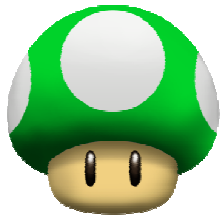


# INF 1771 – Inteligência Artificial

## Aula 12 – Planejamento

Edirlei Soares de Lima  
<elima@inf.puc-rio.br>



# Introdução

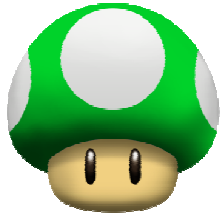
💡 Agentes vistos anteriormente:

💡 **Agentes baseados em busca.**

- 💡 Busca cega;
- 💡 Busca heurística;
- 💡 Busca local;

💡 **Agentes baseados em lógica.**

- 💡 Lógica proposicional;
- 💡 Lógica de primeira ordem;
- 💡 Prolog;

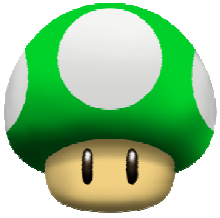


# Introdução

- ❏ **Planejamento** consiste na tarefa de apresentar uma sequência de ações para alcançar um determinado objetivo.

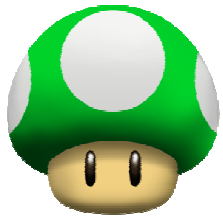
Ir(Mercado), Comprar(Biscoito), Ir(Farmácia), Comprar(Remédio), Ir(Casa)

- ❏ Dado um objetivo, um **agente planejador** deve ser capaz de construir um plano de ação para chegar ao seu objetivo.
- ❏ Após planejar, o agente deve **executar as ações do plano** uma a uma.



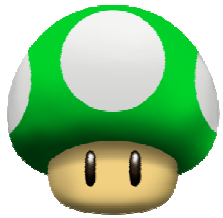
## Funcionamento de um Agente Planejador

- ❏ Inicialmente um agente planejador **gera um objetivo** a alcançar.
- ❏ **Constrói um plano** para atingir o objetivo a partir do estado atual do ambiente.
- ❏ **Executa o plano** do começo ao fim.
- ❏ **Gera um novo objetivo** com base no novo estado do ambiente.



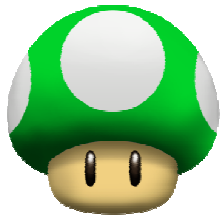
# Planejamento

- ❏ Em **planejamento clássico** o ambiente do problema possui as seguintes características:
  - ❏ Observável;
  - ❏ Determinístico;
  - ❏ Finito;
  - ❏ Estático;



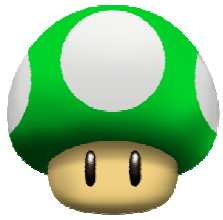
# Resolução de Problemas X Planejamento

- ❏ **Algoritmos de busca** tendem a tomar ações irrelevantes.
  - ❏ Grande fator de ramificação.
  - ❏ Pouco conhecimento para guiar a busca.
  
- ❏ **Planejador** não considera ações irrelevantes.
  - ❏ Faz conexões diretas entre estados (sentenças) e ações (pré-condições + efeitos)
  - ❏ Objetivo: Ter(Leite).
    - ❏ Ação: Comprar(Leite) => Ter(Leite)



## Resolução de Problemas X Planejamento

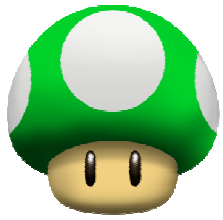
- ❏ Em problemas do mundo real é difícil definir uma boa heurística para **algoritmos de busca heurística**.
- ❏ Um **planejador** tem acesso a representação explícita do objetivo.
  - ❏ Objetivo: conjunção de sub-objetivos que levam ao objetivo final.
  - ❏ Heurística **única**: número de elementos da conjunção não-satisfeitos.



