

IFT 608 / IFT 702
Planification en intelligence artificielle

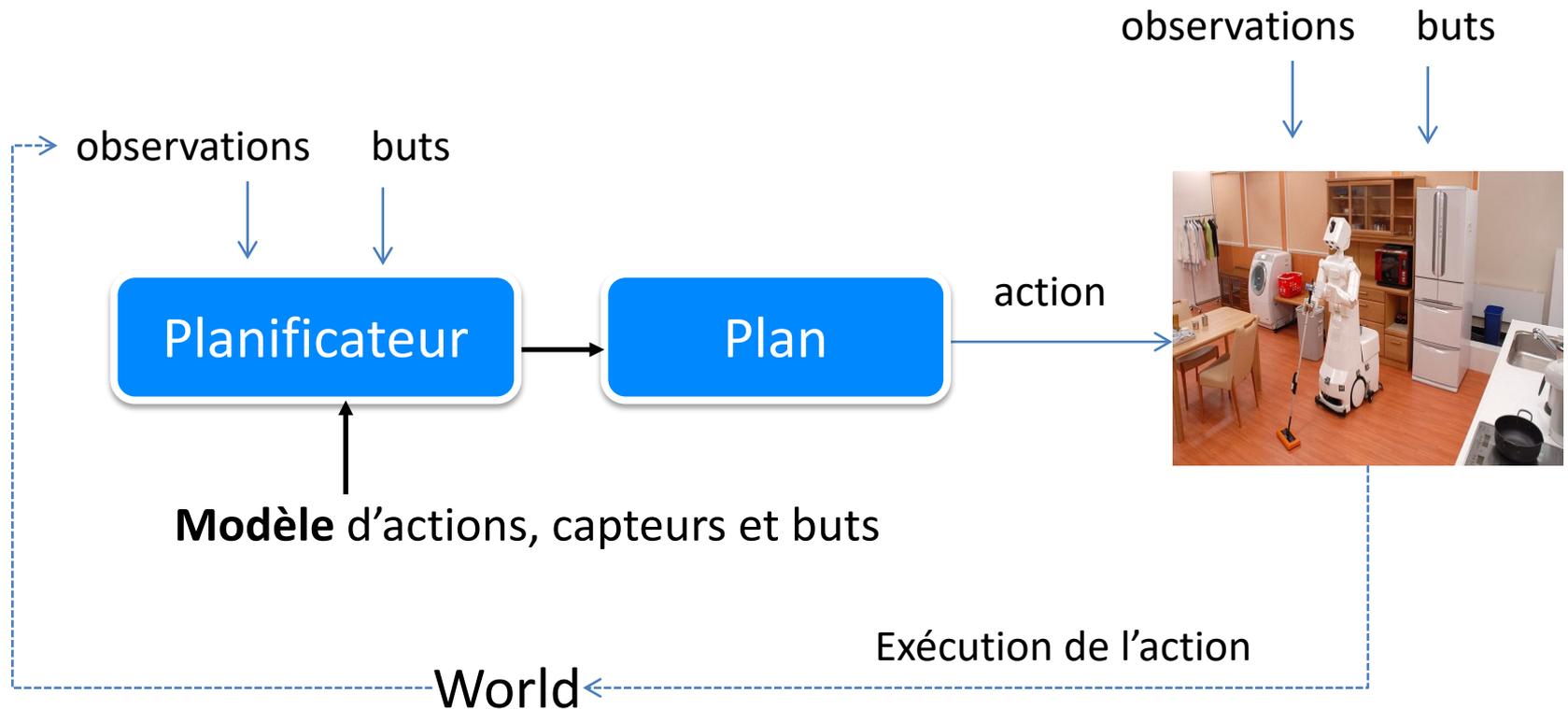
**Planification symbolique déterministe
par recherche heuristique dans l'espace d'états**

Froduald Kabanza
Département d'informatique
Université de Sherbrooke

Contenu

- Rappels
- Architecture d'un planificateur utilisant comme solveur une recherche dans un espace d'états

