planeamento

planeamento

anteriormente:

- planeamento de ações num espaço de estados específico
- planeamento de ações numa KB, baseado em lógica

planeamento clássico

- estado do mundo com representação fatorizada em variáveis

planeamento – assunções

- o mundo é determinístico

    não há acontecimentos externos às ações do agente que façam alterar o estado do mundo

- o agente sabe o estado em que está

    o mundo é completamente observável

- o tempo é discreto orientado por eventos $t \rightarrow t+1$ (passo seguinte)

    independentemente do tempo que demora cada passo

- os objetivos são predicados de estados que devem ser atingidos

o planeamento é encontrar uma sequência de ações para atingir o objetivo

representação de estados

em PDDL (*Planning Domain Definition Language*) simples

estado é representado por uma conjunção de predicados, objetos ou variáveis

mundo fechado
átomos não mencionados
são falsos

noite

em(*casa, manuel*)

aberta(*porta*) ∧ *acesa*(*luz*) ∧ *sentado*(*manuel*) ∧ *ler*(*manuel*)

¬ *aberta*(*janela*) ∧ *ligado*(*rádio*)

ações

representadas por um esquema de ação

nome; lista de variáveis; pré-condições; efeito

ação(*voar*(*P*, *De*, *Para*),

PRÉ-COND: *em*(*P*, *De*) ∧ *avião*(*P*) ∧ *aeroporto*(*De*) ∧ *aeroporto(Para)*

EFEITO: ¬*em*(*P*, *De*) ∧ *em*(*P*, *Para*))

tempo: pré-condições referem-se a tempo *t*
efeitos referem-se a tempo *t+1*

ações instanciadas

ação(*voar*(p_1, LIS, PDL),

PRÉ-COND: *em*(p_1, LIS) ∧ *avião*(p_1) ∧ *aeroporto*(*LIS*) ∧ *aeroporto*(PDL)

EFEITO: ¬*em*(p_1, LIS) ∧ *em*(p_1, PDL))

resultado:

removem-se as afirmações negativas no efeito: DEL(*a*) – lista de remoção

ex: ¬*em*(p_1, LIS)

adicionam-se as afirmações positivas no efeito: ADD(*a*) – lista de adição

ex: *em*(p_1, PDL)

$$\text{RESULT}(s, a) = (s - \text{DEL}(a)) \cup \text{ADD}(a).$$

exemplo – carga aérea

inic(*em*(c_1, LIS) ∧ *em*(c_2, PDL) ∧ *em*(p_1, LIS) ∧ *em*(p_2, PDL) ∧ *carga*(c_1) ∧ *carga*(c_2) ∧ *avião*(p_1) ∧ *avião*(p_2) ∧ *aeroporto*(LIS) ∧ *aeroporto*(PDL))

objetivo(*em*(c_1, PDL) ∧ *em*(c_2, LIS))

ação(*carrega*(*C*, *P*, *A*),

  PRÉ-COND: *em*(*C*, *A*) ∧ *em*(*P*, *A*) ∧ *carga*(*C*) ∧ *avião*(*P*) ∧ *aeroporto(A)*

  EFEITO: ¬ *em*(*C*, *A*) ∧ *no*(*C*, *P*))

ação(*descarrega*(*C*, *P*, *A*),

  PRÉ-COND: *no*(*C*, *P*) ∧ *em*(*P*, *A*) ∧ *carga*(*C*) ∧ *avião*(*P*) ∧ *aeroporto*(*A*)

  EFEITO: *em*(*C*, *A*) ∧ ¬ *no*(*C*, *P*))

ação(*voar*(*P*, *De*, *Para*),

  PRÉ-COND : *em*(*P*, *De*) ∧ *avião*(*P*) ∧ *aeroporto*(*De*) ∧ *aeroporto*(*Para*)