## Resolução de Problemas X Planejamento

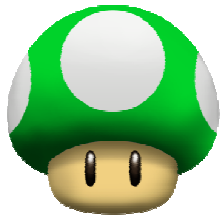
- ❏ **Algoritmos de busca** não tiram proveito da decomposição do problema.
- ❏ **Planejadores** aproveitam a estrutura do problema. É possível decompor com facilidade sub-objetivos.
- ❏ Exemplo:  $\text{Ter}(A) \wedge \text{Ter}(B) \wedge \text{Ter}(C) \wedge \text{Ter}(D)$





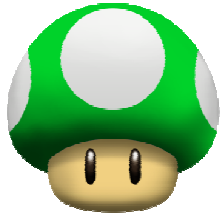
# Linguagem STRIPS

- ❏ **Linguagem formal** para a especificação de problemas de planeamento.
- ❏ **Representação de estados:** conjunção de literais positivos sem variáveis.
  - ❏ **Inicial:** Em(Casa)
  - ❏ **Final:**  $\text{Em(Casa)} \wedge \text{Ter(Leite)} \wedge \text{Ter(Bananas)} \wedge \text{Ter(Furadeira)}$
  - ❏ **Hipótese do mundo fechado:** qualquer condição não mencionada em um estado é considerada negativa.
    - ❏ Exemplo:  $\neg \text{Ter(Leite)} \wedge \neg \text{Ter(Bananas)} \wedge \neg \text{Ter(Furadeira)}$



# Linguagem STRIPS

- ❏ **Objetivos:** conjunção de literais e possivelmente variáveis:
  - ❏  $\text{Em}(\text{Casa}) \wedge \text{Ter}(\text{Leite}) \wedge \text{Ter}(\text{Bananas}) \wedge \text{Ter}(\text{Furadeira})$
  - ❏  $\text{Em}(x) \wedge \text{Vende}(x, \text{Leite})$
- ❏ **Ações** são especificadas em termos de pré-condições e efeitos:
  - ❏ **Descritor da ação:** predicado lógico
  - ❏ **Pré-condição:** conjunção de literais positivos
  - ❏ **Efeito:** conjunção de literais (positivos ou negativos)



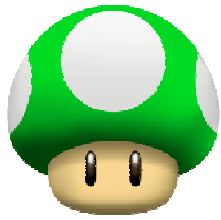
# Linguagem STRIPS

- ❏ Operador para ir de um lugar para outro:

**Ação** Ir(Destino),

**Pré-condição** Em(Partida)  $\wedge$  Caminho(Partida, Destino),

**Efeito** Em(Destino)  $\wedge$   $\neg$  Em(Partida))



## Exemplo - Transporte Aéreo de Carga

Início( $\text{Em}(C1, \text{SFO}) \wedge \text{Em}(C2, \text{JFK}) \wedge \text{Em}(A1, \text{SFO}) \wedge \text{Em}(A2, \text{JFK}) \wedge \text{Carga}(C1) \wedge \text{Carga}(C2) \wedge \text{Avião}(A1) \wedge \text{Avião}(A2) \wedge \text{Aeroporto}(\text{JFK}) \wedge \text{Aeroporto}(\text{SFO})$ )

Objetivo( $\text{Em}(C1, \text{JFK}) \wedge \text{Em}(C2, \text{SFO})$ )

Ação(**Carregar**( $c, a, l$ ))

PRÉ-CONDIÇÃO:  $\text{Em}(c, l) \wedge \text{Em}(a, l) \wedge \text{Carga}(c) \wedge \text{Avião}(a) \wedge \text{Aeroporto}(l)$

EFEITO:  $\neg \text{Em}(c, l) \wedge \text{Dentro}(c, a)$

Ação(**Descarregar**( $c, a, l$ ))

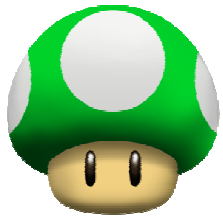
PRÉ-CONDIÇÃO:  $\text{Dentro}(c, a) \wedge \text{Em}(a, l) \wedge \text{Carga}(c) \wedge \text{Avião}(a) \wedge \text{Aeroporto}(l)$

EFEITO:  $\text{Em}(c, l) \wedge \neg \text{Dentro}(c, a)$

Ação(**Voar**( $a, de, para$ ))

