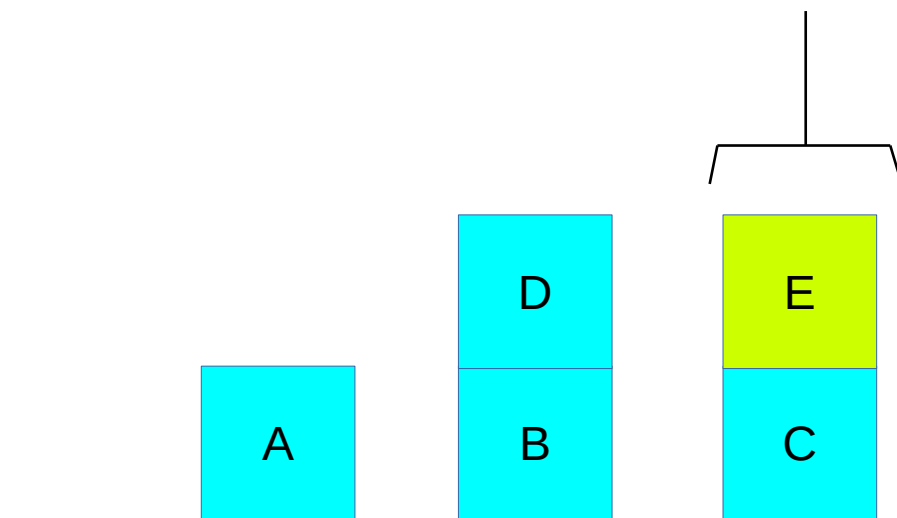
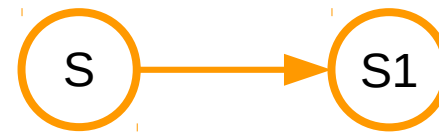


# Frame problem

- **Frame problem (problema della cornice):**  
normalmente le azioni hanno un impatto limitato: come rappresentare ciò che non viene modificato da un' azione?



Quando il braccio afferra E  
l'unica cosa che cambia è che il  
braccio non è più vuoto. Per il resto  
la situazione rimane immutata



Il sistema automatico deve poter inferire  
cosa S1 eredita "as-is" da S

# Frame problem 1: enumerare ciò che non cambia

- **Frame problem (problema della cornice):**  
normalmente le azioni hanno un impatto limitato: come rappresentare ciò che non viene modificato da un' azione?
- 1) Rappresentazione esplicita tramite **assiomi di frame:**  
$$\forall \text{params, vars, s } \text{fluent}(\text{vars}, s) \wedge \text{params} \neq \text{vars} \Rightarrow$$
$$\text{fluent}(\text{vars}, \text{Result}(\text{Action}(\text{params}), s))$$

Per ogni azione viene definito un assioma di questo tipo per ogni fluente

# Frame problem 1: enumerare ciò che non cambia

- Rappresentazione esplicita tramite **assiomi di frame**:  
$$\forall \text{params, vars, s } \text{fluent}(\text{vars}, s) \wedge \text{params} \neq \text{vars} \Rightarrow$$
$$\text{fluent}(\text{vars}, \text{Result}(\text{Action}(\text{params}), s))$$

Esempio: azione: Move, vars = {x, y}, params = {z, w}

$$\forall \text{params, vars, s } \text{on}(\text{x}, \text{y}, \text{s}) \wedge \text{x} \neq \text{z} \Rightarrow$$
$$\text{on}(\text{x}, \text{y}, \text{Result}(\text{Move}(\text{z}, \text{w}), \text{s}))$$
$$\forall \text{params, vars, s } \text{clear}(\text{x}, \text{s}) \wedge \text{x} \neq \text{w} \Rightarrow$$
$$\text{clear}(\text{x}, \text{Result}(\text{Move}(\text{z}, \text{w}), \text{s}))$$

# Frame problem

- **Problema:**  
occorre introdurre frame axioms per **tutti** i fluenti che non sono modificati da ciascuna azione!
- **Esempio**  
Se i blocchi ora possono essere dipinti di un colore, rivestiti di un materiale, ecc. occorrerà specificare frame axiom per ciascuna di queste proprietà:

$\forall \text{params, vars, s colore}(x, y, s) \wedge x \neq z \Rightarrow$   
 $\text{colore}(x, y, \text{Result}(\text{Move}(z, w), s))$   
 $\forall \text{params, vars, s materiale}(x, y, s) \wedge x \neq z \Rightarrow$   
 $\text{materiale}(x, y, \text{Result}(\text{Move}(z, w), s))$

...

