

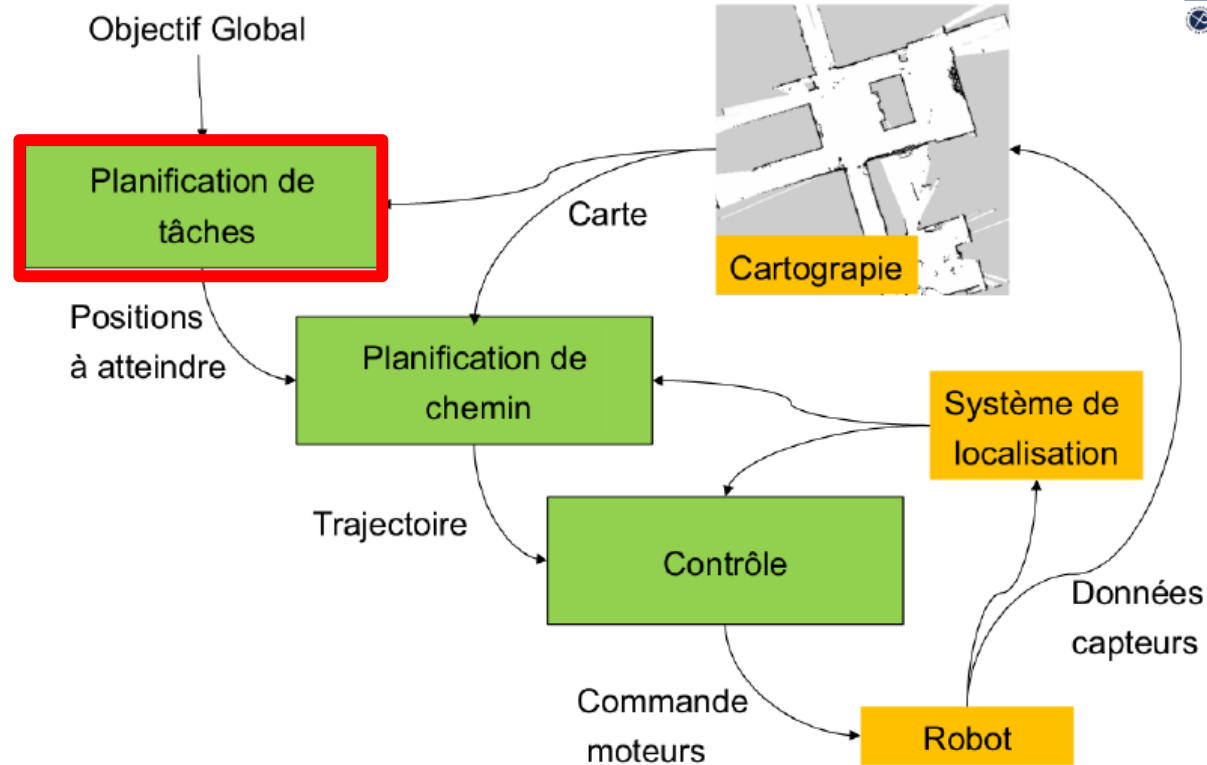
# Planification d'actions

## Heure 1

Philippe Morignot  
pmorignot@yahoo.fr

# Situation dans le cours ROB316

## Objectifs du cours



# Plan de ce cours pour ROB316

## Lundi 8 janvier 2024

1. Notions de base
2. Algorithmes pour planifier
3. Planification et exécution

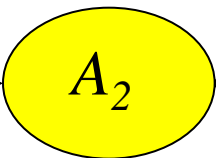
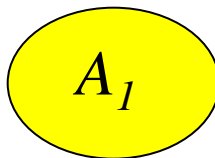
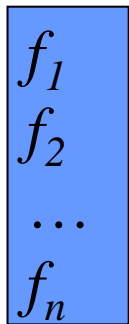
## Lundi 15 janvier 2024

4. Fin du cours éventuellement ...
5. TP ou début de projet en CPT version 2

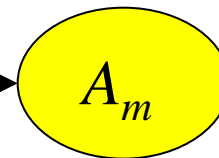
# Énoncé

« *Etant donnés des actions génériques possibles,  
un état et des buts,  
trouver une séquence d'actions instantiées,  
menant l'état initial à un état (final)  
contenant les buts.* »

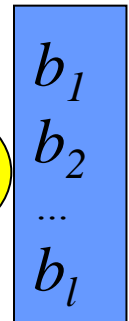
**Etat**



...

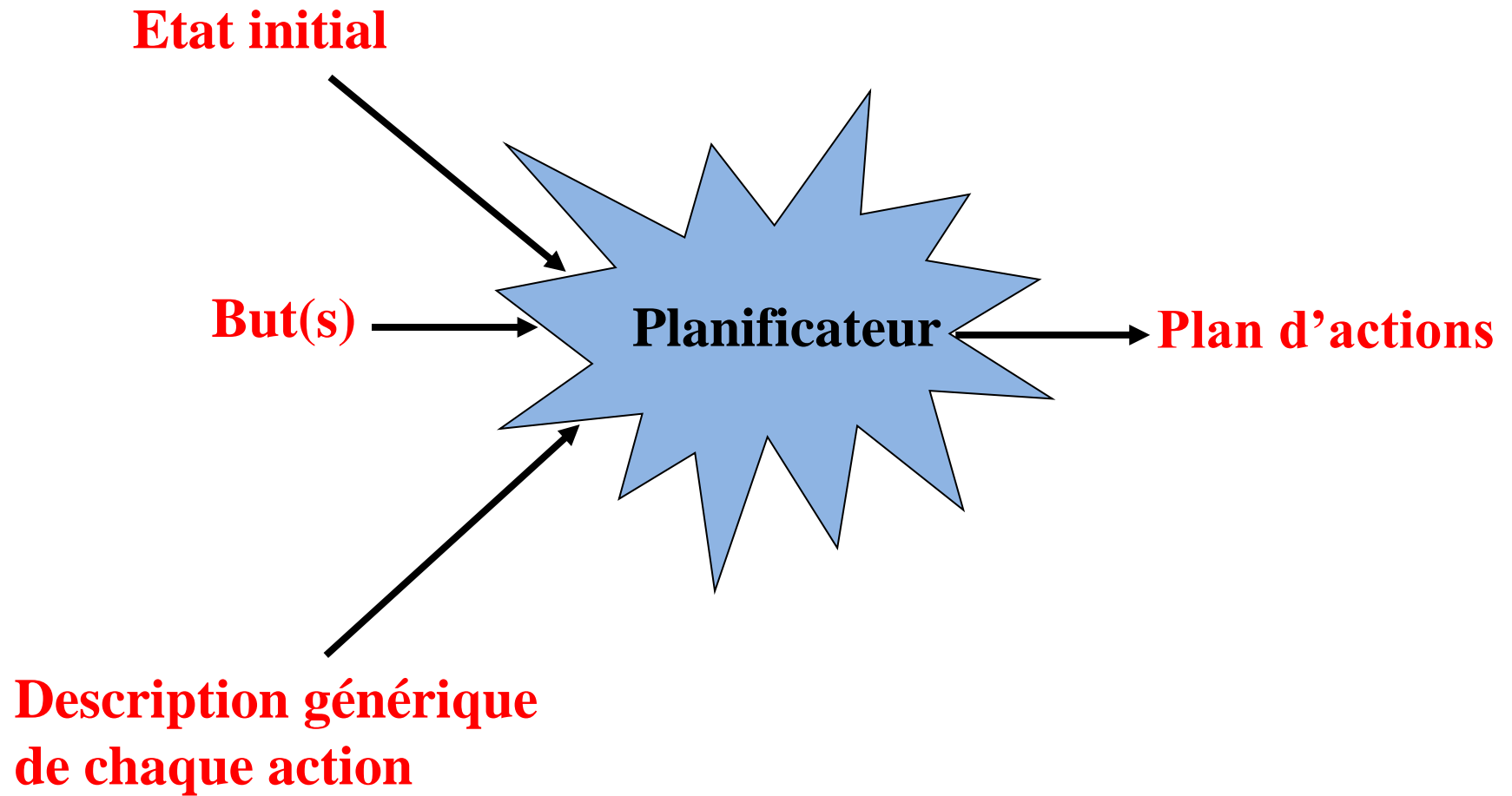


**Buts**



- « **Planification d'actions** » / « **planification de tâches** » / « **synthèse de plan** » / « **génération de plans d'actions** » : activité de construction d'un plan.
- « **Planificateur** » / « **planificateur de tâches** » / « **planificateur d'actions** » : programme informatique qui résout ce problème.
- Différent de « planificateur de chemin » en Robotique.