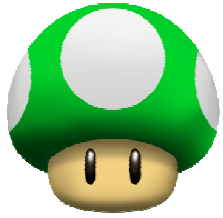
PRÉ-CONDIÇÃO:  $\text{Em}(a, de) \wedge \text{Avião}(a) \wedge \text{Aeroporto}(de) \wedge \text{Aeroporto}(para)$

EFEITO:  $\neg \text{Em}(a, de) \wedge \text{Em}(a, para)$



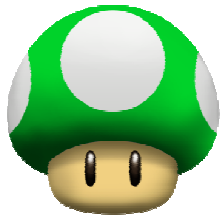
# Tipos de Planejadores

- ❏ Formas de Buscas de Planos:
  - ❏ **Progressivo:** estado inicial -> objetivo.
  - ❏ **Regressivo:** objetivo -> estado inicial.
    - ❏ mais eficiente (há menos caminhos partindo do objetivo do que do estado inicial)
  
- ❏ Espaços de busca:
  - ❏ **Espaço de situações:** Funciona da mesma forma que na resolução de problemas por meio de busca.
  
  - ❏ **Espaço de planos:** planos parciais.
    - ❏ mais flexível.



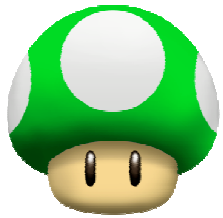
# Busca em Espaço de situações

- ❗ A **busca em espaço de situações** é **ineficiente** devido a ela não considerar o problema das ações irrelevantes. Todas as opções de ações são testadas em cada estado.
- ❗ Isso faz com que a complexidade do problema cresça muito rapidamente.
- ❗ **Solução?** Busca no espaço de planos parciais (**planejamento de ordem parcial**).



# Planejamento de Ordem Parcial

- ❏ **Subdivisão do problema.**
- ❏ **Ordem de elaboração do plano flexível.**
- ❏ **Compromisso mínimo.**
  - ❏ Adiar decisões durante a procura.
- ❏ O planejador de ordem parcial pode inserir duas ações em um plano sem especificar qual delas deve ser executada primeiro.



# Exemplo dos Sapatos

Inicio()

Objetivo(SapatoDireitoCalçado^SapatoEsquerdoCalçado)

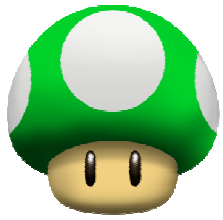
Ação(SapatoDireito,  
PRECOND: MeiaDireitaCalçada,  
EFFECT: SapatoDireitoCalçado)

Ação(MeiaDireita,  
EFFECT: MeiaDireitaCalçada)

Ação(SapatoEsquerdo,  
PRECOND: MeiaEsquerdaCalçada,  
EFFECT: SapatoEsquerdoCalçado)

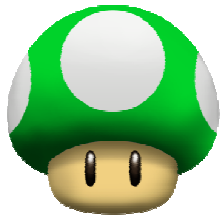
Ação(MeiaEsquerda,  
EFFECT: MeiaEsquerdaCalçada)





