

Intelligence Artificielle

Planification Logique

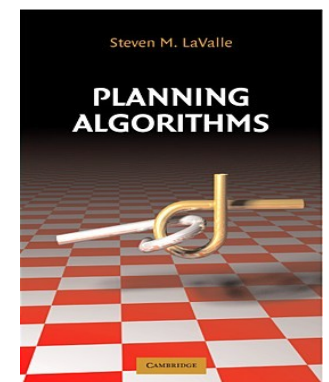
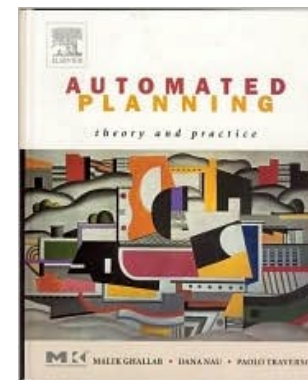
Frederic Koriche

koriche@lirmm.fr

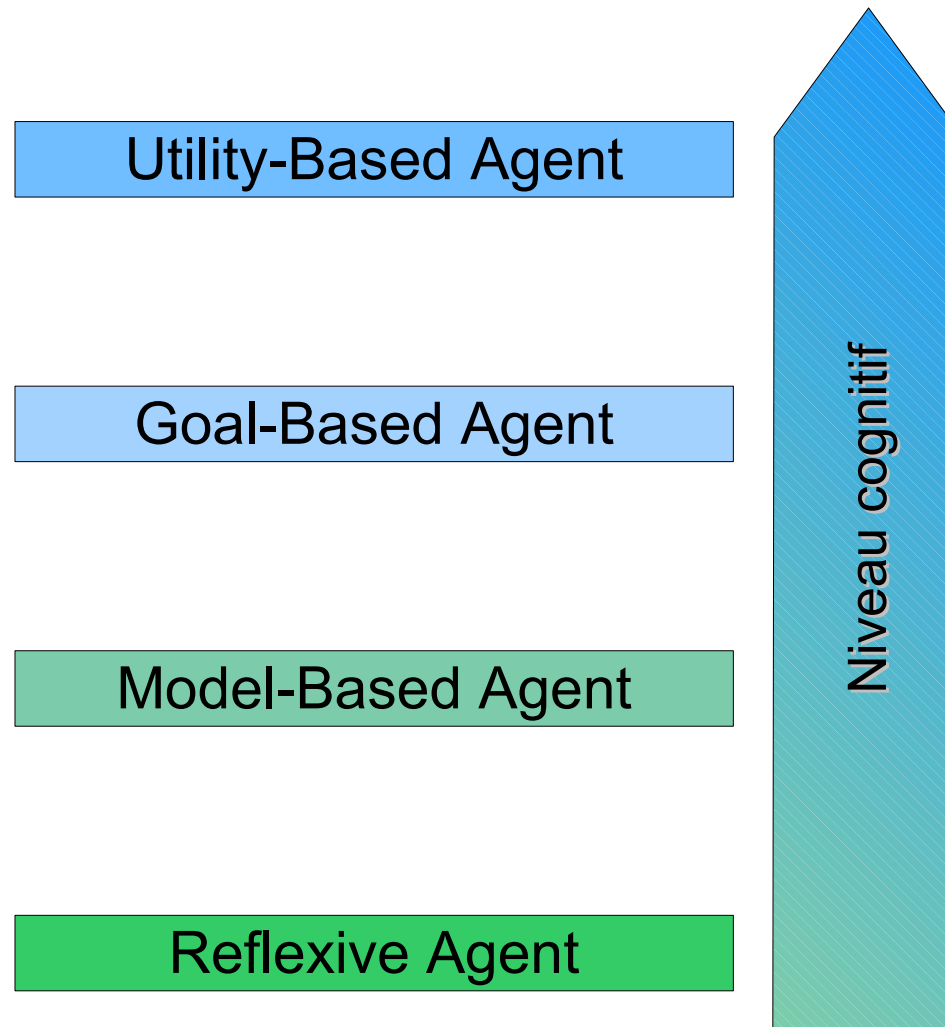
Plan

1. Agents planificateurs
2. Raisonnement sur l'action
3. Planification arborescente
4. SAT-Plan
5. Graph-Plan

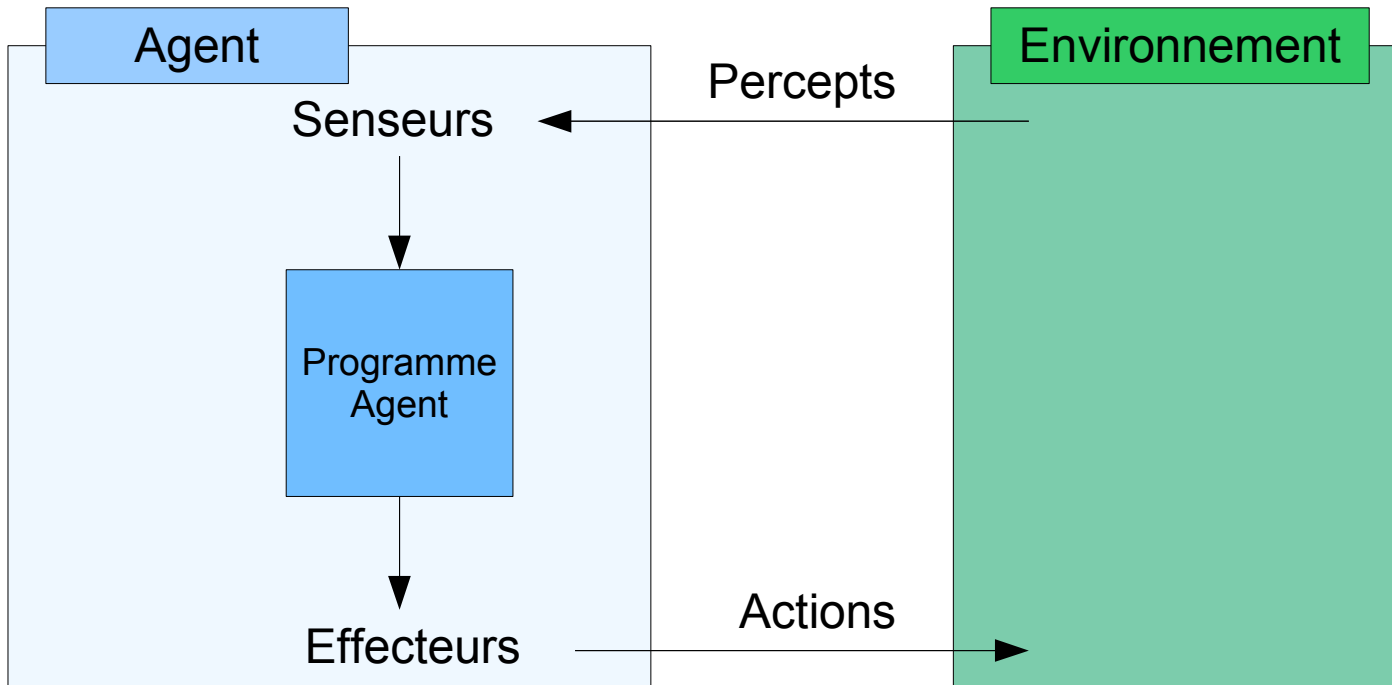
Ouvrages de référence



Agents Planificateurs

