
- a eficiência de planejamento depende tanto do método implementado (sistemático ou redundante) quanto das características específicas do domínio;
- um planejador pode ser calibrado, de acordo com as características do domínio, de modo a ter sempre o melhor desempenho.



IME-USP



Trabalhos futuros

- planejamento hierárquico
- programação de agentes (robótica cognitiva)
- outros:
 - representação ADL
 - multi-contribuidores
 - consumo de recursos
 - mudança contínua



IME-USP

Fim

