

The Frame Problem

IA&Cognition

Présentée par:

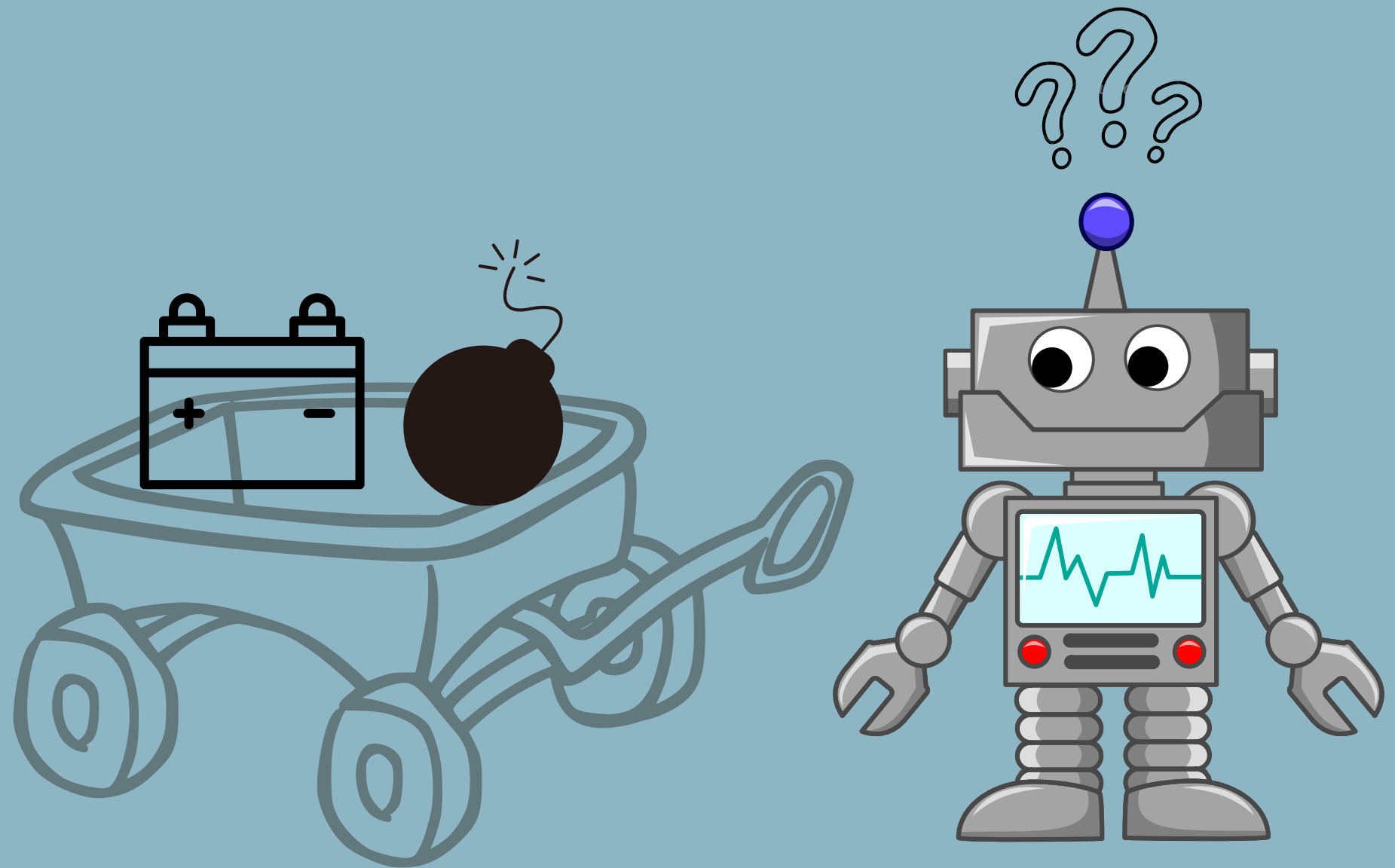
RÖTH Oscar

RIDOUAN Aafaf

HAOUAS Aida

MESSIKH Nassim

The frame problem



Sommaire

de la présentation

01

Introduction

04

Frame problem & Cognition

02

Illustration du frame problem

05

Débat & Réflexion

03

Solutions & Evolution du problème

06

Synthèse

Problématique : Comment un système intelligent peut-il prendre des décisions sans être submergé par des informations non pertinentes ?

Apparition : En 1969 par John McCarthy et Patrick Hayes dans le cadre du calcul des situations

Le problème englobe des questions philosophiques profondes sur la nature de l'intelligence, la pertinence et la cognition humaine.

Point de vue philosophique

Dennett et Jerry Fodor Le problème ne se limite pas aux robots. Il touche à la nature même de l'intelligence :

Hubert Dreyfus critique l'approche de l'IA traditionnelle, le problème du cadre montre les limites d'une vision purement logique et symbolique de l'intelligence.