Rappel – un planificateur est un solveur de modèle



Rappel – Hypothèses sur le domaine

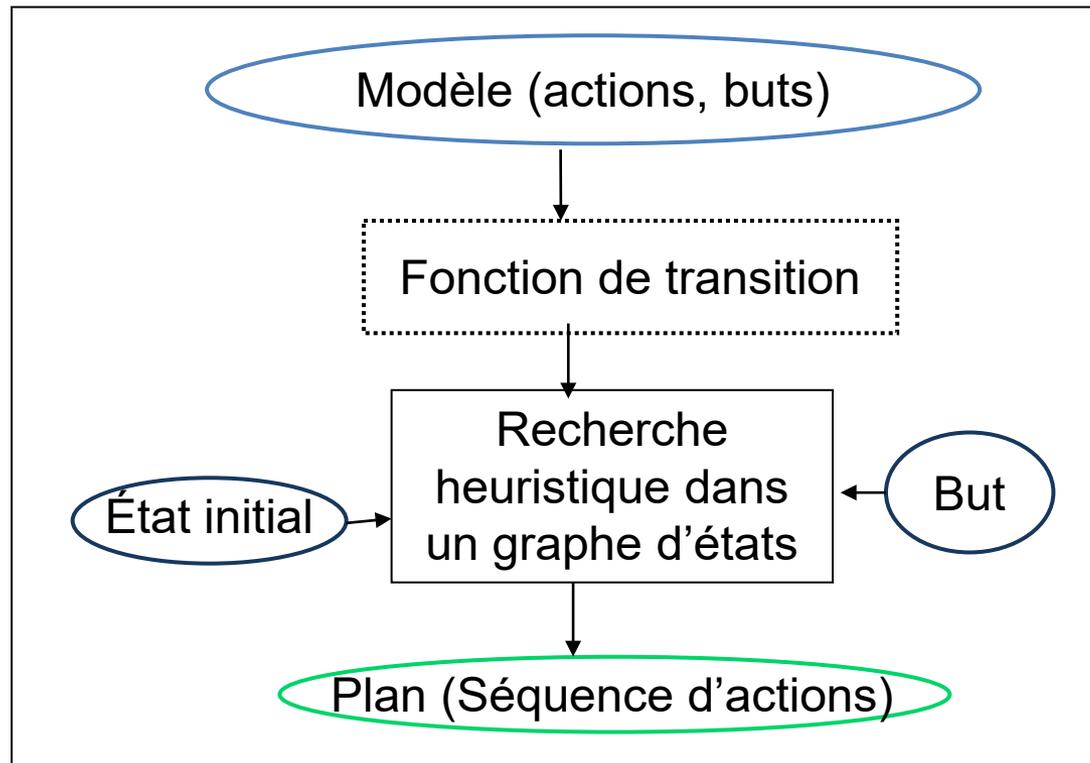
Les hypothèses du domaine à considérer sont:

- **Un seul agent** au lieu de plusieurs agents
- **Déterministe** au lieu de stochastique
- **Complètement observable** au lieu de partiellement observable
- **Séquencement d'actions** ayant des liens de causalité

Un algorithme défini avec ces hypothèses peut dans une certaine mesure être appliqué dans un environnement ne satisfaisant pas les deux premières hypothèses:

- Un planificateur déterministe centralisé peut planifier pour plusieurs agents
- L'incertitude est gérée par l'architecture décisionnelle en re-planifiant

Architecture d'un planificateur symbolique déterministe par recherche dans un espace d'états



- Le modèle ne décrit pas les capteurs puisque l'environnement est déterministe.
- Le modèle est transformé en fonction de transition pour un graphe d'états.

