

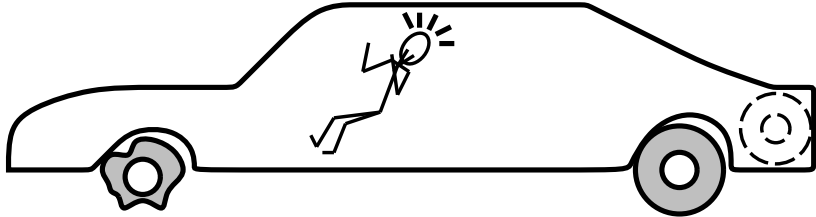
# PIANIFICAZIONE II

CORSO DI SISTEMI INTELLIGENTI, CAPITOLO 12

# Outline

- ◇ Il mondo reale
- ◇ Pianificazione condizionale
- ◇ Monitoraggio e ripianificazione

# Il mondo reale



**START**

*~Flat(Spare) Intact(Spare) Off(Spare)  
On(Tire1) Flat(Tire1)*

*On(x) ~Flat(x)*

**FINISH**

*On(x)*  
**Remove(x)**

*Off(x) ClearHub*

*Off(x) ClearHub*  
**Puton(x)**

*On(x) ~ClearHub*

*Intact(x) Flat(x)*  
**Inflate(x)**

*~Flat(x)*

# Problemi!!

## *Informazione incompleta*

Precondizioni sconosciute, e.g., *Intact(Spare)*?

Effetti disgiuntivi, e.g., *Inflate(x)* causa

$Inflated(x) \vee SlowHiss(x) \vee Burst(x) \vee BrokenPump \vee \dots$

## *Informazione non corretta*

Stato corrente non corretto, e.g., ruota di scorta NON intatta

Precondizioni negli operatori mancanti/scorretti

## Problema della Qualifica:

non si finisce mai di elencare tutte le precondizioni necessarie e presenza di possibili esiti condizionali per le azioni

# Soluzioni

## Conformant o pianificazione senza sensori

progettare un piano trascurando lo stato o l'esito delle azioni

*Tali piani possono non esistere*

## Pianificazione condizionale

Piano per ottenere l'informazione (azioni di osservazione)

Sottopiano per ogni contingenza, p.e.,

[*Check(Tire1)*, **if** *Intact(Tire1)* **then** *Inflate(Tire1)* **else** *CallAAA*

*Costoso perchè si pianifica considerando molti casi poco probabili*

## Monitoraggio/Ripianificazione

Assume stati normali, esiti

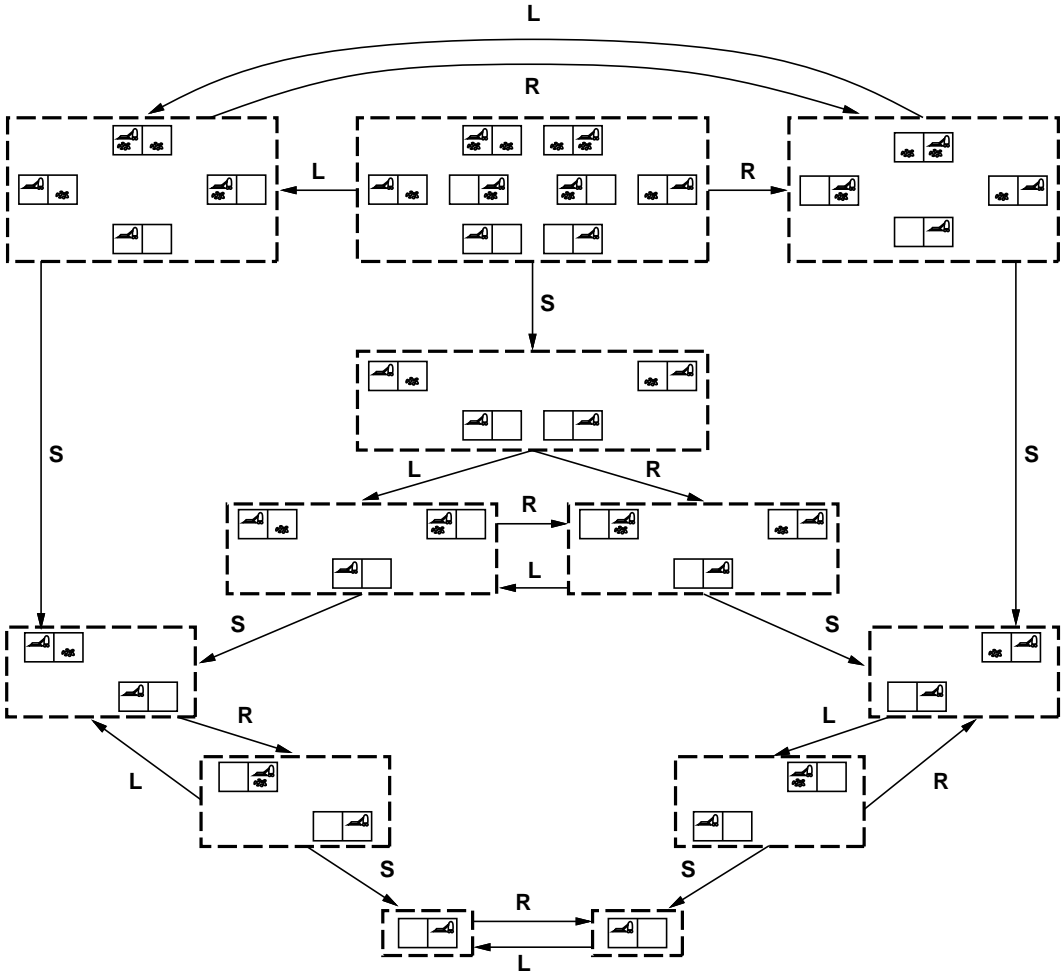
Controlla il progresso del piano *durante l'esecuzione*, ripianificare se necessario

*Esiti non previsti possono condurre a fallimenti (p.e., nessuna AAA card)*

(C'è bisogno di una combinazione; pianificare per eventi probabili/sicuri, trattare gli altri eventi quando accadono)

