MAC5788
Planejamento em IA

Linguagens de Planejamento

Leliane Nunes de Barros

Linguagens de Planejamento
Expressividade e extensões

- Restrições da linguagem STRIPS
  => uso de algoritmos simples e eficientes
  => fácil descrever problemas reais
  => restrição importante: literais livres de funções
  - uma ação schema pode ser traduzida para um conjunto finito de ações proposicionais (sem variáveis)
- Problema: STRIPS não é suficientemente expressivo para alguns domínios reais.
- Novas versões de linguagens para planejamento:
  => ADL (Action Description Language)
  => PDDL (Planning Domain Definition Language

Linguagem STRIPS	**Linguagem** ADL
Somente literais positivos no estado: $Pobre \wedge Desconhecido$	Literais positivos e negativos no estado: $\neg Rico \wedge \neg Famoso$
Suposição de Mundo Fechado (*CWA*): Literais não presentes são falsos	Suposição de Mundo Aberto: Literais não presentes são desconhecidos
Efeito $P \wedge \neg Q$ significa: adicione P e elimine Q	Efeito $P \wedge \neg Q$ significa: adicione P e $\neg Q$ e elimine $\neg P$ e Q
Somente literais sem variáveis na meta: $Rico \wedge Famoso$	Variáveis quantificadas na meta: $\exists x\ Em(P_1, x) \wedge Em(P_2, x)$ é uma meta para P_1 e P_2 estarem no mesmo lugar.
Metas são conjunções: $Rico \wedge Famoso$	Metas são conjunções e disjunções: $\neg Pobre \wedge (Famoso \vee Inteligente)$
Efeitos são conjunções.	Efeitos condicionais: **Quando** $P : E$, significa E é um efeito da ação somente se P for verdade.
Não permite igualdade.	Predicado igualdade ($x = y$) pode ser usado nas pré=condições.
Não permite tipos de variáveis.	Variáveis podem ter tipos, por exemplo: (x : *Localidade*)

Exemplos

- Ação STRIPS:

  Action(Fly(p,from,to),

  Precond: $At(p, from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)$

  Effect: $\neg At(p, from) \wedge At(p, to)$

- Ação ADL:

  Action(Fly(p : Plane, from : Airport, to : Airport),

  Precond: $At(p, from) \wedge (from \neq to)$

  Effect: $\neg At(p, from) \wedge At(p, to)$

Problema da ramificação

- STRIPS e ADL: não permitem representar a **ramificação** de ações de uma maneira natural. Exemplo da valise.
- Planejamento clássico não trata o problema => propriedades não representadas podem fazer com que as ações falhem.
- Necessidade de **restrições de estados**

PDDL

- Sintaxe padrão para permitir a troca de problemas *benchmark* e comparar resultados (competição de planejamento)
- PDDL inclue Strips, ADL, Redes de Tarefas Hierárquicas, recursos, tempo e metas com otimizações.