# Un planificateur d'actions



# Difficulté : explosion combinatoire

- Domaine du grutier :
  - 1 grue,  $a$  lieux,  $b$  camions,  $c$  piles de conteneurs,  $d$  conteneurs.



- Si  $a = 5$ ,  $b = 3$ ,  $c = 3$ ,  $d = 100$ , alors  $\sim 10^{277}$  états.
- La planification classique est non-déterministe polynomiale (NP).
- **On ne peut pas expliciter tous les états.**

# Hypothèses

- **Hyp. 1 : *l'agent est la seule cause de changement dans l'environnement.***
  - Pas d'autre agent, artificiel ou humain.
- **Hyp. 2 : *l'environnement est totalement observable, l'agent en a une connaissance parfaite.***
  - L'agent ne raisonne (e.g., planifie) pas sur des choses qu'il ne connaît pas.
- **Hyp. 3 : *l'environnement est statique.***
  - Même si l'environnement peut avoir des lois de comportements, il ne bouge pas spontanément.
- **Hyp. 4 : *le nombre d'objets considérés est fini.***
  - Logique des propositions.

# Planning Domain Definition Language (PDDL) (1 / 2)

- Langage de représentation pour définir :
  - un **domaine** : opérateurs
  - un **problème** : état et buts.
- Un opérateur est composé de :
  - **Pré-conditions** : termes qui doivent être vrai pour que l'action puisse s'exécuter.
  - **Effets / post-conditions** : termes que l'exécution de l'action change par rapport à l'état entrant (liste d'ajouts ADD-LIST / de retraits DELETE-LIST).
    - Une post-condition peut être positive ou négative.
- Un terme peut être parfois vrai, parfois faux, suivant l'instant où on le considère dans le plan.
  - Connecteur « *not* ». Ex. : (**not** (SUR SOURIS TAPIS))
  - « *Fluent* » (*littéral*). Ex. : (SUR SOURIS TAPIS)

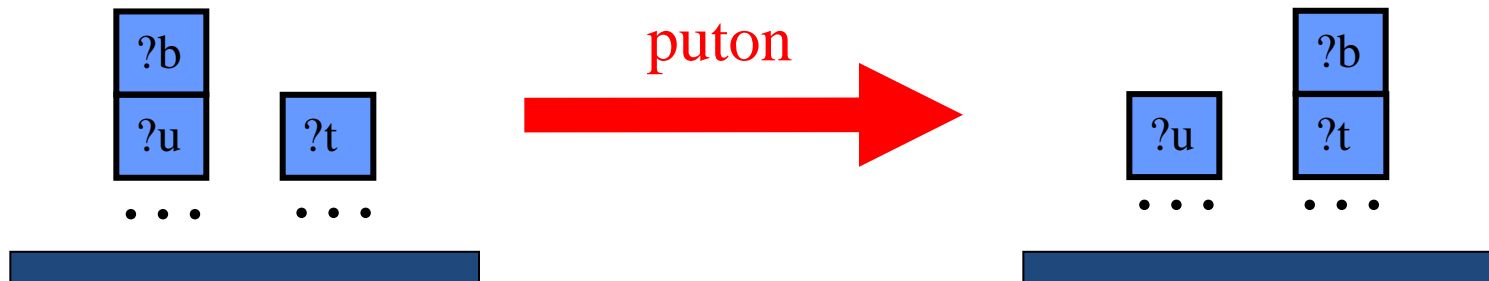


# PDDL : exemple de domaine

## Le monde des cubes

- Opérateur : **(:action puton**  
**:parameters (?b ?u ?t - block)**  
**:precondition (and (clear ?b)**  
**(on ?b ?u)**  
**(clear ?t))**  
**:effect (and (not (on ?b ?u)) (clear ?u)**  
**(on ?b ?t) (not (clear ?t))))**

puton ?b ?u ?t	
(clear ?b)	(not (on ?b ?u))
(on ?b ?u)	(clear ?u)
(clear ?t)	(on ?b ?t)
	(not (clear ?t))



- Mais comment représenter des extensions : Et la table ? Et le bras ? Et si plusieurs bras ? Et si les cubes sont colorés ? Ou avec une encoche ? Ou de dimensions variables ?
- Conditionnelles ? Quantification universelle ?

# PDDL : exemple de problème

## Le monde des cubes

```
(define (problem blocks-24-1)
  (:domain blocks)
  (:objects X W V U T S R Q P O N M L K J I H G F E D C A B)
  (:init
    (CLEAR K) (CLEAR I) (ONTABLE C) (ONTABLE O)
    (ON K F) (ON F T) (ON T B) (ON B G) (ON G R)
    (ON R M) (ON M E) (ON E J) (ON J V) (ON V N)
    (ON N U) (ON U H) (ON H C) (ON I A) (ON A P)
    (ON P Q) (ON Q D) (ON D W) (ON W X) (ON X S)
    (ON S L) (ON L O) (HANDEEMPTY))
  (:goal (and
    (ON L C) (ON C P) (ON P Q) (ON Q M) (ON M B)
    (ON B G) (ON G F) (ON F K) (ON K E) (ON E R)
    (ON R A) (ON A W) (ON W T) (ON T N) (ON N J)
    (ON J U) (ON U S) (ON S D) (ON D H) (ON H V)
    (ON V O) (ON O I) (ON I X))))
```



# PDDL : Représentation (1 / 2)

- Problème de la **qualification** : en pratique, on ne peut pas lister toutes les pré-conditions dans un opérateur.
- Problème de la **ramification** : en pratique, on ne peut pas lister toutes les post-conditions dans un opérateur.
- Exemple : opérateur « Démarrer une voiture »
  - Pré-conditions : clé-dans-le-barillet ET clé-tournée
  - Post-conditions : moteur-tourne

# PDDL : Représentation (2 / 2)

- Résolution du **problème du cadre** : lors de l'exécution d'un opérateur, ce qui n'est pas explicitement changé par une post-condition est considéré comme inchangé.
- Hypothèse du **monde clos** : dans un état, un terme qui n'est pas mentionné est considéré comme étant *faux*.
  - Par opposition à l'hypothèse du **monde ouvert** (ontologies) : dans un état, ce qui n'est pas mentionné est considéré comme étant *inconnu*.