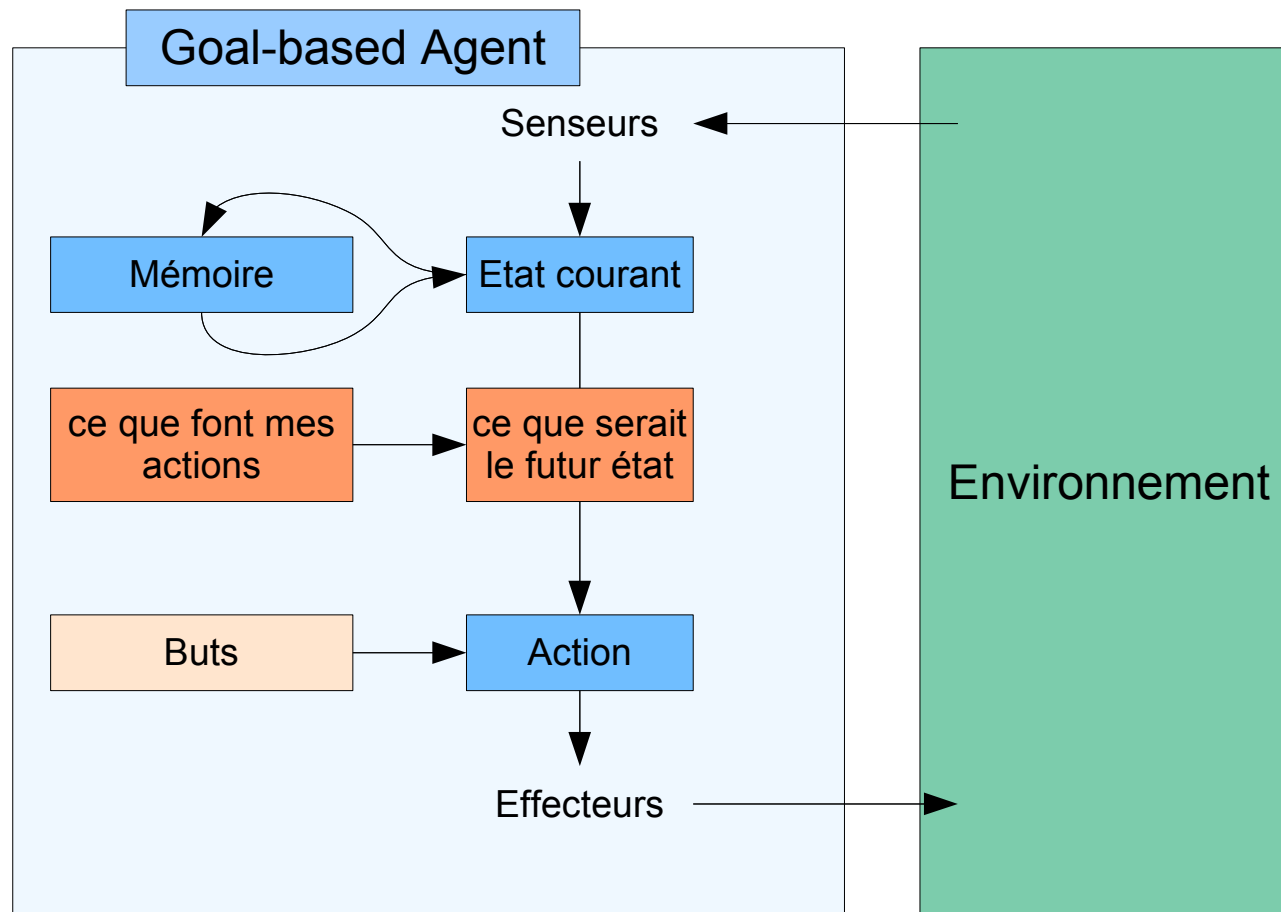


Agents Planificateurs



- ♦ **Agent**: entité percevant son environnement par ses senseurs et agissant sur cet environnement par ses effecteurs

Agents Planificateurs



- ♦ **Agent goal-based**: l'agent décide d'atteindre un état dans lequel son but est vérifié ; l'agent utilise son modèle d'actions et explore l'espace des états (**search, planning**)

