UOM2 : Intelligence Artificielle

Planification
vs
Calcul des situations
(Situation Calculus)

Philippe Lamarre

Faculté des Sciences et Techniques de Nantes

La planification

Le fait de faire un plan pour atteindre certains objectifs est une marque certaine d'intelligence.

La planification est utilisée dans de nombreux domaines.

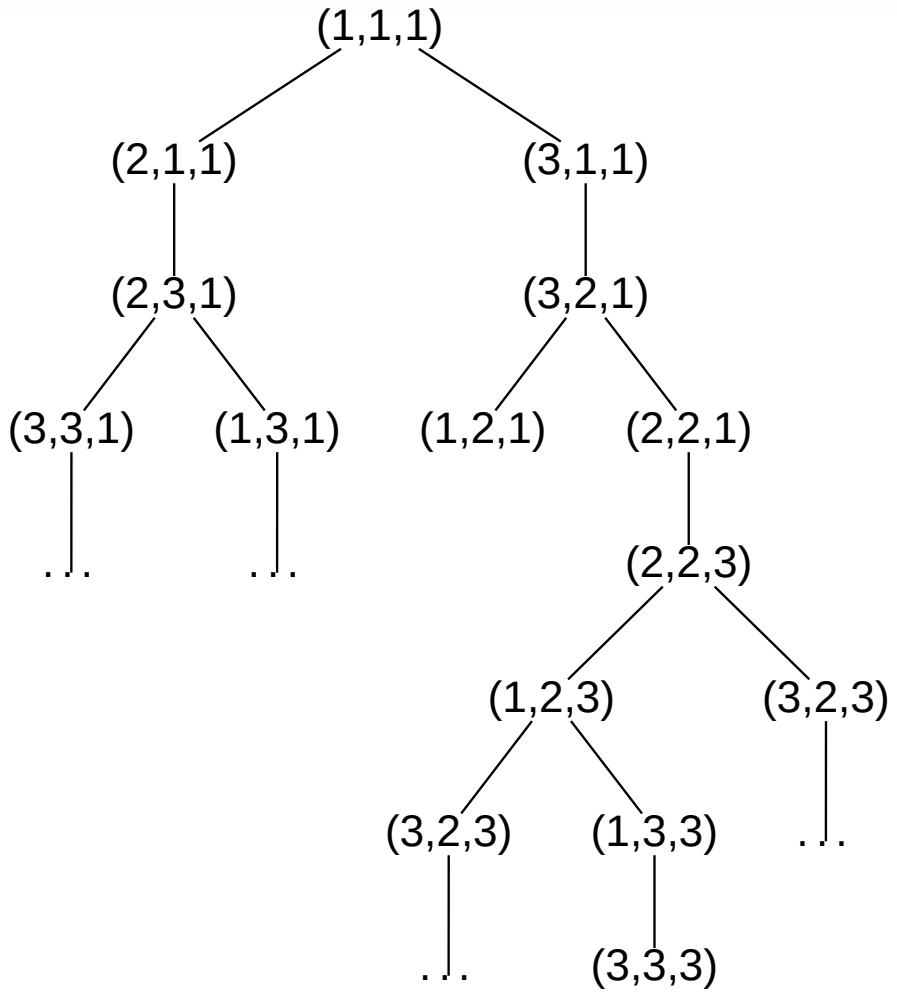
- Dans des applications particulières
  - assemblage de pièces
  - découpage
  - positionnement de machines
  - . . .
- Actions de robots
- Discours (que dire dans quel ordre. . .)
- . . .

Hypothèses de l'étude.

- Connaissance totale de l'environnement.
- Actions discrettes.
- Maîtrise totale de l'environnement.

Rappel : Graphes d'états.

Rappel : Graphes d'états.



Plannification et graphes d'états

- Représentation d'un état
- Représentation d'un but
- Représentation du point de départ
- Représentation d'une action
- Représentation d'un plan

Plannification et graphes d'états

- Représentation d'un état
  Liste de variables avec des valeurs assignées directement dans chaque état.
- Représentation d'un but
  C'est un état particulier, ou une caractérisation.
- Représentation du point de départ
  Description d'un état particulier.
- Représentation d'une action
  C'est une transition permettant de passer d'un état à l'autre : modifie les affectations des variables.
- Représentation d'un plan
  Chemin dans le graphe d'états.

Quelques objectifs.

- Avoir un même langage pour décrire
  - les états,
  - les actions et leurs effets

  Et donc une théorie uniforme.
- Utiliser des méthodes de démonstration connues pour résoudre le problème.

La base : la logique classique + situations.

La logique classique fournit un outil permettant de représenter un état (avec les restrictions d'usage et la prudence nécessaire).

Le calcul des situations (McCarthy and Hayes 1969) est la rencontre de la logique classique et de la notion de situation.

