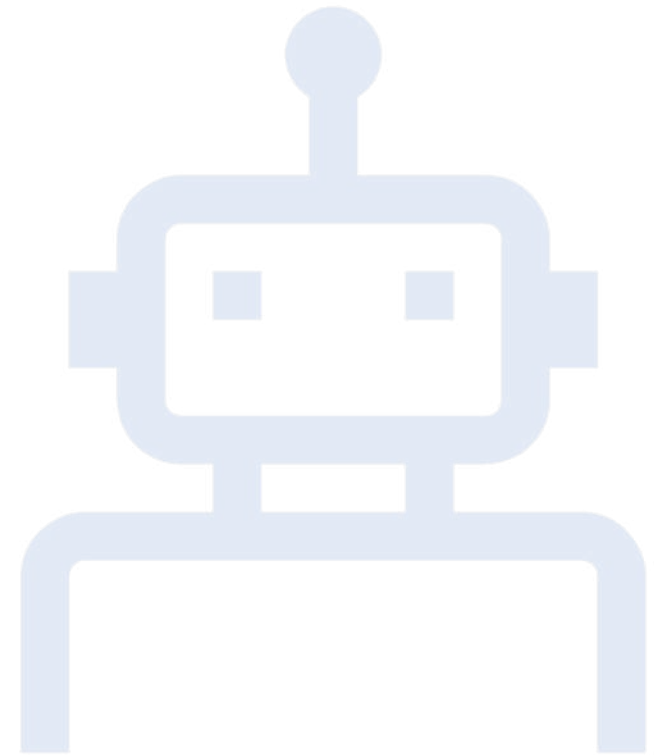


Inteligência Artificial



Anna Helena Reali Costa
PCS – EPUSP

Planejamento Clássico





Planejamento

1. Introdução
2. Planejamento e Resolução de Problemas
3. STRIPS
4. POP
1. Conclusões

Conceitos Básicos

- Planejador: mecanismo que permite encontrar/gerar um **plano** que permita a um agente atingir um objetivo
- Plano: **sequência ordenada de ações**
 - problema: obter banana, leite e uma furadeira
 - plano: ir ao supermercado, ir à seção de frutas, pegar as bananas, ir à seção de leite, pegar uma caixa de leite, ir ao caixa, pagar tudo, ir a uma loja de ferramentas, ..., voltar para casa.

Planejamento e Resolução de Problemas

- **Representação em RP**

- **Ações** : programas que geram o estado sucessor
- **Estados** : descrição completa
 - problemático em ambientes inacessíveis
- **Objetivos**: função de teste e heurística
- **Planos**: totalmente ordenados e criados incrementalmente a partir do estado inicial (ex. posições das peças de um jogo)

- Exemplo do supermercado

- **estado inicial**: em casa, sem objetos desejados
- **estado final**: em casa com objetos desejados
- **operadores**: tudo o que o agente pode fazer
- **heurística**: número de objetos ainda não possuídos

Exemplo de Resolução de Problemas



Limitações de Resolução de Problemas

- Fator de ramificação grande
- A função heurística apenas escolhe o estado mais próximo do objetivo. Não permite descartar ações a priori
- Não permite abstração dos estados parciais
- Considera ações a partir do estado inicial, uma após a outra
- Objetivo é testado para cada estado, e para cada novo estado gerado um teste idêntico deve ser realizado
- **Ideia:** combinar busca com uma descrição mais rica dos estados, baseada em conhecimento, e criar um algoritmo eficiente para processá-la!

Planejamento: 3 ideias principais

- Representação dos estados, objetivos e ações usando **lógica** (descrições parciais dos estados)
 - pode conectar diretamente estados e ações
 - ex. estado: Have(Milk), ação: Buy(milk) → Have(Milk)
- Liberdade de adicionar ações ao plano quando forem necessárias
 - ordem de planejamento \neq ordem de execução
 - primeiro, coloca-se o que é importante (ex. Buy(Milk)) mesmo sem saber quando esta ação será executada!
 - diminui fator de ramificação
- Utilizar a estratégia de dividir para conquistar, resolvendo sub-objetivos
 - sub-plano supermercado, sub-plano loja de ferramentas

Planejamento clássico: Ambientes

- Consideramos aqui ambientes para planejamento clássico, que são:
 - Totalmente observáveis
 - Determinísticos
 - Finitos
 - Estáticos
 - Discretos (no tempo, nas ações, nos objetos e nos efeitos)



Uma Proposta: STRIPS & POP

- Criar uma linguagem especializada



STRIPS: Stanford Research Institute Problem Solver (Fikes e Nilsson, 1971)

- Criar um algoritmo para planejar



POP: Partial Order Planning

STRIPS: Estados

- Estados: **conjunção de literais sem variáveis**
 - **Inicial**: $At(Home)$
 - Por default, literal não representado é **falso** (**hipótese do mundo fechado**); assim, não precisa escrever: $\neg Have(Milk) \wedge \neg Have(Bananas) \wedge \neg Have(Drill)$
 - **Final**: $At(Home) \wedge Have(Milk) \wedge Have(Bananas) \wedge Have(Drill)$

Mundo fechado (closed world assumption): assume que o modelo do mundo do problema contém tudo que for necessário para o planejamento (não há surpresas!)

STRIPS: Objetivos

- Objetivos: **conjunção de literais e/ou variáveis**
 - $\text{At(Home)} \wedge \text{Have(Milk)} \wedge \text{Have(Bananas)} \wedge \text{Have(Drill)}$
 - $\text{At}(x) \wedge \text{Sells}(x, \text{Milk})$ x é uma variável

STRIPS: Ações

- Ações:
 - Descritor da **ação**: predicado lógico
 - **Pré-condição**: conjunção de literais positivos
 - **Efeito**: conjunção de literais, podendo ser:
 - positivos (adicionados a uma lista)
 - negativos (retirados de uma lista)

STRIPS: Ações

- Exemplo: Operador para ir de um lugar a outro
 - Op(ACTION: Go(there),
 PRECOND: At(there) \wedge Path(there, there),
 EFFECT: ADD: At(there), DEL: \neg At(there))
- Notação alternativa *At(there), Path(there, there)*

Go(there)

At(there), \neg At(there)

Tipos de Planejadores

- **Controle**

- **Progressivo**: estado inicial → objetivo

- **Regressivo**: objetivo → estado inicial

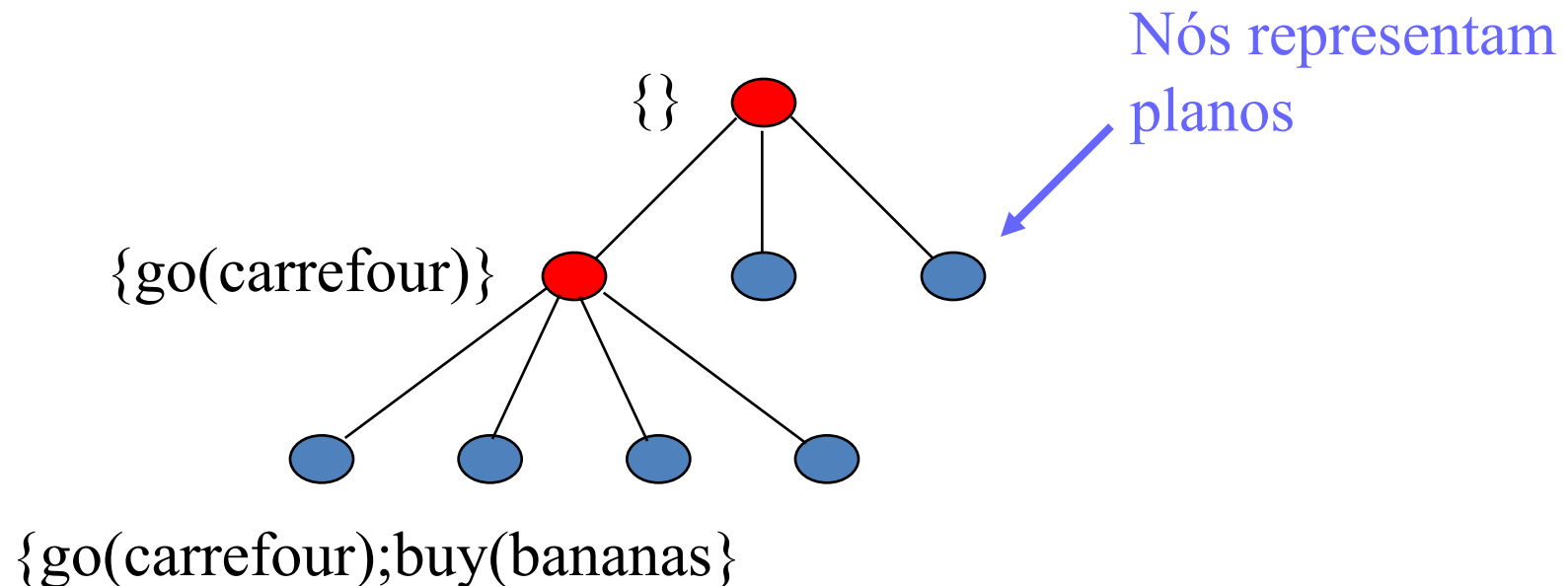
- mais eficiente (há menos caminhos partindo do objetivo do que do estado inicial), mas é problemático se existem múltiplos objetivos

Tipos de Planejadores

- **Espaços de busca**
 - **Espaço de situações** (como em resolução de problemas): vértices na árvore de busca representam estados do mundo
 - **Espaço de planos** (planos parciais): vértices na árvore de busca representam planos parciais
 - mais flexível, evita engajamento prematuro

Busca no Espaço de Planos

- Ideia: buscar um plano desejado em vez de uma situação desejada (espécie de meta-busca)
- Parte-se de um plano inicial (parcial), e aplica-se os operadores até chegar a um plano final (completo)



Busca no Espaço de Planos: Nós

Cada nó representa um plano parcial e contém:

- **Ações** = $\{A1, A2, A3, \dots, A_n\}$
- **Restrições de Ordem** = $\{A1 < A2, \dots, A3 < A_n\}$,
- **Ligações Causais** = $\{A_i \xrightarrow{c} A_j, \dots\}$
 - efeito de A_i adiciona c que é uma pré-condição de A_j
- **Pré-Condições Abertas**: conjunto de pré-condições que ainda não tem ligações causais
- **Instanciação de Variáveis** = $\{x = \text{cte1}, y = z\}$

Busca no Espaço de Planos: Operadores

Os operadores possíveis no espaço de planos são:

- **Adicionar uma ação** para eliminar uma pré-condição aberta
- **Adicionar um link causal** de uma ação já existente para uma pré-condição aberta
- **Adicionar uma restrição de ordem** de uma ação em relação a outra
- **Instanciar uma variável**

Busca no Espaço de Planos: Plano Final

- **Plano inicial**

- passos **Start** e **Finish**
- Start tem como efeitos o estado inicial do mundo
- Finish tem como pré-condições o objetivo do plano

- **Plano final**

- **Completo** – toda a pré-condição de toda ação tem uma ligação causal para alguma outra ação
- **Consistente** – não há contradições
 - nos ordenamentos das ações
 - nas atribuição de variáveis

Exemplo: Plano para calçar meias e sapatos

- Plano inicial
 - Start
 - Finish (pré-condição: estar com meias e sapatos)
- Operadores
 - calçar meia direita (pré-condição: pé direito descalço; efeito: pé direito com meia)
 - calçar sapato direito (pré-condição: pé direito com meia; efeito: pé direito com meia e sapato)
 - calçar meia esquerda...
 - calçar sapato esquerdo...
- Plano final?
 - Existem vários possíveis....
 - Como representar isto?

Exemplo: Plano para calçar meias e sapatos

- **Objetivo:** $\text{RightShoeOn} \wedge \text{LeftShoeOn}$
- **Operadores:**

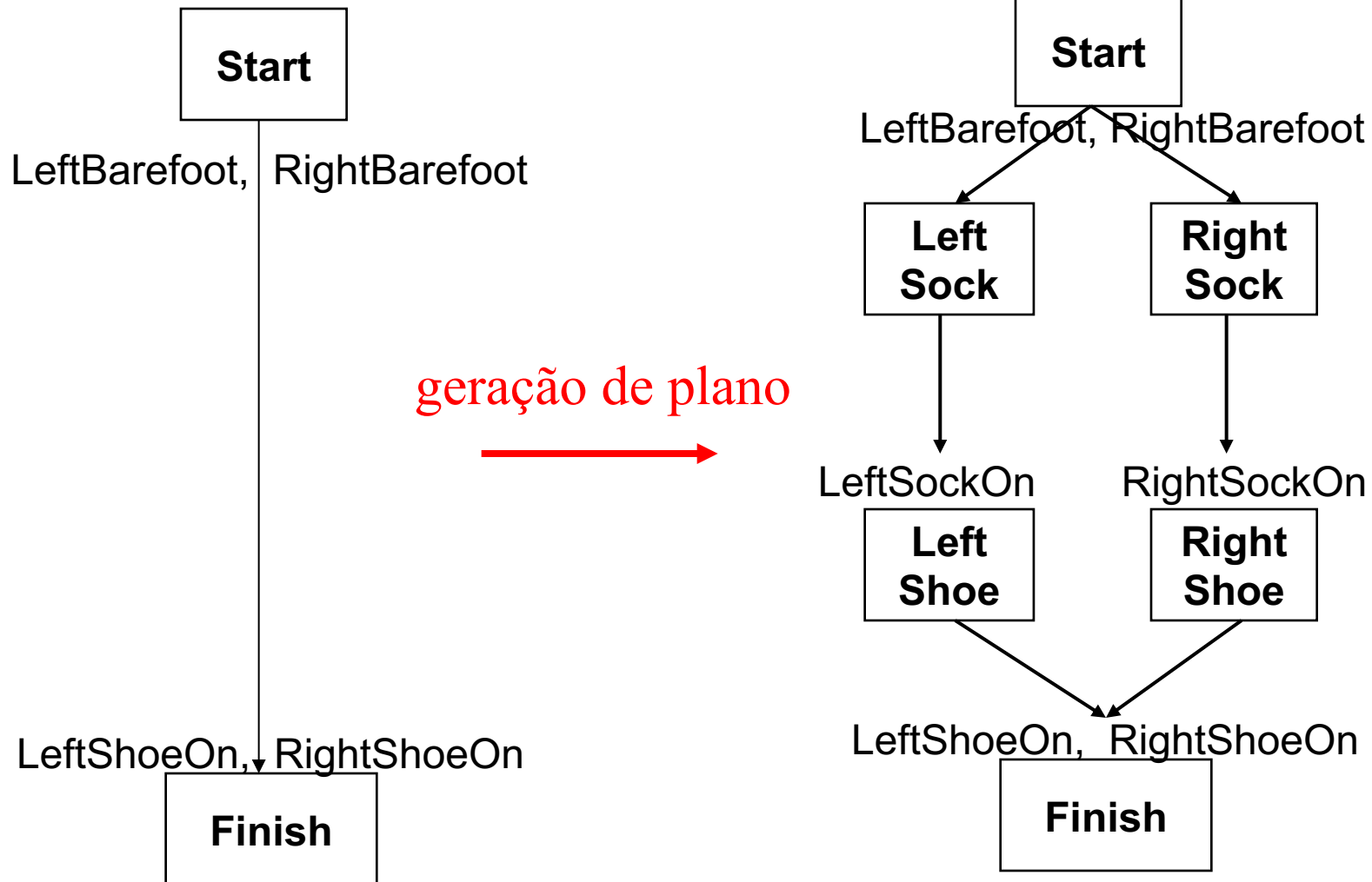
Op(ACTION:RightShoe PRECOND: RightSockOn , EFFECT: ADD: RightShoeOn)	Op(ACTION: RightSock PRECOND: RightBarefoot , EFFECT: ADD: RightSockOn)
Op(ACTION:LeftShoe PRECOND: LeftSockOn , EFFECT: ADD: LeftShoeOn)	Op(ACTION: LeftSock PRECOND: LeftBarefoot , EFFECT: LeftSockOn)

Exemplo: Plano para calçar meias e sapatos

- Plano inicial

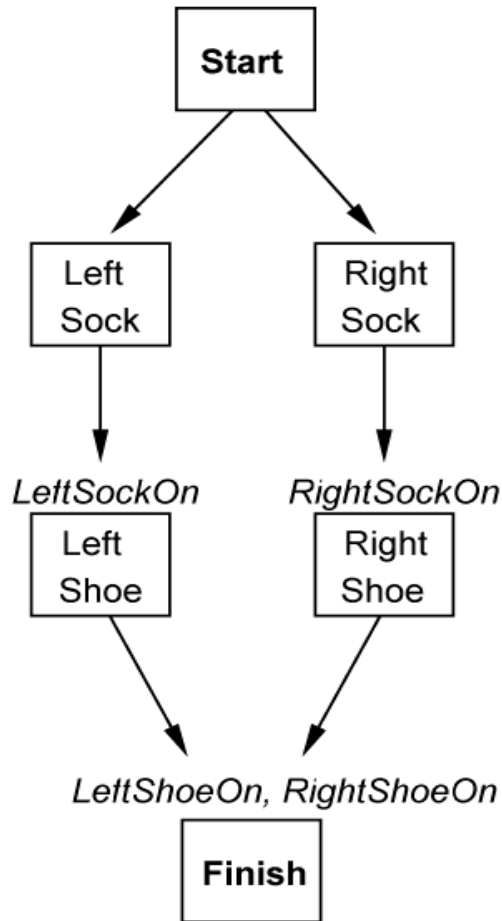
Plan(**ACTIONS** = {S1: Op(ACTION: Start,
EFFECT: RightBarefoot \wedge
LeftBarefoot),
S2: Op(ACTION: Finish,
PRECOND: RightShoeOn \wedge
LeftShoeOn)},
ORDERINGS: { S1 < S2 }, // relações de ordem
BINDINGS: {}, // instâncias de variáveis
LINKS: {}) // conjunto de links causais