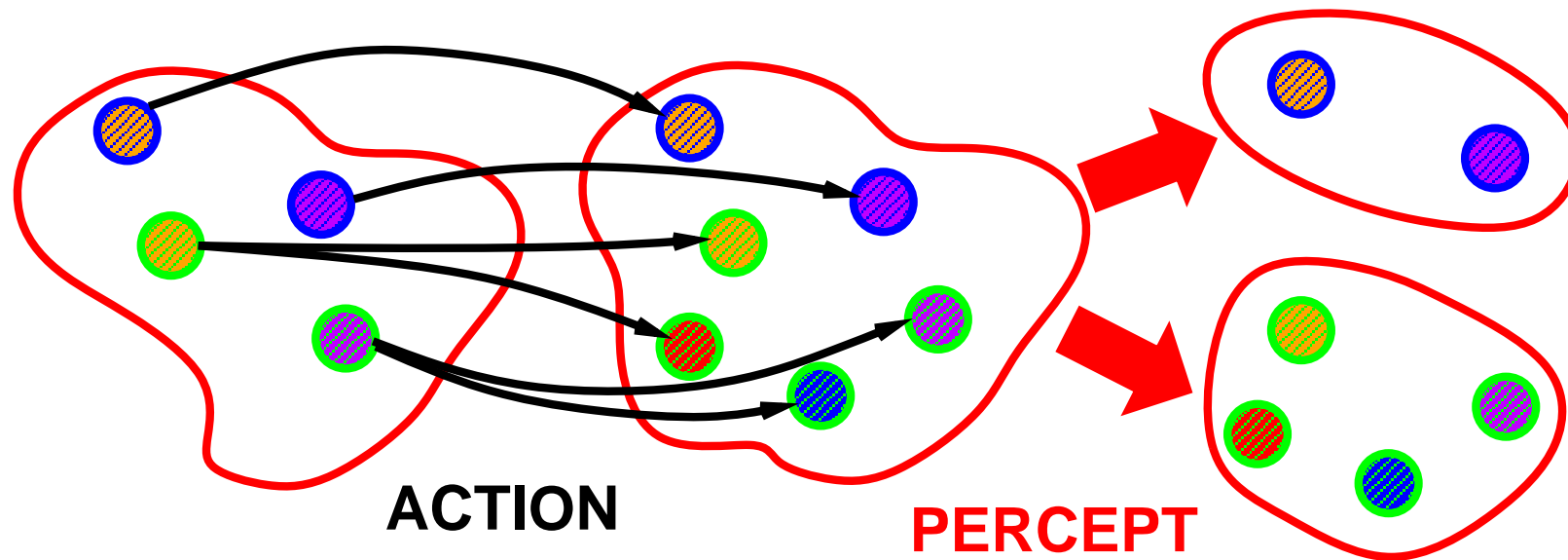
# Conformant planning

Cercare nello spazio di **stati di credenza** (insiemi di possibili stati effettivi)



# Pianificazione Condizionale

Se il mondo è nondeterministico o parzialmente osservabile  
la percezione di solito *fornisce informazioni*,  
cioè, *suddivide* lo stato di credenza



## Pianificazione Condizionale

Piani condizionali controllano (ogni conseguenza della KB +) percezioni

[... , **if**  $C$  **then**  $Plan_A$  **else**  $Plan_B$ , ...]

Esecuzione: controlla  $C$  rispetto alla KB corrente, esegue “then” o “else”

Necessita di *qualche* piano per *ogni* possibile percezione

(Cf. giochi: *qualche* risposta per *ogni* mossa dell'opponente)

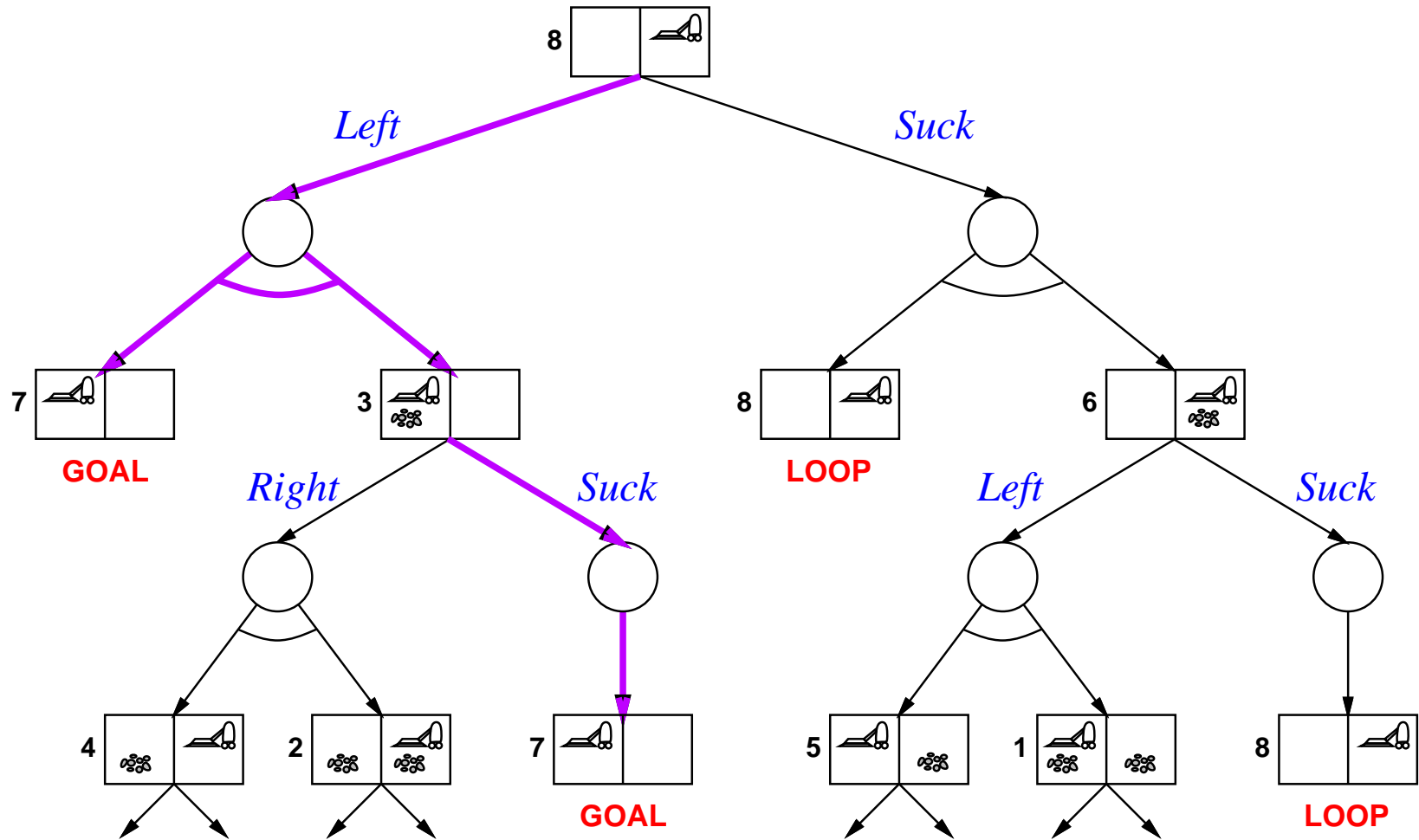
(Cf. backward chaining: *qualche* regola tale che *ogni* premessa è soddisfatta)