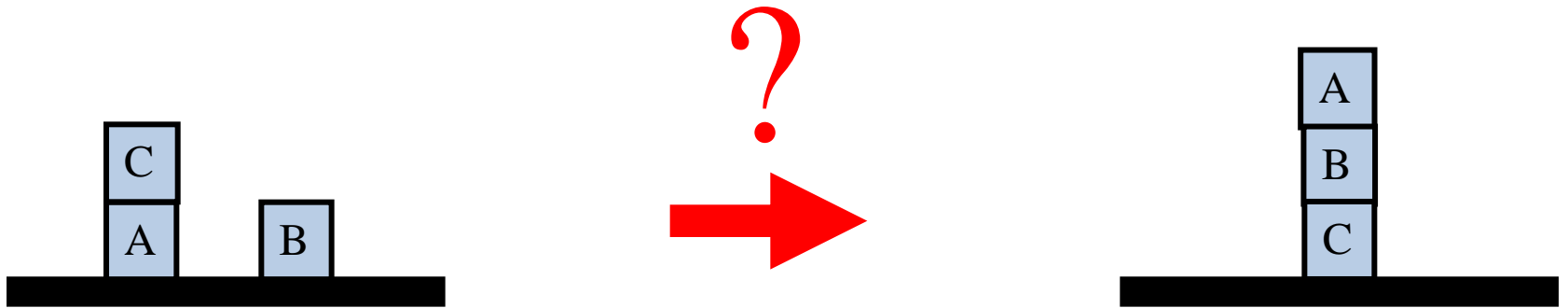
# Expressivité : STRIPS et ADL

**STRIPS** = Stanford Research Institute Problem Solver

**ADL** = Action Description Language

Littéraux positifs seulement dans les états : Riche $\wedge$ Célèbre	Littéraux positifs et négatifs dans les états : $\neg$ Riche $\wedge$ $\neg$ Célèbre
Hypothèse du monde fermé	Hypothèse du monde ouvert.
Effet $P \wedge \neg Q$ signifie ajoute P et détruit Q	Effet $P \wedge \neg Q$ signifie ajoute P et $\neg Q$ et détruit $\neg P$ et Q
Propositions seulement dans les buts Riche $\wedge$ Célèbre	Variables quantifiées dans les buts : $\exists \mathbf{x}, AT(\text{Avion1}, \mathbf{x}) \wedge AT(\text{Avion2}, \mathbf{x})$
Les buts sont des conjonctions : Riche $\wedge$ Célèbre	Les buts sont des conjonctions et disjonctions : $\neg$ Pauvre $\wedge$ (Riche $\vee$ intelligent)
Les effets sont des conjonctions	Effets conditionnels : QUAND P : E
Pas d'égalité	Egalité ( $x = y$ ) prédéfinie
Pas de types	Les variables peuvent être typées

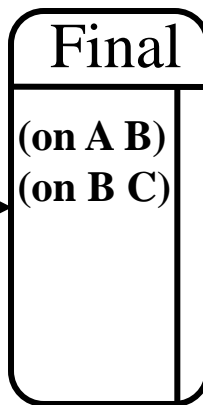
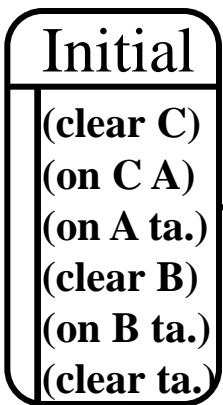
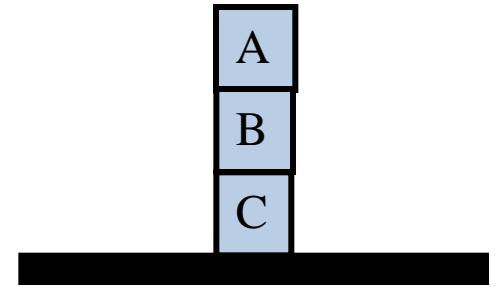
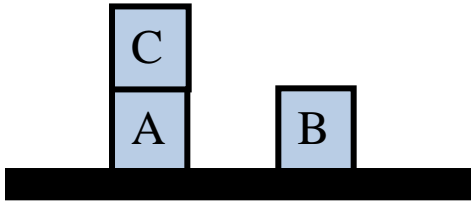
# L'anomalie de Gerald Jay Sussman (1/16)



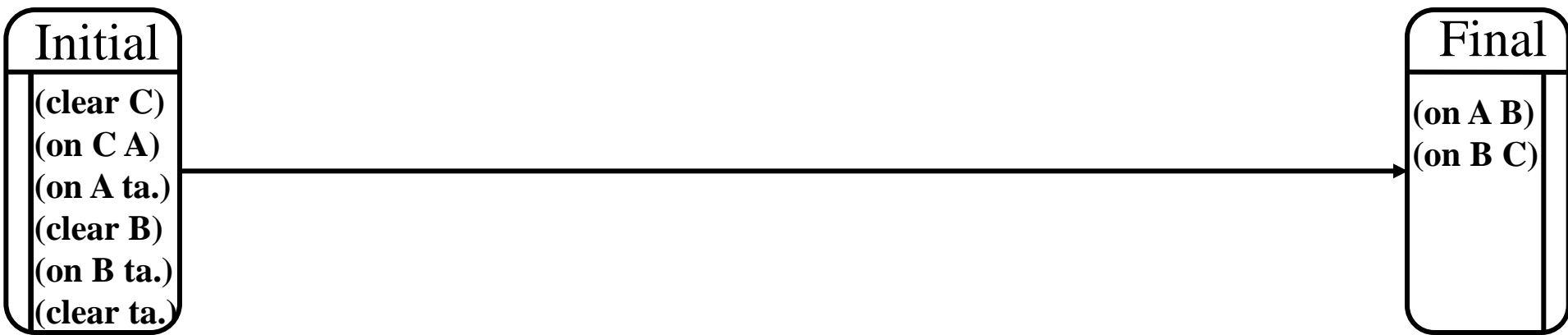
avec :

puton ?b ?u ?t	
(clear ?b)	(not (on ?b ?u))
(on ?b ?u)	(=> (<> ?u table))
(clear ?t)	(clear ?u)
	(on ?b ?t)
	(=> (<> ?t table))
	(not (clear ?t))