Plano de Ordem Parcial

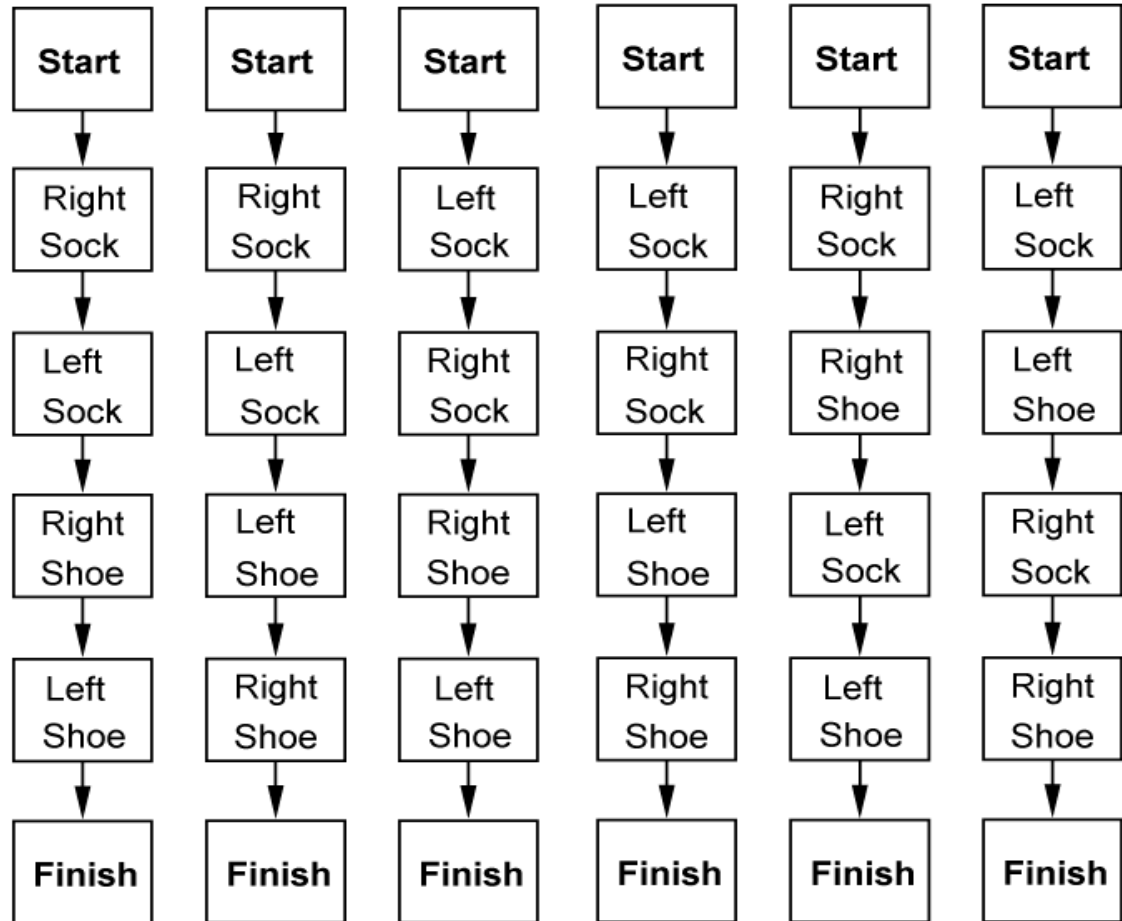


Plano de Ordem Parcial e Total

Partial Order Plan:



Total Order Plans:



Princípio do menor engajamento

POP (Partial Order Planning)

- Características do POP

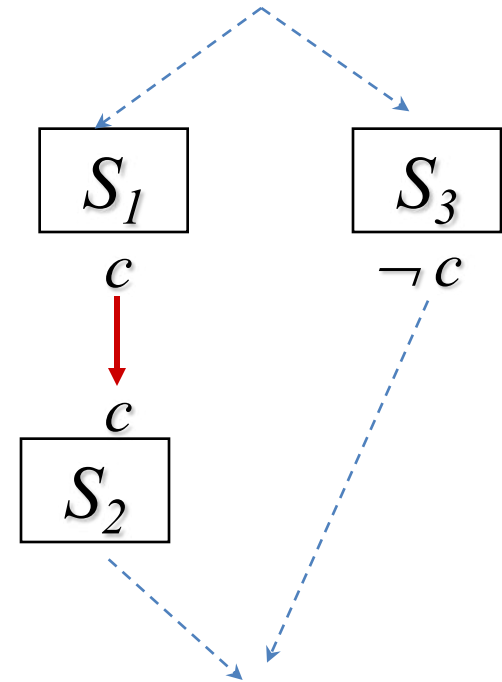
- algoritmo não determinístico
- a inserção de uma ação só é considerada se atender uma pré-condição **aberta**
- planejador **regressivo** (do objetivo para o início)
- é correto e completo, assumindo busca em largura ou em profundidade iterativa

- Ideia do algoritmo

1. identifica ação com pré-condição aberta
2. introduz ação cujo efeito é satisfazer esta pré-condição
3. instancia variáveis e atualiza os links causais
4. verifica se há **ameaças** e corrige o plano se for o caso

POP: Problema da Ameaça

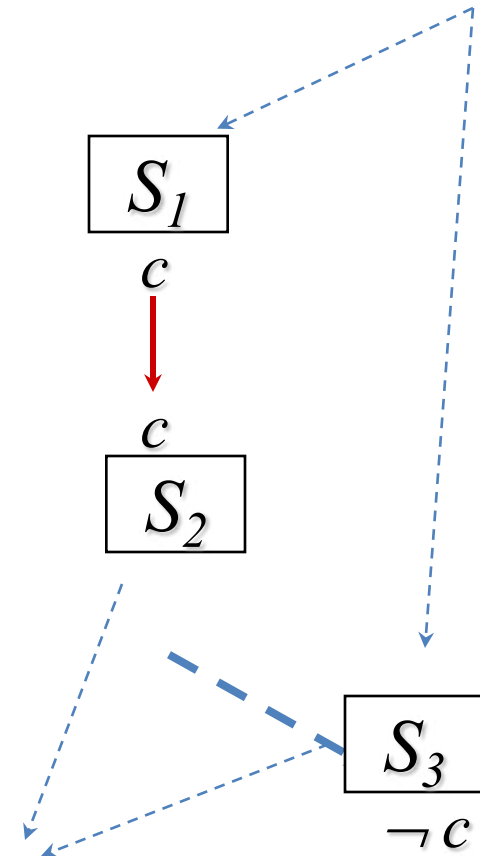
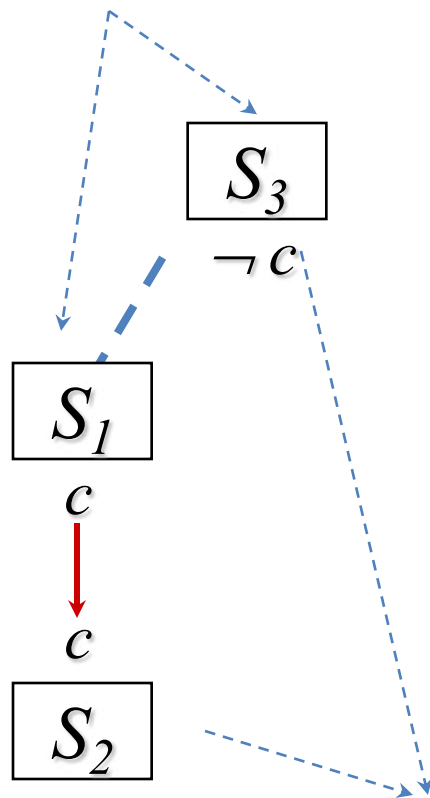
- Ameaça
 - ocorre quando os efeitos de um passo põem em risco as pré-condições de outro
- Como testar?
 - O efeito do novo passo é inconsistente com condição protegida
 - O passo antigo é inconsistente com nova condição protegida



S3 ameaça a condição c estabelecida por $S1$ e protegida pelo link causal $S1$ para $S2$