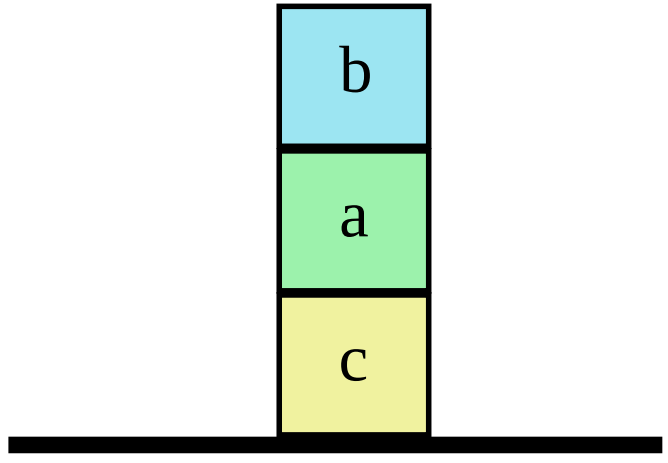
Introduction.

- Représentation d'un état
- Représentation d'un but
- Représentation du point de départ
- Représentation d'une action
- Représentation d'un plan

Introduction.

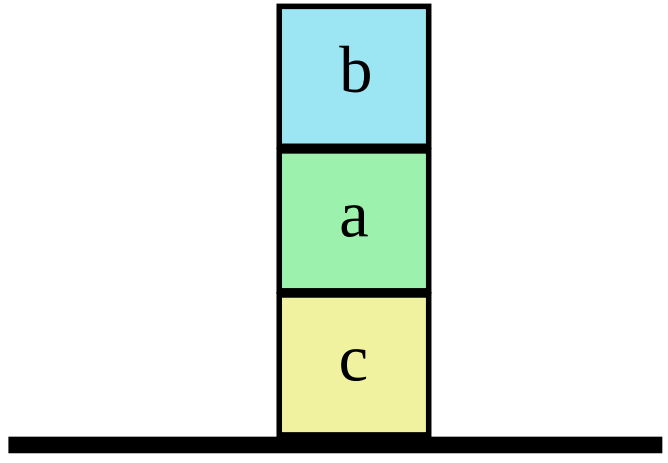
- Représentation d'un état
  Formule logique.
- Représentation d'un but
  Formule logique qui décrit l'objectif.
- Représentation du point de départ
  Formule logique.
- Représentation d'une action
  Ajout/suppression de formules logiques décrivant le monde.
- Représentation d'un plan
  Ensemble d'actions totalement ou partiellement ordonné.

Les étapes



1. Définir un langage de description du monde
   $on/2; clear/1 \ldots$
2. Utiliser le langage pour décrire le monde
   $on(b,a) \wedge on(a,c) \wedge on(c,sol) \wedge clear(b) \wedge clear(sol)$

Les étapes

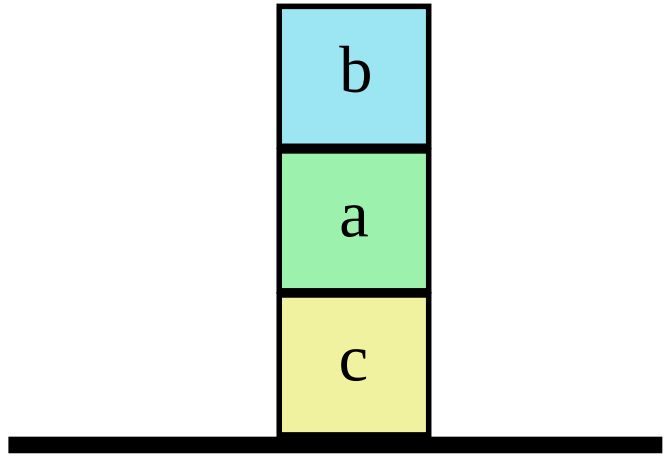


3 Réïfier (i.e. considérer comme concret quelque chose qui est abstrait, ici la notion de situation).

$on(X, Y) \rightarrow on(X, Y, s0)$

$clear(X) \rightarrow clear(X, s0)$

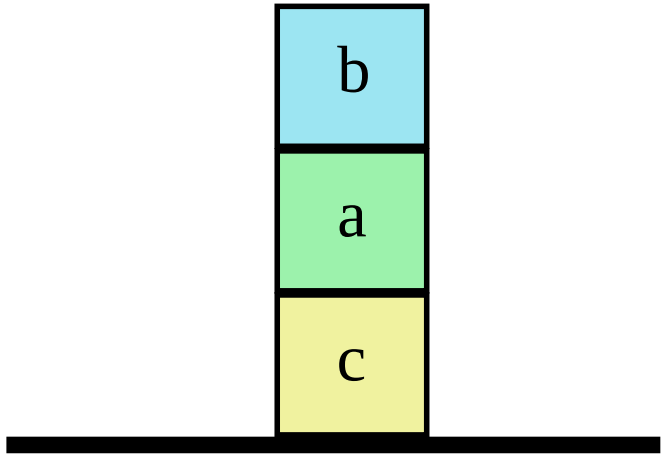
Les étapes



4 Rechercher les axiomes qui sont vrais dans toutes les situations.

$$\forall X, Y, S((on(X, Y, S) \wedge (Y \neq sol)) \rightarrow \neg clear(Y, S))$$