# L'anomalie de Gerald Jay Sussman (2/16)

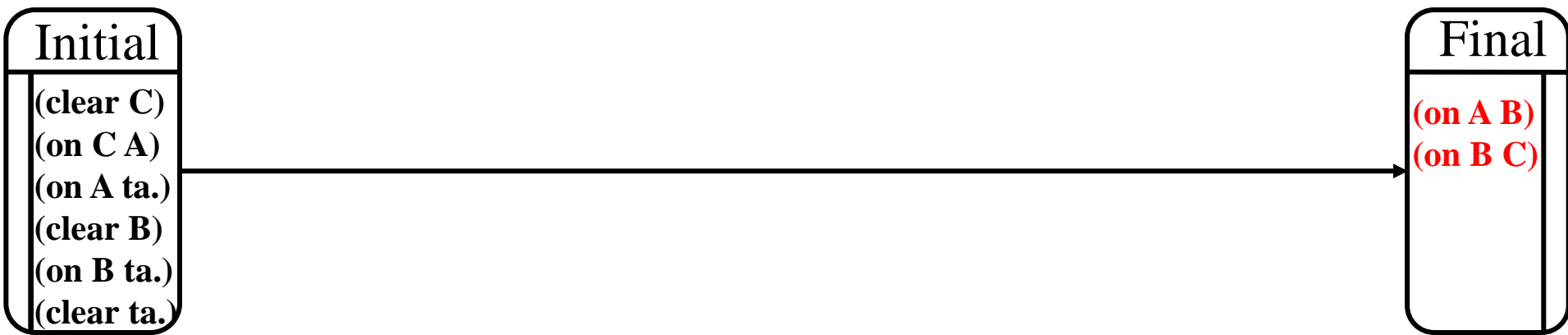


# L'anomalie de Gerald Jay Sussman (2/16)

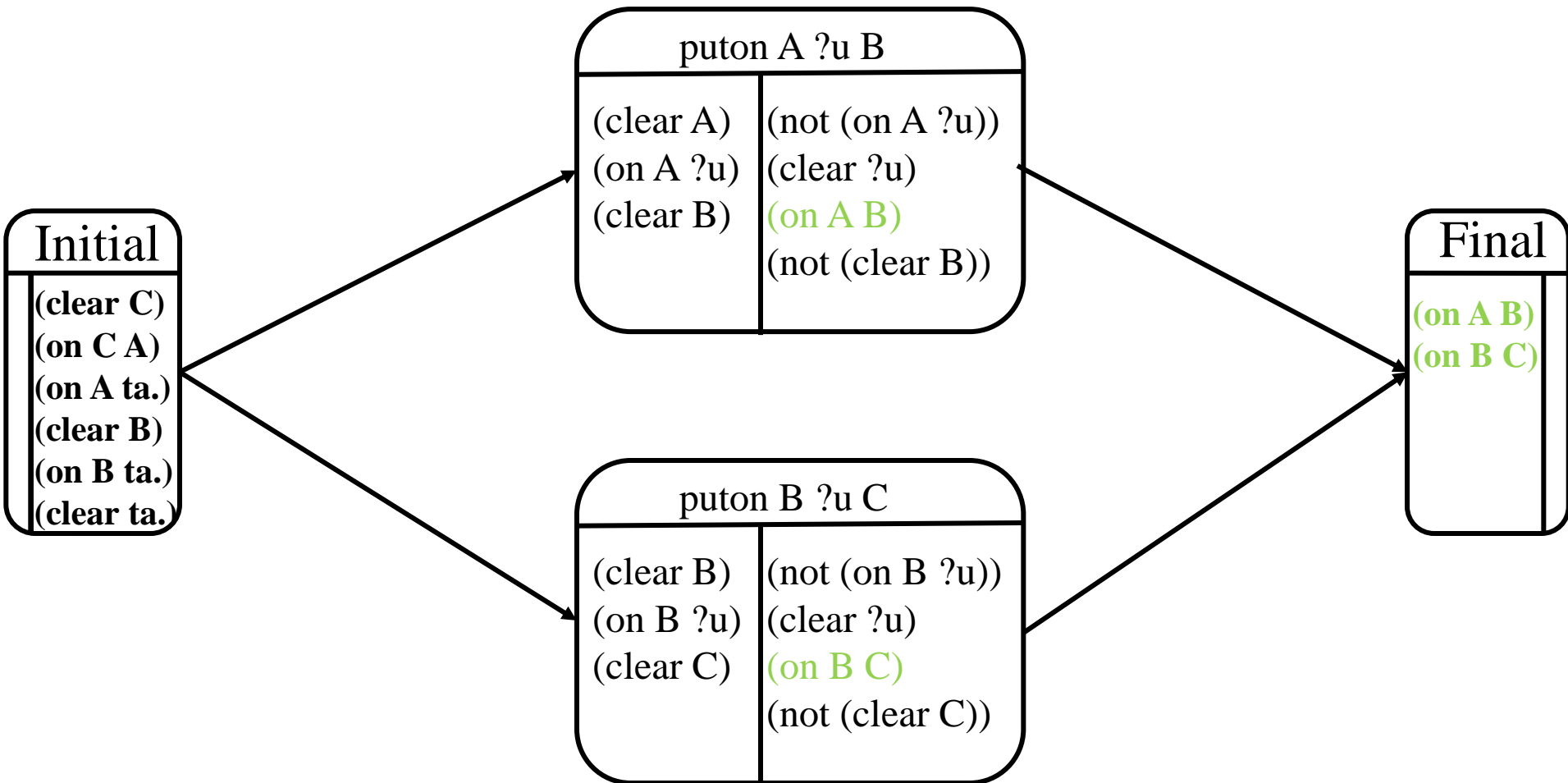




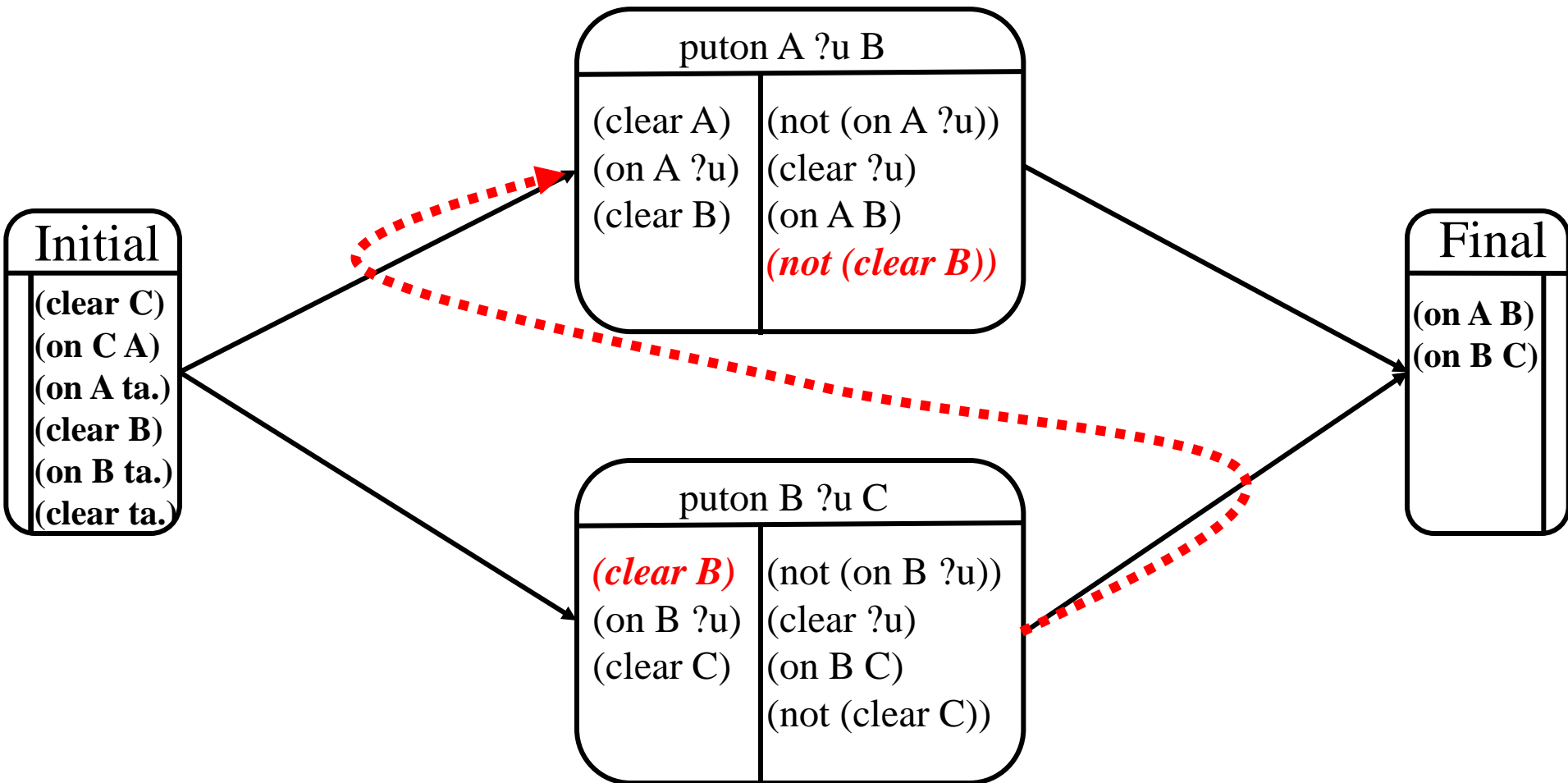
# L'anomalie de Gerald Jay Sussman (3/16)