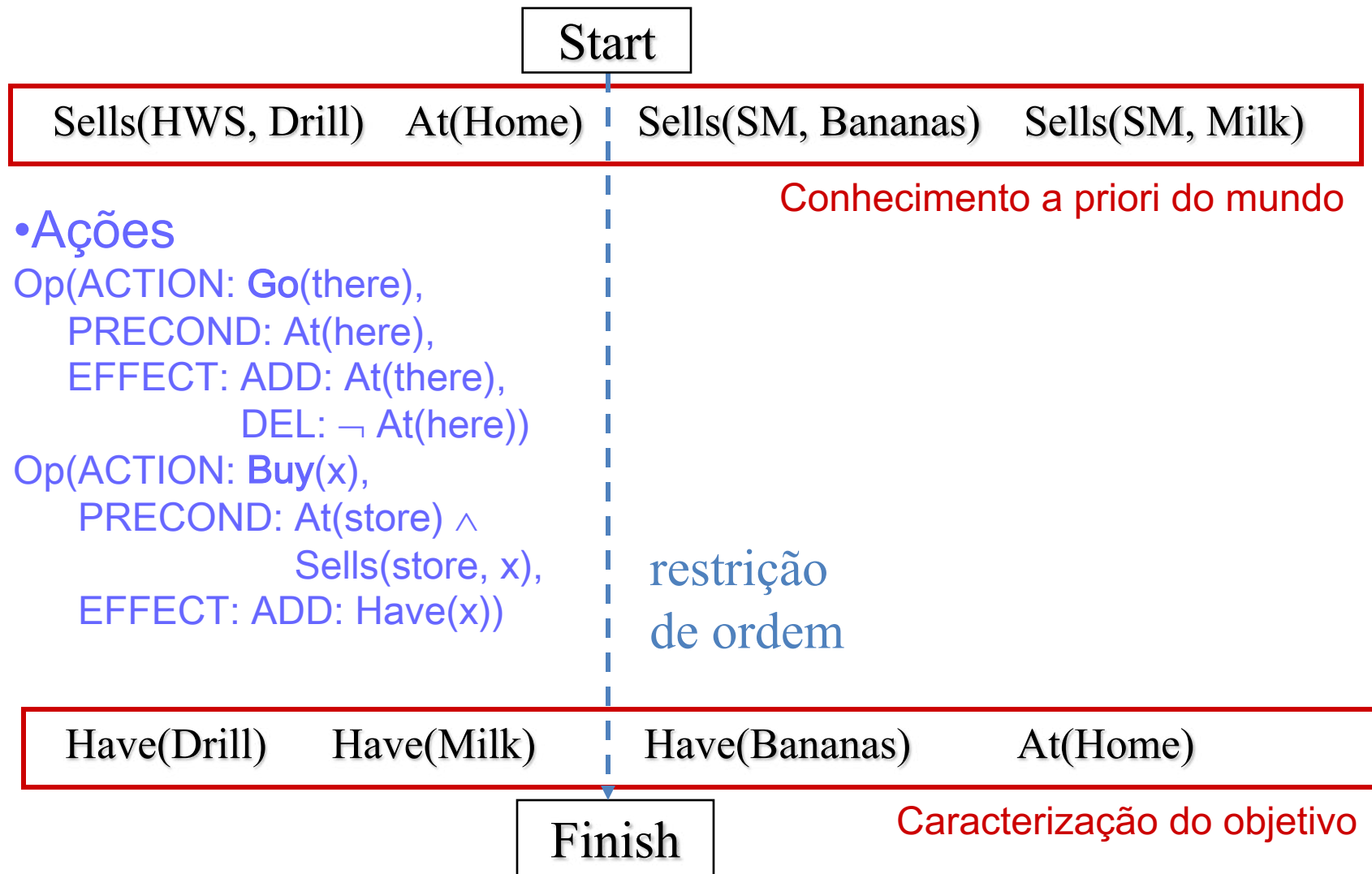
POP: Solução do Problema da Ameaça

Demoção: adiciona uma restrição de ordem, forçando S3 a ocorrer antes de S1

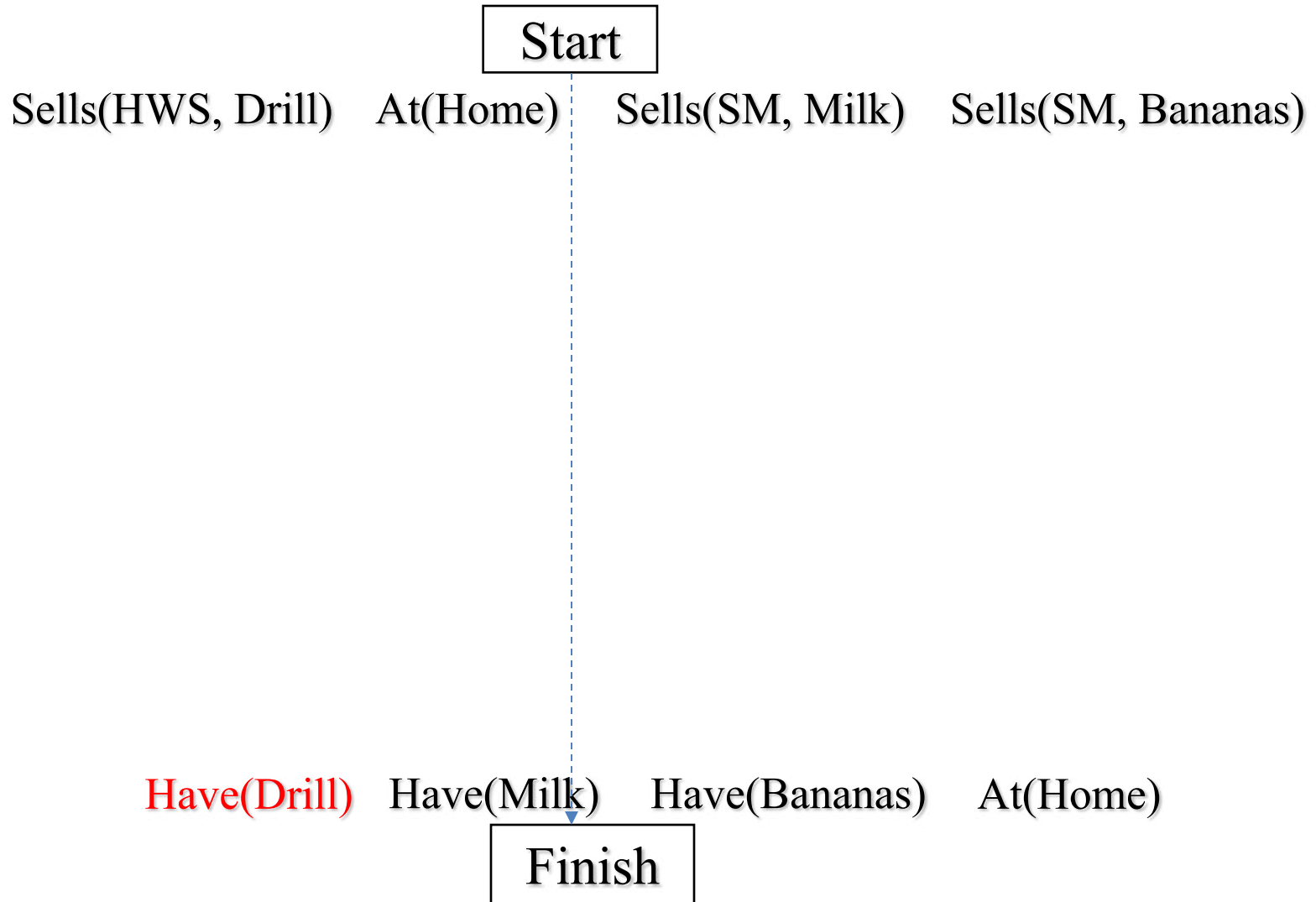


Promoção: adiciona uma restrição de ordem, forçando S3 a ocorrer depois de S2

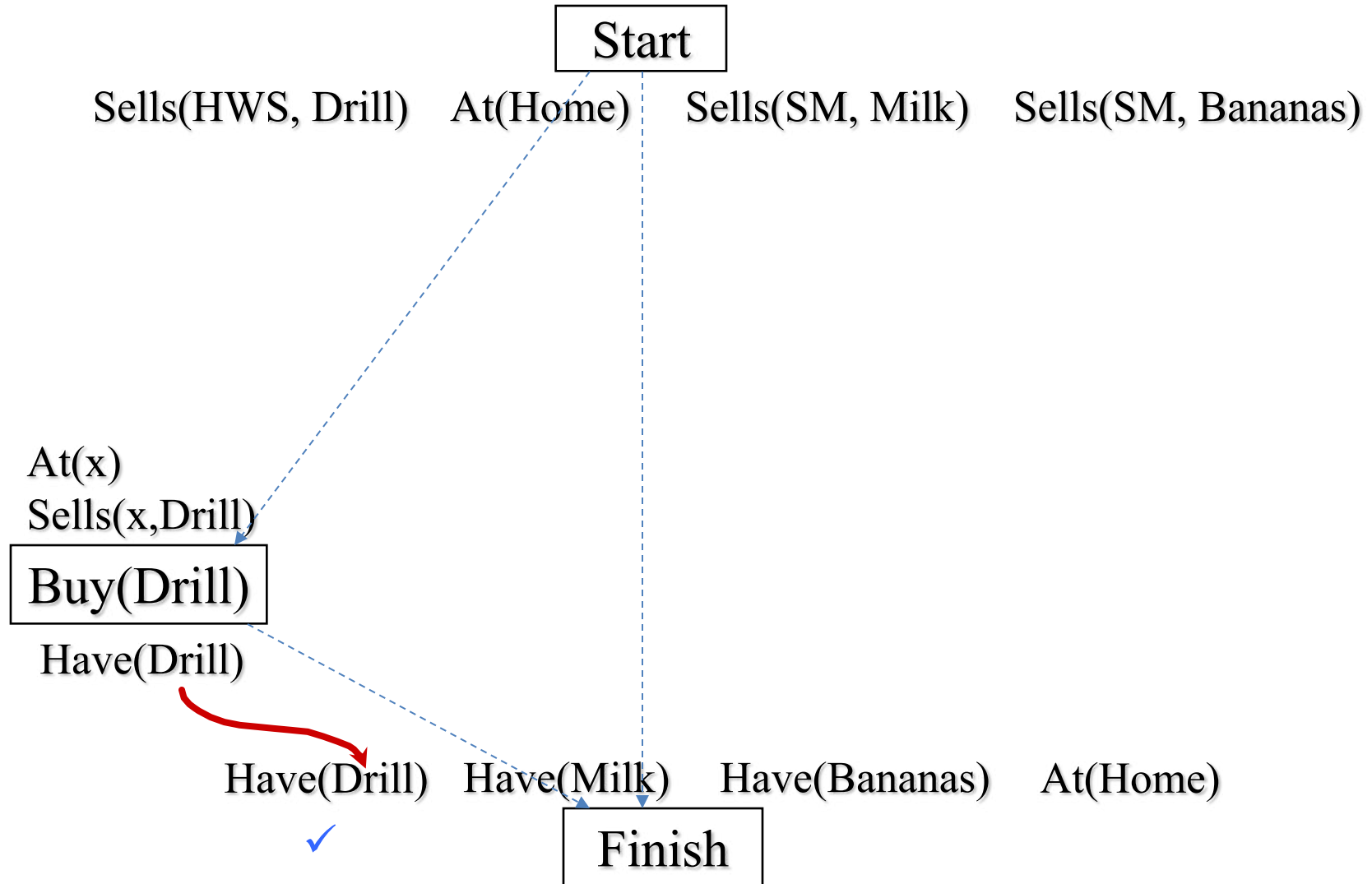
POP: Exemplo das Compras



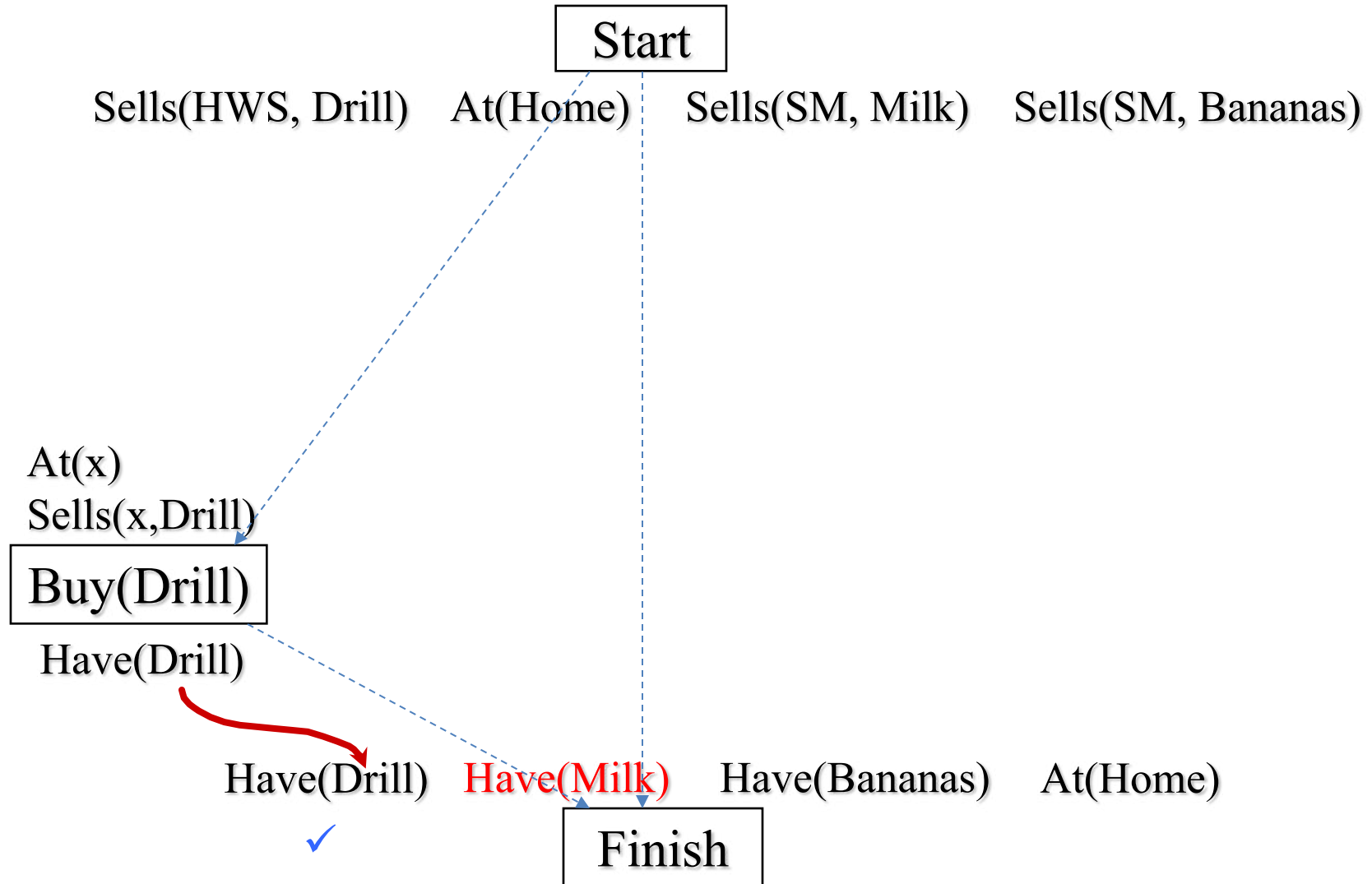
POP: Exemplo das Compras