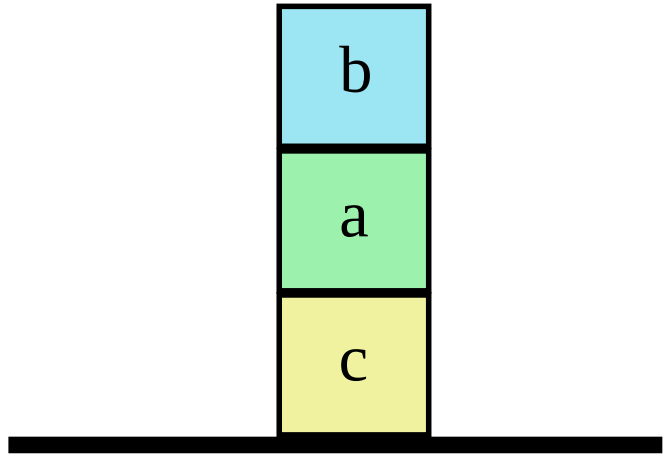
Les étapes



5 Définir les actions

$move(X, Y, Z)$ i.e. déplacer X de Y sur Z.

Les étapes



6 Définir une fonction do/2 qui donne la situation résultant de la réalisation de l'action a dans la situation s.
$do(move(a, n, sol), so)$

Planification dans le calcul des situation

L'objectif est de trouver un plan permettant d'arriver à la situation souhaitée.

Ceci est obtenu en prouvant la démontrabilité de la situation à partir des données initiales et des actions possibles.

Le plan est obtenu par instanciantion d'une variable.

Le “Frame Problem”

- En général, les actions ont des effets locaux (i.e. elles n’affectent pas le monde dans son entier).
- Une action peut modifier certains fait et n’avoir aucune conséquence sur d’autres. Que nous offre la logique ?

Le “Frame Problem”

- En général, les actions ont des effets locaux (i.e. elles n’affectent pas le monde dans son entier).
- Une action peut modifier certains fait et n’avoir aucune conséquence sur d’autres. Que nous offre la logique ?

Logique : si rien n’est dit, rien ne peut être déduit.

Le "Frame Problem"

- En général, les actions ont des effets locaux (i.e. elles n'affectent pas le monde dans son entier).
- Une action peut modifier certains fait et n'avoir aucune conséquence sur d'autres. Que nous offre la logique ?

Logique : si rien n'est dit, rien ne peut être déduit.

Nécessité d'écrire des formules exprimant ce qui est persistant : «frame axioms»

Le “Frame Problem”

Principe :

Les actions spécifient dans leurs post-conditions ce qu’elles produisent.

Les axiomes du cadre spécifient ce qui ne change pas.

Tout doit être déductibe à partir des informations spécifiques.

Le "Frame Problem"

Principe :

Les actions spécifient dans leurs post-conditions ce qu'elles produisent.

$\forall X, Y, Z, S(on(X, Y, S) \wedge clear(X, S) \wedge clear(Z, S) \wedge X \neq Z \rightarrow on(X, Z, do(move(X, Y, Z), S)))$

Les axiomes du cadre spécifient ce qui ne change pas.

$\forall A, B, X, Y, Z, S(on(A, B) \wedge A \neq X \wedge on(X, Y, S) \wedge clear(X, S) \wedge clear(Z, S) \wedge X \neq Z \rightarrow on(A, B, do(move(X, Y, Z), S)))$

Tout doit être déductibe à partir des informations spécifiques.

Le “Frame Problem”

Autres solution plus «opérationnelle» :

Les règles indiquent ce qui est modifié (en positif et en négatif).

Seules les informations présentent dans les postconditions des règles sont modifiées.

Attention ce n’est pas exactement la même chose. Cela sera étudié en TD. Le cours considère l’utilisation de «Frame Axiomes».

Le "Frame Problem"

Autres solution plus «opérationnelle» :

Les règles indiquent ce qui est modifié (en positif et en négatif).

Seules les informations présentent dans les postconditions des règles sont modifiées.