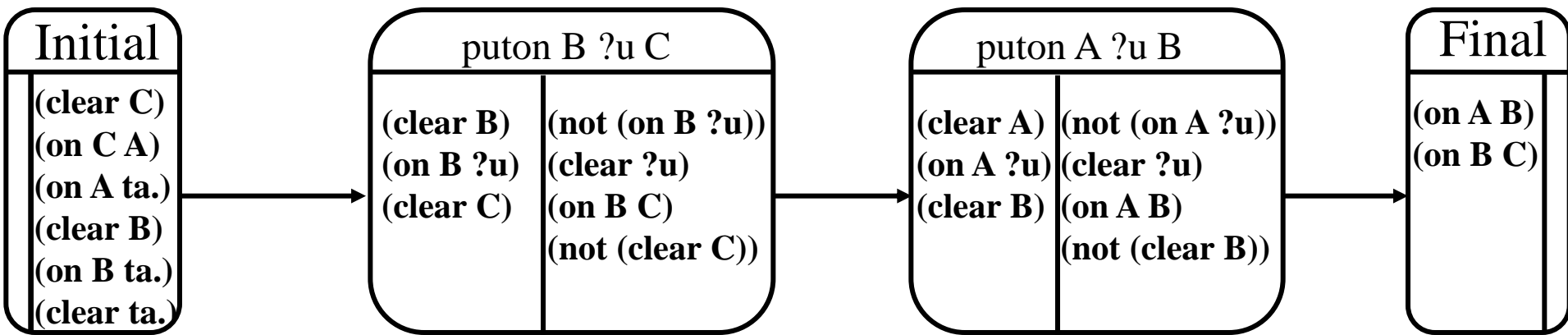
# L'anomalie de Gerald Jay Sussman (4/16)



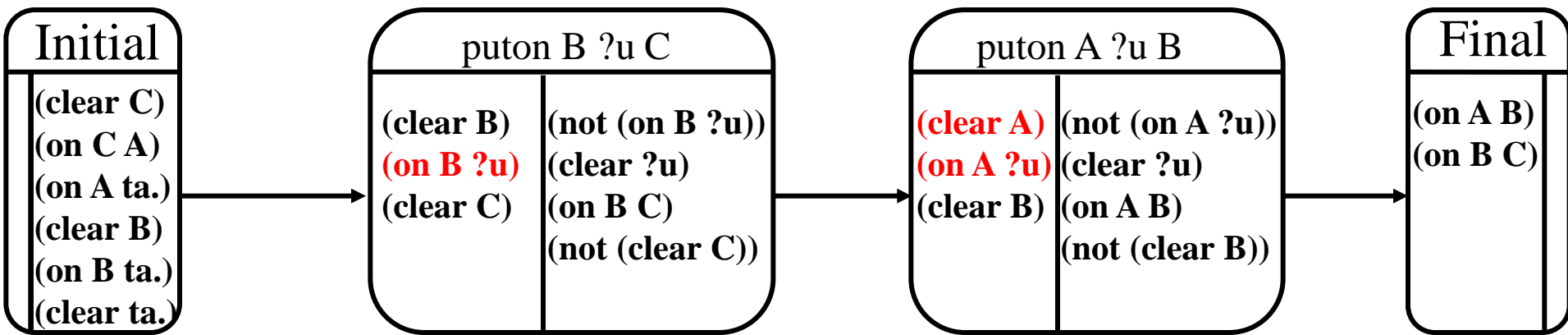
# L'anomalie de Gerald Jay Sussman (5/16)



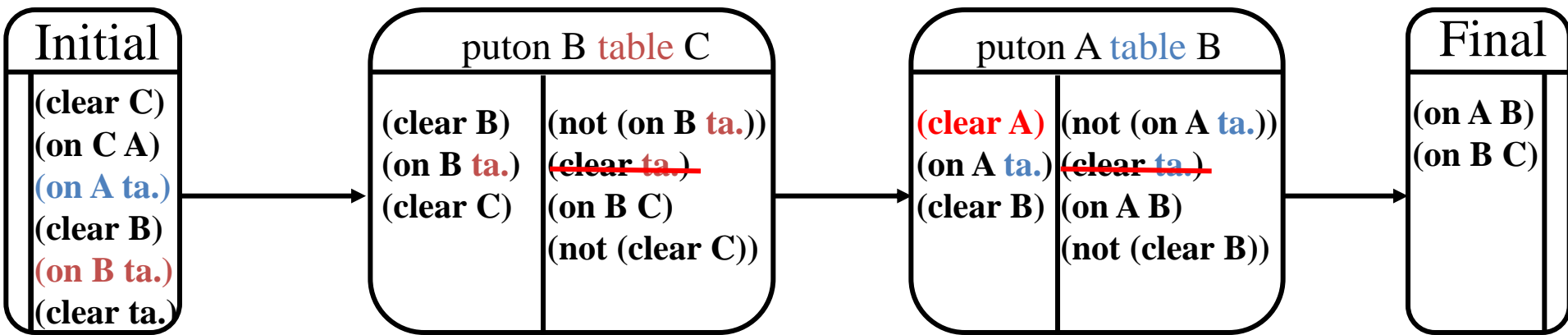
# L'anomalie de Gerald Jay Sussman (6/16)



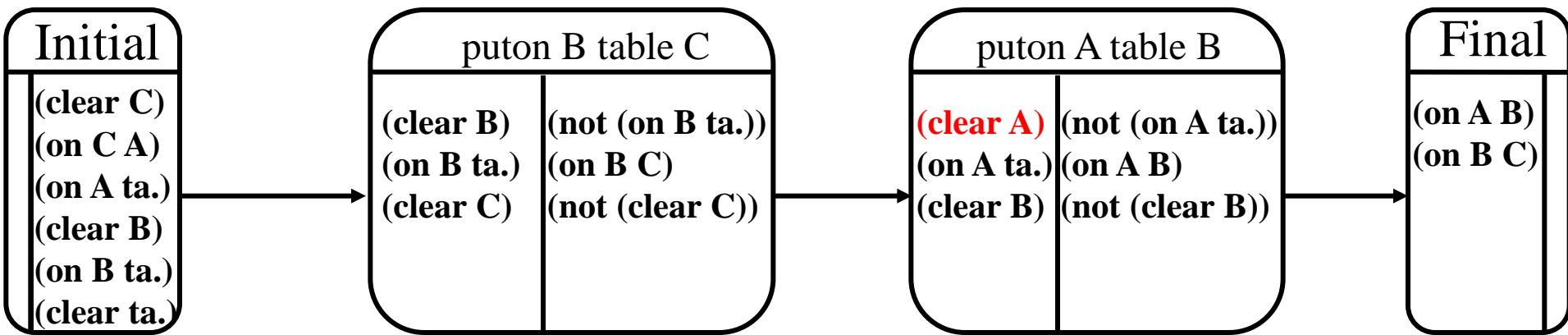
# L'anomalie de Gerald Jay Sussman (7/16)