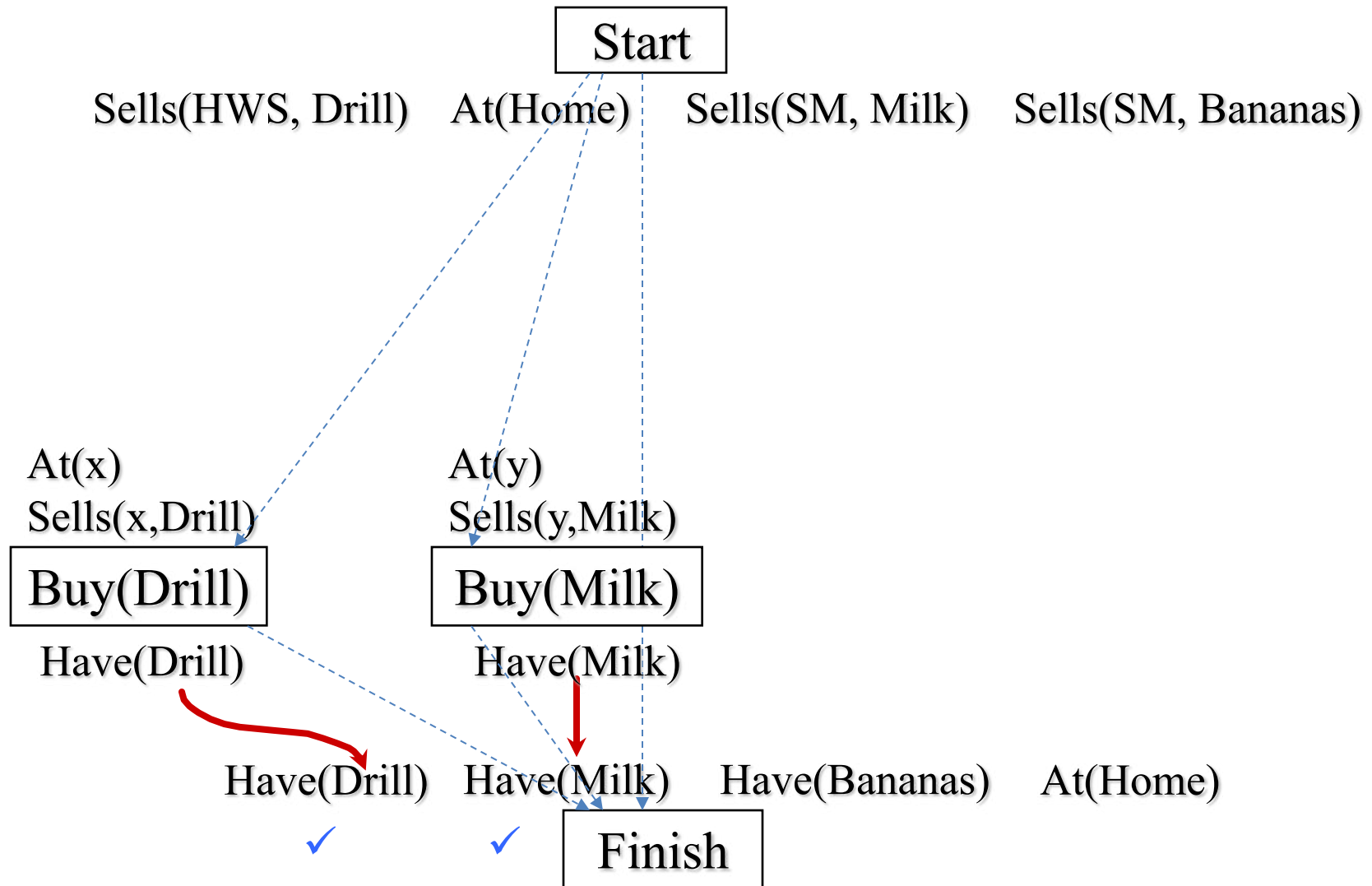
POP: Exemplo das Compras



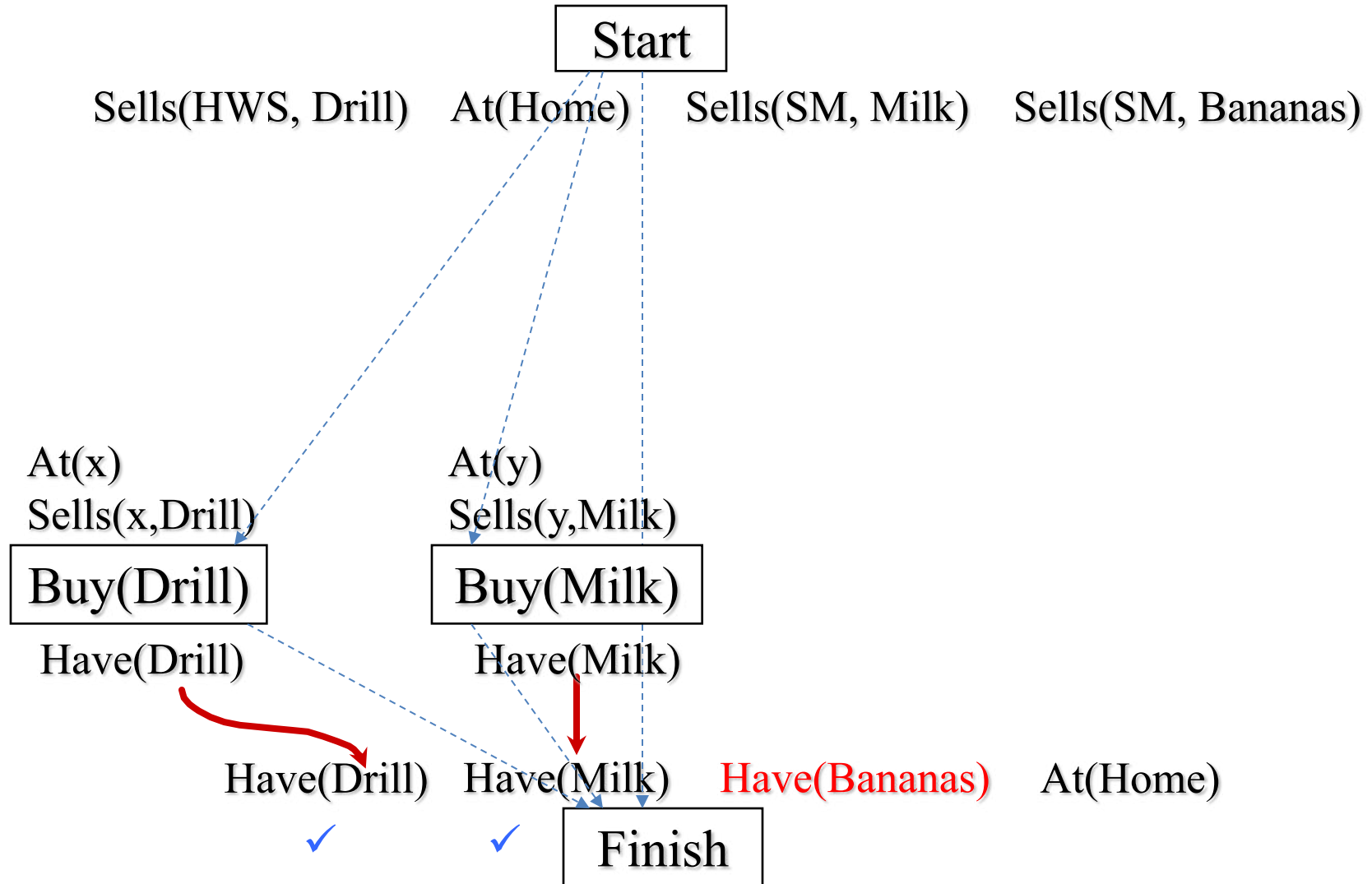
POP: Exemplo das Compras



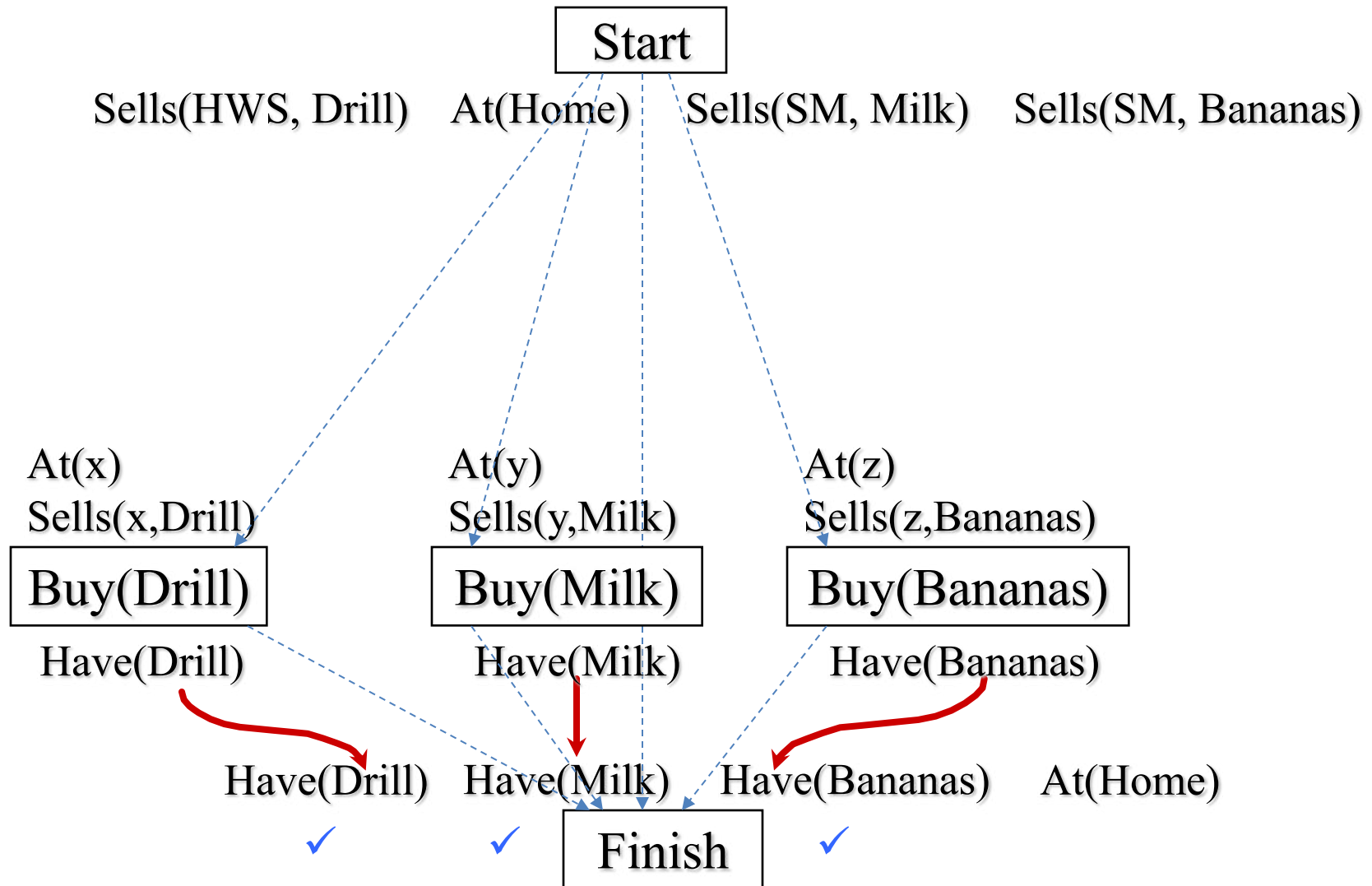
POP: Exemplo das Compras



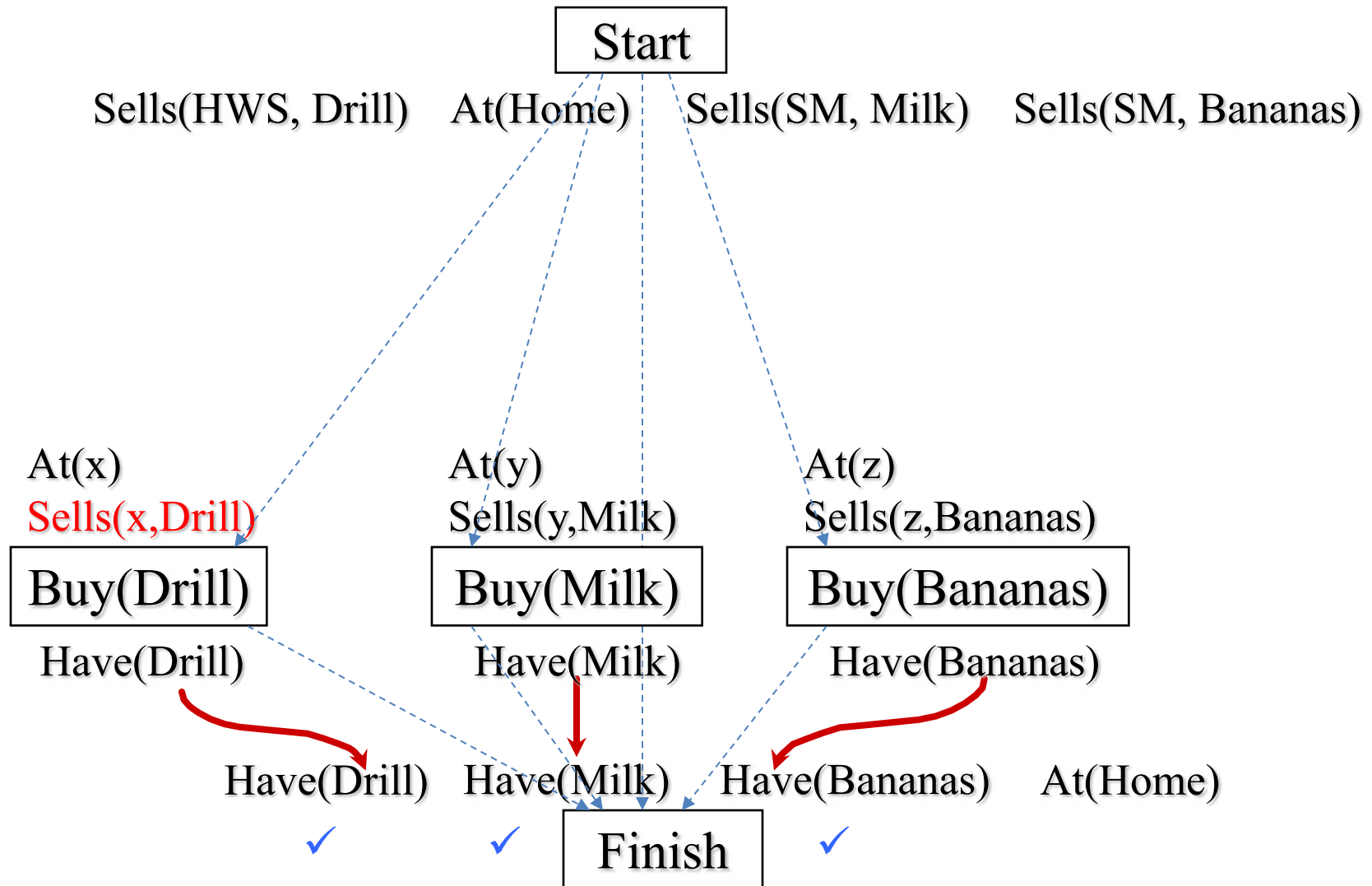
POP: Exemplo das Compras



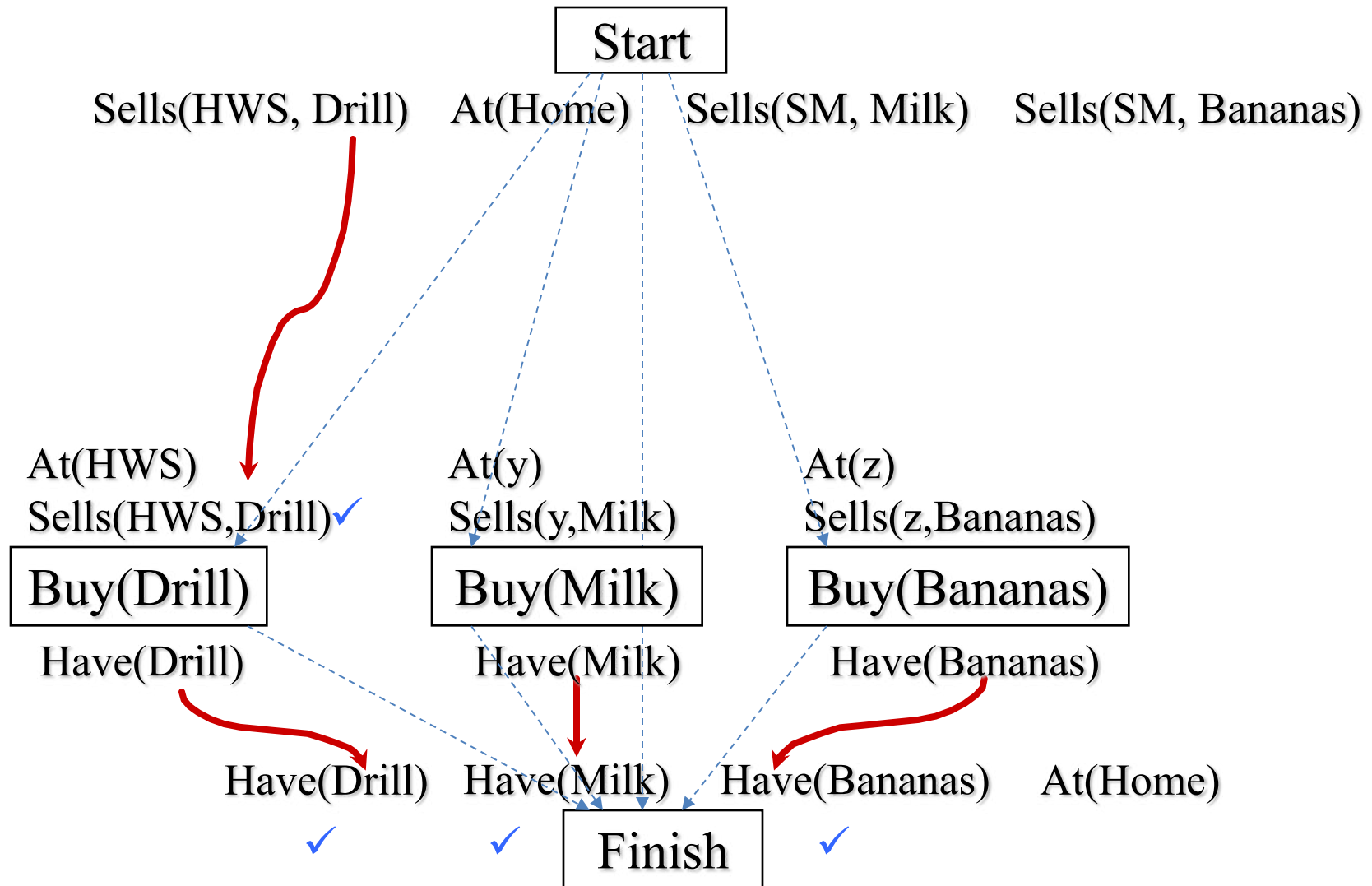
POP: Exemplo das Compras



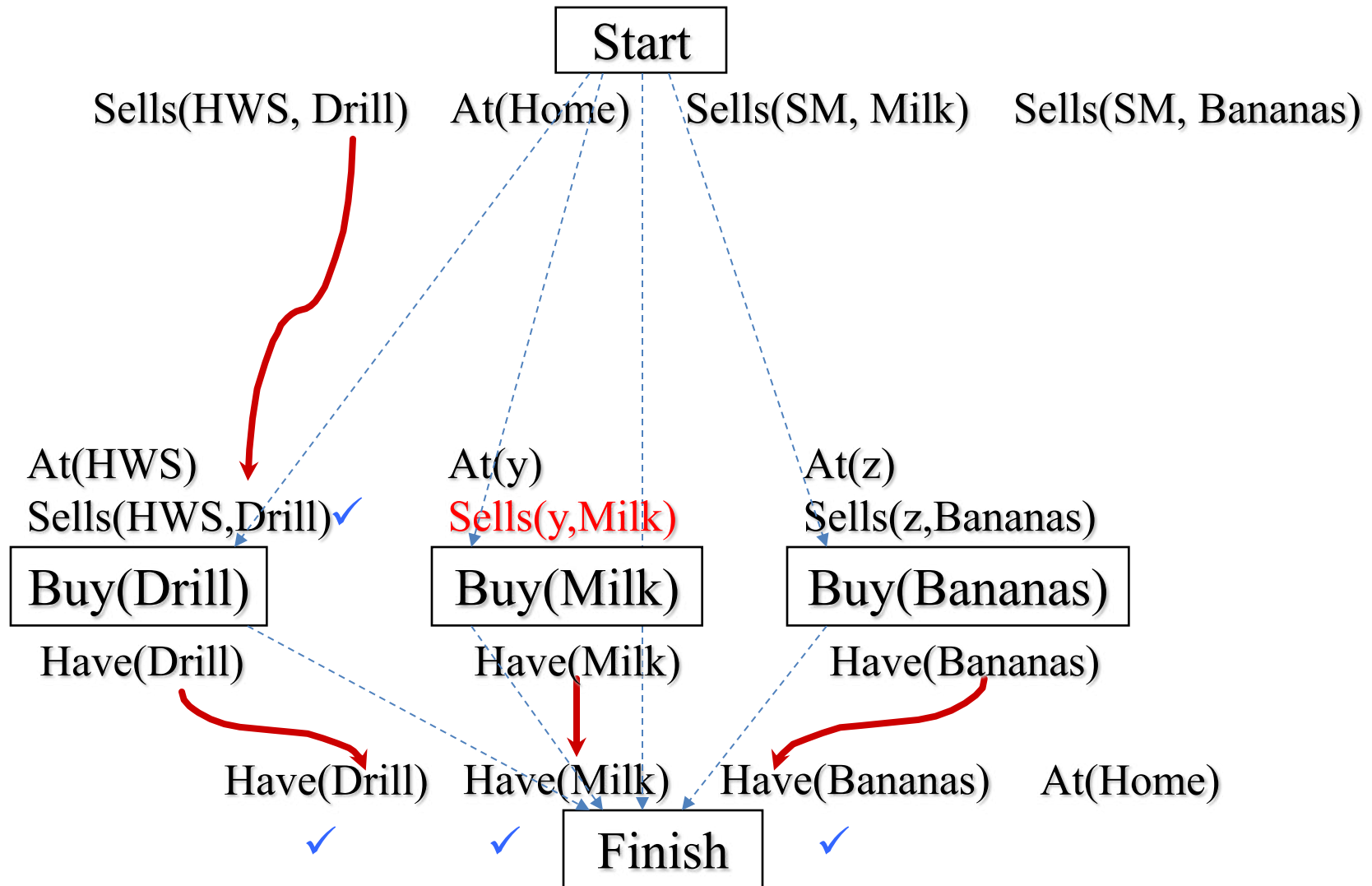
POP: Exemplo das Compras



