Capítulo 13: Planejamento e Ação

Duas formas de lidar com informação **incompleta** e **incorreta**:

- **Planejamento condicional:** plano condicional; uso de **ações sensórias**.
- **Monitoramento da execução:** agente monitora o que está acontecendo e faz **replanejamento**.

Planejamento Condicional

O problema da troca de pneu:

- Estado inicial:

$$
\begin{aligned}
&Op(\text{ACTION}: \mathit{Start},\\
&\quad \text{EFFECT}: \mathrm{inflated(Spare)} \land \mathrm{intact(Spare)} \land\\
&\qquad \mathrm{off(Spare)} \land \mathrm{on(Tire_1)} \land \mathrm{flat(Tire_1)})
\end{aligned}
$$

- Estado meta:

$$
\begin{aligned}
&Op(\text{ACTION}: \mathit{Finish},\\
&\quad \text{PRECOND}: \mathrm{on}(x) \land \mathrm{inflated}(x))
\end{aligned}
$$

O problema da troca de pneu:

- Ações:

$$Op(\text{ACTION}: \mathit{Remove}(x), \text{PRECOND}: \text{on}(x),$$
$$\text{EFFECT}: \text{off}(x) \land \neg\text{on}(x) \land \text{clearhub})$$

$$Op(\text{ACTION}: \mathit{PutOn}(x),$$
$$\text{PRECOND}: \text{off}(x) \land \text{clearhub},$$
$$\text{EFFECT}: \text{on}(x) \land \neg\text{off}(x)) \land \neg\text{clearhub})$$

$$Op(\text{ACTION}: \mathit{Inflate}(x),$$
$$\text{PRECOND}: \text{intact}(x) \land \text{flat}(x),$$
$$\text{EFFECT}: \text{inflated}(x) \land \neg\text{flat}(x))$$

Um plano:

$$[\mathit{Remove}(\text{Tire}_1), \mathit{PutOn}(\text{Spare})]$$

Um plano melhor com um passo **condicional**:

$$\mathit{If}(\mathit{intact}(\text{Tire}_1), [\mathit{Inflate}(\text{Tire}_1)],$$
$$[\mathit{Remove}(\text{Tire}_1), \mathit{PutOn}(\text{Spare})])$$

Um agente de planejamento condicional:

Figura 13.1

Como o agente "sabe" a veracidade de uma condição:

- Consulta a BC.
- Outra possibilidade: Insere ações que possibilitem determinar a condição (**ação sensória**).

  Exemplo:

  $$\begin{aligned}&Op(\text{ACTION}: ChekTire(x),\\&\quad \text{PRECOND}: \text{tire}(x),\\&\quad \text{EFFECT}: \text{KnowsWhether}(\text{"}intact(x)\text{"}))\end{aligned}$$

Gerando planos condicionais:

- **Contexto** de um passo: união das condições para o passo ser executado.

Plano inicial:

Figura 13.2

Satisfazendo as 2 precondições abertas de *Finish*:

- *on*(x) via *Start*; e
- *inflated*(x) via *inflate*(Tire_1).

e satisfazendo a precondição *flat*(Tire_1) de *inflate*(Tire_1):

Figura 13.3

Não há ação com *effect* $intact(x)$:

- Planejador padrão: *backtracking*.
- Planejador condicional: a ação sensória $ChekTire(x)$ permite saber a veracidade de uma proposição que unifica com $intact(\text{Tire}_1)$.

  Adiciona-se o **passo condicional** $ChekTire(\text{Tire}_1)$:

Figura 13.4

Adicionando outro *Finish* com contexto oposto:

Figura 13.5

Precondição de *Inflate*(Tire_1) contradiz contexto de *Finish*. Resta usar *Start*:

Figura 13.6

Os passos *Remove*(Tire_1) e *PutOn*(Spare) são adicionados para satisfazer on(Spare) com contexto *True*.

O passo *Remove*(Tire_1) ameaça $Start \stackrel{\text{on}(\text{Tire}_1)}{\longrightarrow} Finish$.
Condicionamento (alternativa a rebaixamento e promoção):

- Achar um passo condicional que faça os contextos dos dois passos incompatíveis.

Figura 13.7

O algoritmo CPOP:

Figura 13.8

Monitorando a execução:

- Verificar se precondições se tornaram satisfeitas/insatisfeitas.

Monitoramento da execução:

Verifica *precondições restantes do plano.*

(prevê falhas futuras)

Monitoramento da ação:

Verifica *precondições da próxima ação* a executar.

(mais simples)

Em ambos os casos: *replanejamento* é necessário.

Replanejamento:

- Replanejar a partir do zero.
- Aproveitar o máximo do plano original.

Planejamento e execução integrados

Exemplo: mundo dos blocos

$$\begin{aligned}
&Op(\text{ACTION}: Move(x,y),\\
&\quad \text{PRECOND}: \text{clear}(x) \wedge \text{clear}(y), \text{on}(x,z),\\
&\quad \text{EFFECT}: \text{on}(x,y) \wedge \text{clear}(z) \wedge \neg \text{on}(x,z) \wedge \neg \text{clear}(y))
\end{aligned}$$

Figura 13.10

Estado inicial: (a). Plano:

Figura 13.11

Um agente externo muda estado para (b); clear(B) e on(D,G) não são mais verdadeiros:

Figura 13.12

A ligação causal para on(D, B) de *Finish* pode ser suprida por *Start*; remove-se o passo redundante $Move(D, B)$:

Figura 13.13

O agente retira $Move(C, D)$ do plano para ser executado, mas deixa cair C sobre A ao invés de D (estado (c)):

Figura 13.14

A precondição aberta on(C, D) de *Finish* é satisfeita por $Move(C, D)$:

Figura 13.14

Desta vez o agente é mais cuidadoso. Resultado: estado (d). O passo é retirado:

Figura 13.15