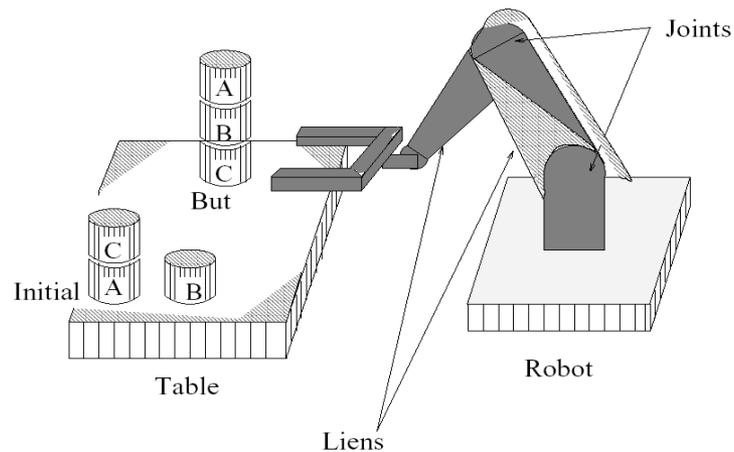
Prérequis IFT615

- IFT 615 (Recherche heuristique avec A*)
 - Algorithme A*
 - Rôle de la fonction heuristique dans A*

Exemple 1: Monde des blocs

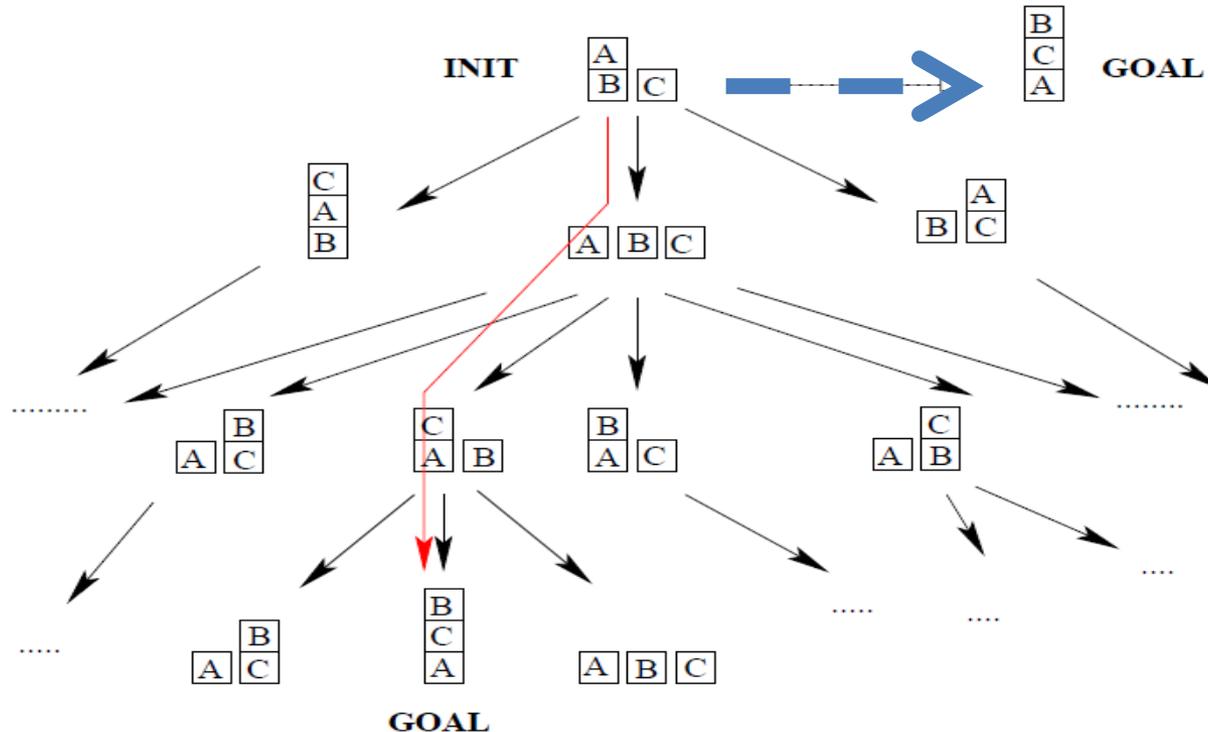
Un robot doit empiler des blocs dans une configuration indiquée.