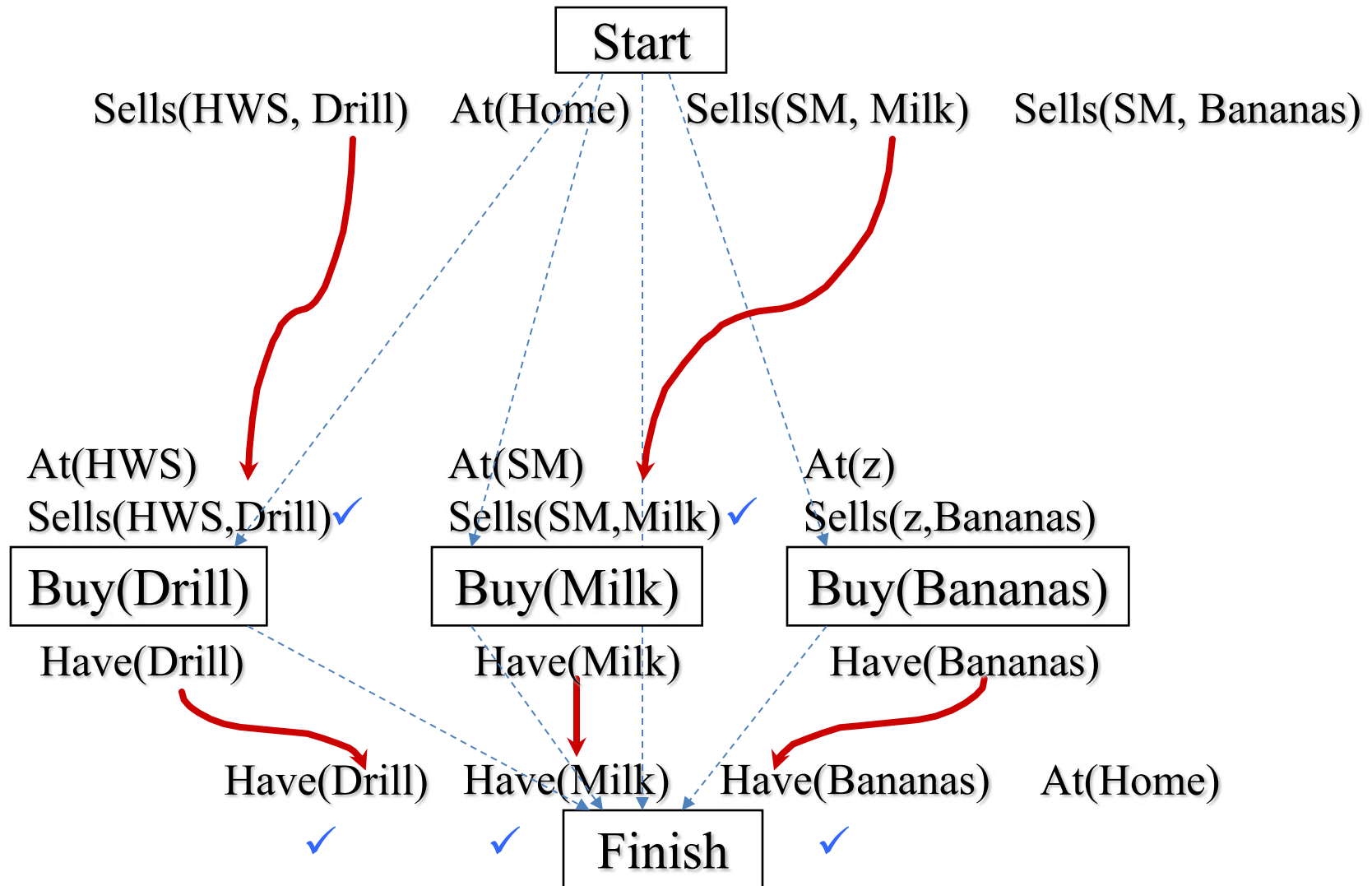
POP: Exemplo das Compras



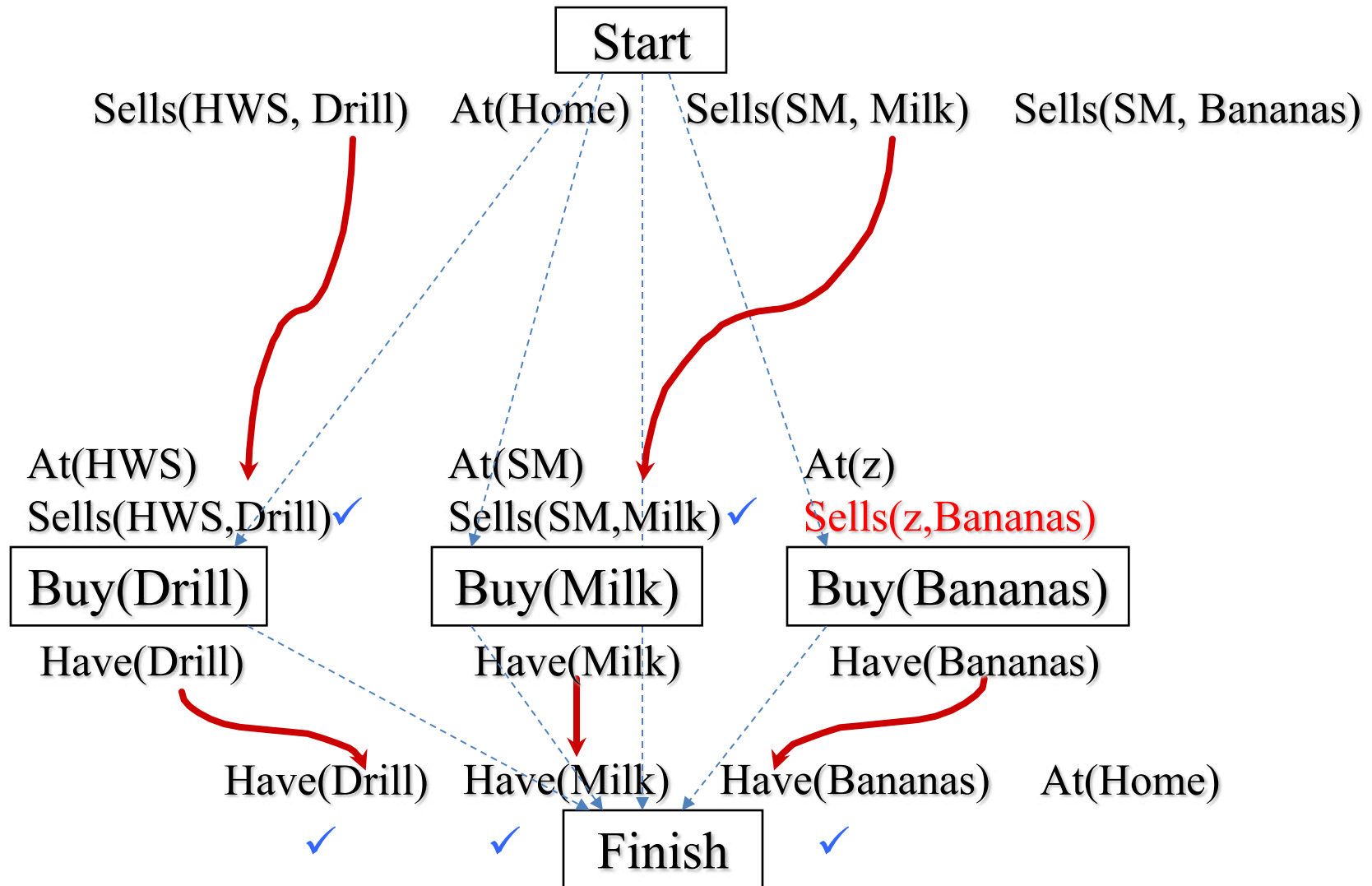
POP: Exemplo das Compras



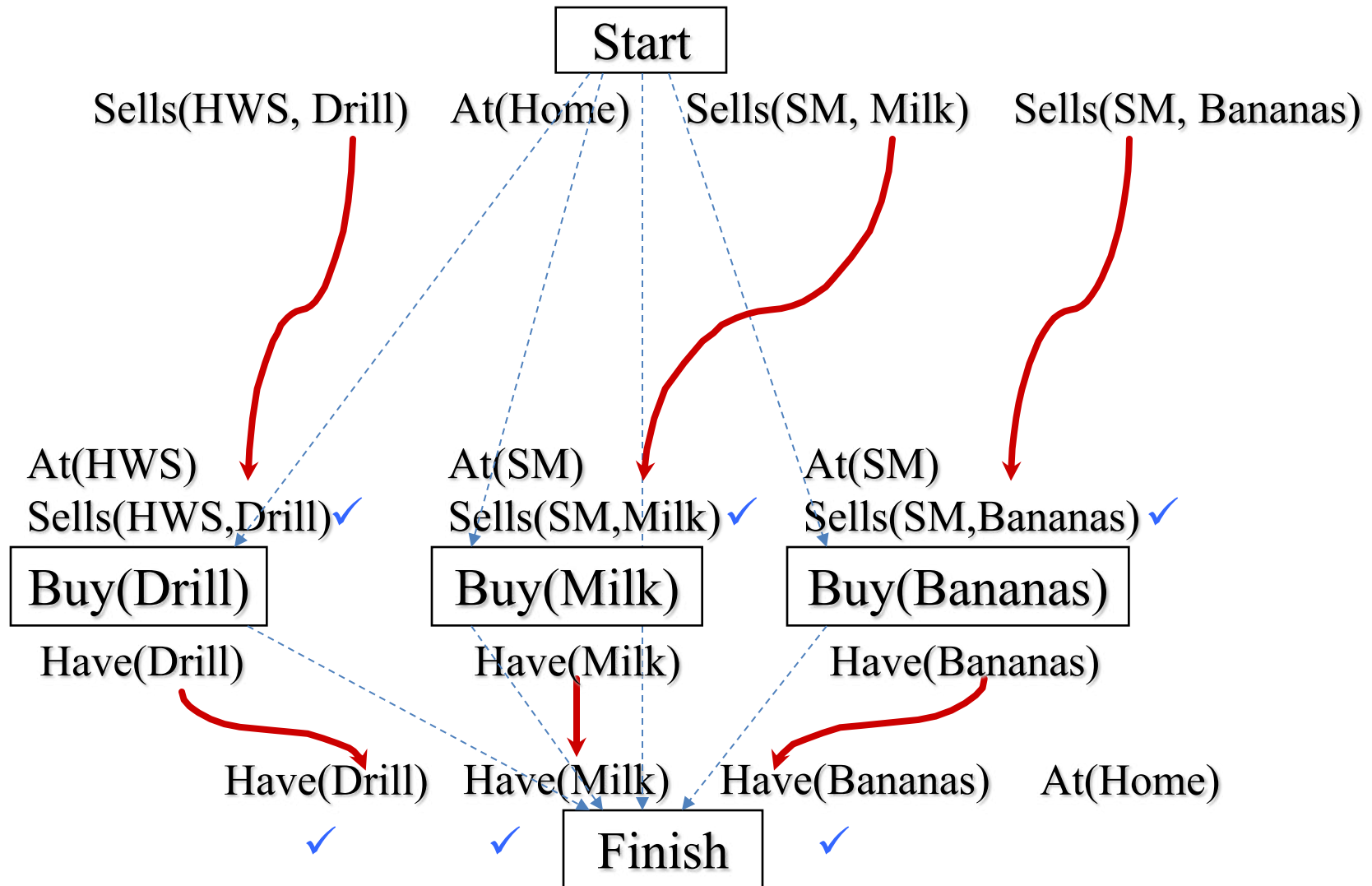
POP: Exemplo das Compras



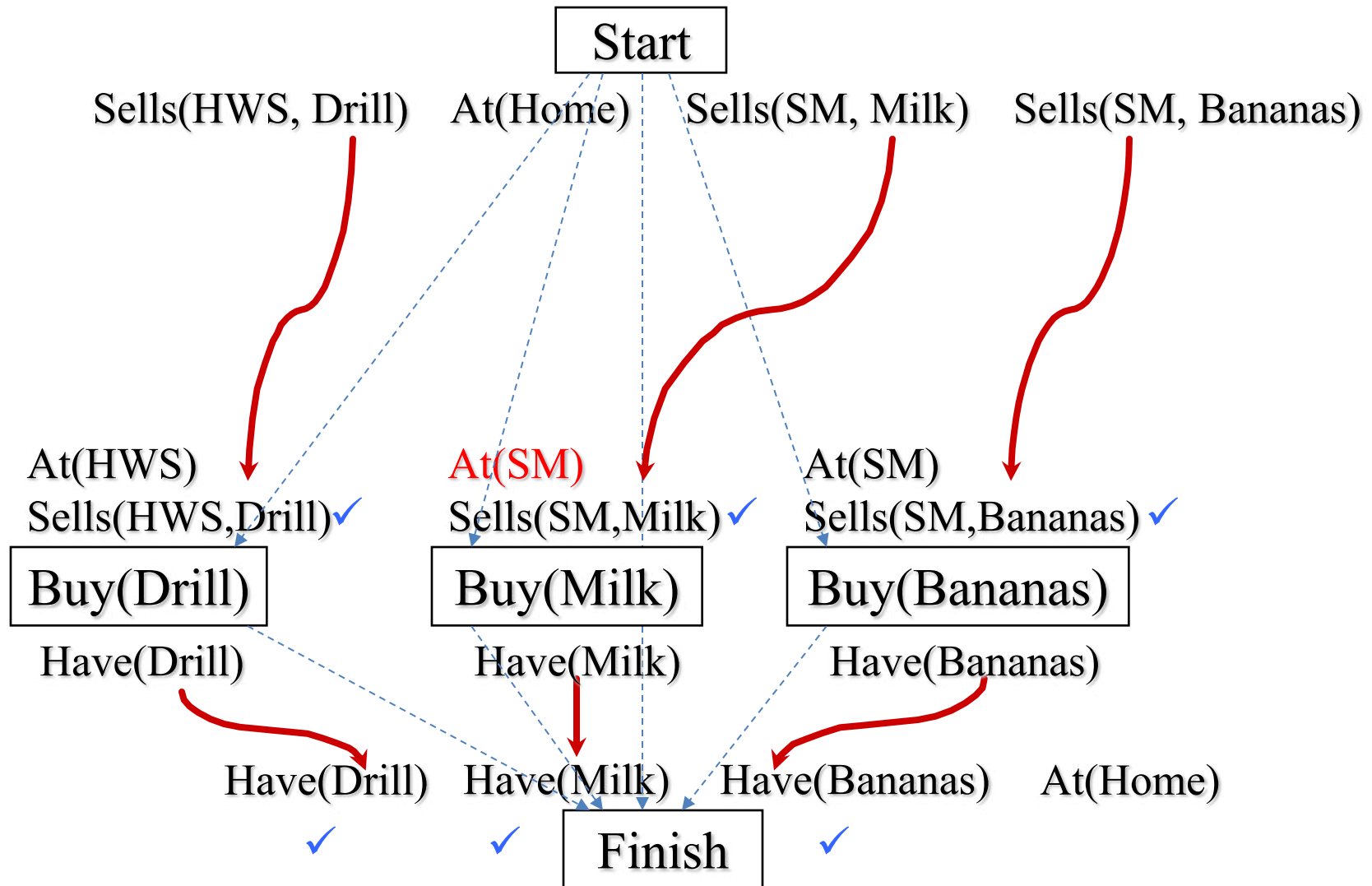
POP: Exemplo das Compras



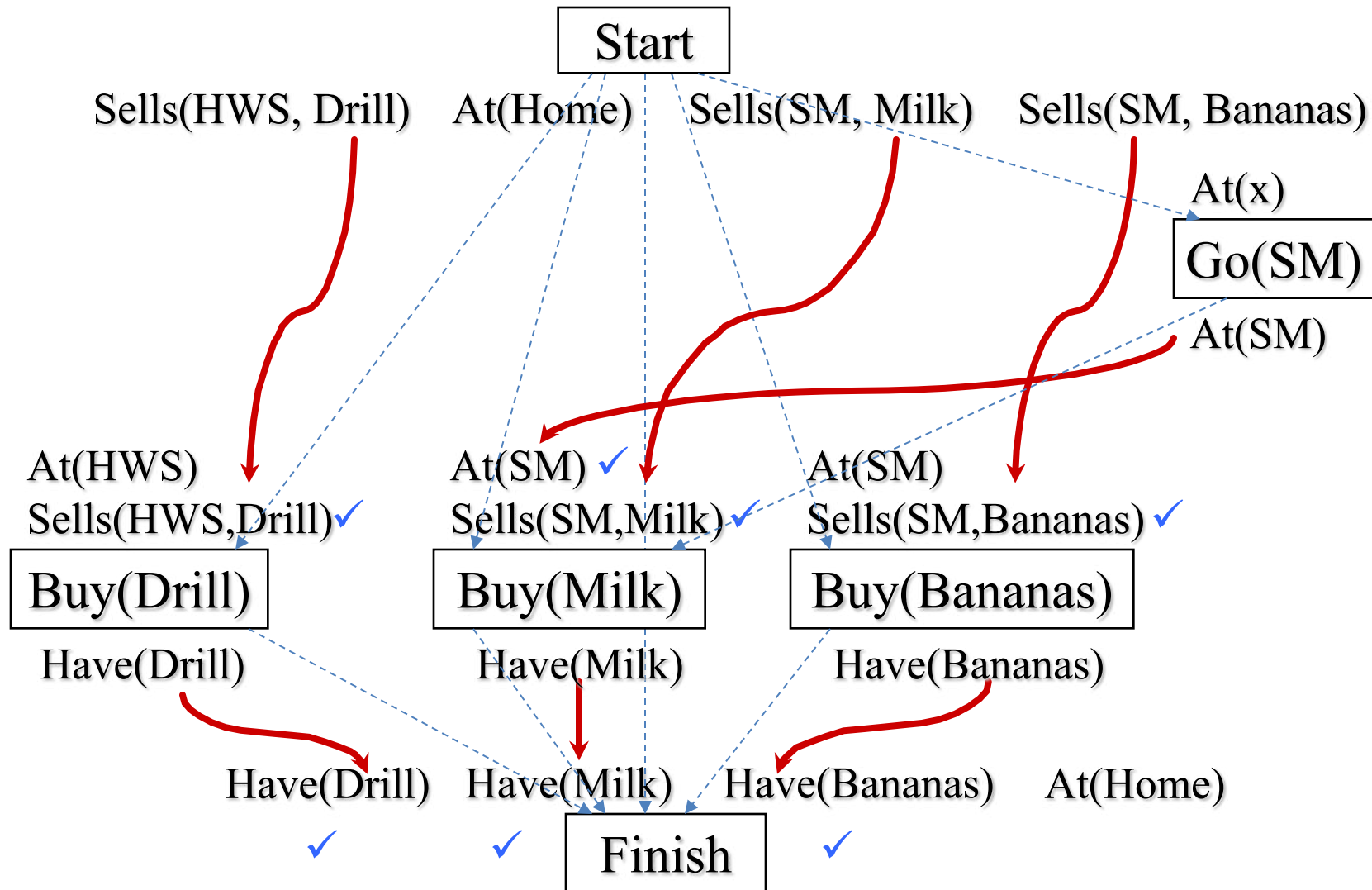
POP: Exemplo das Compras



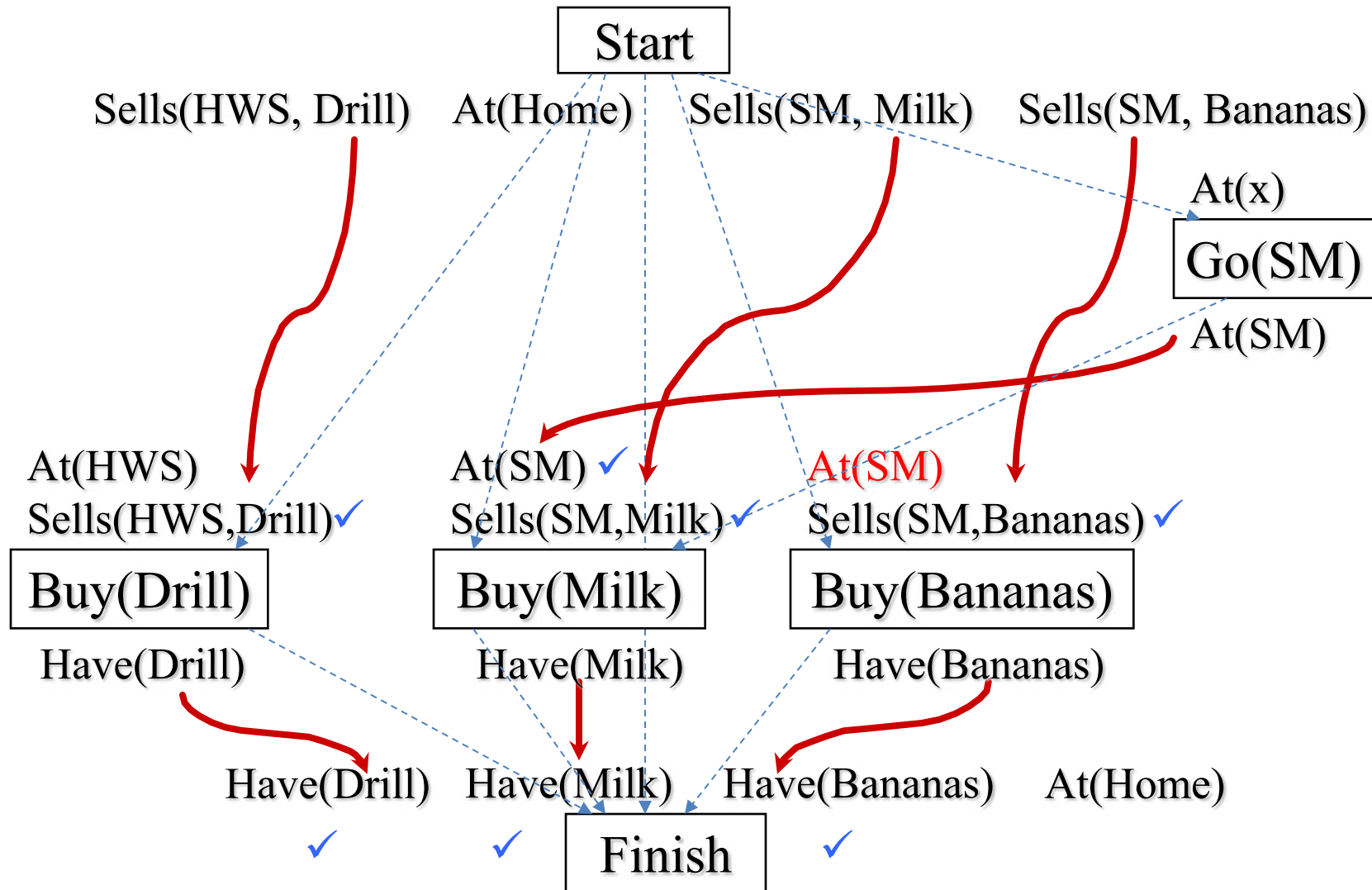
POP: Exemplo das Compras



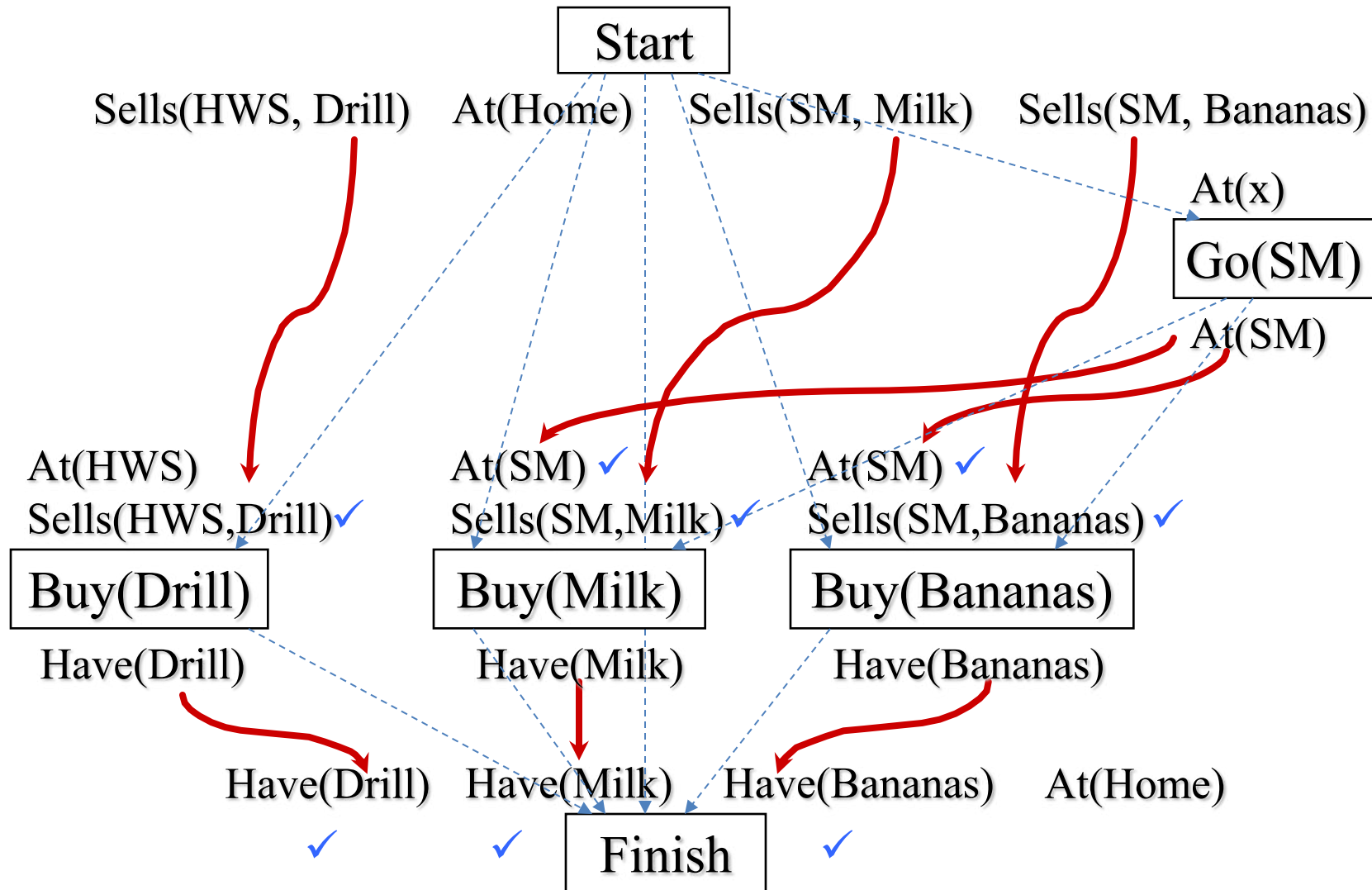