$move(X, Y, Z) \Rightarrow$

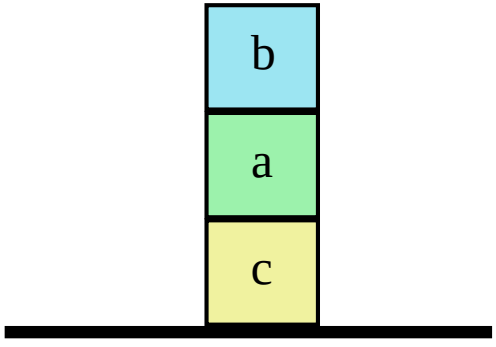
- Assert $on(X, Z)$, $clear(Y)$
- Retract $on(X, Y)$, $clear(z)$

Attention ce n'est pas exactement la même chose. Cela sera étudié en TD. Le cours considère l'utilisation de «Frame Axiomes».

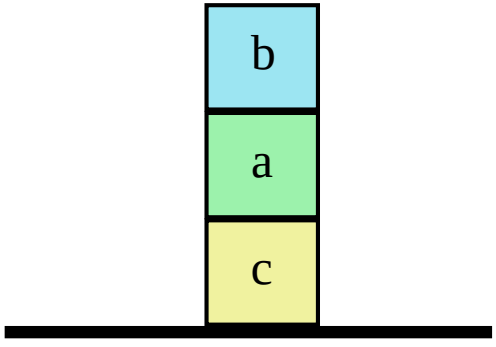
Récapitulatif

1. Représenter le monde (langage, faits explicites et implicites)
2. Reïfier (introduire la notion de situation dans les prédicats de description du monde).
3. Recherche des axiomes vrais dans toutes les situations
4. Définir les actions.
   - Noms
   - Effets produits (faits modifiés)
   - Faits persistants (faits non modifiés - axiomes du cadre).
5. Ecrire les axiomes du cadre.

Un exemple

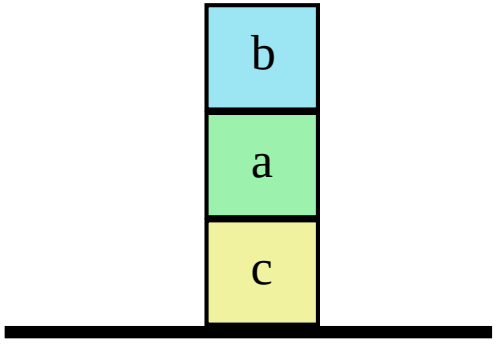


Un exemple



$on(c, sol),\ on(a, c),\ on(b, a),\ on(c, sol,$
$clear(sol),\ clear(b),\ couleur(c, jaune),$
$couleur(a, vert),\ couleur(b, bleu)$

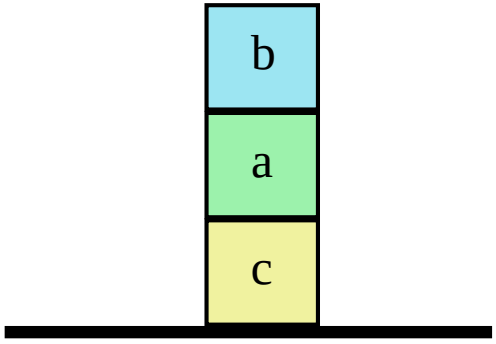
Un exemple



$on(c, sol),\ on(a, c),\ on(b, a),\ on(c, sol,$
$clear(sol),\ clear(b),\ couleur(c, jaune),$
$couleur(a, vert),\ couleur(b, bleu)$

$\forall X, Y(on(X, Y) \rightarrow on(X, Y, s0)), \forall X, Y(couleur(X, Y) \rightarrow couleur(X, Y, s0)),$
$\forall X(clear(X) \rightarrow clear(X, s0))$

Un exemple



$on(c, sol),\ on(a, c),\ on(b, a),\ on(c, sol,$
$clear(sol),\ clear(b),\ couleur(c, jaune),$
$couleur(a, vert),\ couleur(b, bleu)$

$\forall X, Y(on(X, Y) \rightarrow on(X, Y, s0)), \forall X, Y(couleur(X, Y) \rightarrow couleur(X, Y, s0)),$
$\forall X(clear(X) \rightarrow clear(X, s0))$

$\forall X, Y, S((on(X, Y, S) \wedge (Y \neq sol)) \rightarrow \neg clear(Y, S)), \forall S(couleur(c, jaune, S)),$
$\forall S(couleur(a, vert, S)), \forall S(couleur(b, bleu, S))$

Un exemple

$\forall X, Y(on(X, Y) \rightarrow on(X, Y, s0)), \forall X, Y(couleur(X, Y) \rightarrow couleur(X, Y, s0)),$
$\forall X(clear(X) \rightarrow clear(X, s0))$

$\forall X, Y, S((on(X, Y, S) \wedge (Y \neq sol)) \rightarrow \neg clear(Y, S)), \forall S(couleur(c, jaune, S)),$
$\forall S(couleur(a, vert, S)), \forall S(couleur(b, bleu, S))$