Monde des blocs (*Blocksworld* en anglais)



Micro-environnement didactique, couramment utilisé en IA.

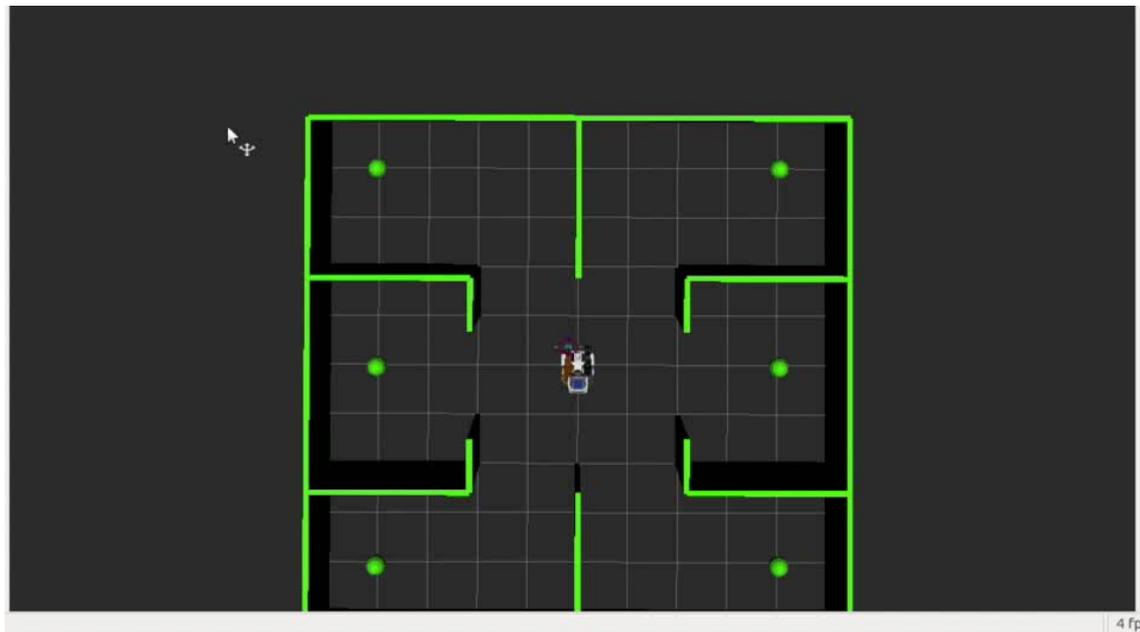
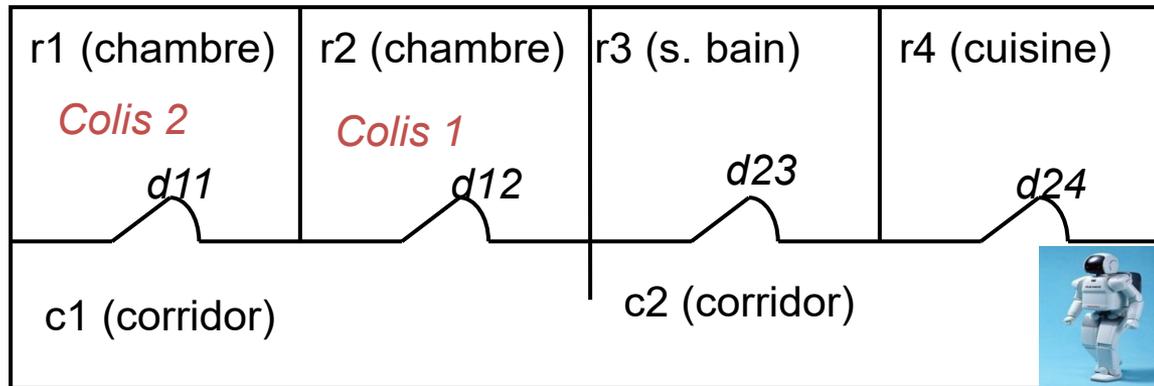
Exemple 1 : Empiler des blocs



- Étant donné un **modèle d'actions** primitives (prendre un block, relâcher un bloc, etc.), **trouver un plan** pour atteindre le but.
- Le problème est transformé en un problème de **trouver un chemin** dans un **graphe dirigé**.

