

Planificació deliberativa

- Dimensions
- Cerca de través de l'espai d'estats

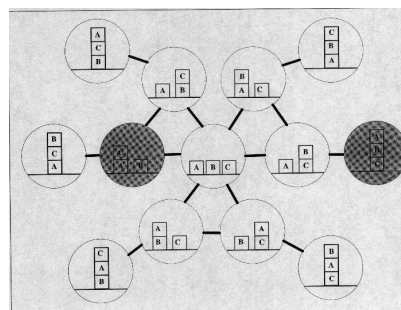
Dimensions

- Estratègies de construcció
 - Refinament: afegir gradualment components
 - Transformació
 - Retracció: eliminar components
- Grandària del component
 - Generatiu: construcció des de 0
 - Basats en casos: construcció a partir de l'experiència

Cerca de través de l'espai d'estats

- Node: un estat del món
- Arc: connecta els estats del món als quals es pot arribar si s'executa una acció.
- Avantatge: amb aquesta aproximació es poden aplicar tots els algorismes de cerca coneguts, tant els de força bruta com els que usen heurístiques
- Algorismes generals:
 - Progressió
 - Regressió

Cerca de través de l'espai d'estats



Cerca de través de l'espai d'estats Progressió

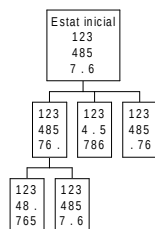
- Aplicació d'un operador en direcció cap endavant, transformant un estat en un altre
 - Donat un estat inicial S,
Un operador $\alpha=(P,A,D)$
 - P: precondicions
 - A: Add-list
 - D: Delete-list
- es pot aplicar en direcció cap endavant si $P \subset S$.
Estat resultant: $A \cup (S - D)$
- Means-ends analysis: es selecciona l'operador que redueix les diferències entre l'estat resultant i l'objectiu.

Cerca de través de l'espai d'estats Progressió. Algorisme

```

funció planificador (estat_inicial, objectius, bc, pla) retorna pla
variables:
  a: acció;
  s: estat;
inici
  si satisfà(estat_inicial, objectius)
    llavors retorna(pla);
  sinó a:=seleccionar_acció(estat_inicial);
    si a={}
      llavors fail;
    sinó s:= aplicar(a, estat_inicial);
      retorna (planificador(s, objectius, bc, concatenar(pla, a)));
    fsi;
fsi;
fplanificador;
    
```

Cerca de través de l'espai d'estats Progressió. Exemple



Cerca de través de l'espai d'estats Regressió

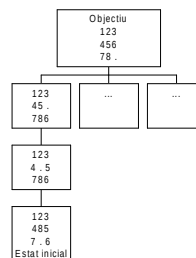
- Aplicació d'un operador en direcció cap enrera, transformant un objectiu en un altre objectiu.
- Donat un objectiu G,
Un operador $\alpha=(P,A,D)$ P: precondicions
 A: Add-list
 D: Delete-list
es pot aplicar sobre un objectiu G, si $D \cap G = \emptyset$
Estat resultant: $P \cup (G - A)$

Cerca de través de l'espai d'estats Regressió. Algorisme

```

funció planificador (estat_inicial, objectius, bc, pla) retorna pla
variables:
  a: acció;
  g: objectius;
inici
  si satisfà(estat_inicial, objectius)
    llavors retorna(pla);
  sinó a:=seleccionar_acció(estat_inicial, objectius);
       g:=regressió(objectius, a);
       si a={} o g={} o objectius  $\subset$  g
         llavors fail;
       sinó retorna (planificador(estat_inicial, g, bc, concatenar(a, pla)));
  fsi;
fplanificador;
    
```

Cerca de través de l'espai d'estats Regressió. Exemple



Cerca de través de l'espai d'estats Progressió i regressió

- Són complets: si existeix un pla, existeix la seqüència de selecció no-determinista que el fa possible
- Són robustos: si retorna un pla, aquest funcionarà.
- Quin és més ràpid? Regressió
 - Justificació:
 - n: nombre de seleccions que calen per resoldre el problema
 - b: nombre d'opcions (operadors, instanciacions,...)
 - Regressió: b és menor que en progressió
 - En regressió poden haver molts operadors aplicables que siguin irrelevants a l'objectiu.