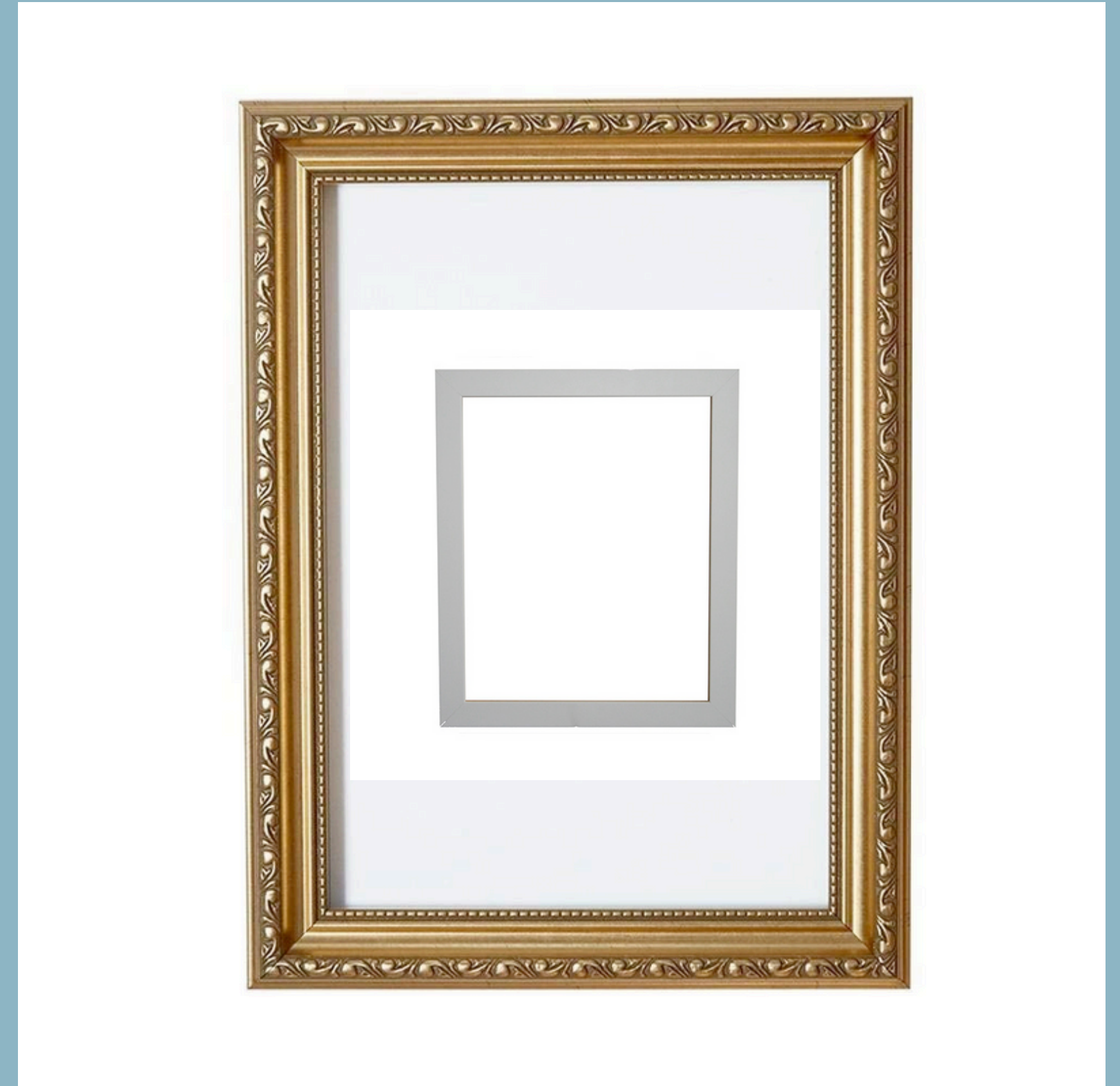
01

Introduction

le frame problem illustre les difficultés rencontrées par l'IA pour imiter le raisonnement humain.

Pose des questions philosophiques et technologiques

Défi technique et philosophique



02

Illustration du frame problem

Robot R1 : un seul objectif, se débrouiller. Un jour il se rend compte que sa batterie est dans une chambre fermée à clé. Il ouvre la porte, et trouve la batterie sur un wagon. Il utilise alors l'action prendre (Wagon, Chambre, t) pour sortir la batterie de la chambre. Mais la bombe est aussi sur la wagon donc...

R1D1 : calcule toutes les implications de ses actions. Après une longue réflexion la bombe explose

R2D1 : peut trier les implications à ignorer. Il décide de ne rien faire car il trouve cela inutile.

R2D2 : il faut résoudre le « frame problem » déjà, pour pouvoir faire ce robot issu de la culture populaire.

02

Illustration du frame problem

Illustration d'entrée dans une pièce

Solutions & Evolution du problème

Idée

Dans un environnement où les actions peuvent changer l'état du monde, le principal défi est de préciser non seulement ce qui change après une action, mais aussi ce qui ne change pas. Les axiomes du cadre fournissent des règles explicites qui énoncent les aspects du monde qui restent inchangés lorsqu'une action est effectuée.

composantes

Frame axioms : Ce sont des énoncés logiques qui déclarent, pour chaque action, quelles propriétés ou fluents (valeurs de vérité des propositions) ne changent pas.