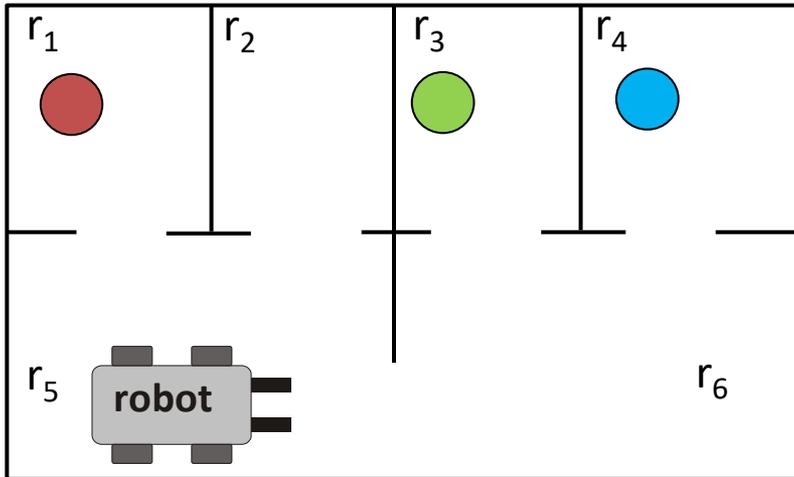
Exemple 2: Livraison de colis

Un robot doit recevoir des commandes de livraisons de colis et les exécuter.