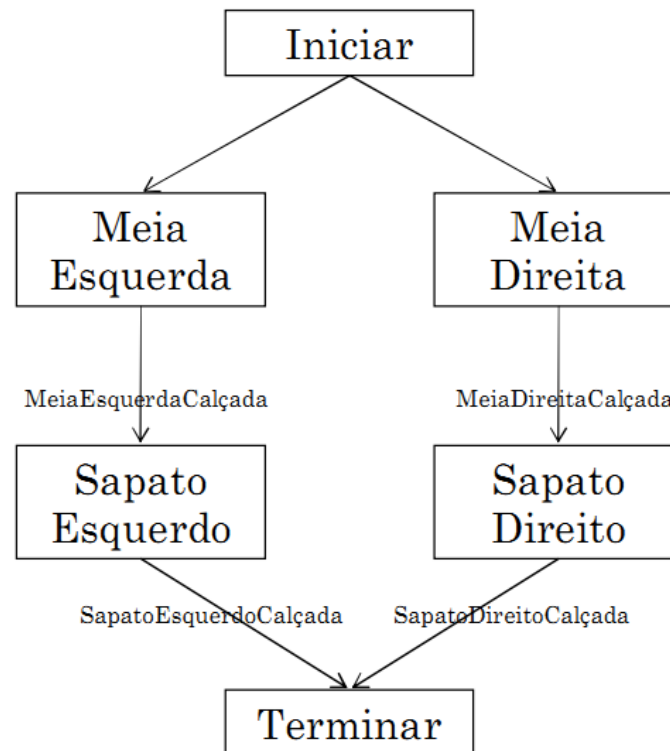
# Exemplo dos Sapatos

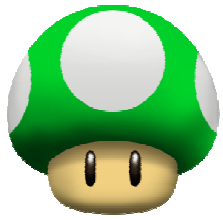
- ❏ Um planejador de ordem parcial deve ser capaz de chegar a **duas sequências de ações**:
  - ❏ MeiaDireita seguido por SapatoDireito;
  - ❏ MeiaEsquada seguido por SapatoEsquerdo.
- ❏ As duas sequências podem ser **combinadas** para produzir o plano final.



# Exemplo dos Sapatos

## ❏ Plano de Ordem Parcial

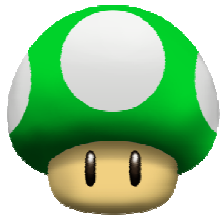




# Exemplo dos Sapatos

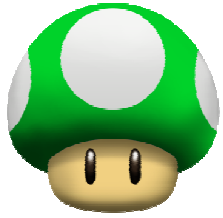
## 💡 Plano de Ordem Total





# Planejamento de Ordem Parcial

- ❏ O planejamento de ordem parcial pode ser implementado como uma **busca no espaço de ordem parcial de planos**.
- ❏ **Idéia:**
  - ❏ Busca-se um plano desejado em vez de uma situação desejada (meta-busca).
  - ❏ Parte-se de um plano inicial (parcial) e aplica-se 2 tipos de **operadores** até chegar a um plano final (completo)
- ❏ **Plano Final:**
  - ❏ **Completo:** todas as pré-condições de todas as ações são alcançada por meio de alguma outra ação.
  - ❏ **Consistente:** não há contradições.



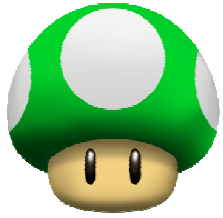
# Planejamento de Ordem Parcial

## ❏ **Operador de refinamento:**

- ❏ adicionar novo passo;
- ❏ instanciar variável;
- ❏ ordenar passos;

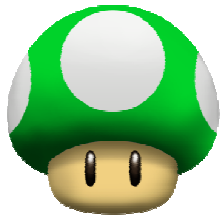
## ❏ **Operador de modificação:**

- ❏ operadores de revisão (para corrigir planos);
- ❏ operadores de decomposição hierárquica;



# Planejamento de Ordem Parcial

- Na estratégia de **compromisso mínimo** a ordem e instanciações totais são decididas quando necessário.
- Exemplo:**
  - Para objetivo **Ter(Leite)**, a ação **Comprar(Produto, Loja)**, instancia-se somente item: **Comprar(Leite, Loja)**
  - Para o problema de colocar meias e sapatos: colocar cada meia antes do sapato, sem dizer por onde começar (esquerda ou direita)



# Planejamento de Ordem Parcial

## ❏ Planos são definidos por 4 componentes:

### ❏ **Ações/Passos:**

- ❏ Ação( $x$ , Precondição  $y$ , Efeito  $z$ )

### ❏ **Restrições de Ordenação:**

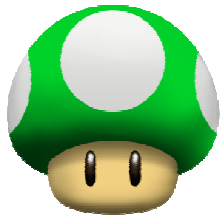
- ❏  $S1 < Sk < Sn$ , o que não significa que entre  $S1$  e  $Sk$  não exista outro passo

### ❏ **Vínculos Causais:** $\{Si \xrightarrow{c} Sj\}$

- ❏ Efeitos da ação  $Si$  = pré-condições da ação  $Sj$ . Não existe nenhum passo entre eles que negue a pré-condição  $c$ .