$\forall X, Y, Z, S(on(X, Y, S) \wedge clear(X, S) \wedge clear(Z, S) \wedge X \neq Z \rightarrow$
$on(X, Z, do(move(X, Y, Z), S)))$
$\forall X, Y, Z, S(on(X, Y, S) \wedge clear(X, S) \wedge clear(Z, S) \wedge X \neq Z \rightarrow$
$clear(Y, do(move(X, Y, Z), S)))$

Un exemple

$\forall X, Y(on(X, Y) \rightarrow on(X, Y, s0)), \forall X, Y(couleur(X, Y) \rightarrow couleur(X, Y, s0)),$
$\forall X(clear(X) \rightarrow clear(X, s0))$

$\forall X, Y, S((on(X, Y, S) \wedge (Y \neq sol)) \rightarrow \neg clear(Y, S)), \forall S(couleur(c, jaune, S)),$
$\forall S(couleur(a, vert, S)), \forall S(couleur(b, bleu, S))$

$\forall X, Y, Z, S(on(X, Y, S) \wedge clear(X, S) \wedge clear(Z, S) \wedge X \neq Z \rightarrow$
$on(X, Z, do(move(X, Y, Z), S)))$
$\forall X, Y, Z, S(on(X, Y, S) \wedge clear(X, S) \wedge clear(Z, S) \wedge X \neq Z \rightarrow$
$clear(Y, do(move(X, Y, Z), S)))$

$\forall A, B, X, Y, Z, S(on(A, B) \wedge A \neq X \wedge on(X, Y, S) \wedge clear(X, S) \wedge clear(Z, S) \wedge X \neq$
$Z \rightarrow on(A, B, do(move(X, Y, Z), S)))$
$\forall A, X, Y, Z, S(clear(A) \wedge A \neq Z \wedge on(X, Y, S) \wedge clear(X, S) \wedge clear(Z, S) \wedge X \neq$
$Z \rightarrow clear(A, do(move(X, Y, Z), S)))$

Un exemple

Comment faire pour que le cube a soit sur le sol?

Un exemple

Comment faire pour que le cube a soit sur le sol?

Existe-t-il une situation S telle que a soit sur le sol ?

Un exemple

Comment faire pour que le cube a soit sur le sol?

Existe-t-il une situation S telle que a soit sur le sol ?

La formule $on(a, sol, S)$ est elle démontrable à partir de la théorie précédente ? Si oui, la forme de la variable S nous donnera un plan permettant de passer de la situation actuelle à la situation où a est sur le sol.

Un exemple

Comment faire pour que le cube a soit sur le sol?

Existe-t-il une situation S telle que a soit sur le sol ?

La formule $on(a, sol, S)$ est elle démontrable à partir de la théorie précédente ? Si oui, la forme de la variable S nous donnera un plan permettant de passer de la situation actuelle à la situation où a est sur le sol.

$\Gamma \vdash^{?} on(a, sol, S)$

Un exemple

Comment faire pour que le cube a soit sur le sol?

Existe-t-il une situation S telle que a soit sur le sol ?

La formule $on(a, sol, S)$ est elle démontrable à partir de la théorie précédente ? Si oui, la forme de la variable S nous donnera un plan permettant de passer de la situation actuelle à la situation où a est sur le sol.

$\Gamma \vdash^{?} on(a, sol, S)$

$S = do(move(a, sol, do(move(b, sol, s0))))$

Un exemple

Comment faire pour que le cube a soit sur le sol?

Existe-t-il une situation S telle que a soit sur le sol ?

La formule $on(a, sol, S)$ est elle démontrable à partir de la théorie précédente ? Si oui, la forme de la variable S nous donnera un plan permettant de passer de la situation actuelle à la situation où a est sur le sol.

$\Gamma \vdash^{?} on(a, sol, S)$

$S = do(move(a, sol, do(move(b, sol, s0))))$

Attention, ce n'est pas la seule solution. . .
$S = do(move(a, sol, do(move(b, sol, do(move(b, a, do(move(b, sol, s0))))))))$
. . . à méditer . . .

Attention: difficultés

Quelques difficulités :