POP: Exemplo das Compras



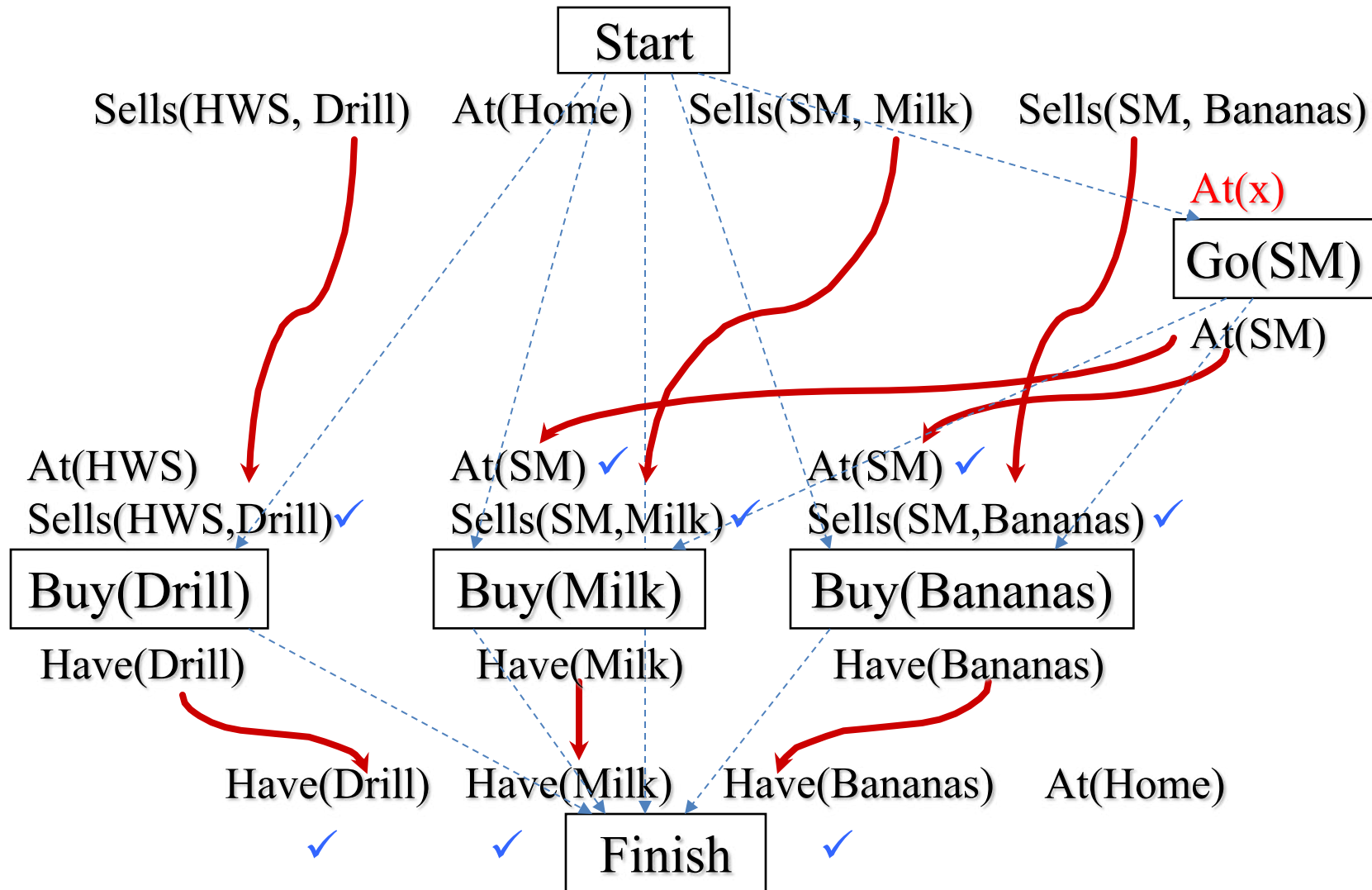
POP: Exemplo das Compras



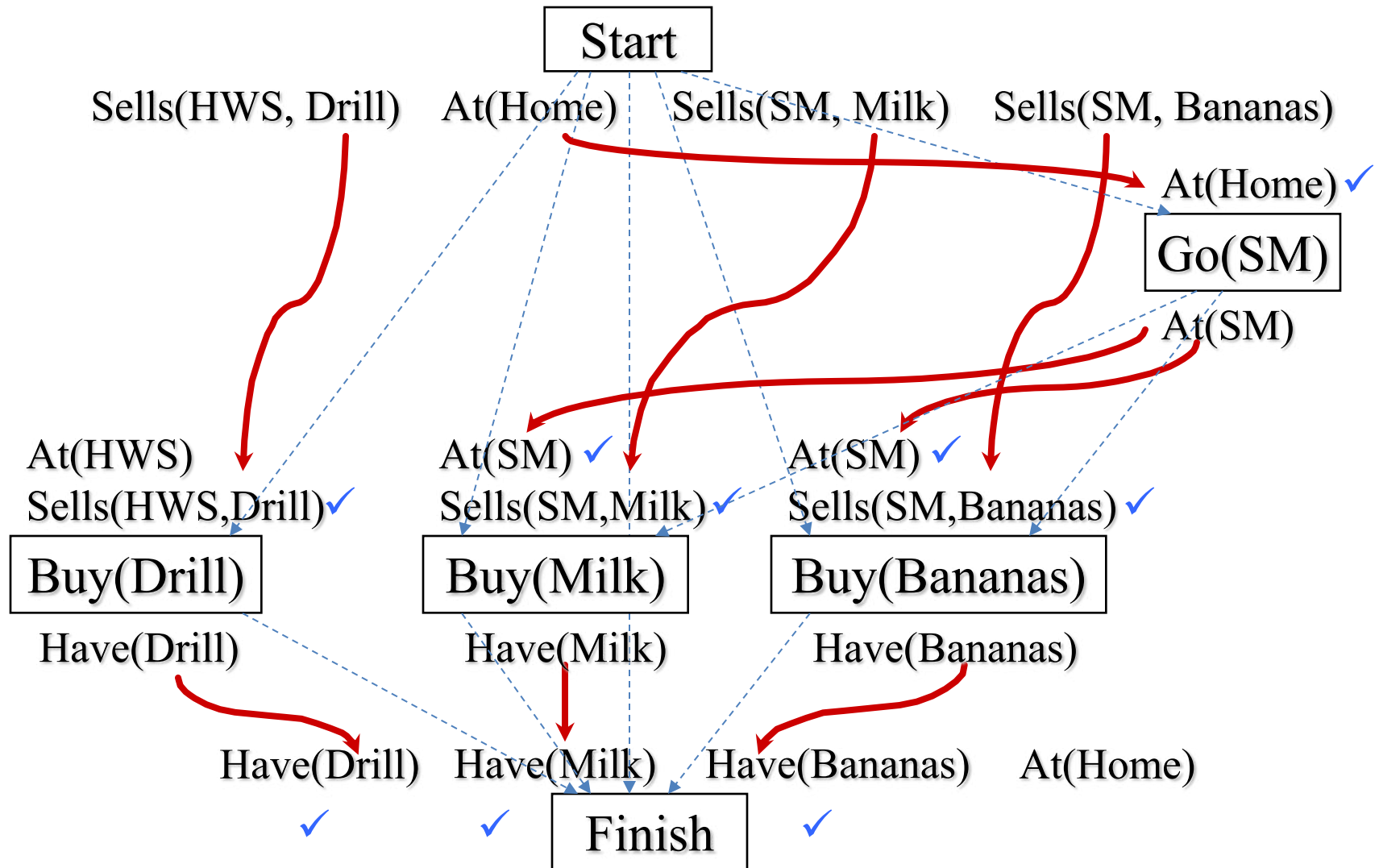
POP: Exemplo das Compras



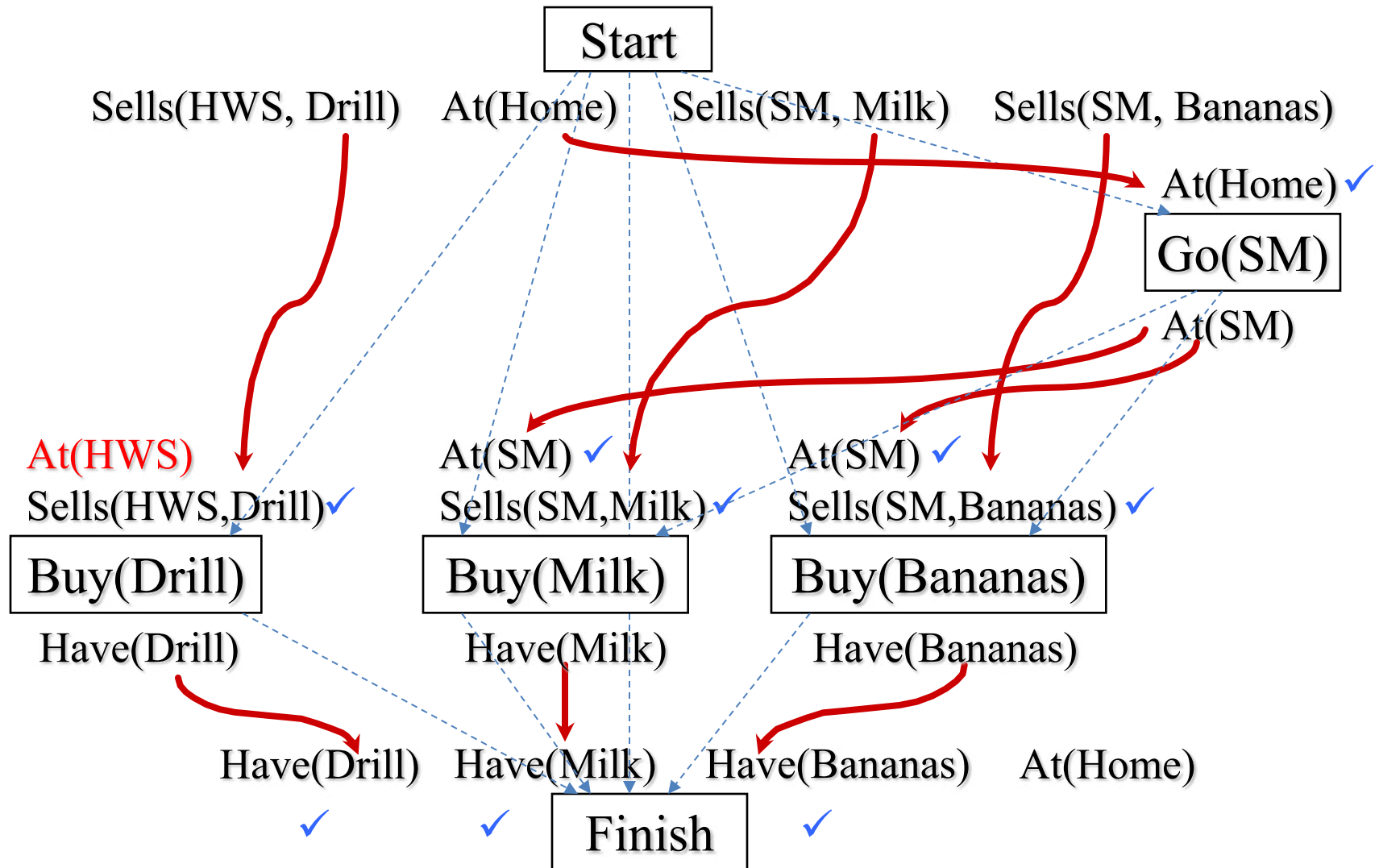
POP: Exemplo das Compras



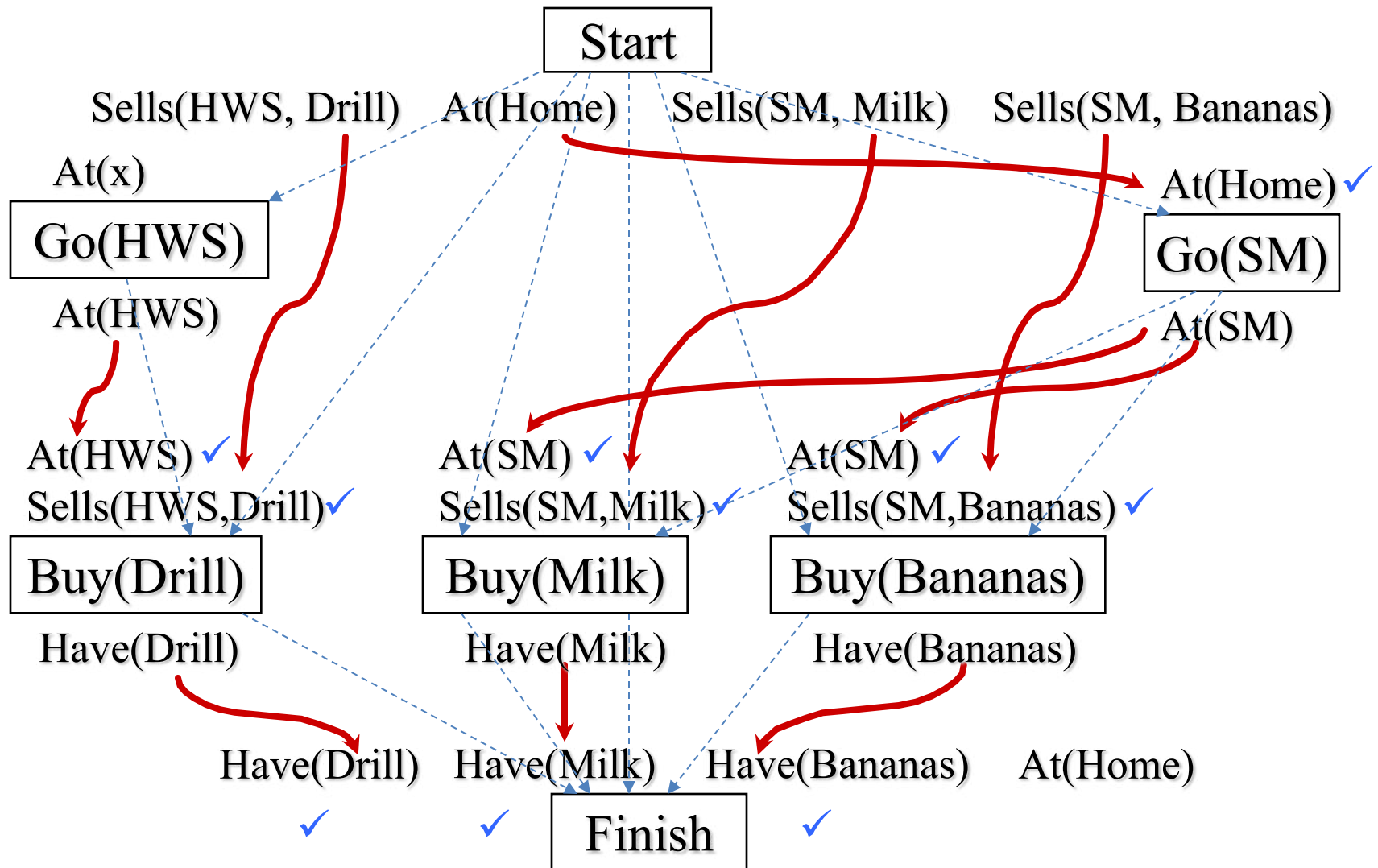
POP: Exemplo das Compras



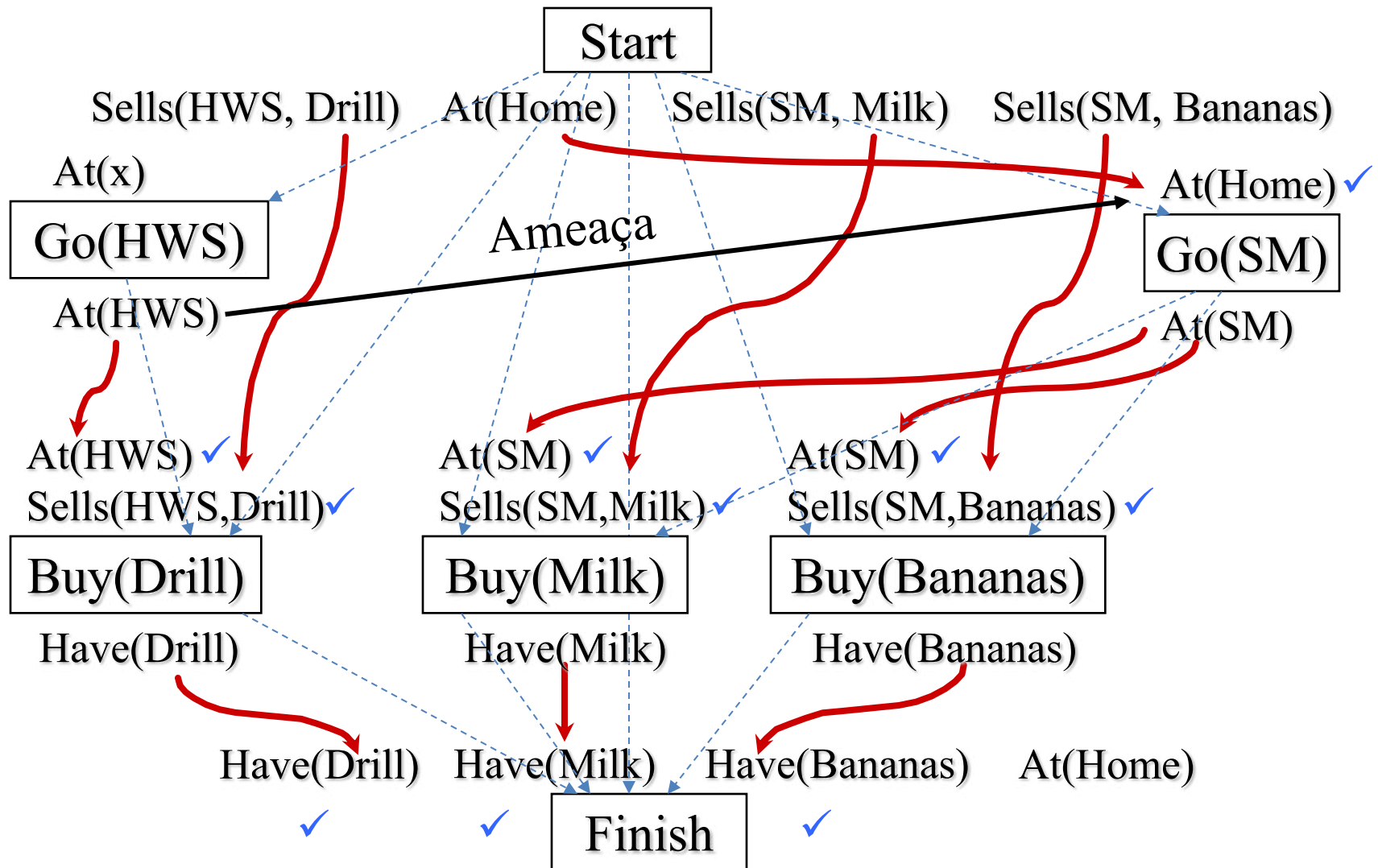
POP: Exemplo das Compras



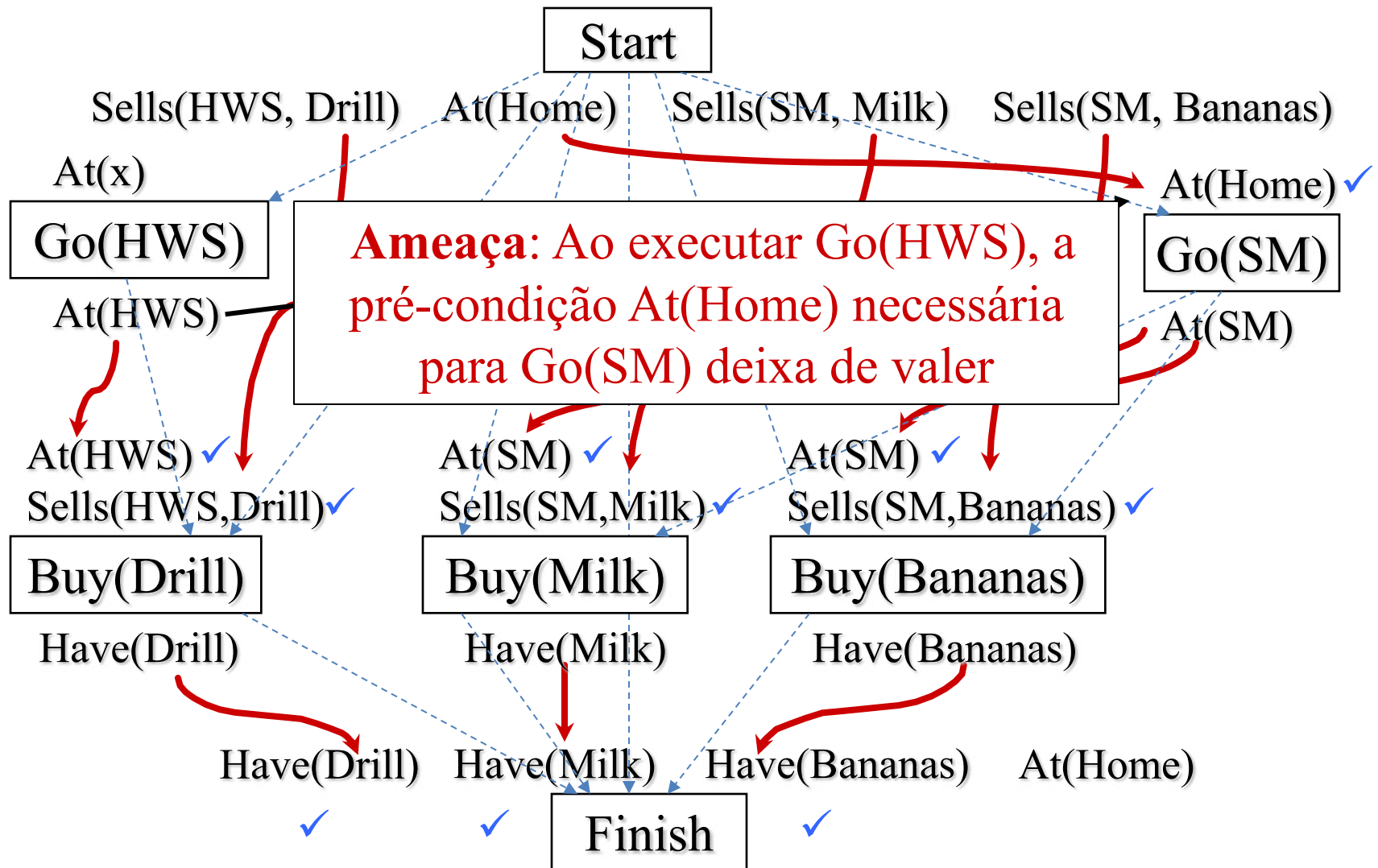