### ❏ **Pré-condições Abertas:**

- ❏ Não é alcançada por nenhuma ação do plano.



# Exemplo dos Sapatos

❏ **Objetivo:** SapatoDireitoCalçado  $\wedge$  SapatoEsquerdoCalçado

❏ **Operadores:**

Ação(SapatoDireito, PRECOND: MeiaDireitaCalçada, EFFECT: SapatoDireitoCalçado)

Ação(MeiaDireita, EFFECT: MeiaDireitaCalçada)

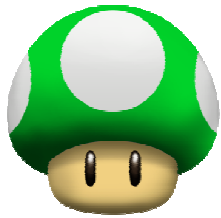
Ação(SapatoEsquerdo, PRECOND: MeiaEsquerdaCalçada, EFFECT: SapatoEsquerdoCalçado)

Ação(MeiaEsquerda, EFFECT: MeiaEsquerdaCalçada)

❏ **Plano Inicial:**

Plano(Passos: {S1: Operador(Ação: Inicio),  
S2: Operador(Ação: Fim,  
Precondição: SapatoDireitoCalçado  $\wedge$  SapatoEsquerdoCalçado)},  
ORDENAÇÃO: { S1 < S2 },  
VINCULOS: {},  
PRECONDIÇÕES\_ABERTAS: {})





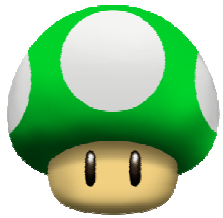
# Planejamento de Ordem Parcial

## ❏ **Características do planejamento de ordem parcial:**

- ❏ A inserção de um passo só é considerada se ele atender uma condição não atingida.
- ❏ O planejador é regressivo.
- ❏ É correto e completo, assumindo busca em largura ou em profundidade iterativa

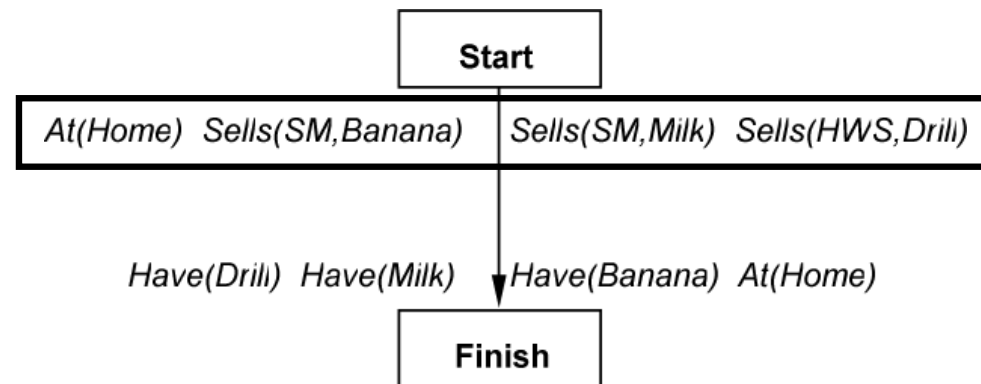
## ❏ **Idéia do algoritmo:**

- ❏ Identificar um passo com a pré-condição (sub-goal) não satisfeita.
- ❏ Introduzir um passo cujo efeito satisfaz a pré-condição.
- ❏ Instanciar variáveis e atualizar as ligações causais.
- ❏ Verificar se há conflitos e corrigir o plano se for o caso.