  EFEITO : ¬ *em*(*P*, *De*) ∧ *em*(*P*, *Para*))

exemplo – carga aérea

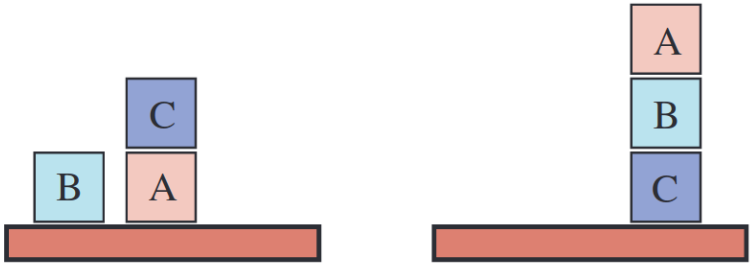
"*em*" - disponível para uso no respetivo local

"*no*" - dentro do avião

um plano

[$carrega(c_1, p_1, \text{LIS})$, $voa(p_1, \text{LIS}, \text{PDL})$, $descarrega(c_1, p_1, \text{PDL})$, $carrega(c_2, p_2, \text{PDL})$, $voa(p_2, \text{PDL}, \text{LIS})$, $descarrega(c_2, p_2, \text{LIS})$].

o mundo dos blocos



Start State

Goal State

inic(*sobre*(*a*, *mesa*) ∧ *sobre*(*b*, *mesa*) ∧ *sobre*(*c*, *a*) ∧ *bloco*(*a*) ∧ *bloco*(*b*) ∧ *bloco*(*c*) ∧ *livre*(*b*) ∧ *livre*(*c*))

objetivo(*sobre*(*a*, *b*) ∧ *sobre*(*b*, *c*))

ação(*move*(*B*, *X*, *Y*),

PRÉ-COND: *sobre*(*B*, *X*) ∧ *livre*(*B*) ∧ *livre*(*Y*) ∧ *bloco*(*B*) ∧ *bloco*(*Y*) ∧ (*B*≠*X*) ∧ (*B*≠*Y*) ∧ (*X*≠*Y*)

EFEITO: *sobre*(*B*, *Y*) ∧ *livre*(*X*) ∧ ¬ *sobre*(*B*, *X*) ∧ ¬ *livre*(*Y*))

ação(*moveParaMesa*(*B*, *X*),

PRÉ-COND: *sobre*(*B*, *X*) ∧ *livre*(*B*) ∧ *bloco*(*B*) ∧ (*B*≠*X*)

EFEITO: *sobre*(*B*, *mesa*) ∧ *livre*(*X*) ∧ ¬ *sobre*(*B*, *X*))

algoritmos de planeamento

procura adiante

- a partir do estado inicial
- usar as ações definidas para avançar até ao objetivo

- espaço de procura cresce exponencialmente com fator de ramificação

- mas podem derivar-se heurísticas fortes independentes do contexto
- ⇒ viabilizam a procura adiante

procura retrógrada ou procura pelos estados relevantes

começa no objetivo

necessita de saber como retroceder no espaço de estados

em PDDL é simples:

dado um estado *s* e uma ação *a* que levou a esse estado, a regressão ao estado anterior *s'* é dada por

$$s' = (s - \textsc{Add}(a)) \cup \textsc{Pre-cond}(a).$$

i.e. os efeitos adicionados pela ação não precisam ser verdadeiros antes mas as pré-condições da ação tinham de ser verdadeiras

lidar com instanciações parciais

ex. entregar uma carga específica em LIS: $em(c_2, \text{LIS})$

$ação(descarrega(c_2, P', \text{LIS}),$

Pré-cond: $no(c_2, P') \land em(P', \text{LIS}) \land carga(c_2) \land avião(P') \land aeroporto(LIS)$

Efeito: $em(c_2, LIS) \land \neg\, no(c_2, P'))$

representa entregar uma carga específica num aeroporto específico por um avião não específico

a regressão sobre o estado dá

$s' = \text{no}(c_2, P') \land em(P', \text{LIS}) \land carga(c_2) \land avião(P') \land aeroporto(LIS)$