POP: Exemplo das Compras



POP: Exemplo das Compras



POP: Exemplo das Compras



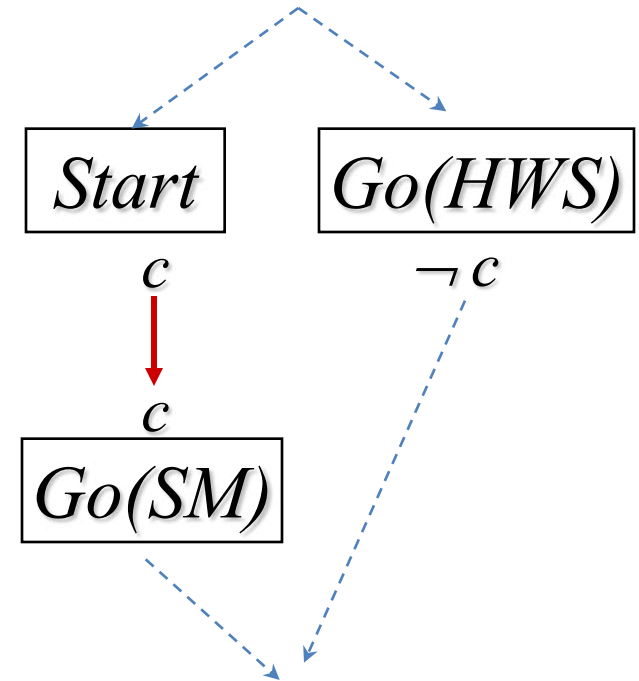
POP: Problema da Ameaça

- Ameaça
- Solução:
 - Demoção

Go(HWS) ocorre antes de Start →
não pode! Nada pode vir antes de Start!

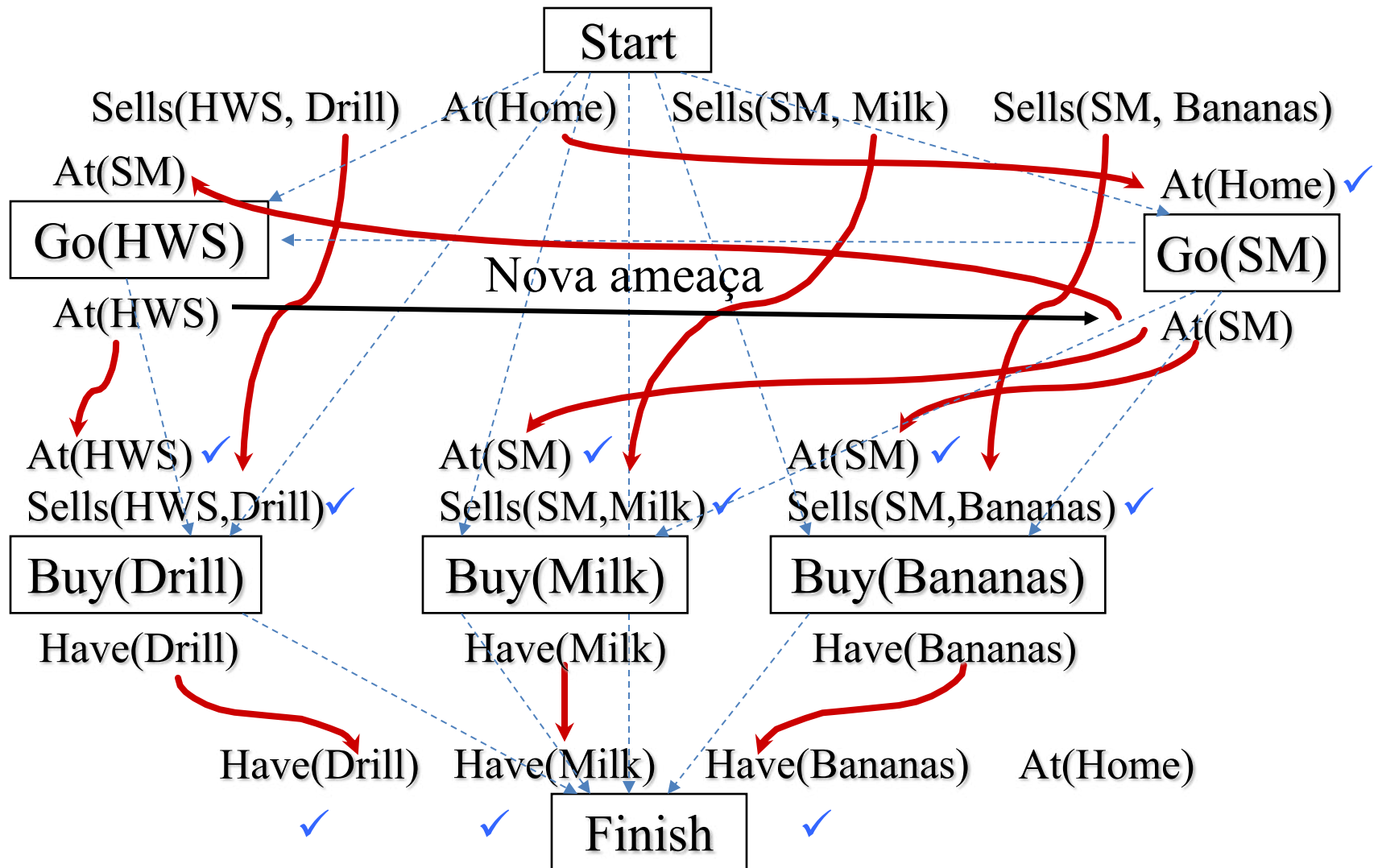
- Promoção

Go(HWS) ocorre depois de Go(SM)
→ insere uma relação de ordem e
link causal entre Go(HWS) e
Go(SM)