# Exemplo

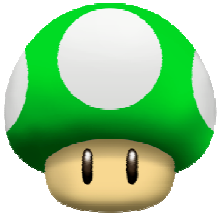
## Plano Inicial:



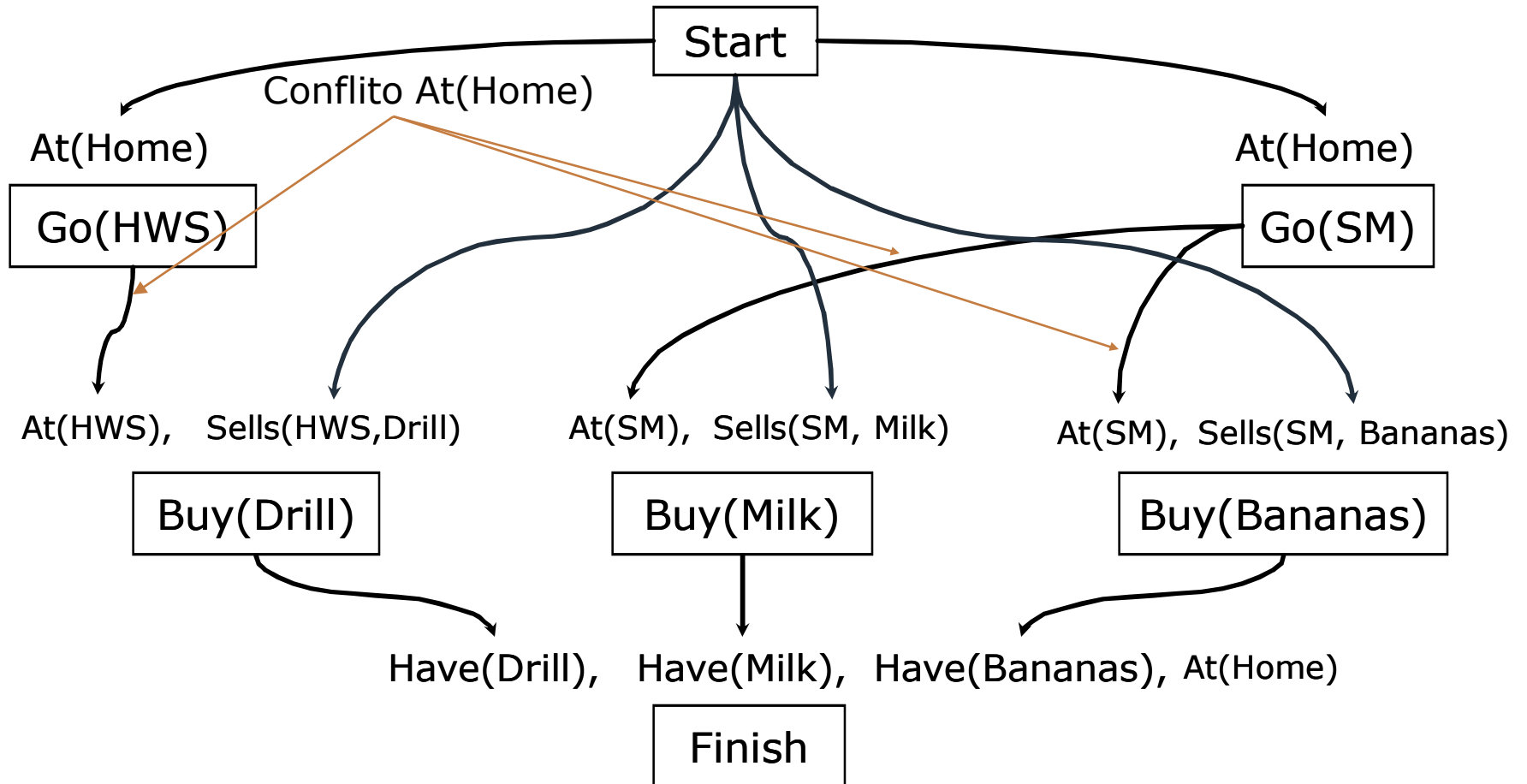
## Ações:

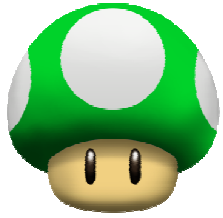
Op(ACTION: **Go**(there),  
PRECOND: At(here),  
EFFECT: At(there)  $\wedge$   $\neg$  At(here))

Op(ACTION: **Buy**(x),  
PRECOND: At(store)  $\wedge$  Sells(store, x),  
EFFECT: Have(x))