Alberi di ricerca AND–OR (molto simile all'algoritmo backward chaining)



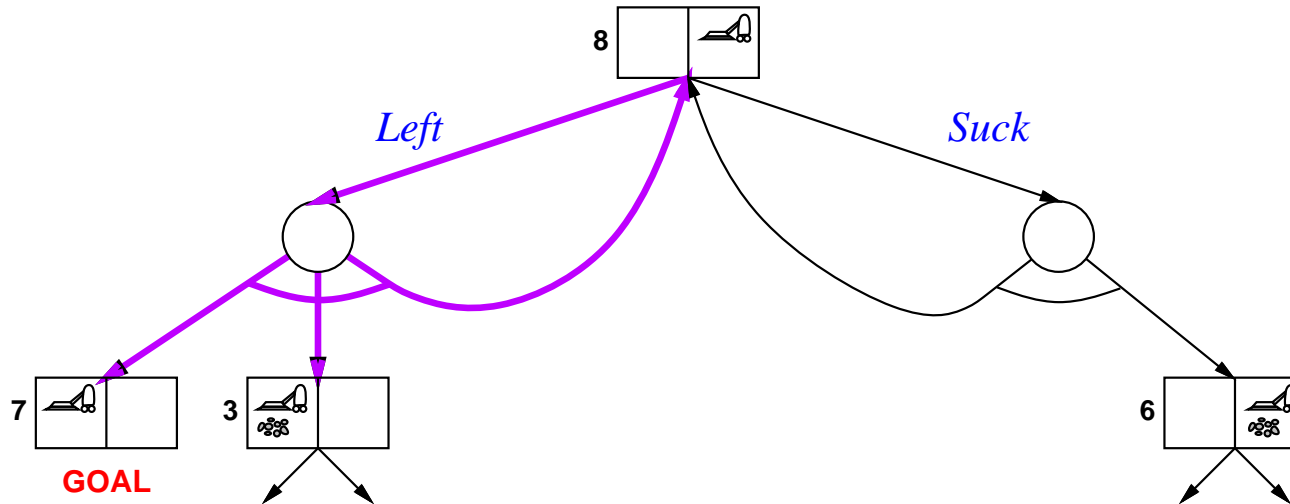
# Esempio

Doppio Murphy: aspirare o arrivare può sporcare un quadrato pulito



# Esempio

Triplo Murphy: qualche volta si resta fermi invece di muoversi



$[L_1 : \textit{Left}, \text{if } \textit{AtR} \text{ then } L_1 \text{ else } [\text{if } \textit{CleanL} \text{ then } [] \text{ else } \textit{Suck}]]$

or  $[\text{while } \textit{AtR} \text{ do } [\textit{Left}], \text{if } \textit{CleanL} \text{ then } [] \text{ else } \textit{Suck}]$

“Ciclo infinito” ma prima o poi si esce, a meno di azione che fallisce sempre

## Monitoraggio della esecuzione

“Fallimento” = precondizioni del *resto del piano* non soddisfatte

Precondizioni del resto del piano

= tutte le precondizioni dei passi rimanenti non raggiunte dai passi rimanenti

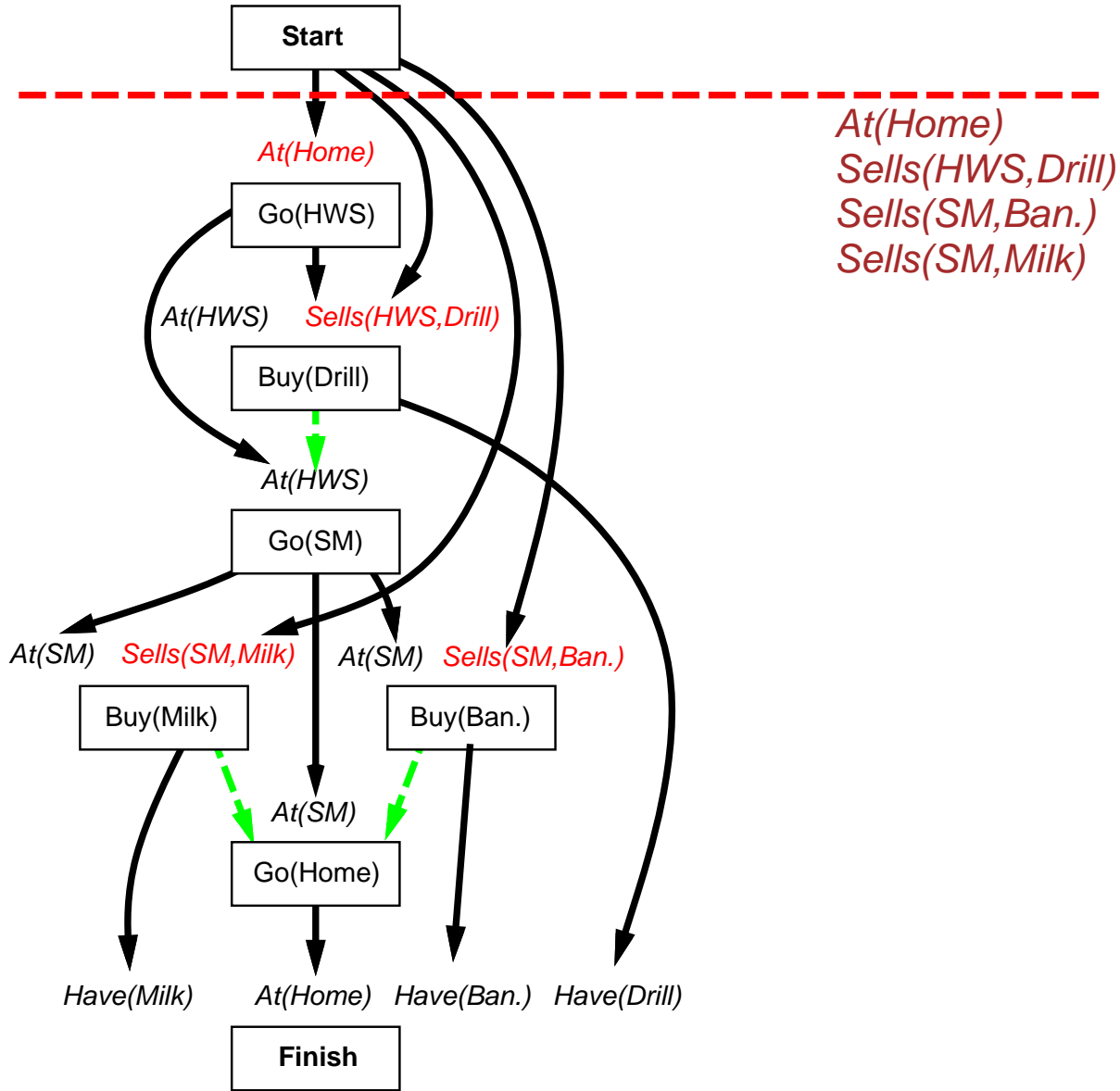
In caso di fallimento, riprendere POP per raggiungere condizioni aperte dallo stato corrente