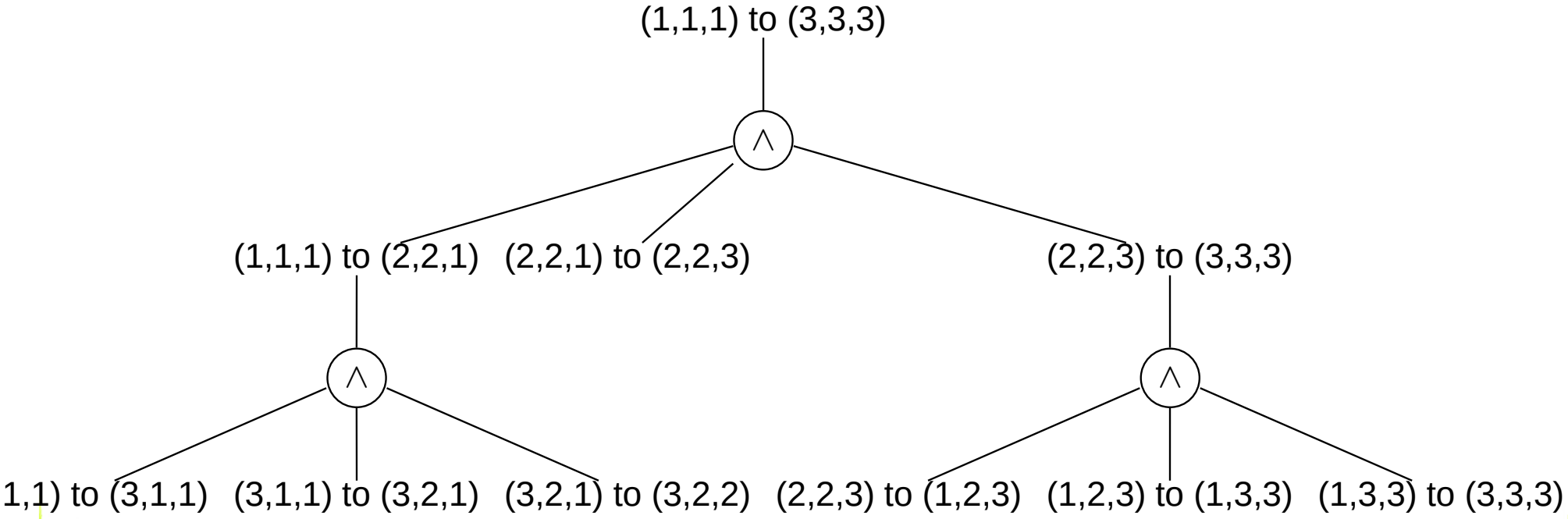
- Problème de ramification.
  Lister toutes les conséquences qui sont liées à un fait.
- Problème de qualification
  Lister toutes les conditions qui font qu'une action peut s'appliquer.
- Temps de calcul pour arriver à un résultat.
  Les calculs nécessaire pour obtenir un plan peuvent être très longs.

Autres techniques.

Rappel : espaces de sous problèmes.

Autres techniques.

Rappel : espaces de sous problèmes.



Autres techniques.

STanford **R**esearch **I**nstitue **P**lanning **S**ystem (Fikes & Nilson début 70): **STRIPS**

Utilisé pour planifier les mouvements du robot Shakey.

Autres techniques.

STanford **R**esearch **I**nstitue **P**lanning **S**ystem (Fikes & Nilson début 70): **STRIPS**

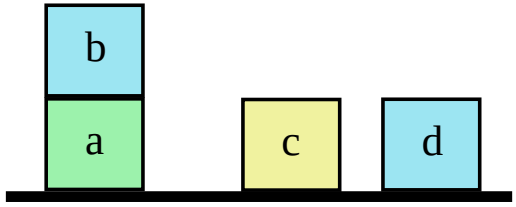
Principes

- Recherche de différence entre l'état courant et l'état recherché.
- Recherche d'un opérateur permettant de réduire cette différence.
- Obtenir un état satisfaisant la précondition de l'opérateur choisi à partir de l'état initial, ET rechercher les transitions permettant de passer de l'état obtenu après application de l'opérateur utilisé sur la situation obtenue pour arriver à l'état souhaité.

STRIPS sur un exemple.

Définition du problème.

État initial



État final

$$---->$$

$sur(b, a)$
$surTable(a)$
$surTable(c)$
$surTable(d)$
$pinceVide$

$sur(c, a)$
$sur(b, d)$
$surTable(a)$
$surTable(d)$

Les axiomes. . . jocker. . .

STRIPS sur un exemple.

Utilisation d'une pile pour mémoriser

$$\left.\begin{array}{c} sur(c,a) \\ sur(b,d) \\ surTable(a) \\ surTable(d) \end{array}\right\}$$

décomposer
empiler

$$sur(c,a) \wedge sur(b,d) \wedge surTable(a) \wedge surTable(d)$$

STRIPS sur un exemple.

Représentation des opérateurs

empiler(x,y) Empiler x sur y

PréCond $libre(y) \wedge tenir(x)$

Delete $libre(y) \wedge tenir(x)$

Add $pinceVide \wedge sur(x, y)$

saisir(x) saisir x sur la table

PréCond $pinceVide \wedge libre(x) \wedge surTable(x)$

Delete $pinceVide \wedge surTable(x)$

Add $tenir(x)$

dépiler (x, y) Dépiler x de y

PréCond $sur(x, y) \wedge pinceVide$

Delete $sur(x, y) \wedge pinceVide$

Add $tenir(x) \wedge libre(y)$

déposer(x) déposer x sur la table

PréCond $tenir(x)$

Delete $tenir(x)$

Add $surTable(x) \wedge pinceVide$

STRIPS sur un exemple.