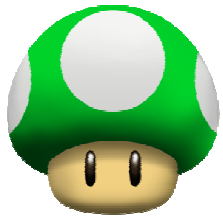
# Exemplo





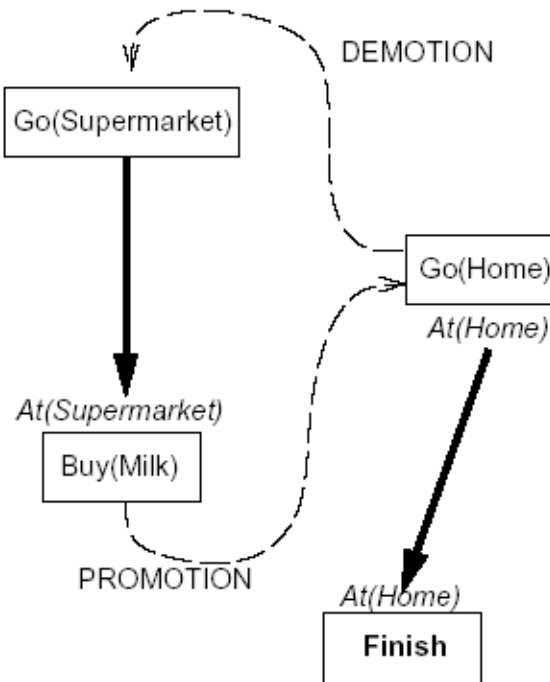
## Conflito em Planejamento de Ordem Parcial

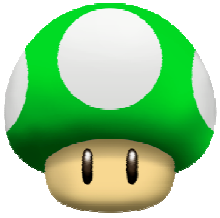
- ❗ Um **conflito** ocorre quando os efeitos de uma ação põem em risco as pré-condições de outra ação.
- ❗ No caso anterior, os operadores  $Go(HWS)$  e  $Go(SM)$  apagam  $At(Home)$ .