IPEM (Integrated Planning, Execution, and Monitoring):

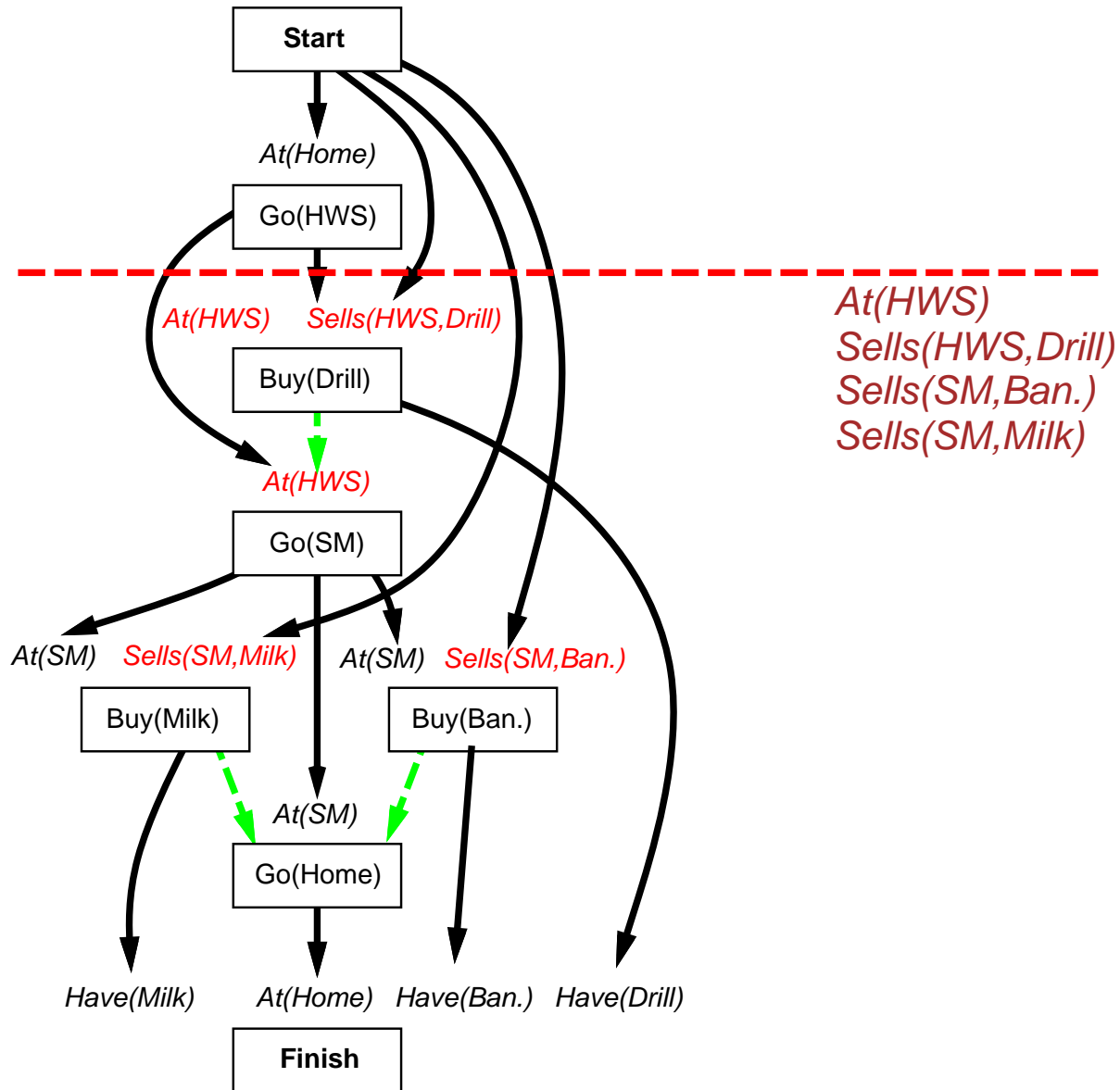
aggiorna continuamente *Start* in modo da rappresentare la situazione corrente