Méthode de traitement : créer des sous-but par un raisonnement «en arrière». Ceci est appelé Régression.

Il est possible de

- réaliser une action. Si l'état courant satisfait la pré-condition de l'action
  - ajouter les éléments de la liste Add à l'état courant
  - supprimer les éléments de la liste Delete de l'état courant
- "défaire" une action (si elle avait été réalisée dans l'état courant).
  - retirer les éléments de la liste Add de l'état courant
  - ajouter les élément de la liste Delete à l'état courant

Attention : cela suppose que. . .

Arrêt lorsque l'état courant correspond à l'état final (états égaux directement ou via une démonstration de théorème d'une propriété caractérisant l'état final).

STRIPS sur un exemple.

Etat final :
$sur(c, a) \wedge sur(b, d) \wedge surTable(a) \wedge surTable(d)$

La dernière partie ($surTable(a) \wedge surTable(d)$) est déjà satisfaite dans l'état initial. On le considère donc comme invariant (i.e. vérifié dans tous les états). Attention, cela suppose. . .

La pile contiendra donc . . . /. . .

STRIPS sur un exemple.

Etat courant = $sur(b,a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d) \wedge pinceVide$

objectifs (mémo)	pile	commentaires
	$sur(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

STRIPS sur un exemple.

Etat courant = $sur(b,a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d) \wedge pinceVide$

objectifs (mémo)	pile	commentaires
	$sur(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	Peut être obtenu par "$empiler(c,a)$"

STRIPS sur un exemple.

Etat courant = $sur(b,a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d) \wedge pinceVide$

objectifs (mémo)	pile	commentaires
$sur(c,a)$	$libre(a)$ $tenir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

STRIPS sur un exemple.

Etat courant = $sur(b,a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d) \wedge pinceVide$

objectifs (mémo)	pile	commentaires
$sur(c,a)$	$libre(a)$ $tenir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	Peut être obtenu par "$depiler(b,a)$"

STRIPS sur un exemple.

Etat courant = $sur(b,a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d) \wedge pinceVide$

objectifs (mémo)	pile	commentaires
$libre(a)$ $sur(c,a)$	$sur(b,a)$ $libre(b)$ $pinceVide$ $sur(b,a) \wedge libre(b) \wedge pinceVide$ $depiler(b,a)$ $tenir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

STRIPS sur un exemple.

Etat courant = $sur(b,a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d) \wedge pinceVide$

objectifs (mémo)	pile	commentaires
$libre(a)$ $sur(c,a)$	[redacted] $depiler(b,a)$ $tenir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$	Directement vérifié dans l'état courant ou déductibles des axiomes.

STRIPS sur un exemple.

Etat courant = $sur(b, a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d) \wedge pinceVide$

Modification de l'état courant : réalisation de l'action $depiler(b, a)$

dépiler (x, y) Dépiler x de y

PréCond $sur(x, y) \wedge pinceVide$

Delete $sur(x, y) \wedge pinceVide$

Add $tenir(x) \wedge libre(y)$

Ajout de Add retrait de Delete.

Etat courant = $tenir(b) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

STRIPS sur un exemple.

Etat courant = $tenir(b) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
$sur(c,a)$	$\left\vert \begin{array}{l} tenir(c) \\ libre(a) \wedge tenir(c) \\ empiler(c,a) \end{array} \right.$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	Peut être obtenu par saisir(c)

actions réalisées = { $depiler(b,a)$ }

STRIPS sur un exemple.

Etat courant = $tenir(b) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
$tenir(c)$ $sur(c,a)$	$surTable(c)$ $libre(c)$ $pinceVide$ $surTable(c) \wedge libre(c) \wedge pinceVide$ $saisir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

actions réalisées = { $depiler(b,a)$ }

STRIPS sur un exemple.

Etat courant = $tenir(b) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
$tenir(c)$ $sur(c,a)$	$surTable(c)$ $libre(a)$ $pinceVide$ $surTable(c) \wedge libre(c)$ $\wedge pinceVide$ $saisir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

actions réalisées = { $depiler(b,a)$ }

STRIPS sur un exemple.

Etat courant = $tenir(b) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
$pinceVide$ $tenir(c)$ $sur(c,a)$	$libre(d)$ $tenir(b)$ $libre(d) \wedge tenir(b)$ $empiler(b,d)$ $surTable(c) \wedge libre(c)$ $\wedge pinceVide$ $saisir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

actions réalisées = { $depiler(b,a)$ }

STRIPS sur un exemple.

Etat courant = $tenir(b) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
$pinceVide$ $tenir(c)$ $sur(c,a)$	[redacted] [redacted] [redacted] $empiler(b,d)$ $surTable(c) \wedge libre(c)$ $\wedge pinceVide$ $saisir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

STRIPS sur un exemple.