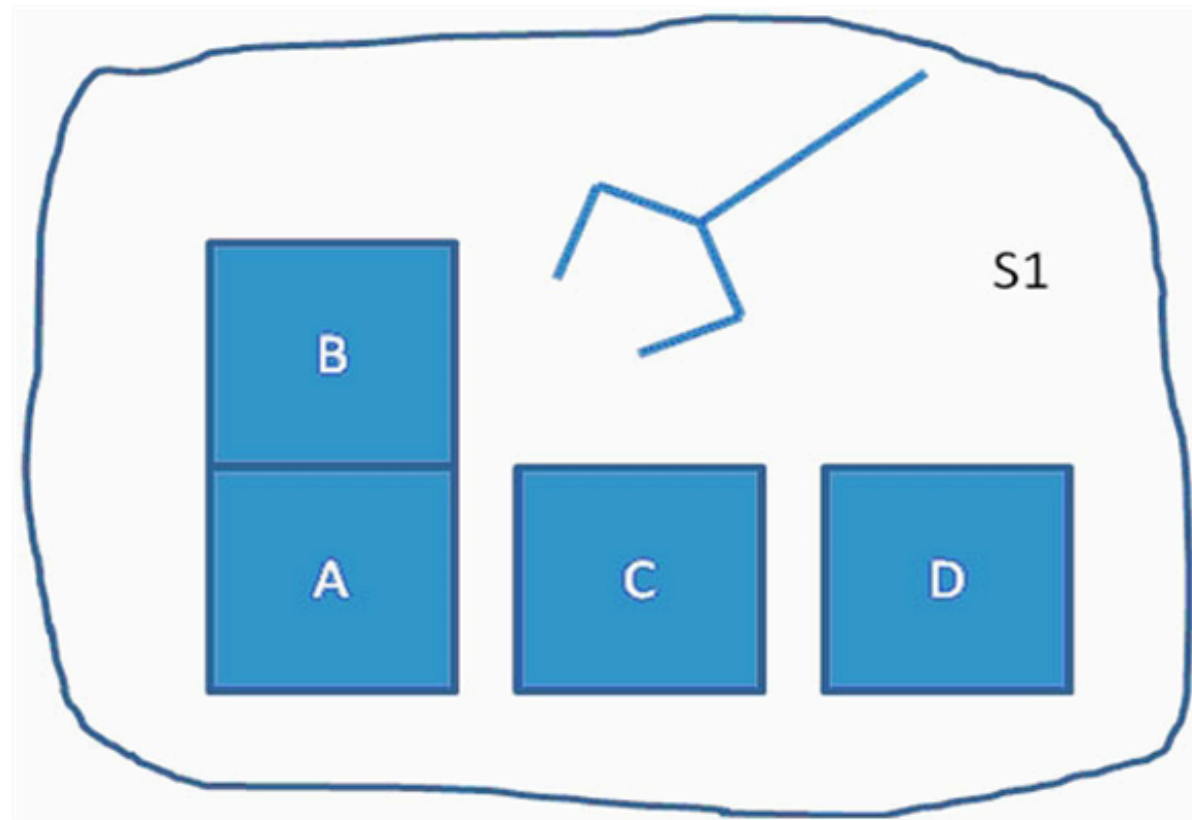
- **Effects** : représente les changements d'action.
- **Non-effects** : ce qui ne change pas dans l'environnement après l'action, capturé à l'aide d'axiomes de cadre.

01

Frame axioms

Exemple

World of blocks



composantes

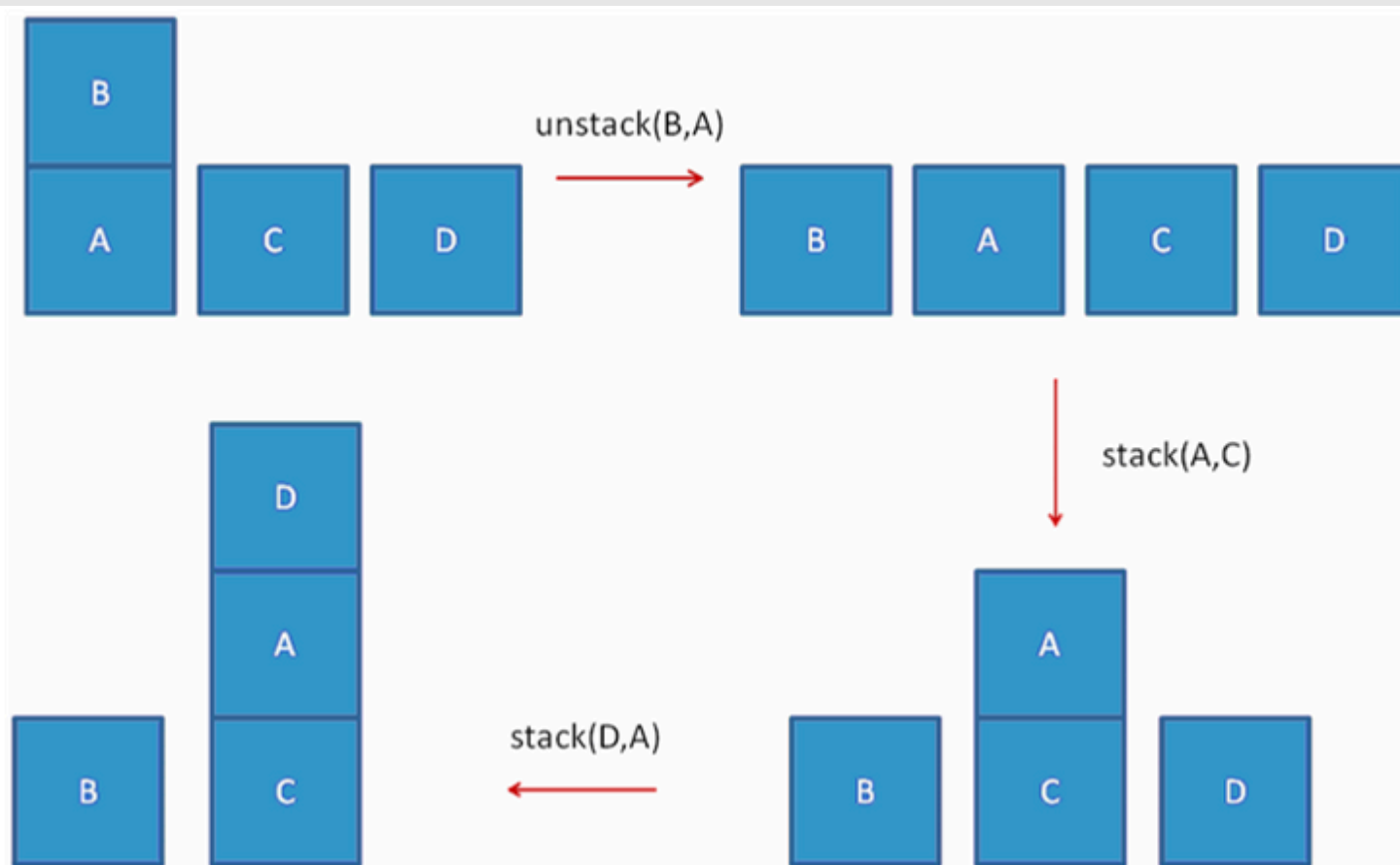
L'ontologie du calcul de situation se compose:

- Action
- Les **situations** sont des instantanés du monde à un moment donné.
- Un **fluents** est quelque chose qui change avec le temps.