PDDL: ação Strips

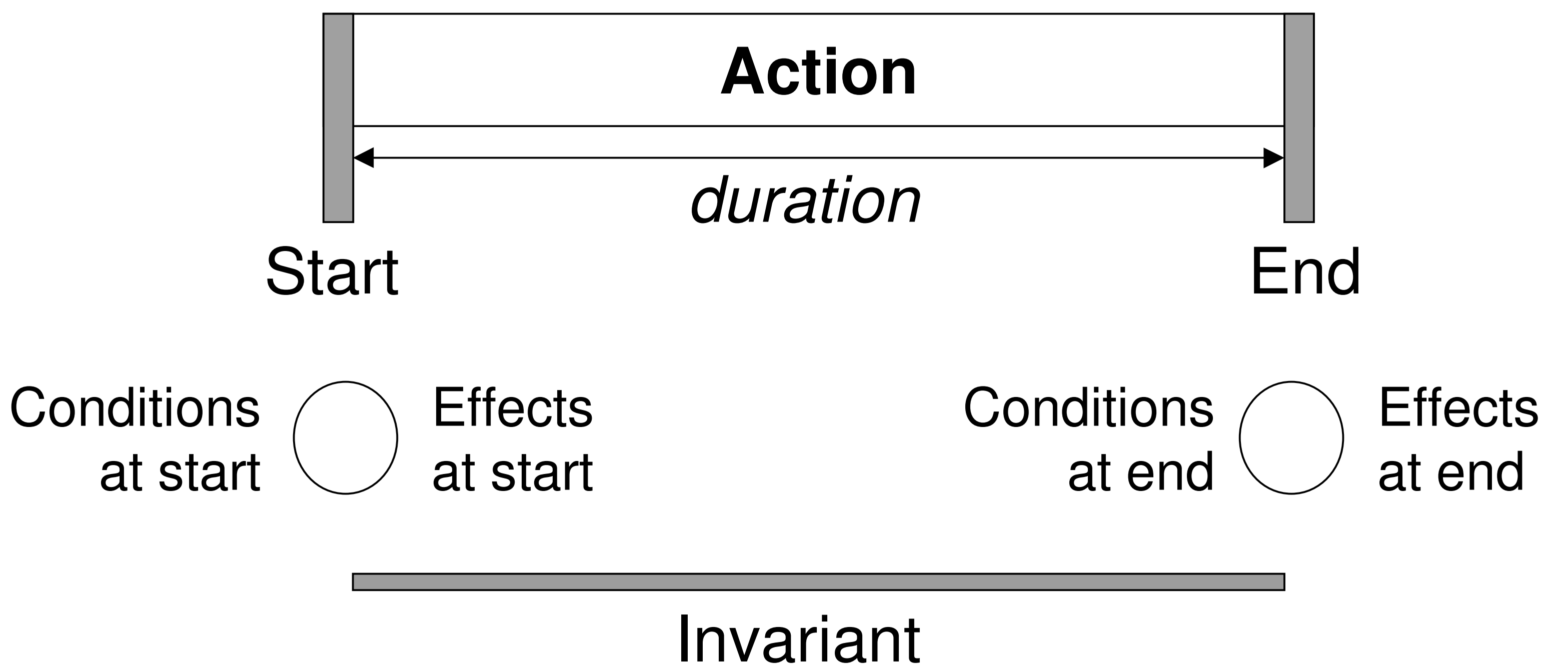
```
(:action turn_to
    :parameters (?s - satellite ?d_new - direction ?d_prev - direction)
    :precondition (and (pointing ?s ?d_prev)
                        (not (= ?d_new ?d_prev))
                        )
    :effect (and  (pointing ?s ?d_new)  (not (pointing ?s ?d_prev))
              )
 )
```

PDDL 2.1: ação Strips-numérico

```
(:action turn_to
   :parameters (?s - satellite ?d_new - direction ?d_prev - direction)
   :precondition (and  (pointing ?s ?d_prev)
                       (not (= ?d_new ?d_prev))
                       (>= (fuel ?s) (slew_time ?d_new ?d_prev))
                   )
   :effect (and    (pointing ?s ?d_new)
                   (not (pointing ?s ?d_prev))
                   (decrease (fuel ?s) (slew_time ?d_new ?d_prev))
                   (increase (fuel-used) (slew_time ?d_new ?d_prev))
              )
 )
```

PDDL 2.1: ação durativa

- Ações durativas (tempo)



PDDL 2.1: ação Strips-temporal

```
(:durative-action turn_to
    :parameters (?s - satellite ?d_new - direction ?d_prev - direction)
    :duration (= ?duration 5)
    :condition (and         (at start (pointing ?s ?d_prev))
                            (over all (not (= ?d_new ?d_prev)))
                )
    :effect (and    (at end (pointing ?s ?d_new))
                    (at start (not (pointing ?s ?d_prev)))
            )
)
```

PDDL 2.1: ação Strips-temporal*

```
(:durative-action turn_to
    :parameters (?s - satellite ?d_new - direction ?d_prev - direction)
    :duration (= ?duration (slew_time ?d_prev ?d_new))
    :condition (and          (at start (pointing ?s ?d_prev))
                             (over all (not (= ?d_new ?d_prev)))
               )
    :effect (and    (at end (pointing ?s ?d_new))
                    (at start (not (pointing ?s ?d_prev)))
            )
)
```

PDDL 2.1: ação Strips-temporal*

```
(:durative-action take_image
    :parameters (?s - satellite ?d - direction ?i - instrument ?m - mode)
    :duration (= ?duration 7)
    :condition (and          (over all (calibrated ?i))
                    (over all (on_board ?i ?s))
                    (over all (supports ?i ?m) )
                    (over all (power_on ?i))
                    (over all (pointing ?s ?d))
                    (at end (power_on ?i))
                    (at start (>= (data_capacity ?s) (data ?d ?m)))
                  )
    :effect (and    (at start (decrease (data_capacity ?s) (data ?d ?m)))
                    (at end (have_image ?d ?m))
                    (at end (increase (data-stored) (data ?d ?m)))        )
```

EP2

- EP2: specificação do problema ROADEF como um problema de planejamento. Escrever um texto detalhado, incluindo uma especificação de ações.
- Ler a especificação do PDDL antes de propor a sua.