Etat courant = $tenir(b) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

Modification de l'état courant : réalisation de l'action $empiler(b, d)$

empiler(x,y) Empiler x sur y

PréCond $libre(y) \wedge tenir(x)$

Delete $libre(y) \wedge tenir(x)$

Add $pinceVide \wedge sur(x, y)$

Ajout de Add retrait de Delete.

Etat courant =

$pinceVide \wedge sur(b, d) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

STRIPS sur un exemple.

Etat courant =

$pinceVide \wedge sur(b,d) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
$tenir(c)$ $sur(c,a)$	$surTable(c) \wedge libre(c) \wedge pinceVide$ $saisir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

actions réalisées = { $depiler(b,a)$, $empiler(b,d)$ }

STRIPS sur un exemple.

Etat courant =

$pinceVide \wedge sur(b,d) \wedge libre(a) \wedge surTable(a) \wedge surTable(c) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
$tenir(c)$ $sur(c,a)$	$saisir(c)$ $libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

actions réalisées = { $depiler(b,a)$, $empiler(b,d)$ }

STRIPS sur un exemple.

Etat courant = $tenir(c) \wedge sur(b,d) \wedge libre(a) \wedge surTable(a) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
$sur(c,a)$	$libre(a) \wedge tenir(c)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

actions réalisées = { $depiler(b,a)$, $empiler(b,d)$, $saisir(c)$ }

STRIPS sur un exemple.

Etat courant = $tenir(c) \wedge sur(b,d) \wedge libre(a) \wedge surTable(a) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
$sur(c,a)$	$libre(\ldots) \wedge \ldots(\ldots)$ $empiler(c,a)$ $sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

actions réalisées = { $depiler(b,a)$, $empiler(b,d)$, $saisir(c)$ }

STRIPS sur un exemple.

Etat courant = $sur(c,a) \wedge pinceVide \wedge sur(b,d) \wedge surTable(a) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires
	$sur(b,d)$ $sur(c,a) \wedge sur(b,d)$	

actions réalisées = { $depiler(b,a)$, $empiler(b,d)$, $saisir(c)$, $empiler(c,a)$ }

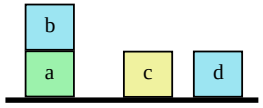
STRIPS sur un exemple.

Etat courant = $sur(c, a) \wedge pinceVide \wedge sur(b, d) \wedge surTable(a) \wedge surTable(d)$

objectifs (mémo)	pile	commentaires

actions réalisées = { $depiler(b, a)$, $empiler(b, d)$, $saisir(c)$, $empiler(c, a)$ }

STRIPS sur un exemple.

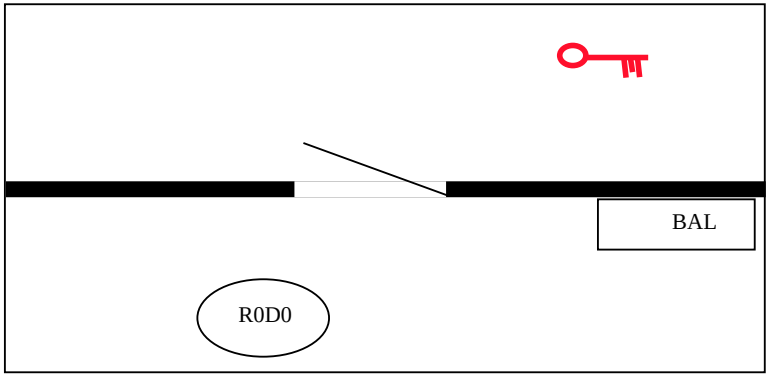
État initial		actions		État final
	$--$	$depiler(b,a),$	$->$	
$sur(b,a)$		$empiler(b,d),\ saisir(c),$		$sur(c,a)$
$surTable(a)$		$empiler(c,a)$		$sur(b,d)$
$surTable(c)$				$surTable(a)$
$surTable(d)$				$surTable(d)$
$pinceVide$				

Limitations.

- Ordre fixé, même lorsque ce n’est pas nécessaire.

Limitations.

- Ordre fixé, même lorsque ce n'est pas nécessaire.
- Difficulté lorsqu'il y a interdépendance des buts.



fermer la porte à clé et mettre la clé dans la BAL.

Tentative de résolution en choisissant le premier sous-but : fermer la porte à clé. Le robot passe dans l'autre pièce, ferme la porte à clé, et. . . ne peut mettre la clé dans la BAL.

Tentative en commençant par l'autre sous-but, aller chercher la clé et la mettre dans la BAL. Mais alors, comment fermer la porte à clé ?

Prespectives

- Planification non linéaire.
- Monde en évolution, re-planification.
- . . .