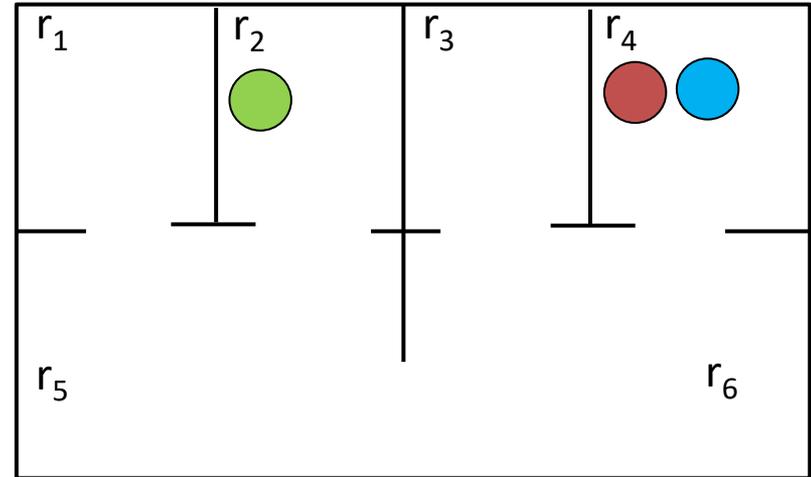


Exemple 2 : Livrer des colis

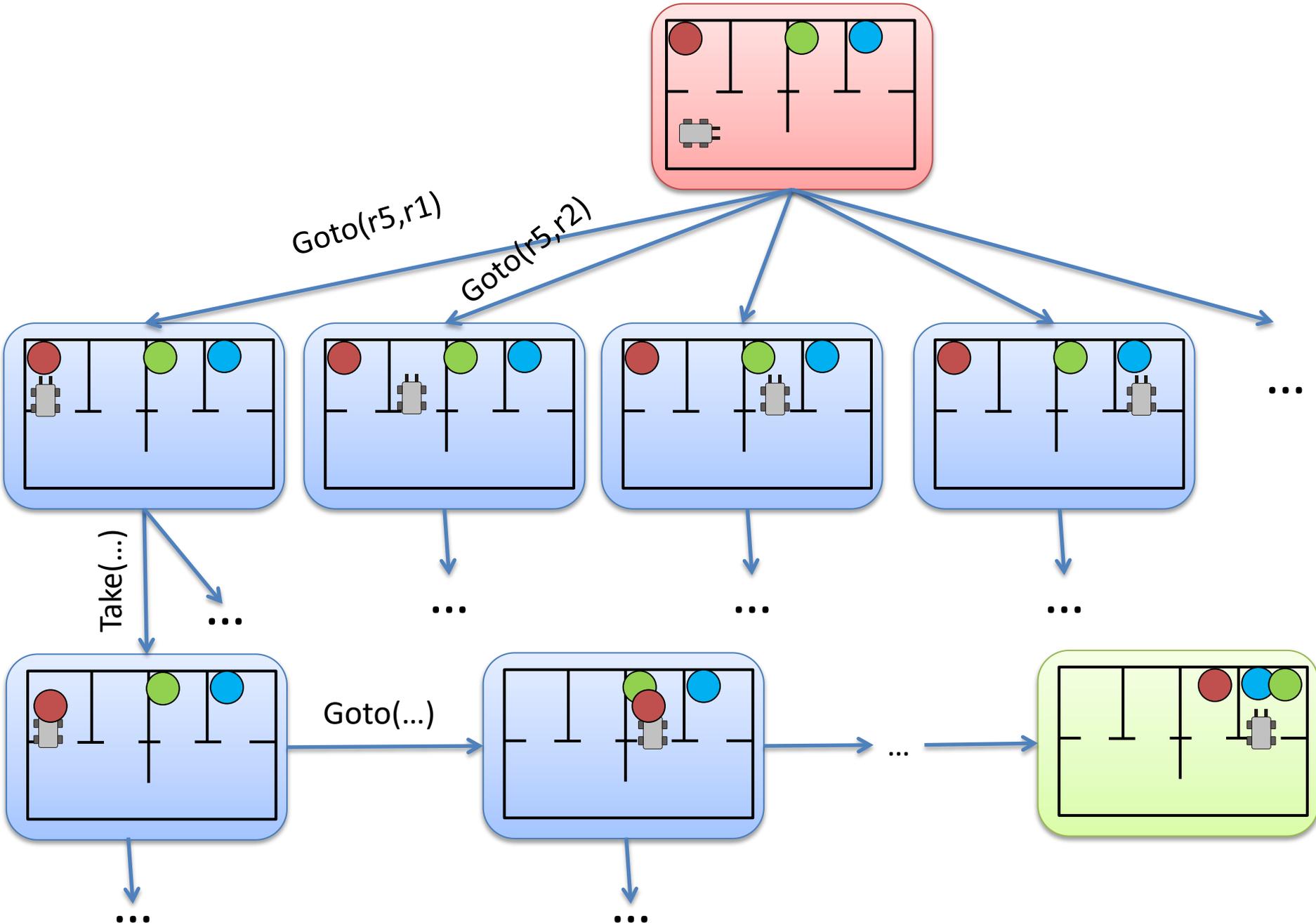
État initial



But



- Étant donné un **modèle d'actions** primitives (prendre un colis, relâcher un bloc, se déplacer d'une pièce à l'autre), **trouver un plan** pour atteindre le but.
- Le problème est transformé en un problème de **trouver un chemin** dans un **graphe dirigé**.



Planification par recherche dans un graphe

- Non informé: Largeur, profondeur, *iterative deepening*, *Dijkstra*, etc.
 - Ces algorithmes ne sont pas efficaces pour des problèmes qui nous intéressent. Ils n'ont aucun sens d'orientation; aucune intuition.
- A*
 - Dijkstra + une direction de recherche donnée par une **heuristique**.
 - Défi: apprendre l'heuristique
- Plusieurs autres approches. Exemple: **PlanSys** (<https://plansys2.github.io/>)
 - N'utilise pas A*
 - Il utilise une recherche dans un graphe différente
 - Le concept fondamental demeure le même.