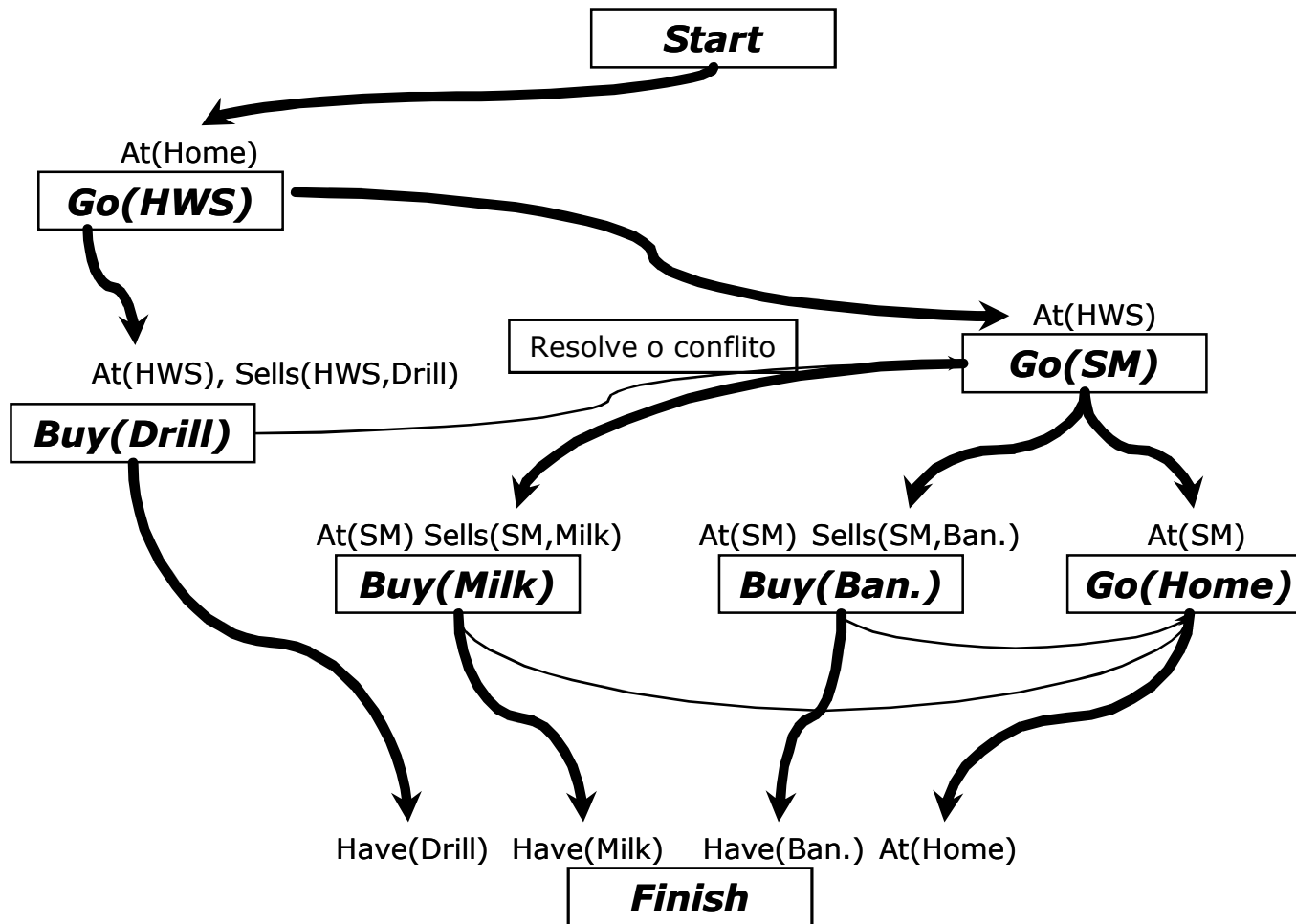
# Solução de Conflitos

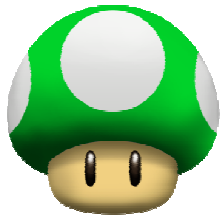
## Demotion e Promotion





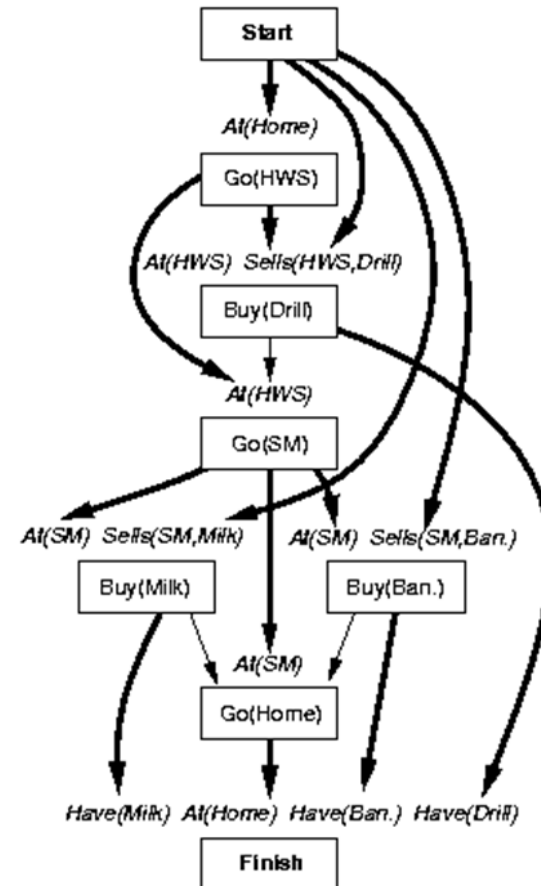
# Exemplo

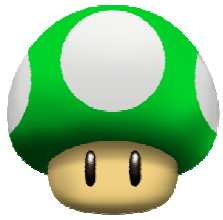




# Exemplo

## ❓ Plano de Ordem Parcial





# Aplicações de Planejamento

- ❏ Qualquer problema que necessite de **passos/ações** para chegar a um determinado **objetivo**.
- ❏ Exemplos:
  - ❏ Robôs que realizam tarefas.
  - ❏ Personagens de jogos direcionados a objetivos.
  - ❏ Geração de histórias para storytelling interativo.