# L'anomalie de Gerald Jay Sussman (8/16)

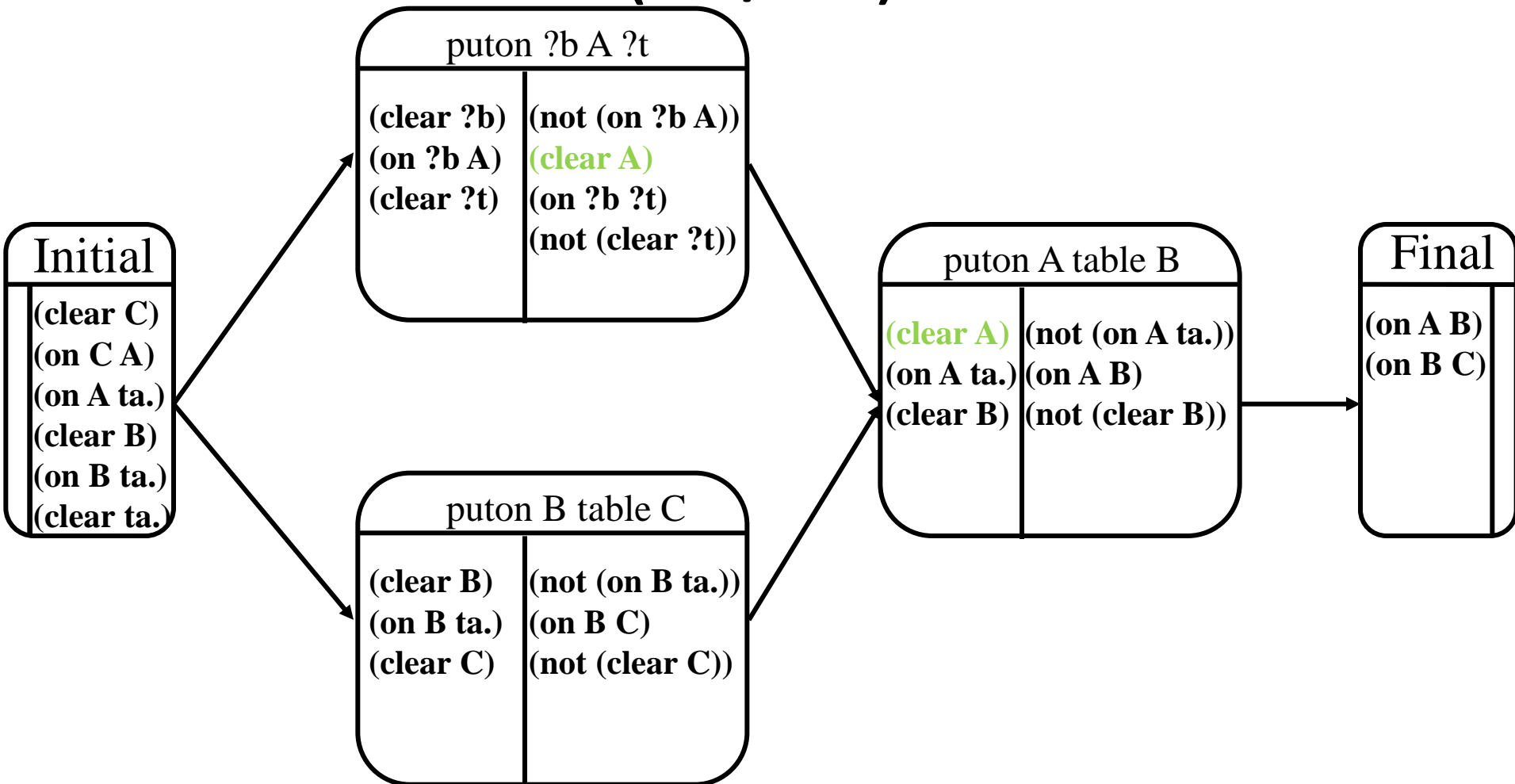


# L'anomalie de Gerald Jay Sussman (9/16)



# L'anomalie de Gerald Jay Sussman

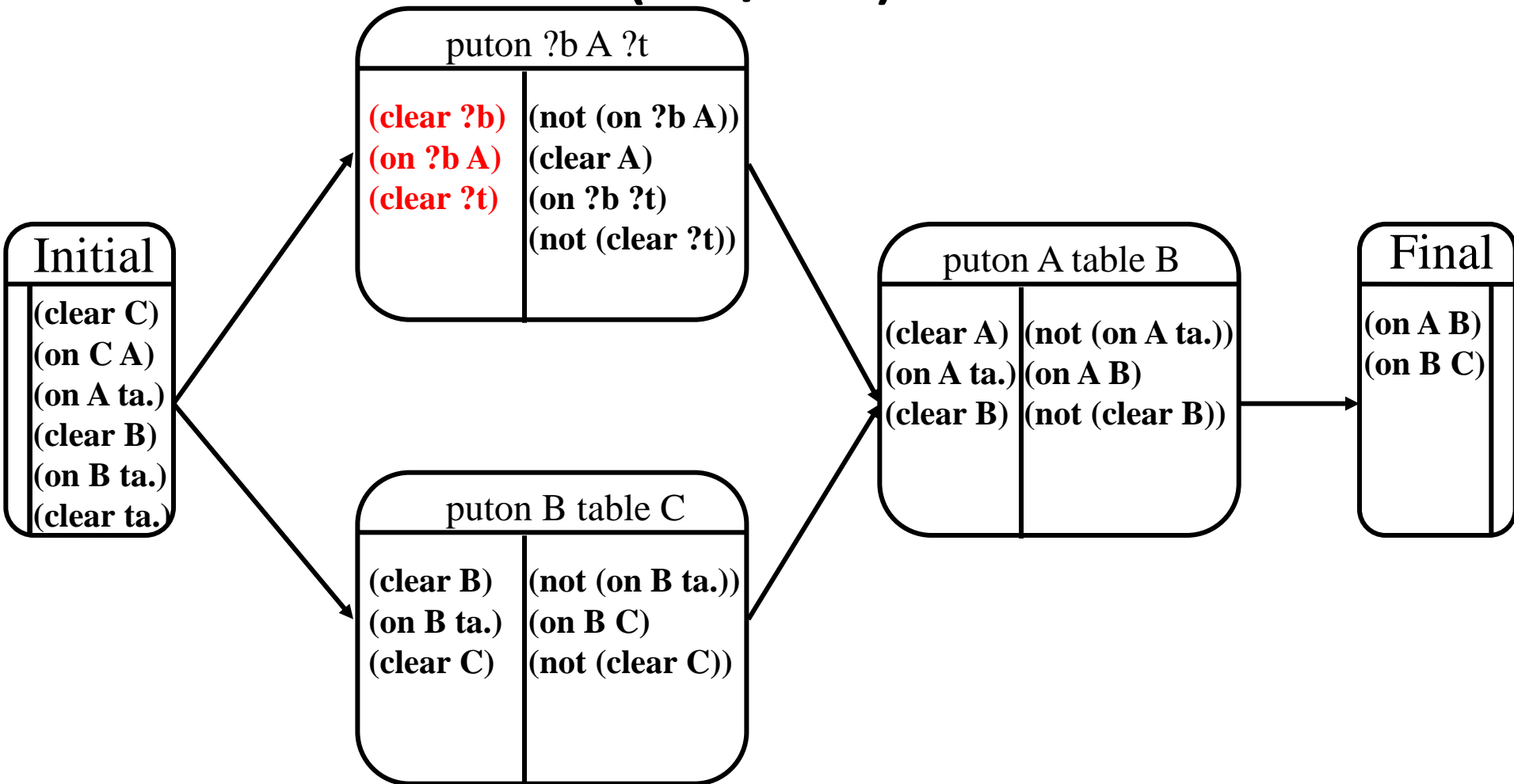
## (10/16)