01

Frame axioms

Exemple



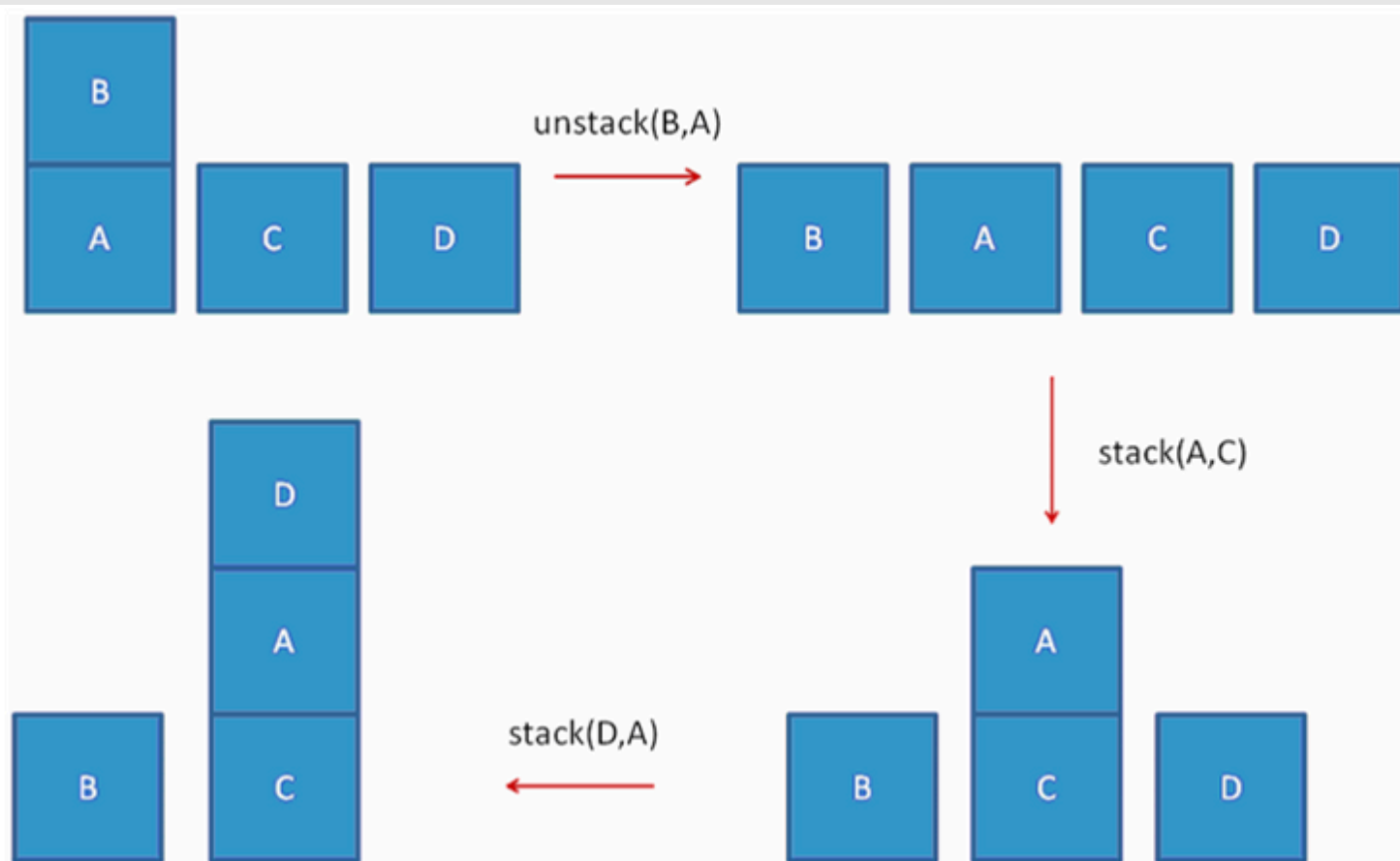
Démarche

- **Objectif:** Placer A sur C et D sur A,
 $Holds(On(A,C), Result(\alpha,SI))$
 $Holds(On(D,A), Result(\alpha,SI))$

Actions possibles :

- **Stack(X,Y)**
- **Unstack(X,Y)**

Exemple



Démarche

- Action 1 : **Unstack(B,A)**
- **Ce qui change:**

$\text{On}(B,A) \wedge \text{Clear}(B) \wedge \text{Clear}(C) \wedge \text{Clear}(D) \rightarrow \text{Holding}(B)$: Robot is holding B.