Raisonnement sur l'action



Le monde du recycleur

Fluents

In: salle A ou B

Dust: poussière dans A ou B

Actions

Go: aller d'une salle à l'autre

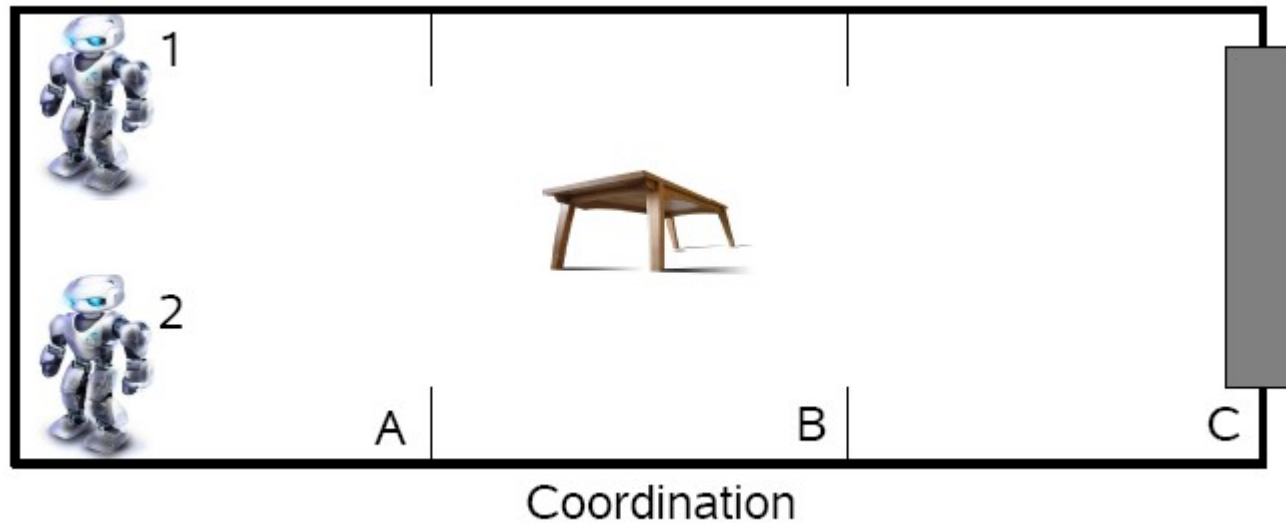
Clean: nettoyer la salle

No: ne rien faire

But

$In(B) \ \& \ \neg Dust(B)$

Raisonnement sur l'action



Fluents

In: 1,2,T dans salle A,B,C

Actions

Go: aller d'une salle à l'autre

Push: pousser un objet d'une salle à l'autre

But

$In(1,C) \ \& \ In(2,C) \ \& \ In(T,C)$

Raisonnement sur l'action

Vocabulaire

- (1) Ensemble fini R de symboles de relations d'arité donnée
- (2) Ensemble fini C de symboles d'objets
- (3) Ensemble énumérable de variables

Eléments

- (1) Terme: variable ou constante
- (2) Atome: $R(t_1, \dots, t_k)$ où R est une relation d'arité k
- (3) Littéral: atome ou sa négation
- (4) Atome de base: atome sans variable

Langage STRIPS

Etat

Ensemble S d'atomes de base
(interprétation de herbrand)

Tout littéral d'apparaissant pas dans
un état est considéré comme négatif

But

Ensemble G de littéraux

Action

Condition: ensemble de littéraux

Effet: ensemble de littéraux

- les littéraux positifs s'ajoutent à l'état courant
- les littéraux négatifs se retranchent à l'état courant

Langage STRIPS

STanford Research Institute Problem Solver



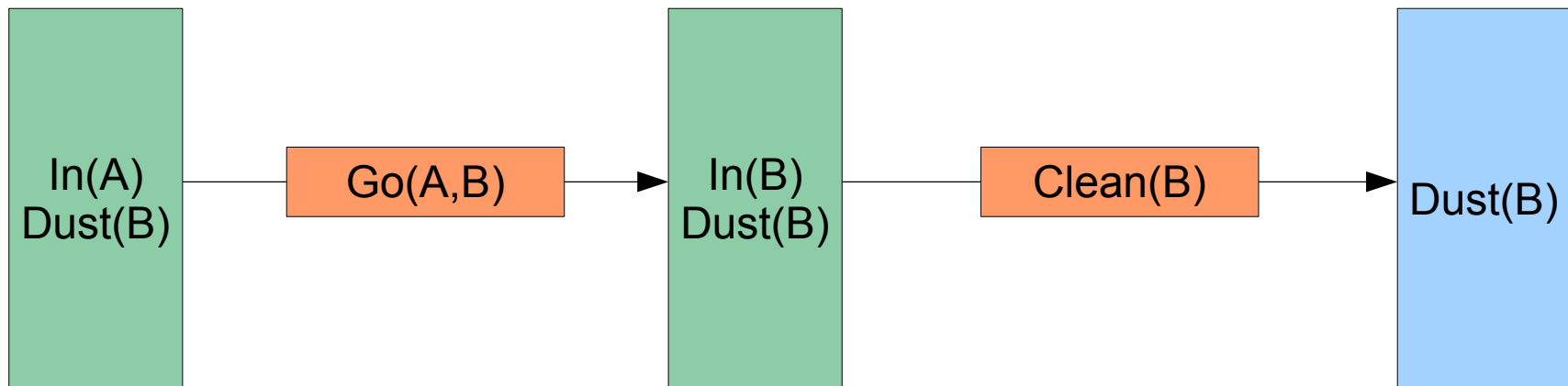
Le monde du recycleur

Go(x,y)

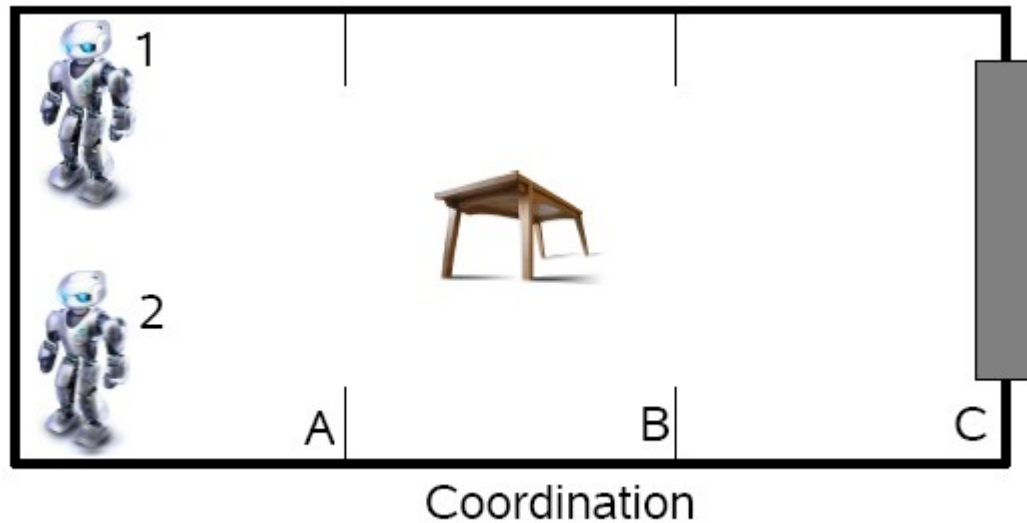
Condition: {In(x)}
Effet: {¬In(x),In(y)}

Clean(x)