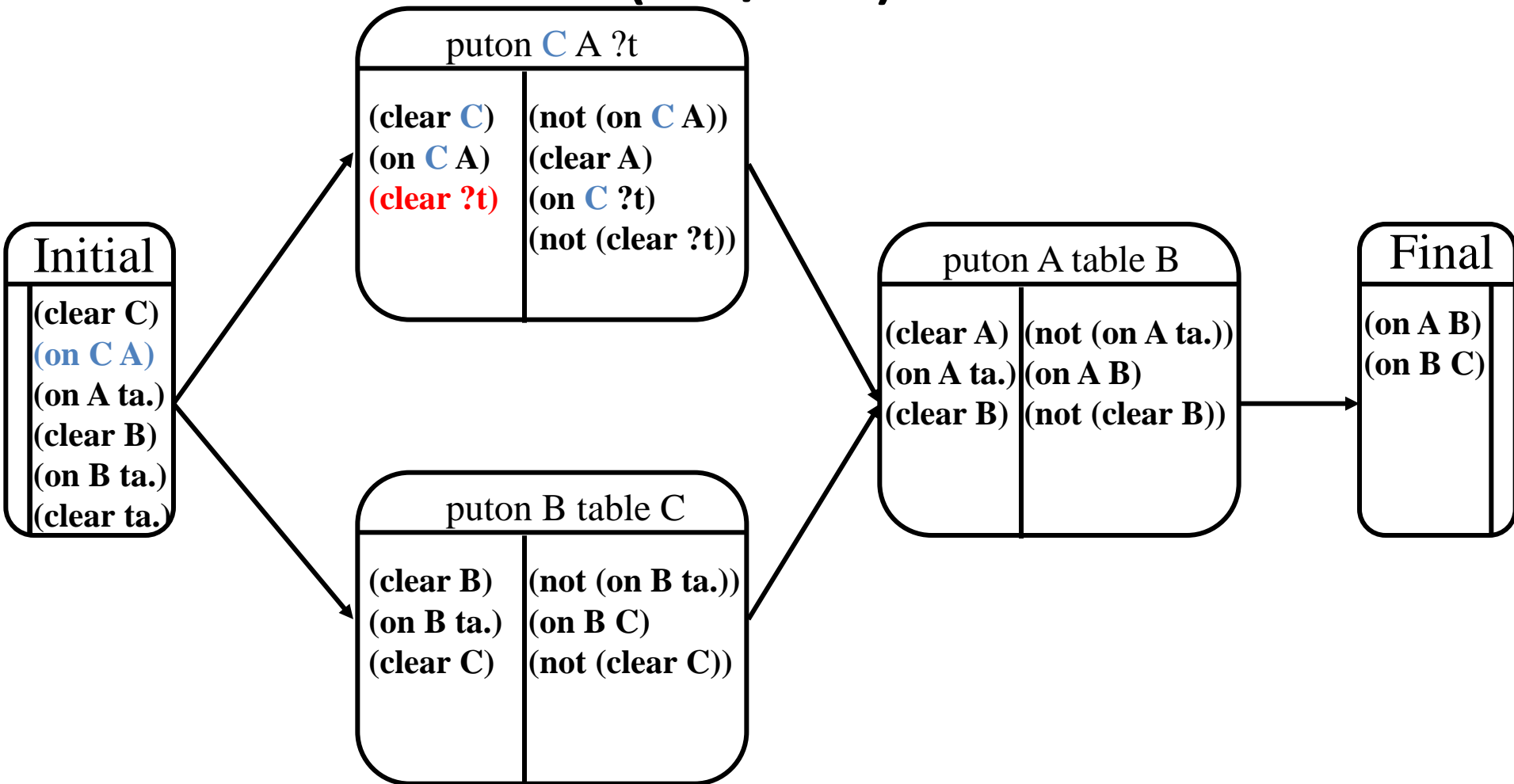
# L'anomalie de Gerald Jay Sussman

(11/16)