$\text{On}(B,A) \rightarrow \neg \text{On}(B,A)$: B is no longer on A

- **Ce qui ne change pas:**

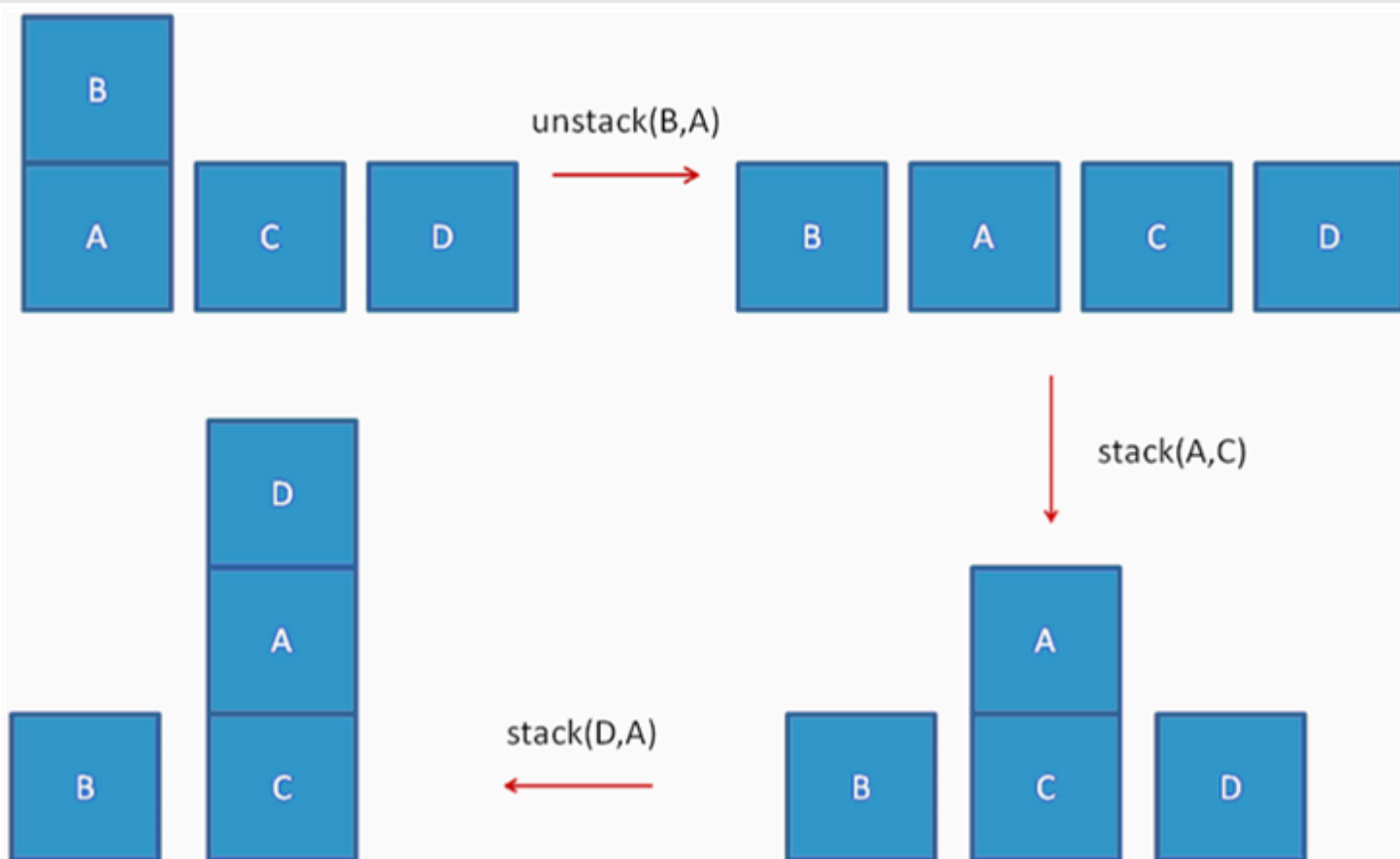
$\text{Holds}(\text{On}(C,\text{Table}),S1) \rightarrow \text{Holds}(\text{On}(C,\text{Table}),S2)$: C remains on the table.

$\text{Holds}(\text{On}(D,\text{Table}),S1) \rightarrow \text{Holds}(\text{On}(D,\text{Table}),S2)$: D remains on the table.

$\text{Holds}(\text{Clear}(C),S1) \rightarrow \text{Holds}(\text{Clear}(C),S2)$: C is still clear.

$\text{Holds}(\text{Clear}(D),S1) \rightarrow \text{Holds}(\text{Clear}(D),S2)$: D is still clear.

Exemple



Démarche

- Action 2 : **Stack(A,C)**
- **Ce qui change:**

$Holding(A) \wedge Clear(C) \rightarrow On(A,C) \wedge \neg Holding(A) \wedge \neg Clear(C)$: A is now on C, and C is no longer clear.

- **Ce qui ne change pas:**

$Holds(On(B,Table),S2) \rightarrow Holds(On(B,Table),S3)$: B remains on the table.

$Holds(On(D,Table),S2) \rightarrow Holds(On(D,Table),S3)$: D remains on the table.

$Holds(Clear(D),S2) \rightarrow Holds(Clear(D),S3)$: D is still clear.

Limitations

- **Explosion combinatoire** : Les axiomes de cadre peuvent entraîner une explosion combinatoire. Pour chaque action dans un système, il est nécessaire de spécifier que les propriétés non affectées par l'action restent inchangées. Cela génère un grand nombre d'axiomes de cadre, rendant le système complexe et difficile à gérer.
- **Complexité de la spécification** : Spécifier manuellement les axiomes de cadre pour chaque action et chaque état dans un environnement est fastidieux et sujet à erreur. Il faut expliciter pour chaque action quelles sont les propriétés de l'environnement qui ne changent pas, ce qui peut devenir très compliqué à grande échelle.