link dalle azioni rimpiazzati da link che partono da *Start* quando eseguite

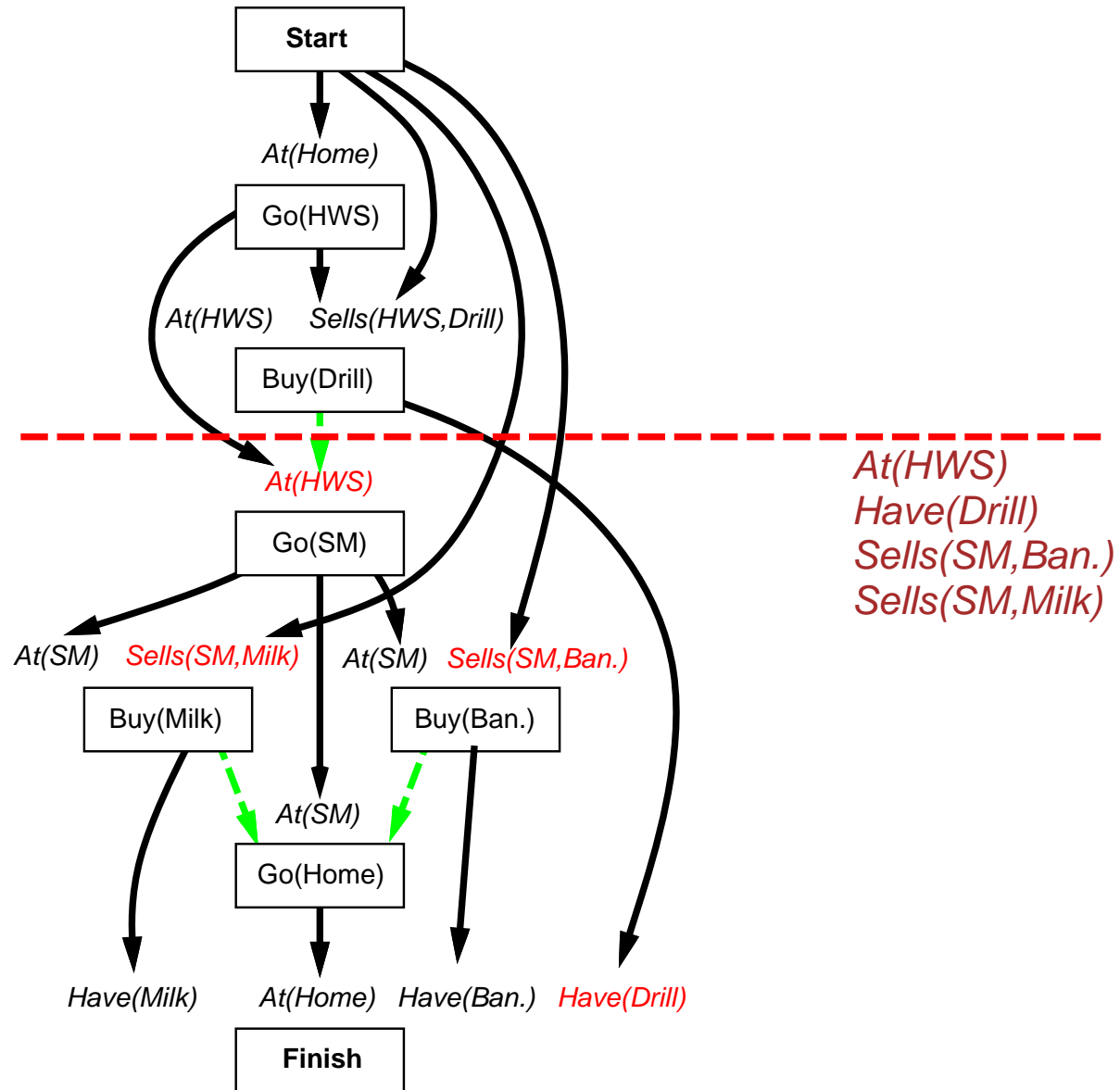
# Esempio



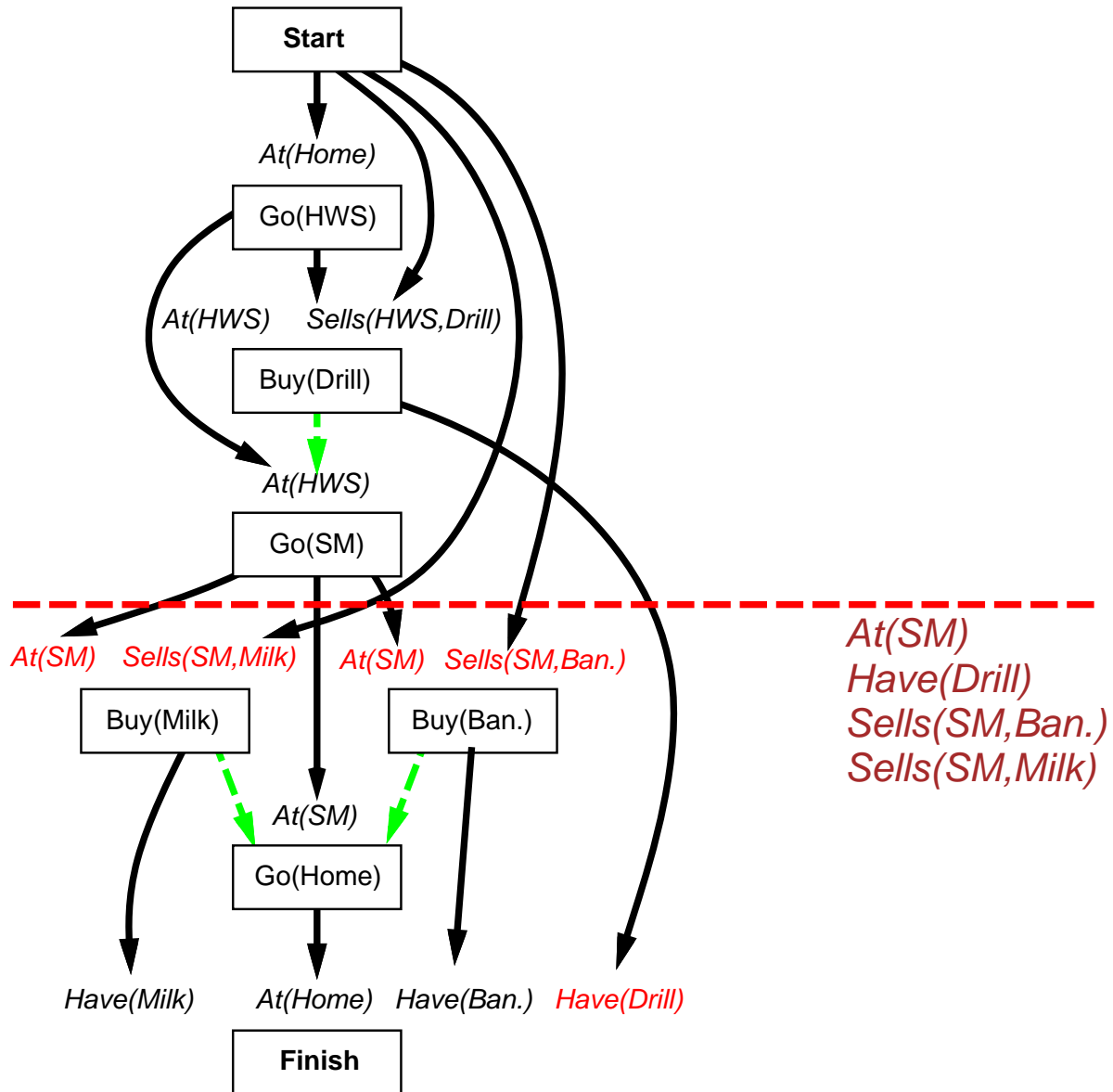