# Frame problem 2: evitare l'enumerazione

- introduciamo un modo per dire al sistema inferenziale che ciò che non è specificamente espresso come effetto è inteso rimanere immutato

# Frame problem

- **ASSIOMA DI STATO SUCCESSORE:**

Azione applicabile  $\Rightarrow$  (fluente vero nella situazione risultante  
 $\Leftrightarrow$  (l' azione lo rende vero  $\vee$   
era vero e l' azione non l' ha reso falso))

- Questo assioma esprime il modo in cui le **situazioni si evolvono** l' una dall' altra a seguito dell' esecuzione delle azioni. In particolare dice quali parti di una situazione sono ereditati dalla precedente

# Unique action names

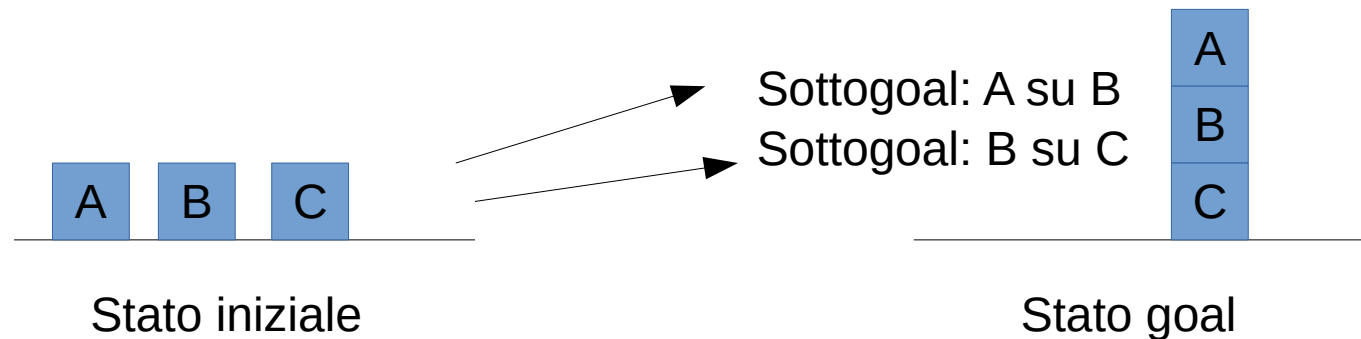
- Il situation calculus è completato da assiomi che catturano:
  - che azioni con nomi diversi sono diverse:  
 $A_i(x, \dots) \neq A_j(y, \dots)$
  - E per ogni nome di azione, che due usi di quel nome sono identici se e solo se gli argomenti sono identici:  
 $A(x_1, \dots, x_n) = A(y_1, \dots, y_n) \Rightarrow x_1=y_1 \wedge \dots \wedge x_n=y_n$

- Mondo dei blocchi (documentazione a parte)