Idée

L'approche Explanation Closure (EC) est une méthode formelle développée pour résoudre le problème du cadre dans l'IA en spécifiant les conditions dans lesquelles les effets d'une action sont expliqués sans avoir besoin d'axiomes de cadre étendus.

composantes

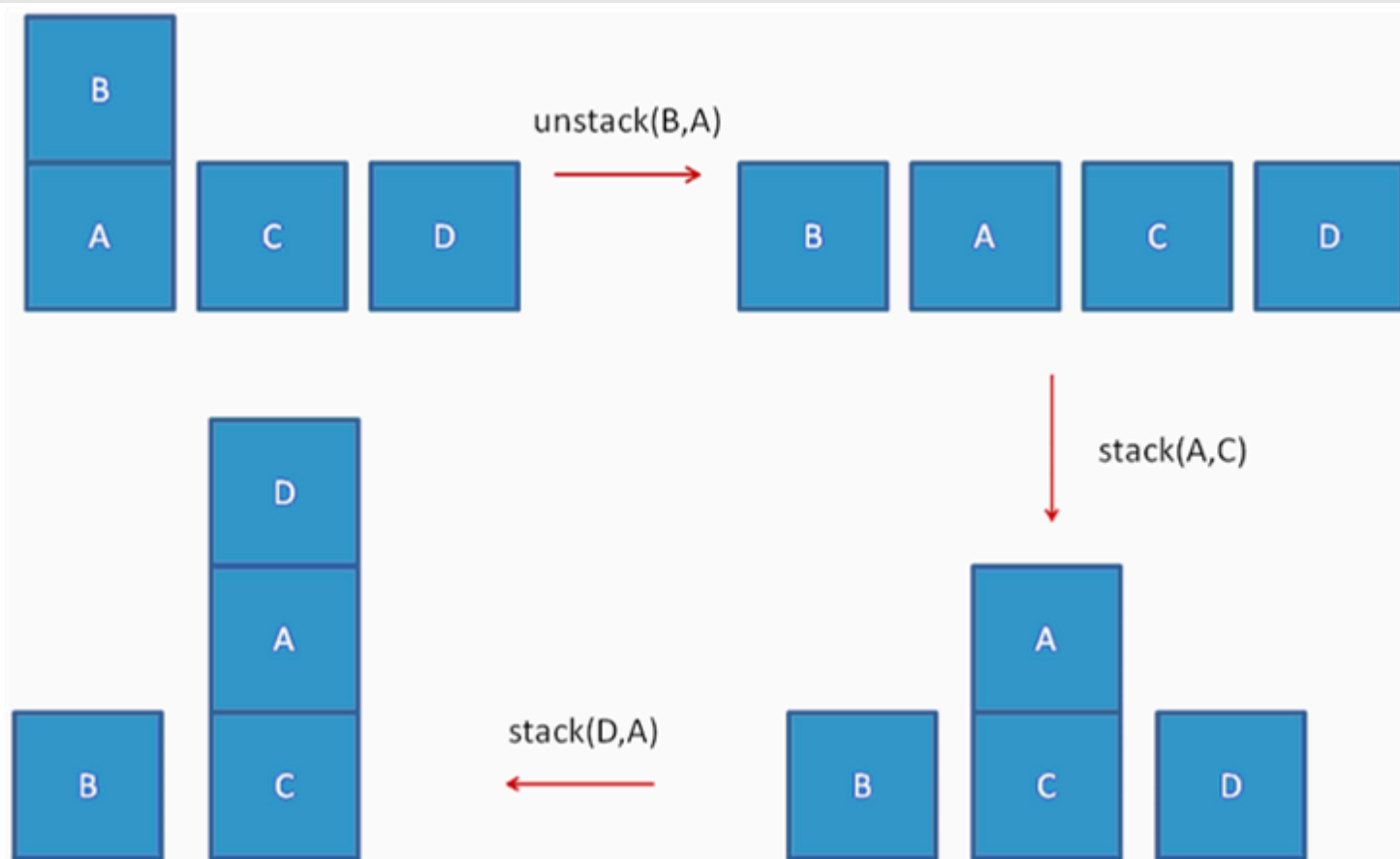
Closure axioms : Ce sont des axiomes qui couvrent tous les changements qu'une action pourrait provoquer. Tout ce qui n'est pas couvert par ces axiomes est supposé rester inchangé.

- **Si une action a un effet sur le monde, elle doit être explicitement expliquée.**
- **Si un aspect du monde n'est pas explicitement expliqué comme ayant été modifié, nous supposons qu'il n'a pas changé.**

02

Explanation Closure (EC)

Exemple



Démarche

- Action 1 : **Unstack(B,A)**

Etat initial:

- B is on A.
- C and D are clear.