Ciências ULisboa

estados candidatos à regressão

têm de ter efeitos da ação que unifiquem com os do estado a regredir, pelo menos 1 deles

e não podem ter efeitos da ação que neguem algum efeito do estado a regredir

ex: $s = b \land c \land d$, e uma ação *a* com efeitos $b \land c \land \neg d$

porque nesse caso a ação *a* não leva ao estado *s* e seria necessária, pelo menos, mais uma ação, para lá chegar

adiante vs. retrógrada

o facto de a procura retrógrada trabalhar com conjuntos de estados, em vez de apenas com estados específicos, torna mais difícil encontrar boas heurísticas

a procura adiante é mais utilizada porque permite aplicar heurísticas gerais (independentes do problema)

heurísticas de planeamento

relembrando

$h(s)$ é admissível se não sobrestimar a distância do estado s ao objetivo

e permite usar o A*

pode construir-se uma heurística admissível a partir do problema relaxado

- o custo real do problema relaxado é uma heurística admissível do problema original

heurísticas

heurística de ignorar pré-condições

todas as ações se podem aplicar em todos os estados

nº de passos ~ nº de efeitos não satisfeitos

mais exatamente:

removem-se todas as pré-condições das ações e todos os efeitos exceto os que são literais no objetivo

e conta-se o nº mínimo de ações para atingir o objetivo

- um algoritmo sôfrego aproxima a contagem bem, mas não garante admissibilidade...

pré-condições parciais

é possível ignorar apenas algumas pré-condições

ex. o puzzle de n mosaicos, com a ação deslizar (***t**ile*, ***s**pace*)

$ação(deslizar(T, S_1, S_2),$

PRÉ-COND: $em(T, S_1) \land tile(T) \land vazio(S_2) \land adjacente(S_1, S_2)$

EFEITO: $em(T, S_2) \land vazio(S_1) \land \neg em(T, S_1) \land \neg \mathrm{vazio}(S_1))$

removendo as pré-cond $vazio(S_2) \land adjacente(S_1, S_2)$

heurística de nº de mosaicos mal colocados

removendo apenas a pré-cond $vazio(S_2)$

heurística de distância de Manhattan

poder da fatorização

heurística de abstração de estados

- permite reduzir muito o nº de estados
  - mapeando vários num único
  - uma solução neste espaço de estados será mais curta do que no problema original
  - ⇒ heurística admissível
  - e é posteriormente fácil alargar a solução ao problema original

heurística da decomposição

decomposição do objetivo em diversos

- usar a heurístia do máximo dos subconjuntos de objetivos
- usar a heurística da soma dos custos dos subconjuntos

    nem sempre é admissível

- usar abstrações e resolver através de bases de dados de padrões

extra

exemplo – pneu furado

$Init(Tire(Flat) \land Tire(Spare) \land At(Flat, Axle) \land At(Spare, Trunk))$
$Goal(At(Spare, Axle))$
$Action(Remove(obj, loc),$
$\quad \text{PRECOND: } At(obj, loc)$
$\quad \text{EFFECT: } \neg At(obj, loc) \land At(obj, Ground))$
$Action(PutOn(t, Axle),$
$\quad \text{PRECOND: } Tire(t) \land At(t, Ground) \land \neg At(Flat, Axle) \land \neg At(Spare, Axle)$
$\quad \text{EFFECT: } \neg At(t, Ground) \land At(t, Axle))$
$Action(LeaveOvernight,$
$\quad \text{PRECOND:}$
$\quad \text{EFFECT: } \neg At(Spare, Ground) \land \neg At(Spare, Axle) \land \neg At(Spare, Trunk)$
$\qquad \land \neg At(Flat, Ground) \land \neg At(Flat, Axle) \land \neg At(Flat, Trunk))$

fonte: AIMA

Ciências ULisboa

exercícios

traduzir o plano anterior

identificar problemas com esse plano

como os colmatar?