Condition: {In(x),Dust(x)}
Effet: {¬Dust(x)}



Langage STRIPS

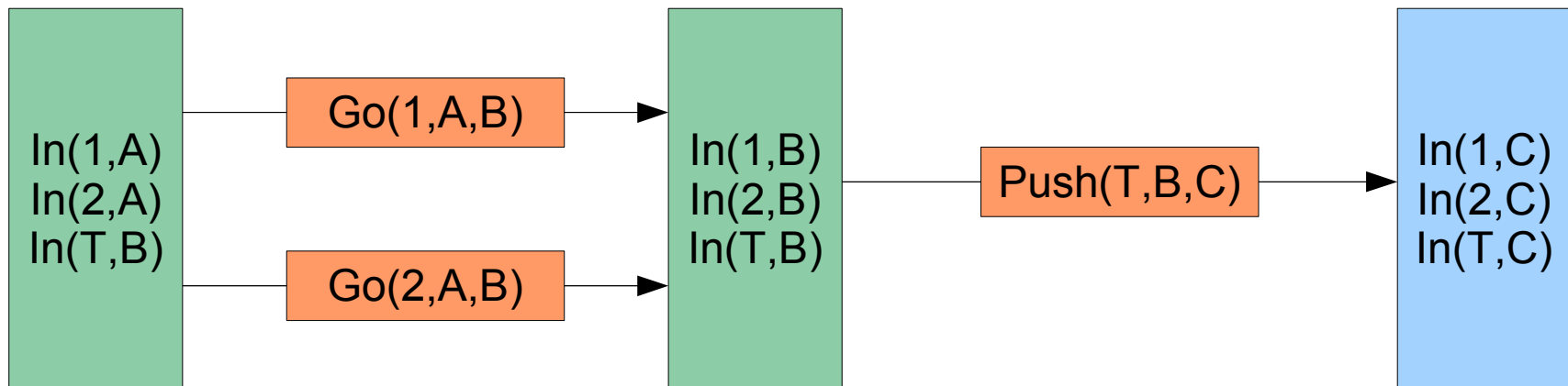


Go(r,x,y)

Condition: $\{In(r,x)\}$
Effet: $\{\neg In(r,x), In(r,y)\}$

Push(o,x,y)

Condition: $\{In(o,x), In(1,x), In(2,x)\}$
Effet: $\{\neg In(o,x), \neg In(1,x), \neg In(2,x), In(o,y), In(1,y), In(2,y)\}$



Langage STRIPS



Sussman Anomaly Problem

- Décrire le langage
- Décrire un plan d'actions

Langage STRIPS



But

Bloquer la sortie

Comment concevoir un agent planificateur dans un univers 3D ?

Langage STRIPS



Etat

```
At(self,1)
Up(1)
Middle(2)
At(w1,3)
At(w2,7)
Down(8)
Exit(8)
...
```

Localisations: données par le graphe des waypoints

Relations:

- Propriétés des localisations
- Présence d'objets sur les localisations
- Propriétés de l'agent

Langage STRIPS



Go(x,y)

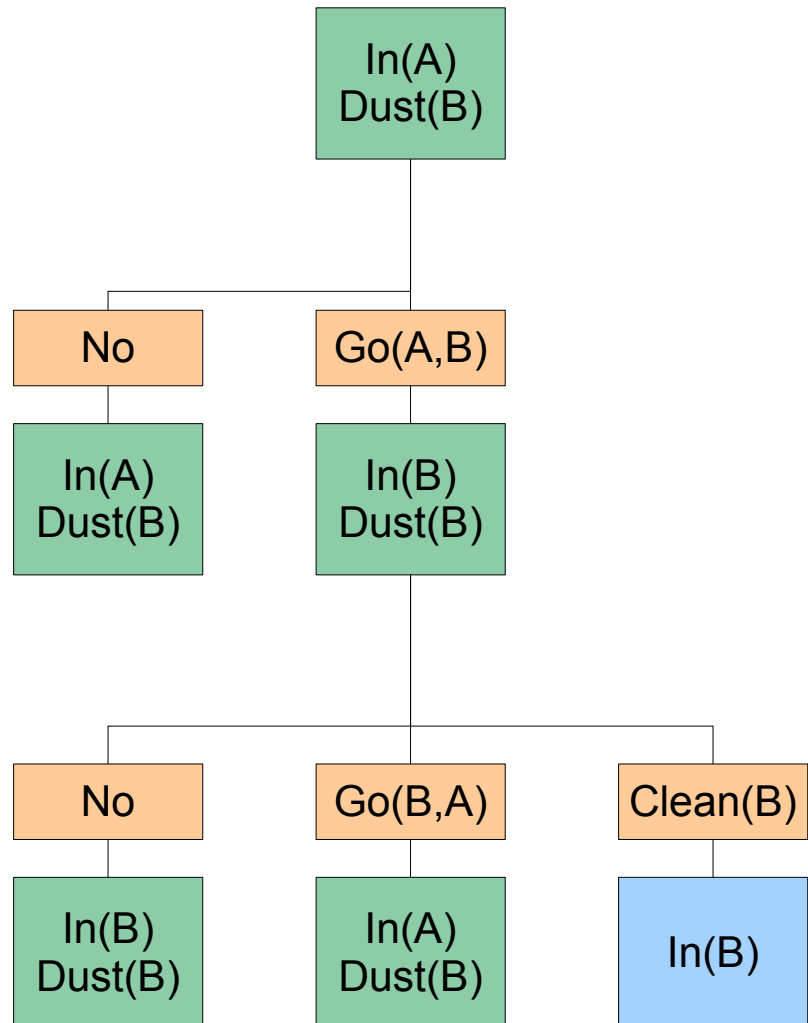
Condition: {At(self,x),Edge(x,y)}
Effet: {¬At(self,x),At(self,y)}

Grab(x,y)