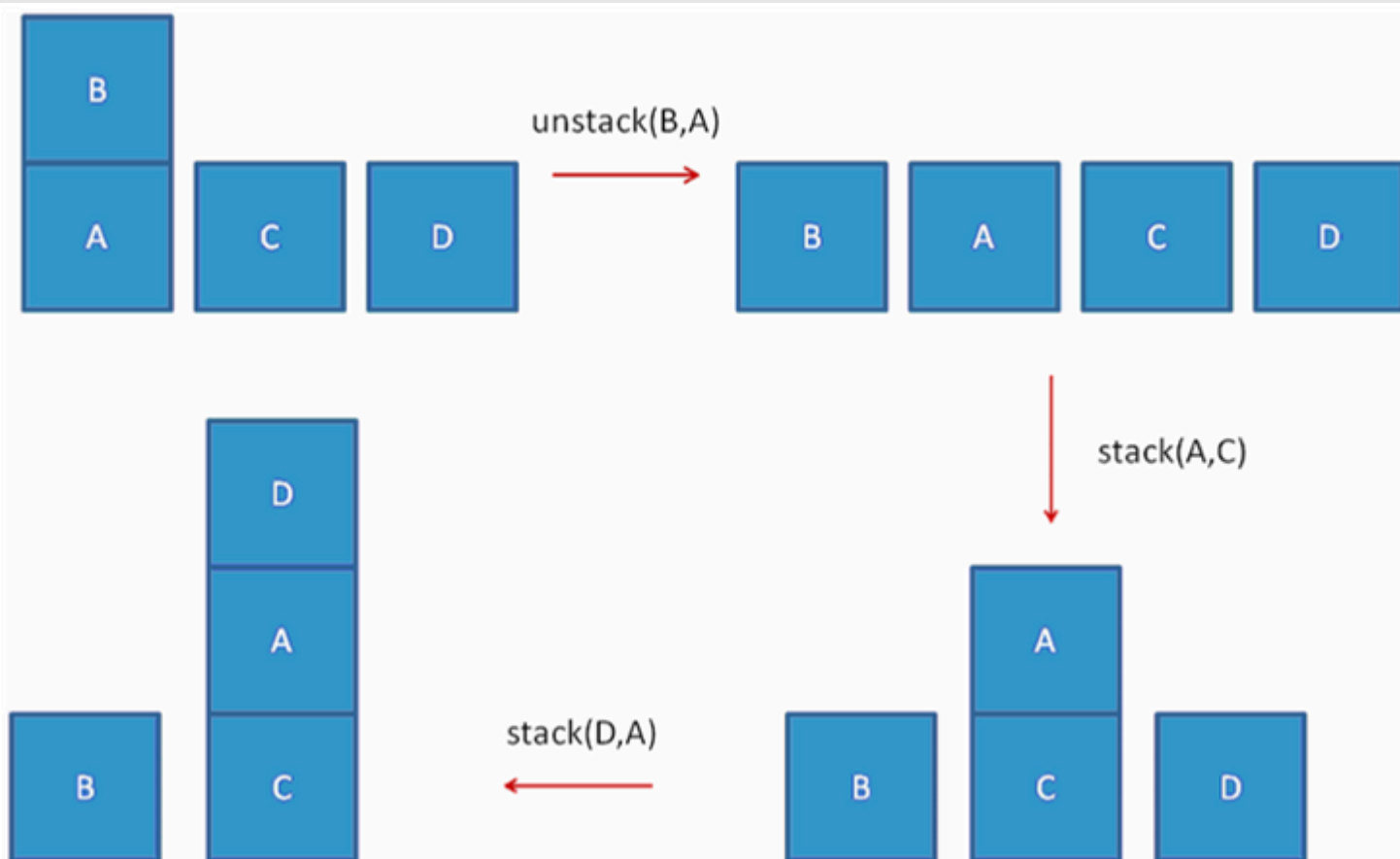
Closure axioms:

- $\text{Unstack}(B,A) \rightarrow \text{Holding}(B)$: The robot is holding block B.
- $\text{Unstack}(B,A) \rightarrow \neg \text{On}(B,A)$: Block B is no longer on A.

02

Explanation Closure (EC)

Exemple



Démarche

- Action 2 : **Stack(A,C)**

Closure axioms:

- $\text{Stack}(A,C) \rightarrow \text{On}(A,C)$: A is now on C.
- $\text{Stack}(A,C) \rightarrow \neg \text{Clear}(C)$: C is no longer clear.

Limitations

- **Non-monotonicité** : La fermeture explicative suppose souvent que l'ajout de nouvelles informations ne modifie pas les conclusions antérieures. Cependant, dans de nombreuses situations réelles, de nouvelles preuves peuvent changer l'interprétation des connaissances existantes, conduisant à un raisonnement non monotone où des conclusions peuvent être révisées.
- **Surcharge d'information** : Dans les systèmes comportant une grande quantité d'information, générer des explications tenant compte de toutes les données pertinentes peut entraîner une surcharge d'information, rendant difficile l'extraction d'insights utiles pour l'utilisateur.

03

Logique non-monotone

Définition

une façon de raisonnement qui nous permet de changer d'avis quand nous apprenons de nouvelles informations.

- Branche de la logique conçue pour le raisonnement défaisable
- Permet de réviser les conclusions face à de nouvelles informations
- Adaptée aux situations d'information incomplète

The frame problem

Analogie du quotidien

Imaginez que vous planifiez un pique-nique :

- Vous décidez d'y aller car la météo prévoit du soleil
- Mais le matin même, vous voyez des nuages menaçants
- Vous changez alors vos plans, même si la prévision initiale n'a pas changé

Logique classique	Logique non-monotone
Une fois une conclusion tirée, elle reste vraie	Les conclusions peuvent changer avec de nouvelles informations

03

Logique non-monotone

Approches principales (1980)

Ray Reiter

Raisonnement par défaut

John McCarthy

Circonscription

Avantages

- Plus proche du raisonnement humain
- Gestion efficace de l'incertitude et de l'information incomplète

The frame problem

Limitation

Nécessité d'axiomes de cadre explicites pour gérer la persistance des états inchangés.

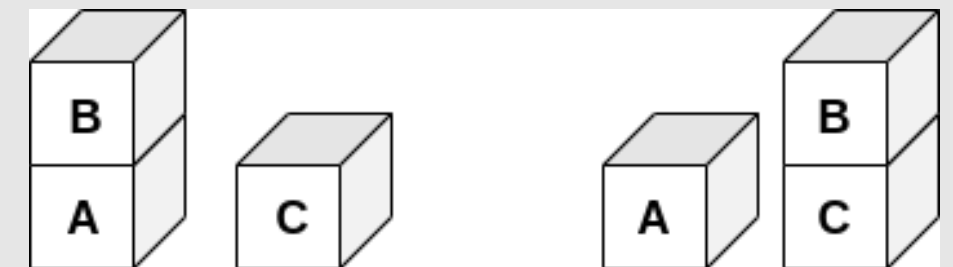
Application au Monde des Blocs

État Initial :

OnTable(A)
On(B, A)
OnTable(C)
Clear(B)
Clear(C)

Action :

Move(B, C)