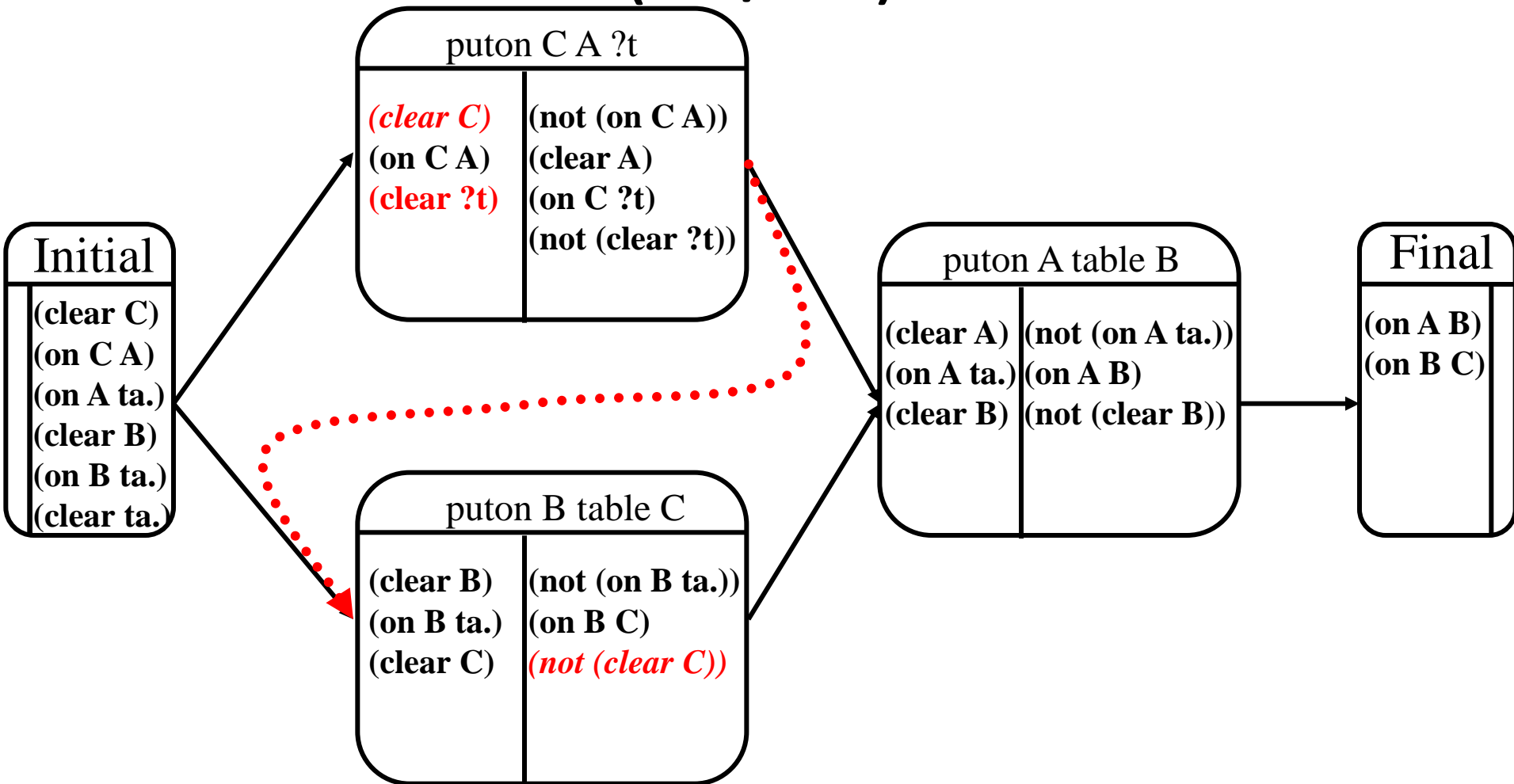
# L'anomalie de Gerald Jay Sussman

## (12/16)

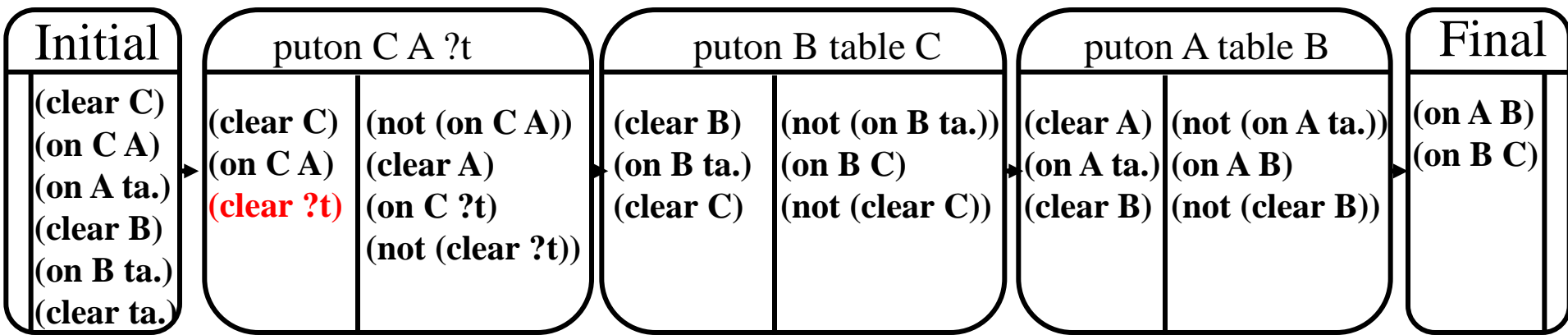


# L'anomalie de Gerald Jay Sussman

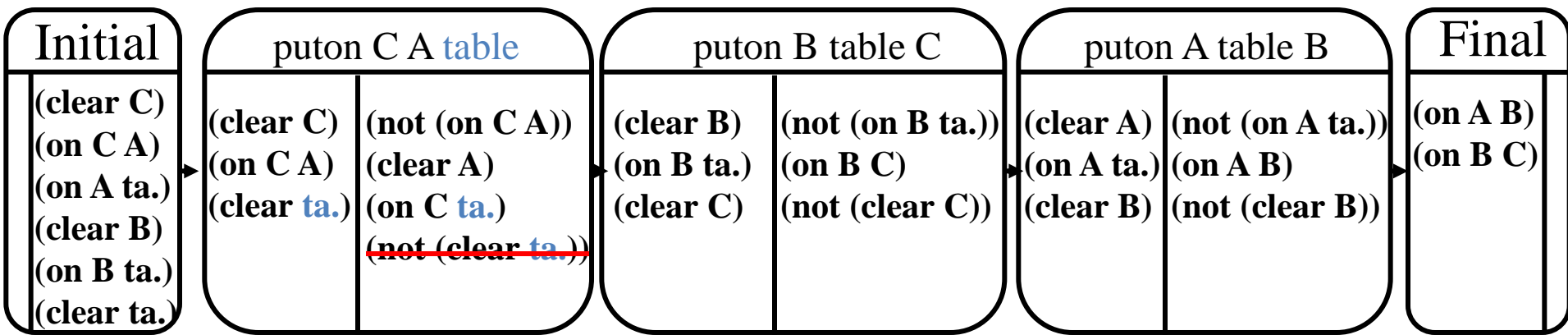
## (13/16)



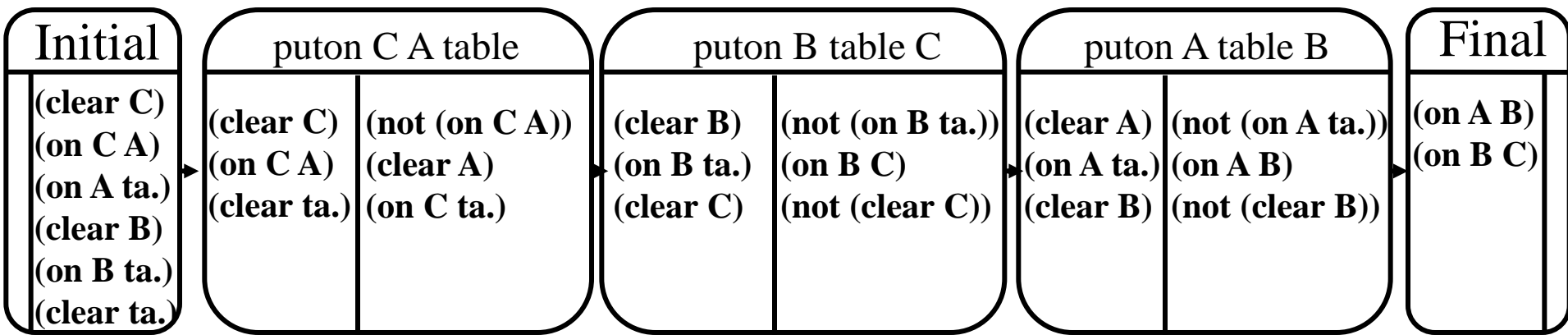
# L'anomalie de Gerald Jay Sussman (14/16)



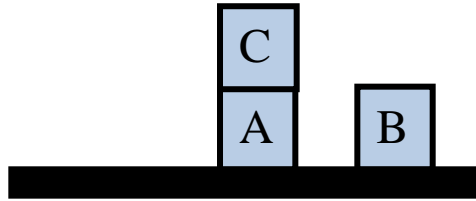
# L'anomalie de Gerald Jay Sussman (15/16)



# L'anomalie de Gerald Jay Sussman (16/16)



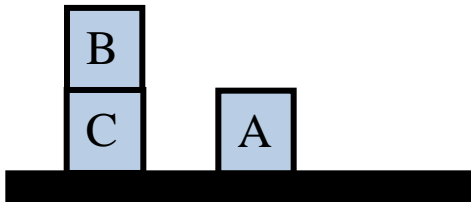
# L'anomalie de Gerald Jay Sussman : solution



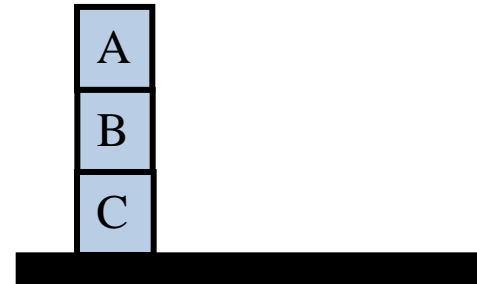
(1)



(2)



(3)



(4)

# Références

- [Weld 94] Daniel Weld, *An Introduction to Least Commitment Planning*, A. I. Magazine, 15(4), pages 27-61, Winter 1994.
- [Russel 2010] Stuart Russell, Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010, 3rd edition. Chapitre 11.
- [Ghallab et al. 04] Malik Ghallab, Dana Nau, Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, San Mateo, CA, May 04, 635 pages.
- PDDL 3.1. <http://ipc.informatik.uni-freiburg.de/PddlExtension>
- Conférences :
  - International Conference on Automated Planning and Scheduling (ICAPS). <http://www.icaps.org>
  - International Joint Conference on A.I. (IJCAI). <http://www.ijcai.org>
  - European Conference on A.I. (ECAI). <http://www.ecai.org>
  - National Conference on A.I. (AAAI). <http://www.aaai.org>
- Journaux :
  - A. I. Journal (AIJ). [http://www.elsevier.com/wps/find/journaldescription.cws\\_home/505601/description#description](http://www.elsevier.com/wps/find/journaldescription.cws_home/505601/description#description)
  - Journal of A.I. Research (JAIR). <http://www.jair.org/>



# Conclusion

- La planification d'actions consiste à trouver une séquence d'opérateurs instantiés (un plan) menant un état initial à des buts.
  - Difficile parce que explosion combinatoire d'ue à l'interaction entre sous buts.
- Les opérateurs peuvent être exprimés en STRIPS / ADL / PDDL.