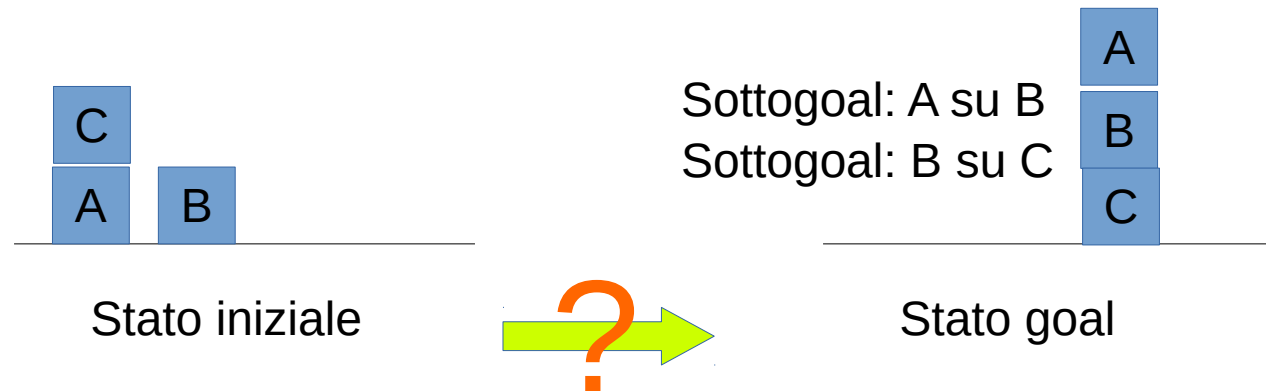
# Perseguire goal complessi

- Un approccio tipico alla pianificazione consiste nello scomporre l'obiettivo in sottoobiettivi e conseguire questi ultimi uno per volta
- **Esempio:**



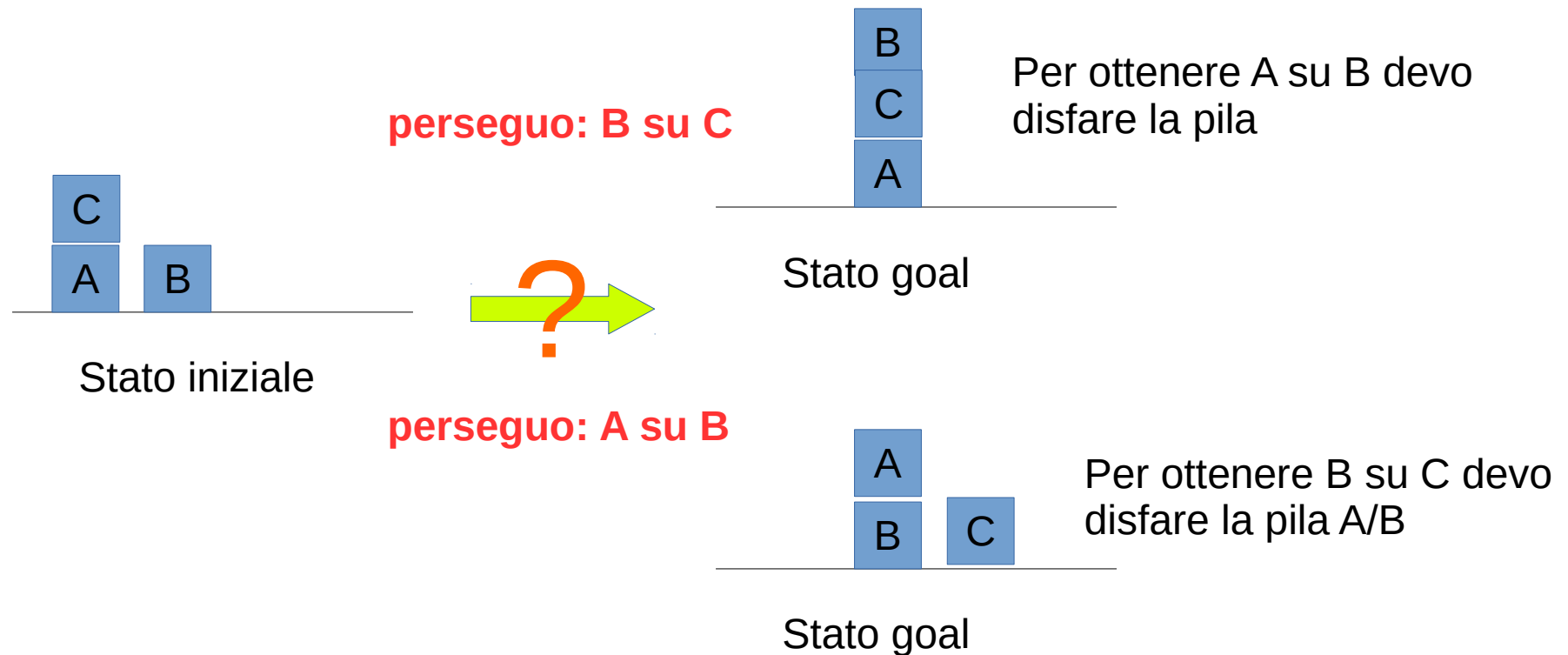
# Anomalia di Sussman

- Non sempre il perseguimento degli obiettivi è sequenzializzabile, esempio:



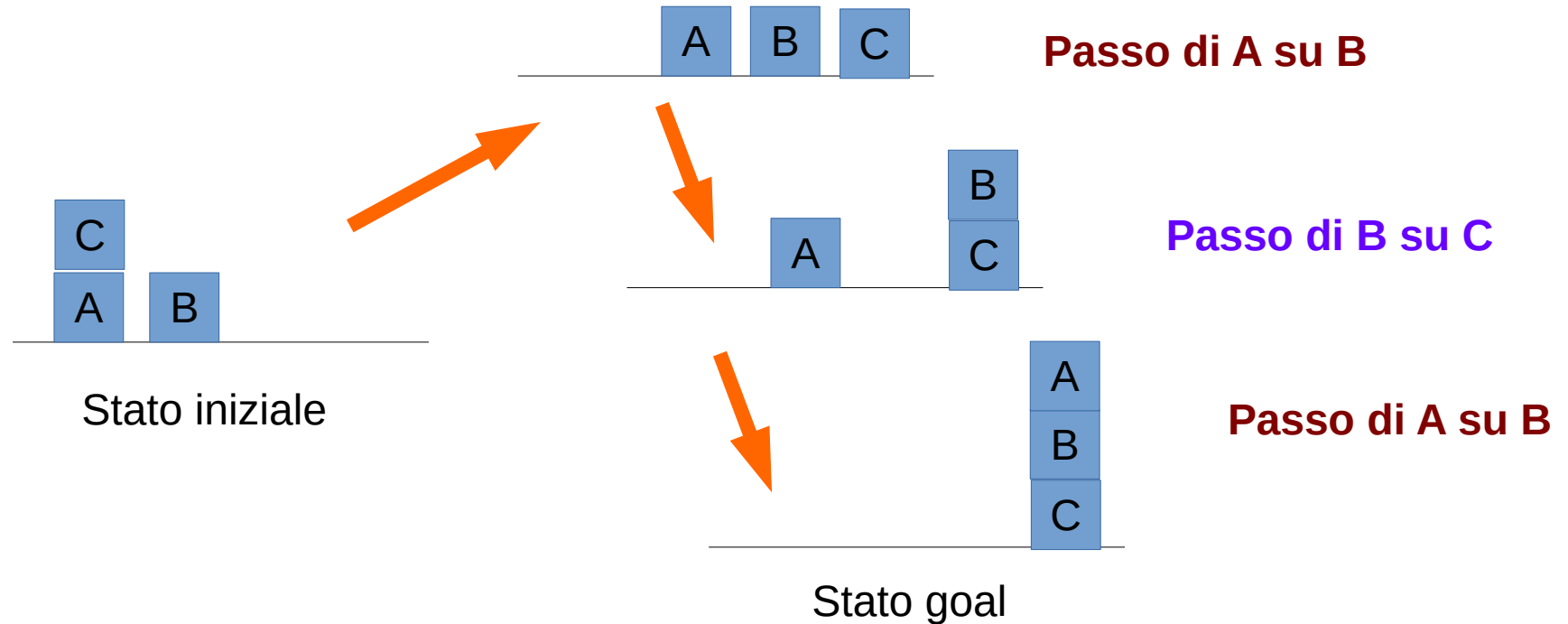
# Anomalia di Sussman

- Non sempre il perseguimento degli obiettivi è sequenzializzabile, anzi alcune volte il perseguimento di un sottogoal può disfare passi effettuati per raggiungerne un altro. Esempio:



# Interleaving dei passi

- Per risolvere efficientemente l' esempio occorre fare interleaving dei passi di soluzione. Esempio:



# Considerazioni

- Il situation calculus permette di usare FOL per problemi di pianificazione
- È stato fondamentale per definire il problema di pianificazione
- Nella pratica non è molto usato perché non esistono euristiche efficienti che guidino la ricerca della soluzione
- La pianificazione (planning) è parte del corso IA e Laboratorio della laurea magistrale in Intelligenza Artificiale e Sistemi Informatici