# Esempio



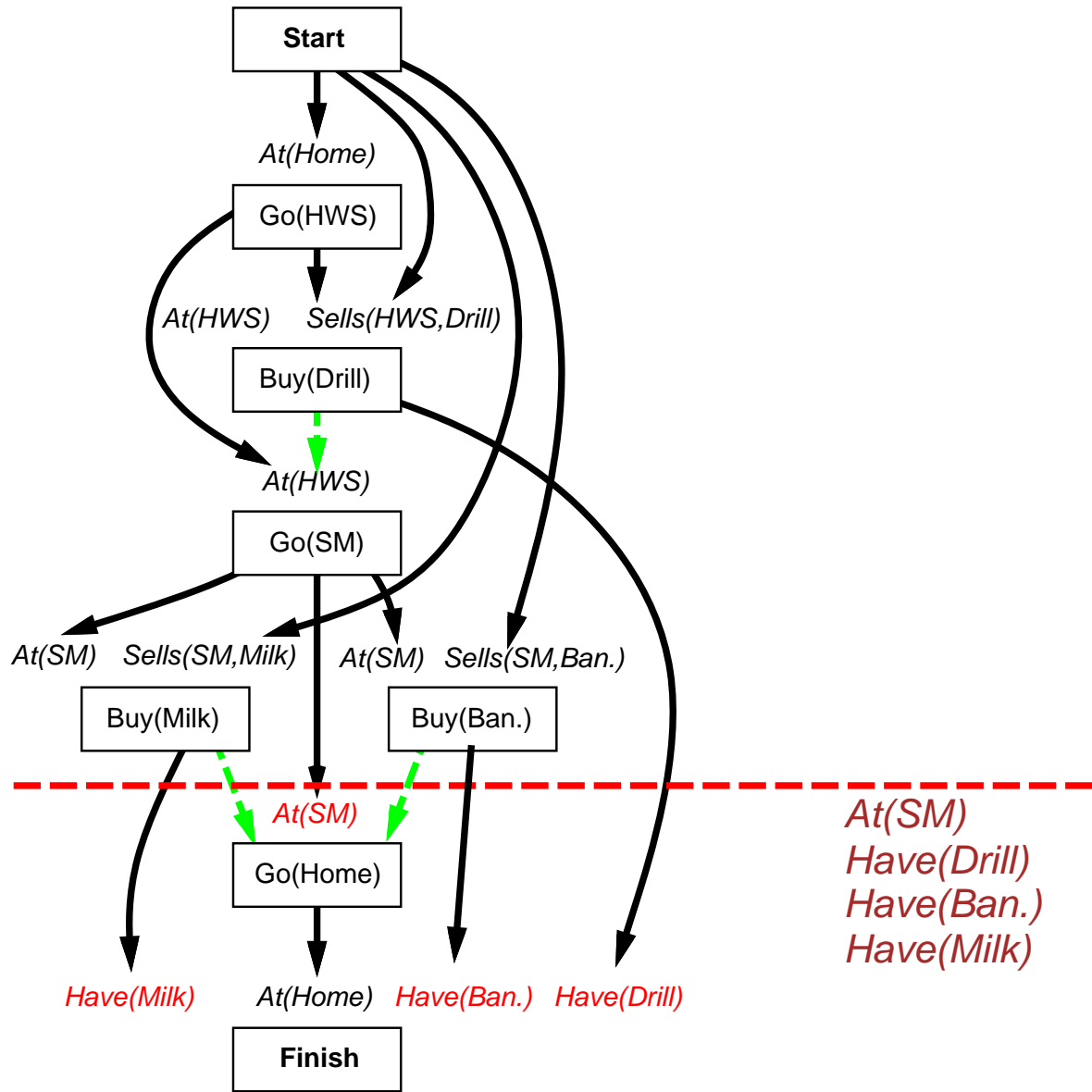
# Esempio



# Esempio

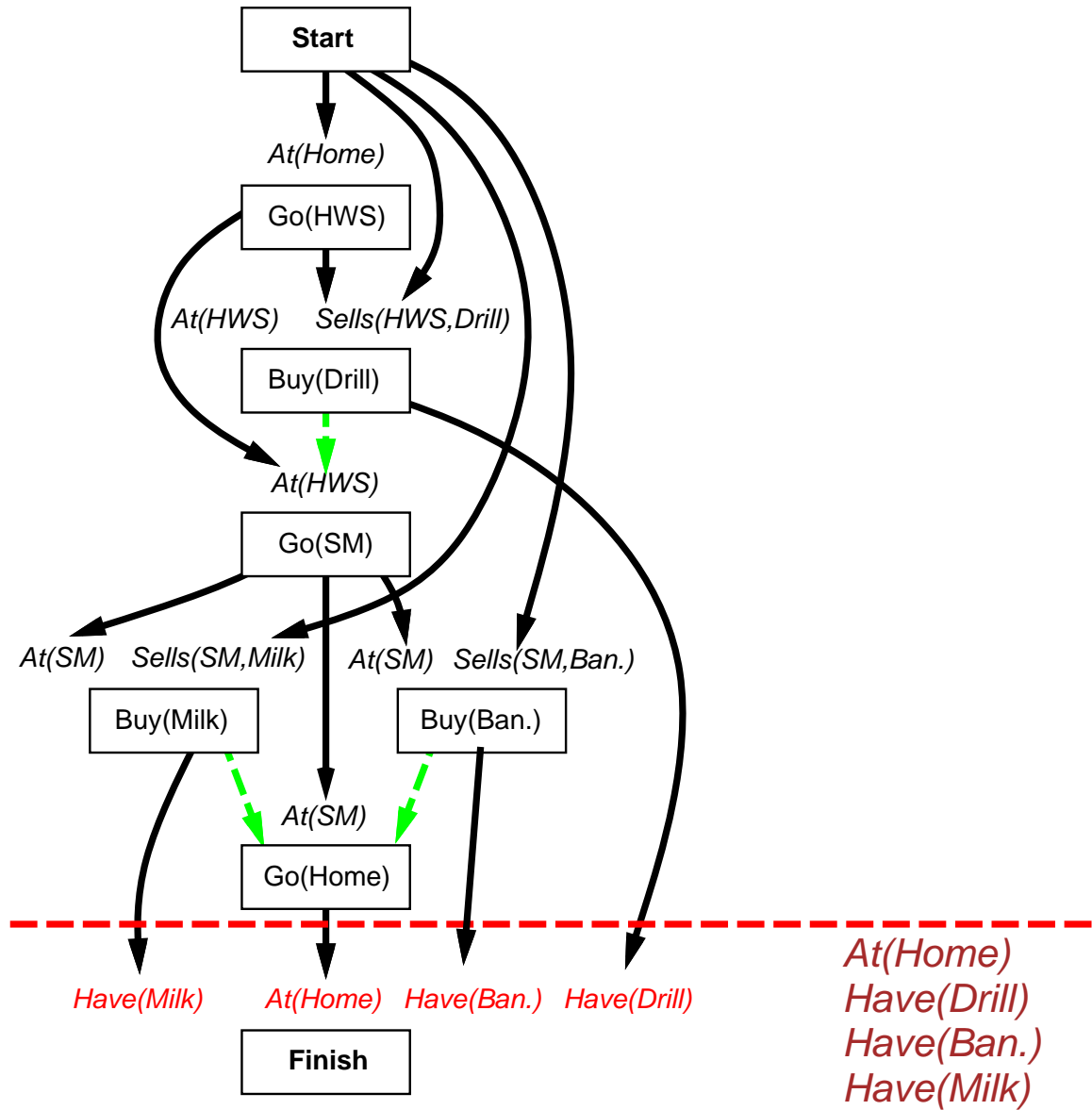


# Esempio





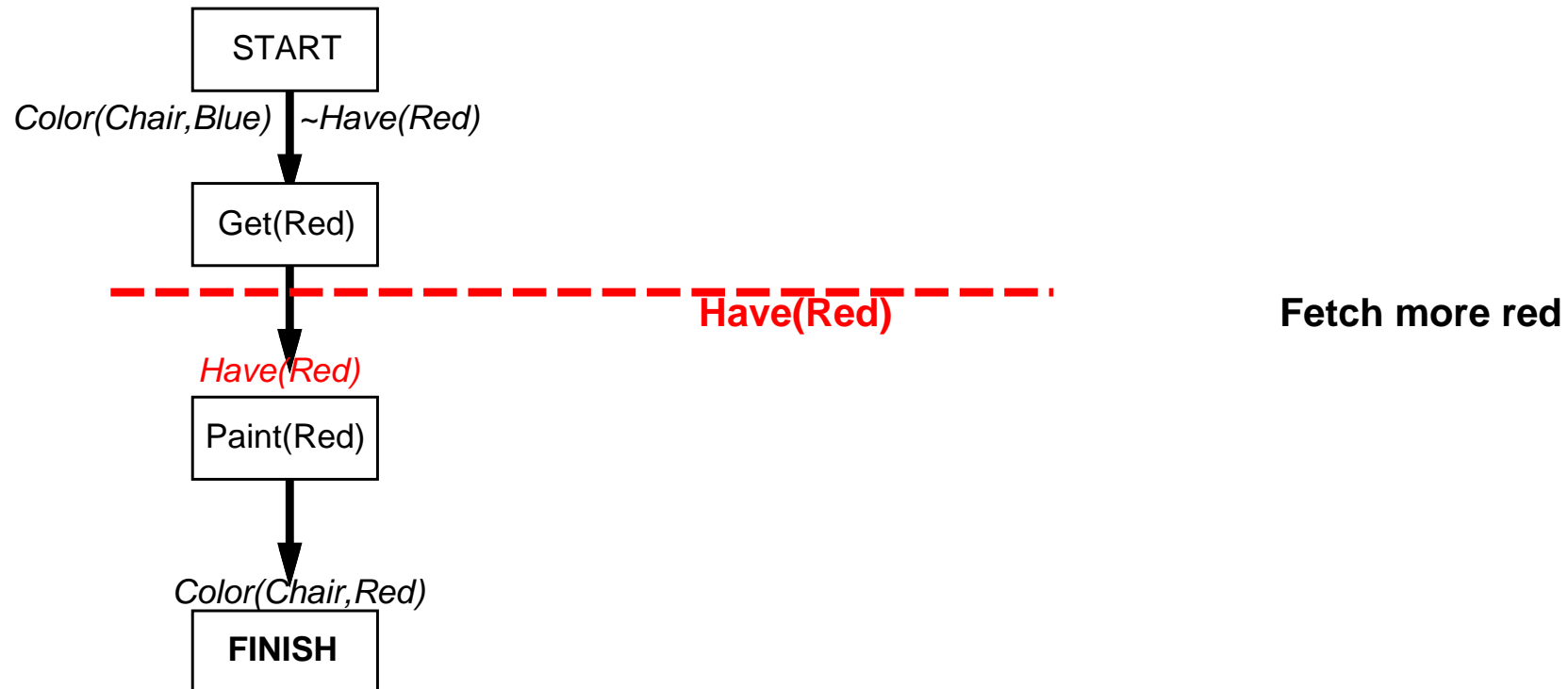
# Esempio



# Comportamento emergente

PRECONDITIONS