Condition: {At(self,x),At(x,y)}
Effet: {¬At(x,y),Get(self,y)}

Actions: données par le profil de l'agent en utilisant le graphe des waypoints

Planification Arborescente



Forward Search

Root: état initial

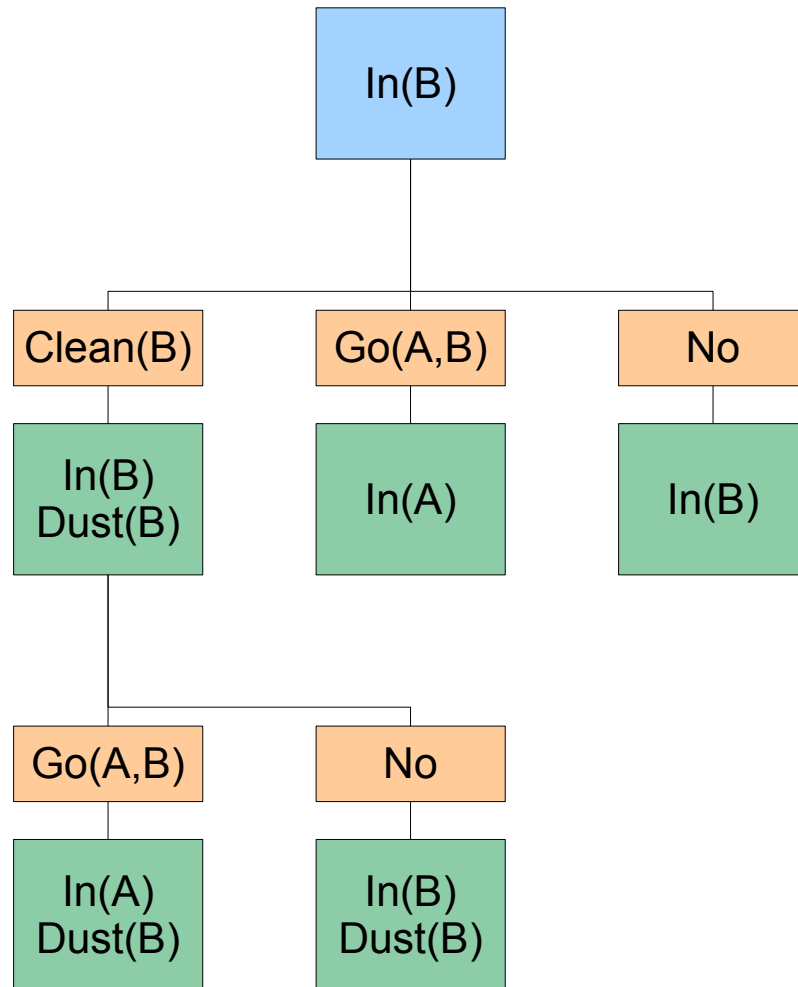
Edges: actions dont les conditions sont applicables sur le noeud parent

Cost: 1 par défaut, dépend du problème

Heuristique

Satisfaire progressivement les buts

Planification Arborescente



Backward Search

Root: état but

Edges: actions dont les effets sont applicables sur le noeud parent

Cost: 1 par défaut, dépend du problème

Heuristique

Eliminer progressivement les buts

Planification Arborescente

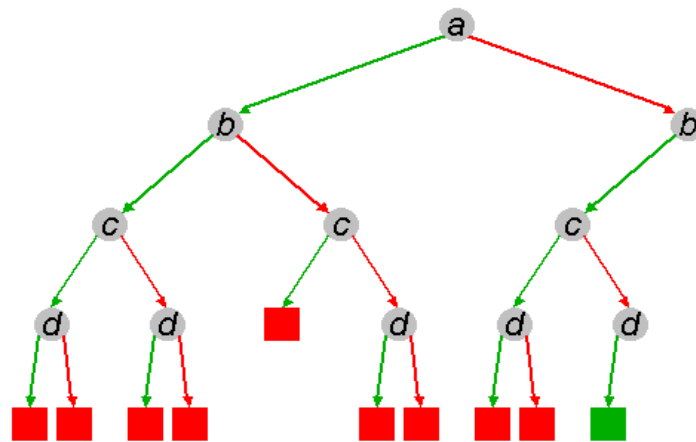
Bidirectional Search

Est-il possible de concevoir une recherche à la fois « forward » et « backward » ?

SAT-Plan

Idée

Coder un problème de planification comme un CSP booléen



SAT-Plan

Etat initial

- Conjonction des littéraux de base vrais dans S associés à l'étape 0
- littéraux positifs: atomes qui sont dans S
 - littéraux négatifs: négation des atomes qui ne sont pas dans S



Le monde du recycleur

$In(A,0) \ \& \ Dust(B,0) \ \& \ \neg In(B,0) \ \& \ \neg Dust(A,0)$

SAT-Plan

Etat but

Conjonction des littéraux de base vrais dans B associés à l'étape K

Si le but contient des variables: le convertir en CNF propositionnelle.



Le monde du recycleur

$In(B,K) \ \& \ \neg Dust(B,K)$

SAT-Plan

Etat but

Conjonction des littéraux de base vrais dans B

Si le but contient des variables: le convertir en CNF propositionnelle.



Le monde du recycleur

$In(B) \ \& \ \neg Dust(B)$

SAT-Plan

Actions

Chaque action est associée à une formule pour toute étape i de 0 à $K-1$:

$$\neg A(i) \vee (P_1(i) \wedge \dots \wedge P_n(i) \wedge E_1(i+1) \dots \wedge E_m(i+1))$$

Où:

- les P_j sont les préconditions de A
- les E_j sont les effets de A

Template de l'action $Go(x,y)$:

$$\neg Go(x,y,i) \vee (In(x,i) \wedge \neg In(x,i+1) \wedge In(y,i+1))$$

Actions propositionnelles: pour chaque i de 0 à $K-1$:

$$\neg Go(A,B,i) \vee (In(A,i) \wedge \neg In(A,i+1) \wedge In(B,i+1))$$

$$\neg Go(B,A,i) \vee (In(B,i) \wedge \neg In(B,i+1) \wedge In(A,i+1))$$

SAT-Plan

Axiomes du cadre

Tout littéral qui n'est pas un effet d'une action doit resté inchangé:

$$L(i) \vee \neg L(i+1) \vee (A_1(i) \vee \dots \vee A_n(i))$$

Où $A_j(i)$ sont les actions contenant $L(i+1)$ comme effet.

Template pour l'axiome du cadre de $\neg In(x)$

$$\neg In(x,i) \vee In(x,i+1) \vee Go(x,y,i)$$

Axiomes propositionnels: pour chaque i de 0 à $K-1$:

$$\neg In(A,i) \vee In(A,i+1) \vee Go(A,B,i)$$

$$\neg In(B,i) \vee In(B,i+1) \vee Go(B,A,i)$$

SAT-Plan

Exclusions

Les actions qui ne peuvent pas s'accomplir en parallèle sont mutuellement exclusives

$$\neg A_p(i) \vee \neg A_q(i)$$

Template de l'exclusion $Go(x,y)$:

$$\neg Go(x,y,i) \vee \neg Go(y,x,i)$$

Exclusions propositionnelles: pour chaque i de 0 à $K-1$:

$$\neg Go(A,B,i) \vee \neg Go(B,A,i)$$

Planification Graphique

Sussman Anomaly

Appliquer SAT-plan sur le problème

Graph-Plan

Graphe

Les noeuds sont séparés en niveaux successifs d'états et d'actions

Expansion: chaque état suivant est formé en appliquant toutes les actions possibles sur l'état courant

Extraction: lorsqu'un état possède un support consistant contenant le but, la solution est extraite

In(a)	\neg In(a)	In(b)	\neg In(b)	Du(a)	\neg Du(a)	Du(b)	\neg Du(b)
-------	--------------	-------	--------------	-------	--------------	-------	--------------