1. Axiome de cadre

Les objets restent immobiles
sauf si déplacés:

- OnTable(A) reste vrai:

$\text{Move}(B, C, t) \rightarrow \text{OnTable}(A, t+1)$

- OnTable(C) reste vrai :

$\text{Move}(B, C, t) \rightarrow \text{OnTable}(C, t+1)$

- Clear(B) reste vrai:

$\text{Move}(B, C, t) \rightarrow \text{Clear}(B, t+1)$

2. Axiomes d'effet

- $\text{Move}(B, C) \rightarrow (\text{On}(B, C) \wedge \neg \text{Clear}(C))$
- $\text{Move}(B, C) \wedge \text{On}(B, A) \rightarrow \text{Clear}(A)$

On(B, C) devient vrai

Clear(C) devient vrai

Clear(A) devient vrai

3. Conditions d'anormalité

$\text{Move}(B, C) \rightarrow \text{Ab}(\text{On}(B, A), \text{Move}(B, C))$

$\text{Move}(B, C) \rightarrow \text{Ab}(\text{Clear}(C), \text{Move}(B, C))$

$\text{Move}(B, C) \wedge \text{On}(B, A) \rightarrow \text{Ab}(\text{Clear}(A), \text{Move}(B, C))$

4. État Résultant

OnTable(A)

OnTable(C)

On(B, C)

Clear(A)

Clear(B)

04

Approches probabilistes et statistiques

Définition

impliquent l'utilisation de modèles mathématiques basés sur la théorie des probabilités et les statistiques pour raisonner sur l'incertitude, faire des prédictions et apprendre à partir des données.

Approches principales

Réseaux Bayésiens	Programmation Probabiliste
<ul style="list-style-type: none">Modèles graphiques représentant des relations probabilistes entre variables	<ul style="list-style-type: none">Langages de programmation conçus pour décrire des modèles probabilistes

Avantages

- Approche quantitative
- Gestion fine de l'incertitude
- Capacité d'apprentissage à partir de données

Limites

- Complexité des calculs
- Dépendance aux données
- Difficulté d'interprétation

1. Définition de l'état initial

La position de chaque bloc est représentée comme une distribution catégorique, et le statut 'Clear' de chaque bloc est représenté comme une distribution de Bernoulli.

- 'A': `pyro.sample('A_position', pyro.distributions.Categorical(probs=torch.tensor([1.0, 0.0, 0.0])))`
- Clear_A: `pyro.sample('Clear_A', pyro.distributions.Bernoulli(probs=torch.tensor(0.0)))`

2. Définition de l'action de déplacement

```
def move(state, block, target):
    success_prob = 0.95 # 95% chance of successful move
    new_state = state.copy()
    if pyro.sample(f'move_{block}_to_{target}_success',
pyro.distributions.Bernoulli(success_prob)):
        # Update block positions
        new_state[block] = pyro.deterministic(f'{block}_new_position',
torch.tensor([0.0, 0.0, 1.0]))
        # Update clear states
        new_state['Clear_A'] = pyro.deterministic('Clear_A_new',
torch.tensor(1.0))
        new_state['Clear_C'] = pyro.deterministic('Clear_C_new',
torch.tensor(0.0))
    else:
        # If move fails, state remains unchanged
        pass
    return new_state
```

3. Definition du Model

définit le modèle probabiliste. Il configure l'état initial, l'observe (en s'assurant que notre modèle correspond aux conditions initiales connues), puis effectue l'action de déplacement.

```
def model():  
    # Initialisation  
    #observation de l'état initial  
    #action de déplacement
```

4. Inférence et Requêtes

Il interroge une distribution pour répondre à des questions probabilistes sur l'état après l'action de déplacement.

Exemples de requêtes :

- *Probabilité de succès du déplacement*
- *Probabilité que B soit sur C après le déplacement*
- *Probabilité que A soit dégagé après le déplacement*

Evolution du frame problem

**1970-
1980**

Premières Solutions:

- *Axiomes de cadre*
- *Circonscription (McCarthy, 1980)*

**1980-
1990**

Nouvelles Approches:

- *Problème du Yale Shooting*
- *Fermeture explicative (Explanation Closure)*

1990

Développement d'Approches Logiques:

- *Calcul des Situations (Reiter)*
- *Calcul des Événements (Shanahan)*

2000

Méthodes Avancées:

- *Calcul des Fluents (Thielscher)*

Présent

Recherche Continue et Intégration:

- *Combinaison avec des méthodes probabilistes et statistiques*

Frame problem & Cognition

06

Le problème de l'induction



Généralisation à partir de l'observation
Croyances basées sur l'habitude

06

Comparaison avec différentes approches cognitives

Cognitivism

Représentation symbolique des information

Connexionnisme

Réseaux de neurones apprennent à s'adapter

Énactivisme

Emergence de la cognition à travers l'interaction avec l'environnement



Problème de la pertinence : information utile ou à ignorer?

Mise à jour des croyances : Apprendre de nouvelles connaissances

Raisonnement implicite : processus de simplification et habitudes

Décision rapide et flexible : Parallélisation du traitement

Délibération : Décisions cohérentes dans des environnements inhabituels

Débat & Réflexion

Questions

Question 1 : Avec les progrès récents en intelligence artificielle, pensez-vous que le "frame problem" restera un défi fondamental ou sera-t-il surmonté par de nouvelles approches ?

Question 2 : L'être humain semble capable de résoudre des problèmes complexes avec des informations limitées. Selon vous, comment les systèmes d'IA pourraient-ils s'inspirer de la cognition humaine pour surmonter le "frame problem" ?

Question 3 : Peut-on imaginer une solution universelle au "frame problem" ou est-ce que différentes approches seront toujours nécessaires selon les contextes ?

Synthèse

Merci pour votre attention !