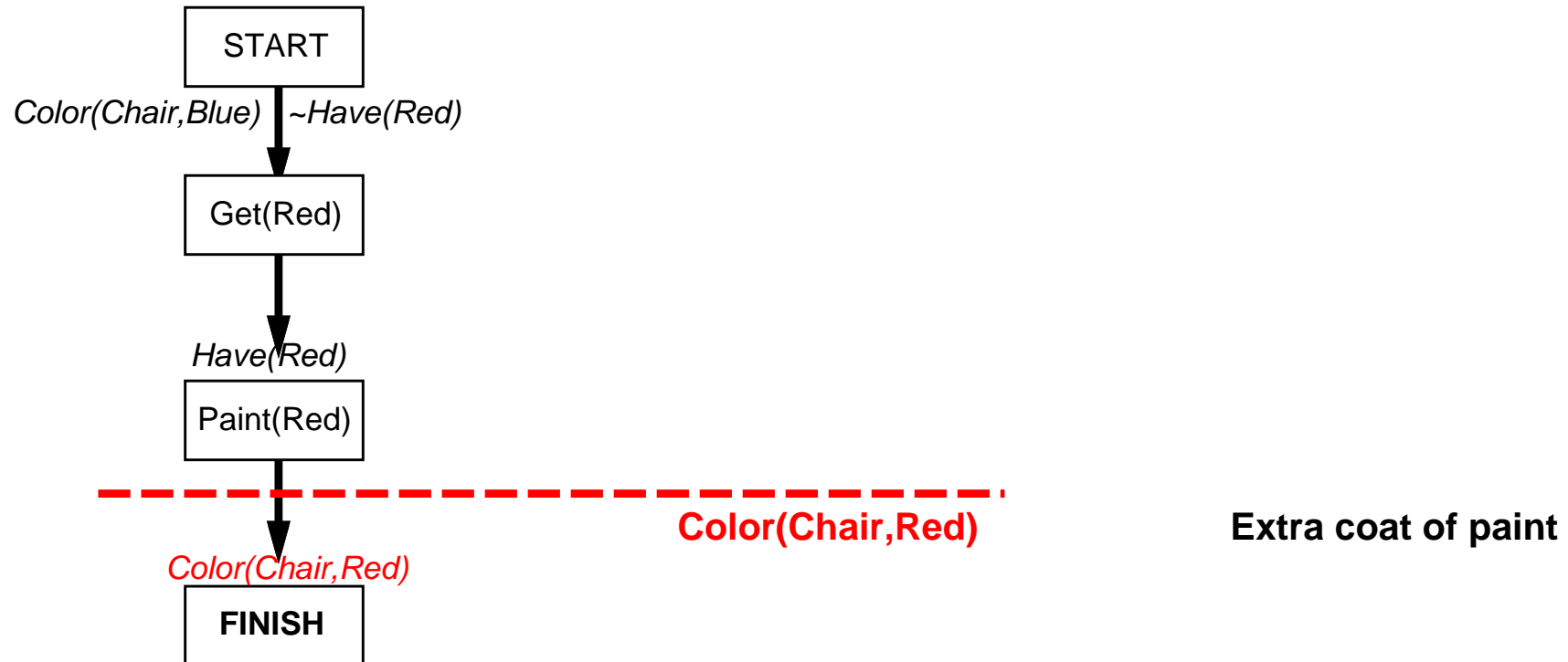
FAILURE RESPONSE



# Comportamento emergente

PRECONDITIONS

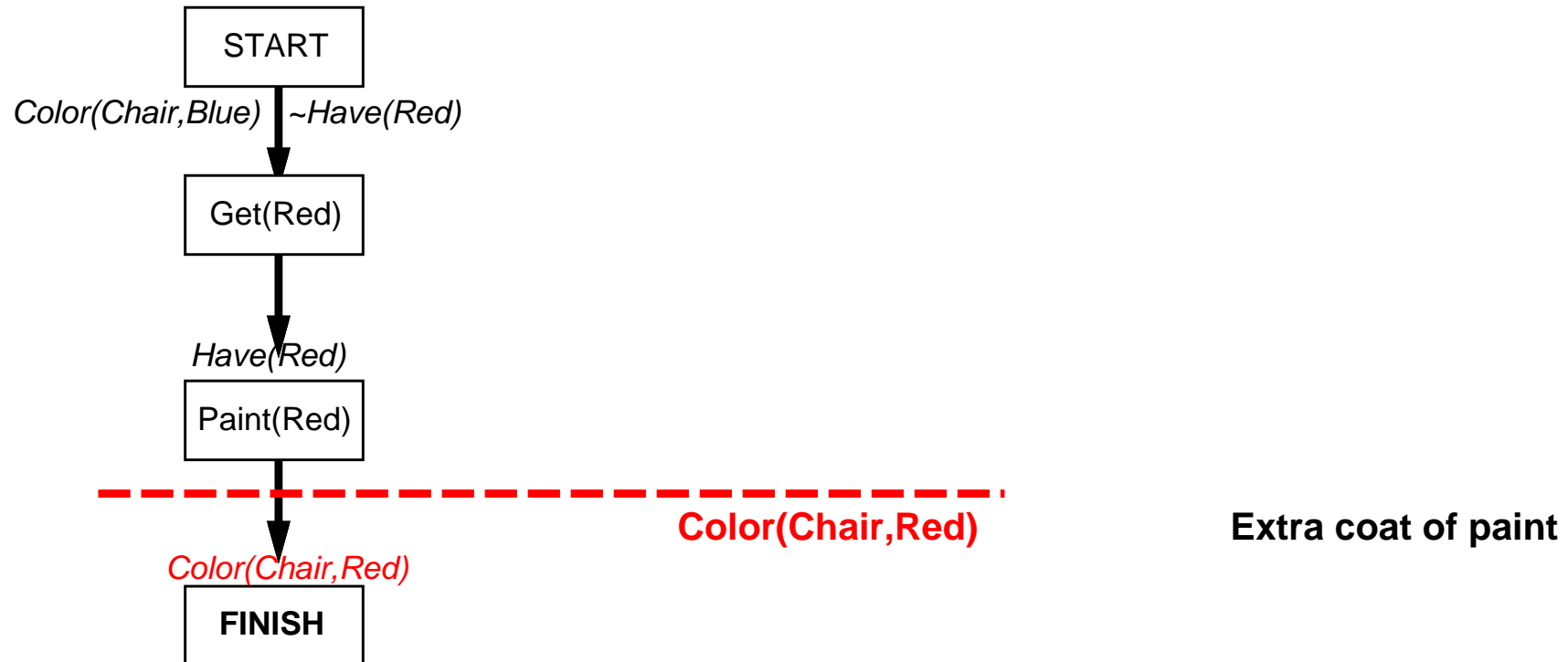
FAILURE RESPONSE



# Comportamento emergente

PRECONDITIONS

FAILURE RESPONSE



Il comportamento “Cicla fino al successo” *emerge* dalla interazione tra agente del tipo monitorizza/riplanifica ed ambiente non cooperativo