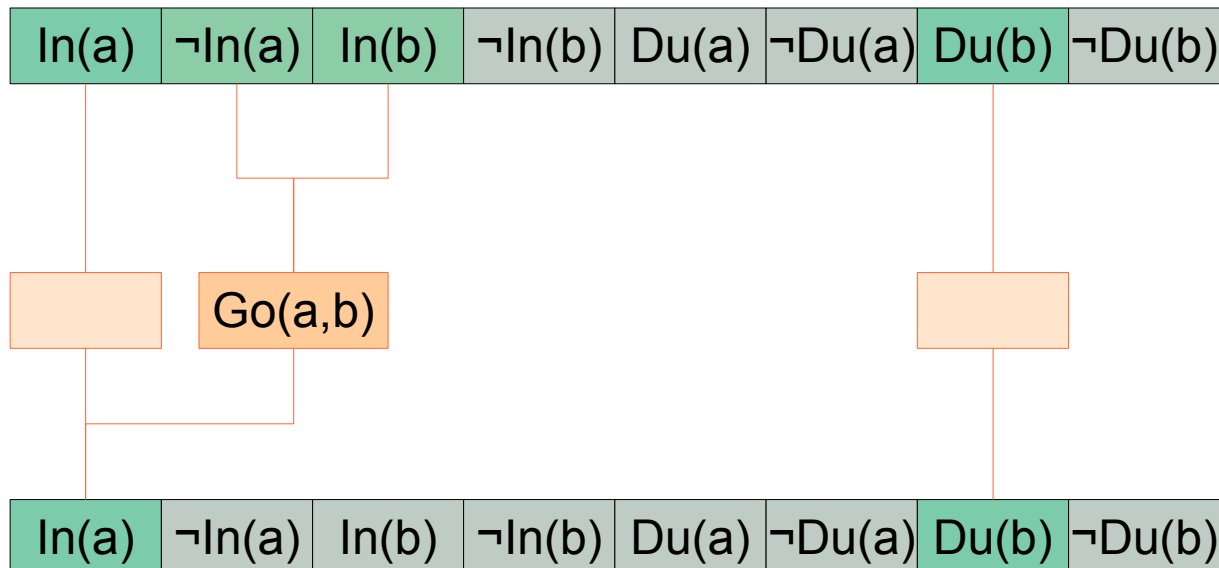
Graph-Plan

Graphe

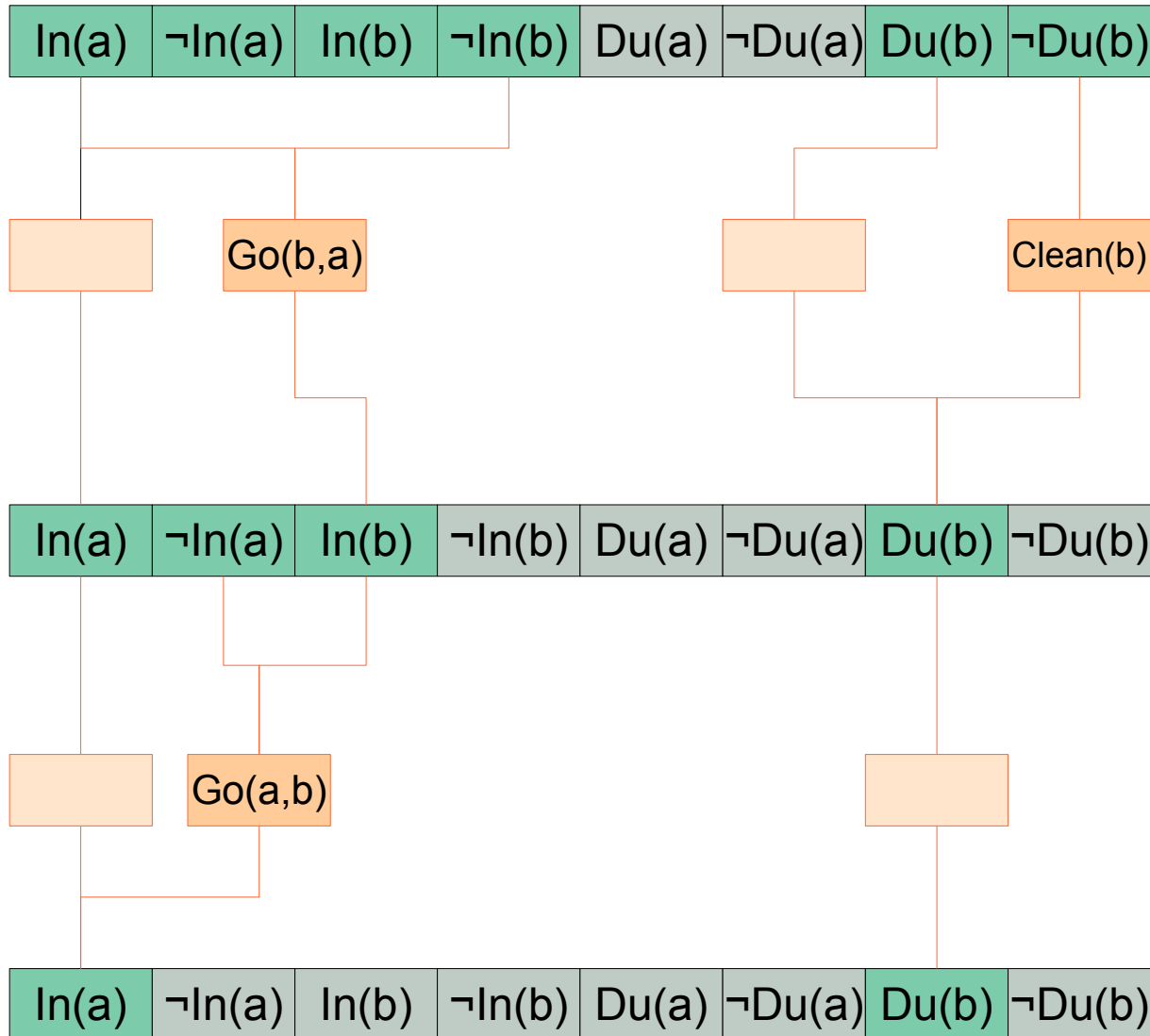
Les noeuds sont séparés en niveaux successifs d'états et d'actions

Expansion: chaque état suivant est formé en appliquant toutes les actions possibles sur l'état courant

Extraction: lorsqu'un état possède un support consistant contenant le but, la solution est extraite



