

Planificacion

Problema de las busquedas ---> Frontera muy grande (Conjunto de estados y movimientos)

2 enfoques:

- Como listas de hechos (facts).
- Como valores de un cjto. de variables

Los distintos movimientos cambian el mundo en que se mueve nuestro agente. Los movimientos no son representados de una forma estructurada.

STRIPS

Representacion: conjunto de hechos (no reglas) y conjunto de movimientos que cambian este conjunto

Problema de planificacion:

1. Estados (mundo) son complejos
2. Acciones tienden a afectar ciertos hechos

Un estado es una asignacion donde todos los atomos (atomo = predicados y objetos de dominio) son verdaderos (los "relevantes").

recordar Interpretacion de una KB.

(Una forma en que el mundo podria ser)

Ej:

aprobado(cc31a), aprobado(cc42a), plan_comun, etc...

No es trivial definir cuales son las acciones a tomar

Ej: aprobar_cc31a o aprobar(X)?

Muchas acciones tienen precondiciones

Ej: mover(pieza, comedor) necesita accesible(pieza, comedor)

Las acciones cambian el estado del mundo:

Ej: ramos aprobados ya no son cursados

Problema: numero de estados posibles muy grande (2^n con n numero de atomos basicos).

Sin embargo, sabemos que las acciones tienden a afectar pocos atomos, no a todos.

Principios basicos:

- 1) Representacion facil de utilizar de los estados
- 2) Representacion de las acciones que aproveche el hecho que las acciones afectan pocos estados locales
- 3) Considerar algoritmos de planificacion que aprovechen esa localidad de los efectos de las acciones

Representacion de estados:

Representacion Explicita (verdadero = {a, b, c, ...} falso = {d, e, f, ...})

Representacion de Mundo cerrado {loc(X), right, up, al_lado(X, Y)}

!verdadero => falso

Cuando un Atomo es valido?

holds(Atom, State) :- member(Atom, State).

holds(neg(Atom), State) :- not(member(Atom, State)).

Nota: S lista de estados

HOLD : meta interprete

Def,

NOT (X) :- X, !, fail.

NOT(X).

Existen hechos derivables

suelomojado <- llovio

Separamos las relaciones derivadas de las basicas (porque?)

baseRel(rhk).

baseRel(loc(X)).

....

derivedRel(suelomojado(X), [llovio_anoche(X)]).

derivedRel(licenciado(X), [plan_comun, aprobado_cc42a(X)]).

Luego, un meta interprete para CWR-D seria

holds(F, S) :- basicRel(F), member(F, S).

holds(neg(F), S) :- basicRel(F), not(member(F, S)).

holds(F, S) :- derivedRel(F, BodyList), holdsAll(BodyList, S).

holds(neg(F), S) :- derivedRel(F, BL), not(holds(F, S)).

Hechos estaticos:

Ej: derivedRel(adyacente(X, Y), []).

Actions:

Action Schema ej: mover(X, Y). Instancias de mover son las verdaderas acciones

Def: Vecino: es un estado T tal que se puede llegar desde S aplicando una accion A

Es necesario describir cuando es A aplicable y cuales son los efectos en S

precondiciones (listado de literales basicos. ademas puede incluir las variables del schema).

Efectos:

Se asume que hechos que no se afectan no se mencionan
efectos positivos se suman a la lista de estados
efectos negativos se eliminan de la lista de estados

Representacion de STRIPS

addlist

dellist

Ej:

addlist(givecoffe, [chc, craighappy]).

deletelist(move(X, Y), [loc(X)]).

NO DEBEN INCLUIR RELACIONES DERIVADAS

Descripcion del dominio:

especificacion de las acciones requeridos en el dominion de planificacion.
por cada accion debe definirse precondition, addlist, deletelist, en KB

en el caso de CRW-D, se deben tener los nombres (schemas) de todas las acciones en la KB

por lo tanto se define canApply(action, S) para cada accion
definir newState(S, Action, NewState).

definir vecino

nb(S, Neighbours) :- findAll(NewS, canApply(A, S), newState(S, A, NewS)),
Neighbours.

Finalmente:

Dado precon, deletelist, addlist, actionlist, derivedRel, baseRel como una KB

Dado un estado inicial s en CWR o CWR-D

Dado un conjunto de meta G

Encontrar una secuencia de acciones a1, .., an tal que
results(a1, s, s1), results(a2, s1, s2), , results(an, sn-1, sn)
y holdsall(G, sn) sean verdaderos