Graph-Plan



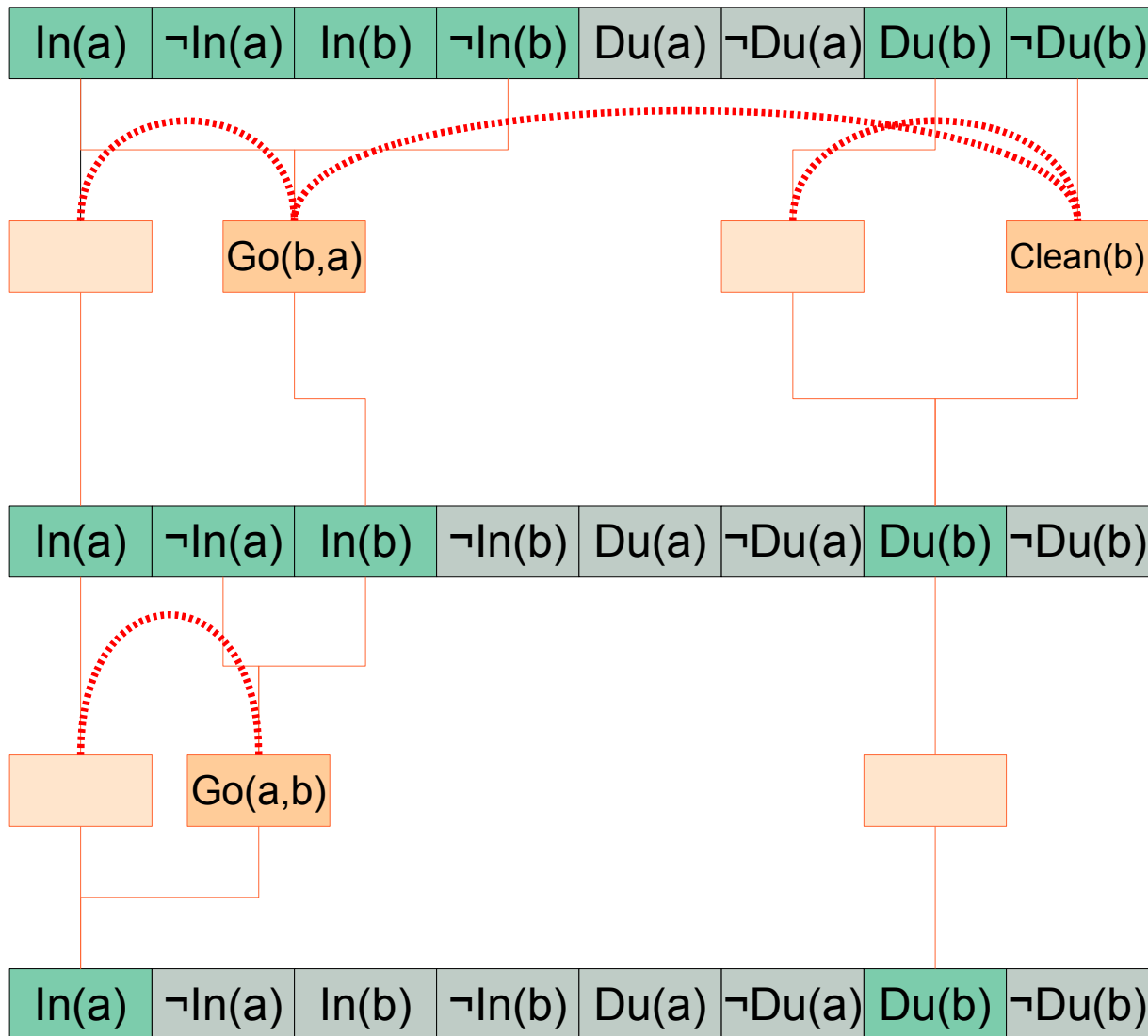
Graphe

Les noeuds sont séparés en niveaux successifs d'états et d'actions

Expansion: chaque état suivant est formé en appliquant toutes les actions possibles sur l'état courant

Extraction: lorsqu'un état possède un support consistant contenant le but, la solution est extraite

Planification Graphique



Mutex

Actions:

deux actions d'un même niveau son mutex si

- leurs effets interfèrent
- leurs conditions interfèrent
- l'effet d'une action est la condition d'une autre

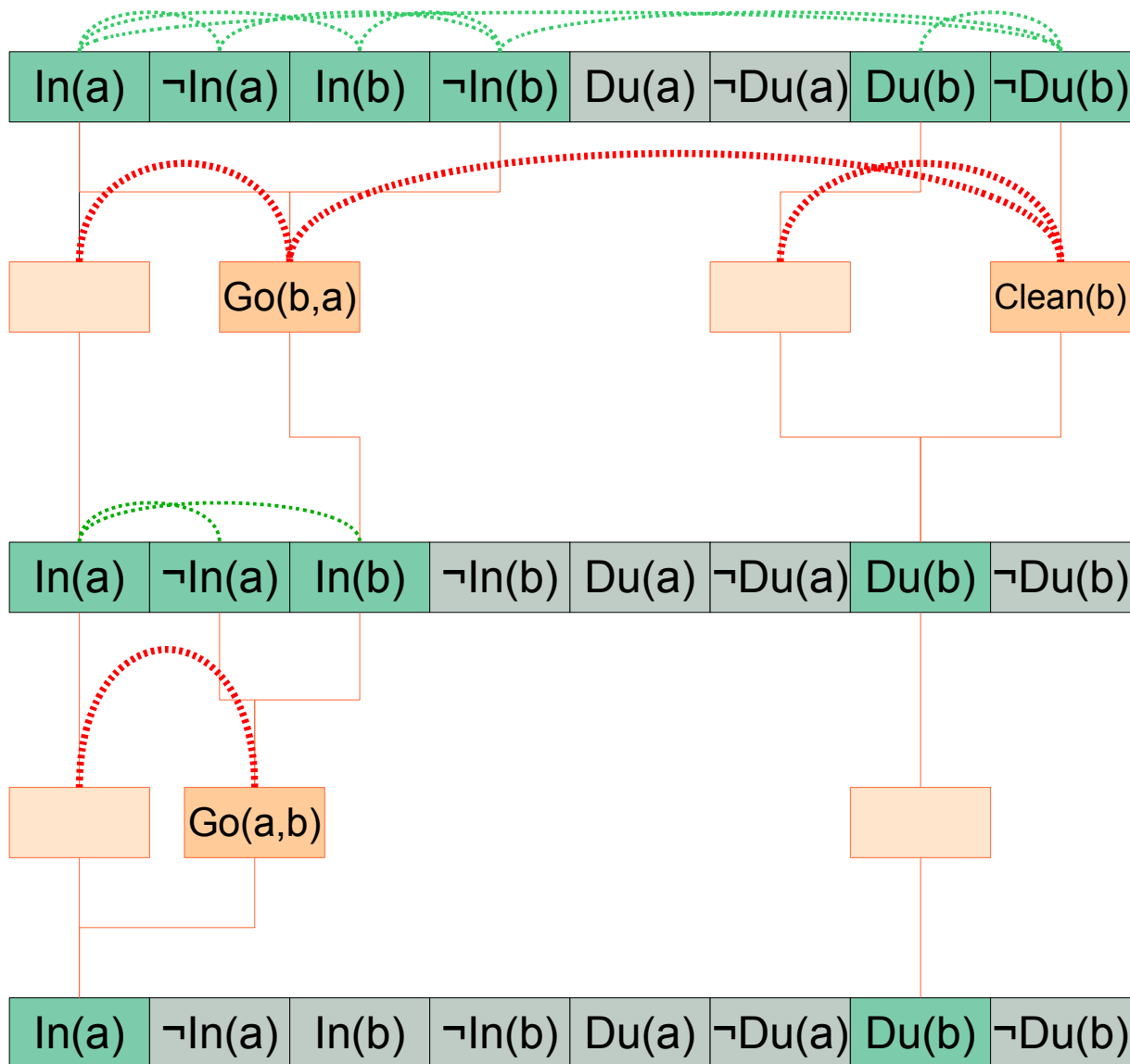
Littéraux:

deux littéraux sont mutex si toute paire d'actions aboutissant à ces littéraux est mutex

Support consistant:

ensemble de littéraux non mutex

Graph-Plan



Mutex

Actions:

deux actions d'un même niveau son mutex si

- leurs effets interfèrent
- leurs conditions interfèrent
- l'effet d'une action est la condition d'une autre

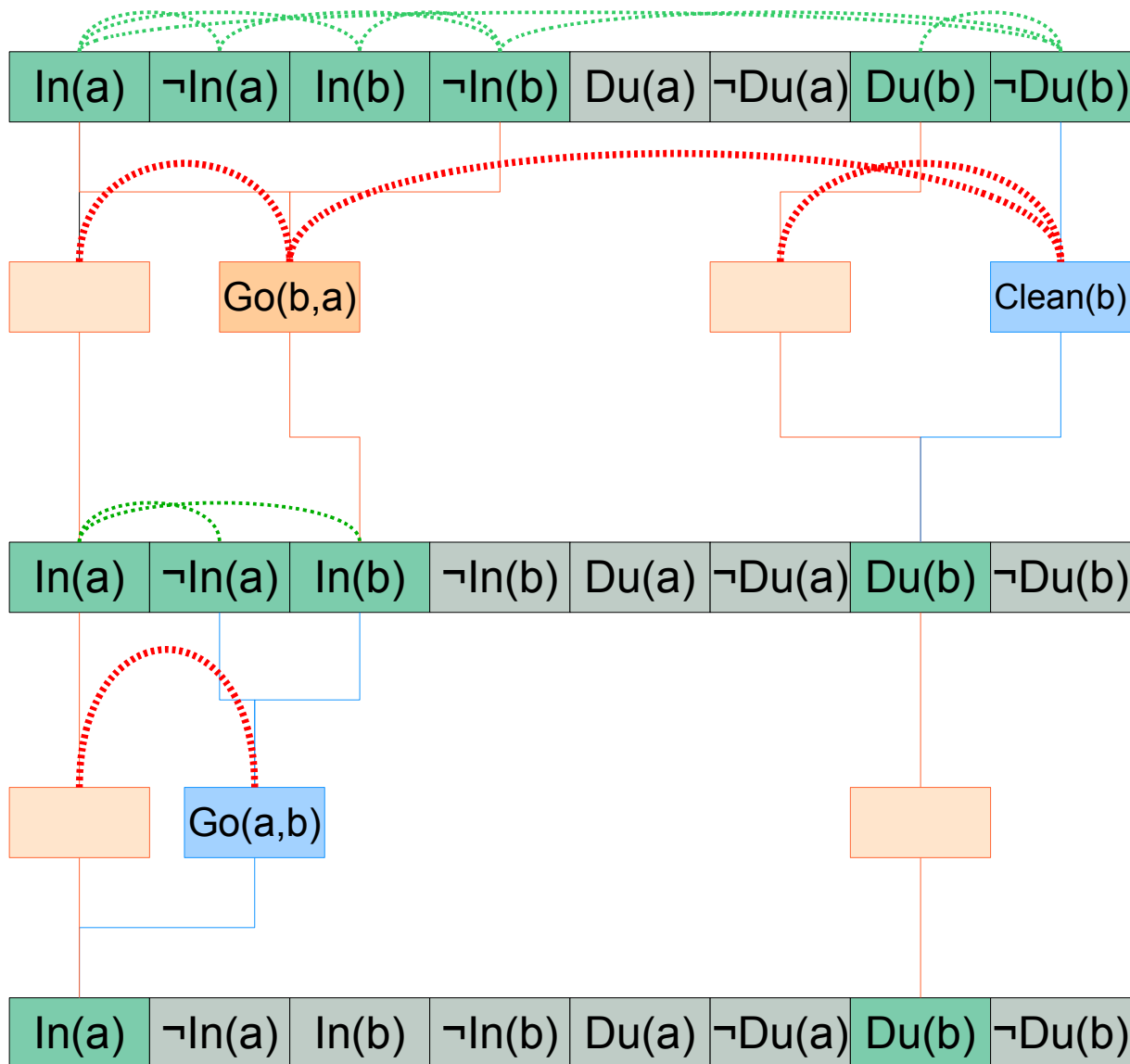
Littéraux:

deux littéraux sont mutex si toute paire d'actions aboutissant à ces littéraux est mutex

Support consistant:

ensemble de littéraux non mutex

Graph-Plan



Extraction

Partir du dernier niveau « N » du graphe

Le niveau « N-1 » est formé par un ensemble d'actions dont les effets couvrent le support consistant

L'opération est répétée jusqu'au niveau « 0 »

Extensions

1995-20..	Decision Theoretic Planning MDP, POMDP, Hybrid POMDP, First-Order MDP, etc.
1990-20..	Motion Planning Geometric, combinatorial, sampling-based motion planning
1980-2000	Neo-Classical Planning SAT-plan, Graph-plan, CSP-plan
1970-1985	Classical Planning A* search, branch & bound



Applications



Autonomous Remote Agents
(NASA deep space)



Robot Motion



Manufacturability Analysis



Strategy Games: STR, Pocker, Bridge, ...