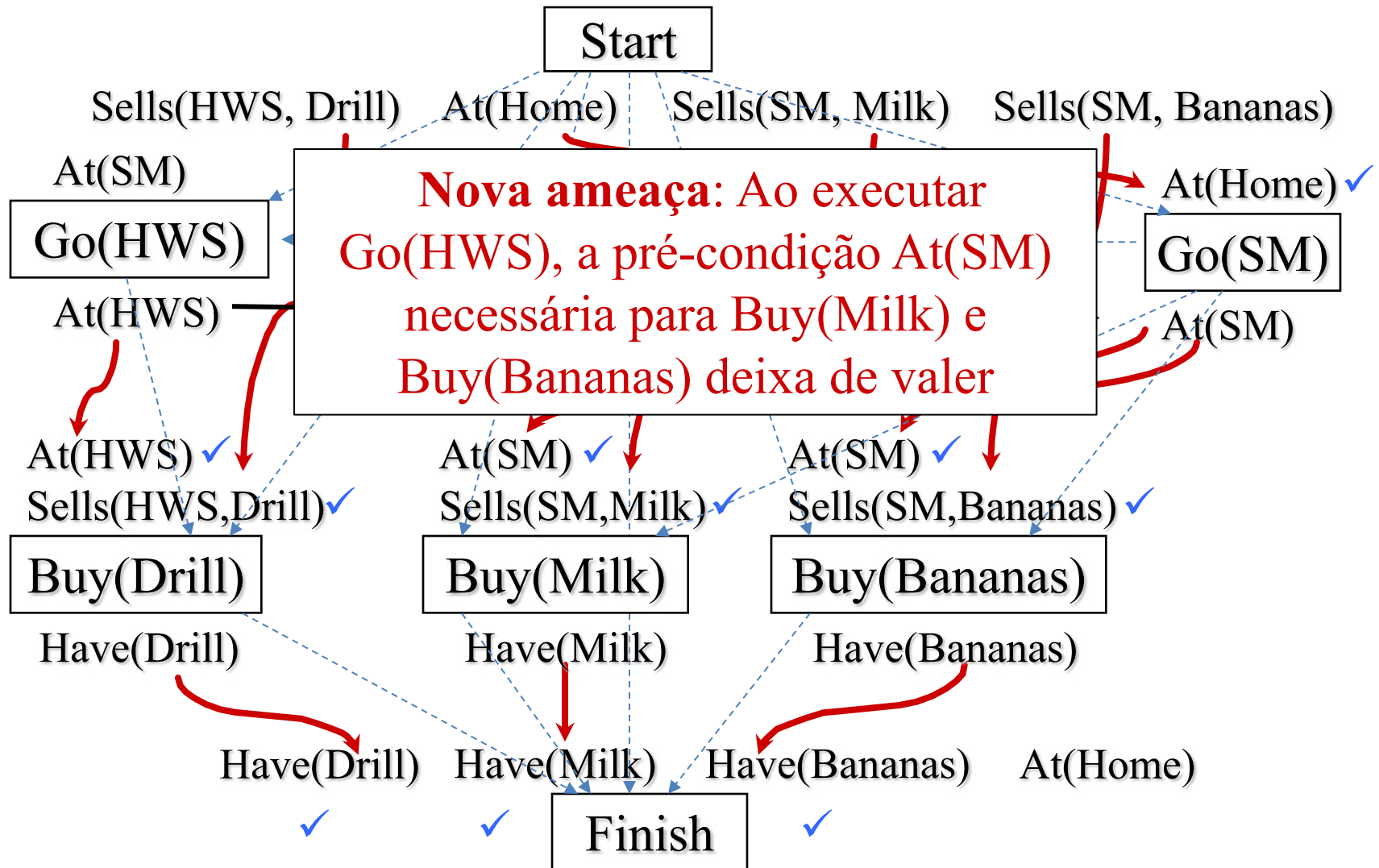


S3 ameaça a condição c estabelecida por S1 e protegida pelo link causal S1 para S2

POP: Exemplo das Compras



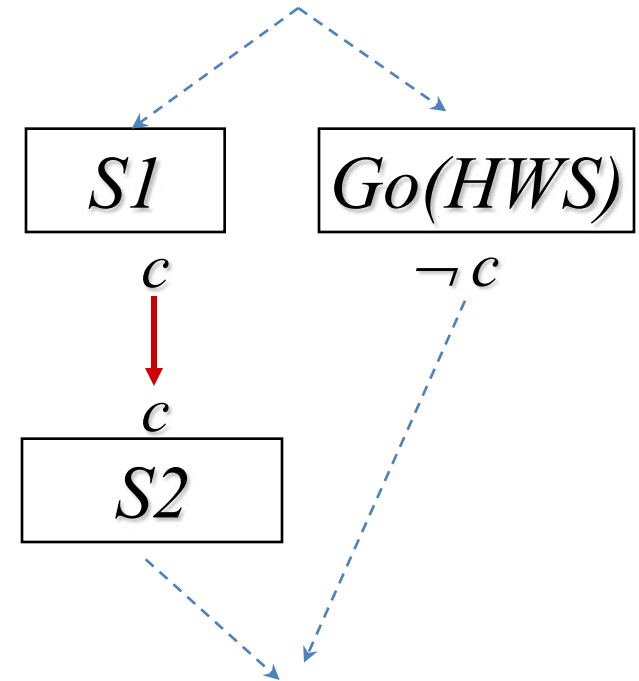
POP: Exemplo das Compras



POP: Problema da Ameaça

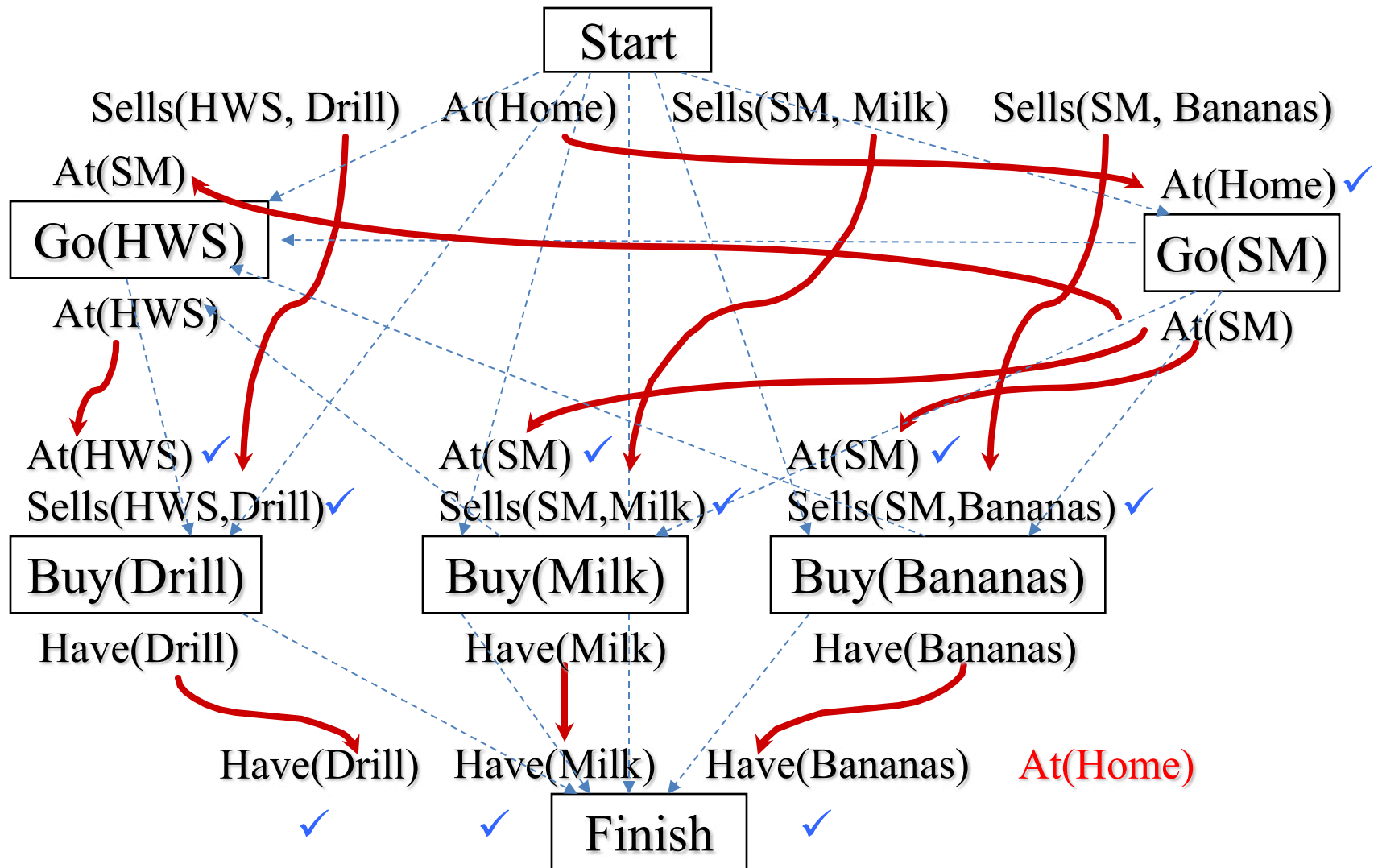
- Ameaça
 - Solução:
 - Demoção
- Não pode!**
- Promoção

Go(HWS) ocorre depois de S2
(Buy(Milk) e Buy(Bananas)) → insere
uma relação de ordem entre Go(HWS)
e (Buy(Milk) e Buy(Bananas))

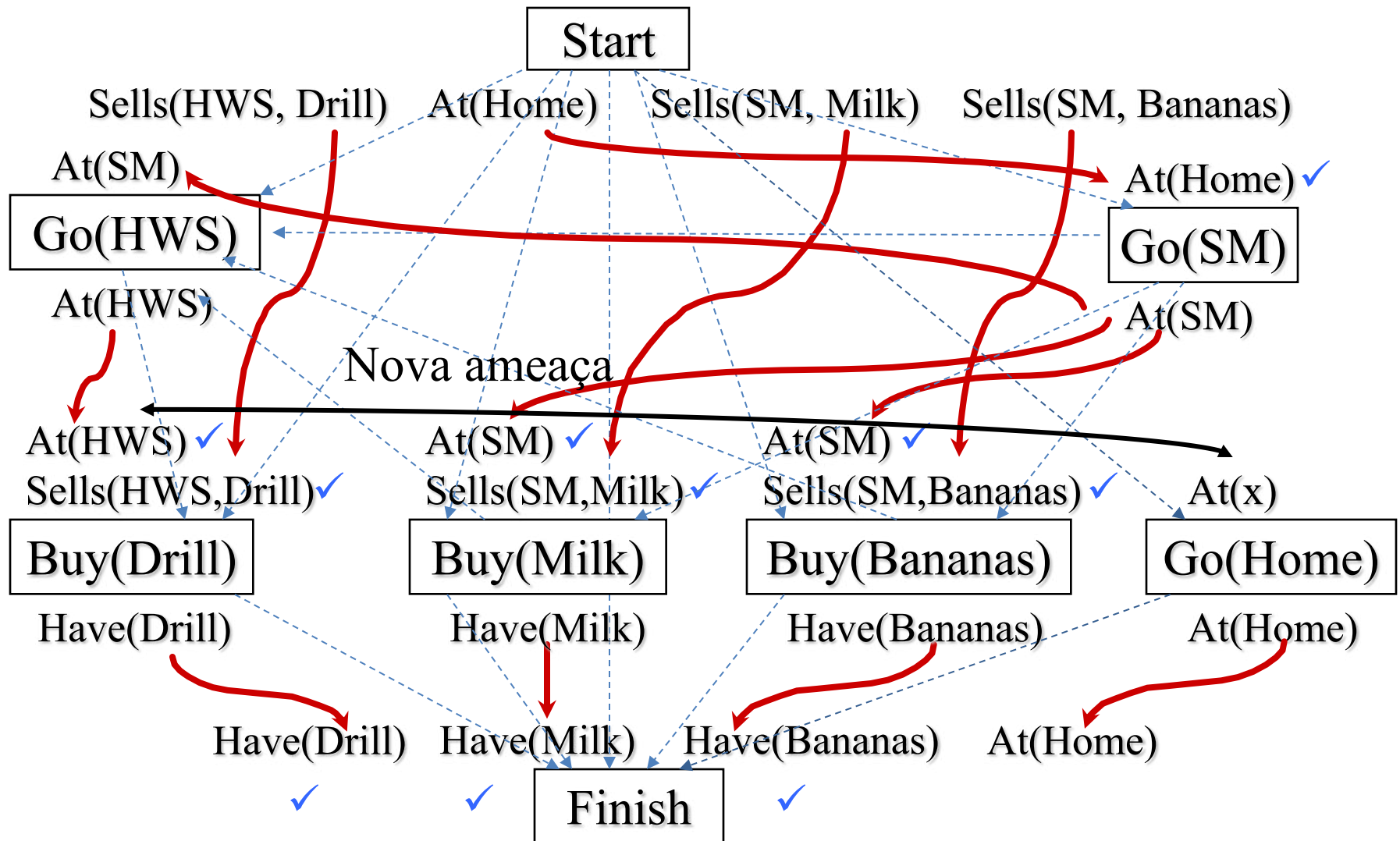


S3 ameaça a condição c estabelecida por S1 e protegida pelo link causal S1 para S2

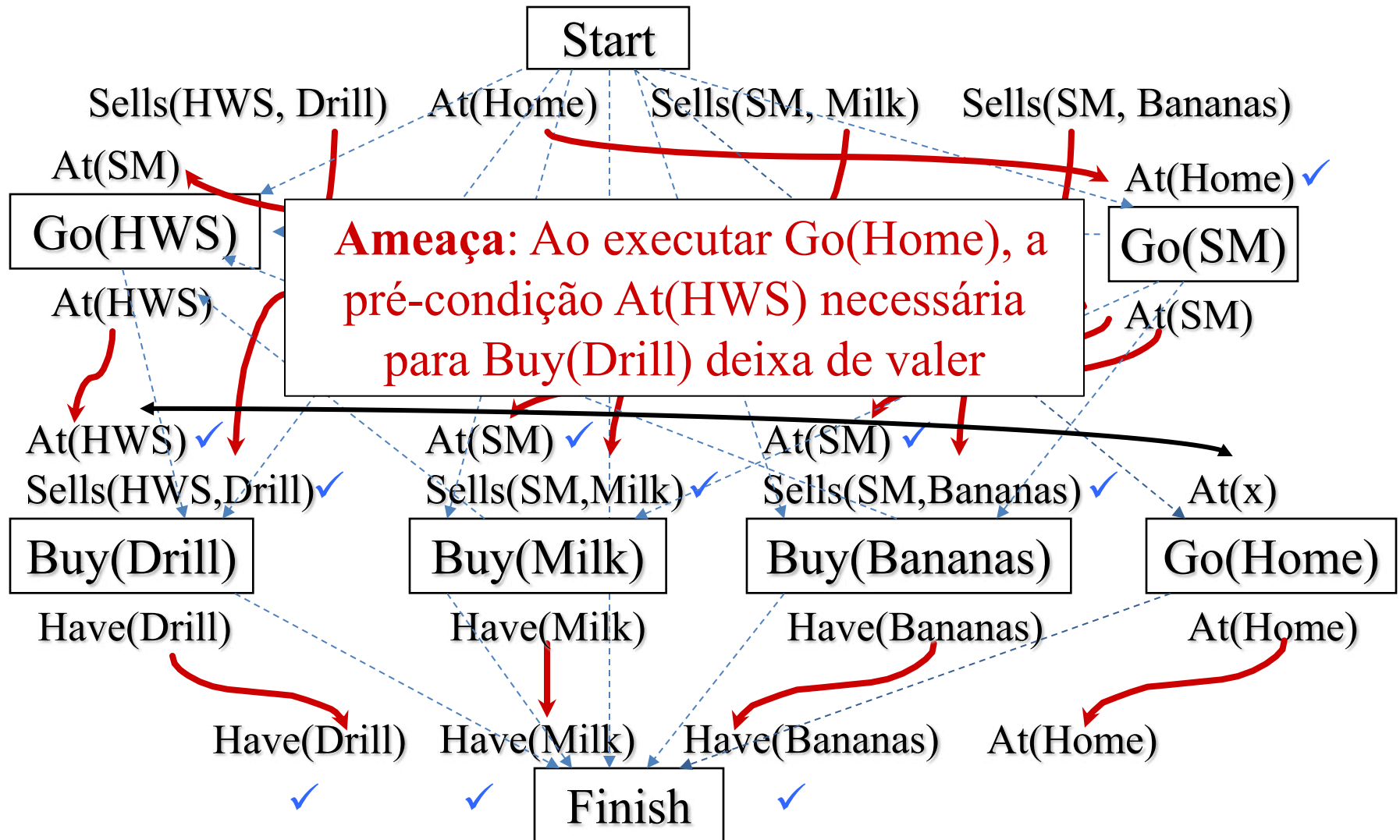
POP: Exemplo das Compras



POP: Exemplo das Compras



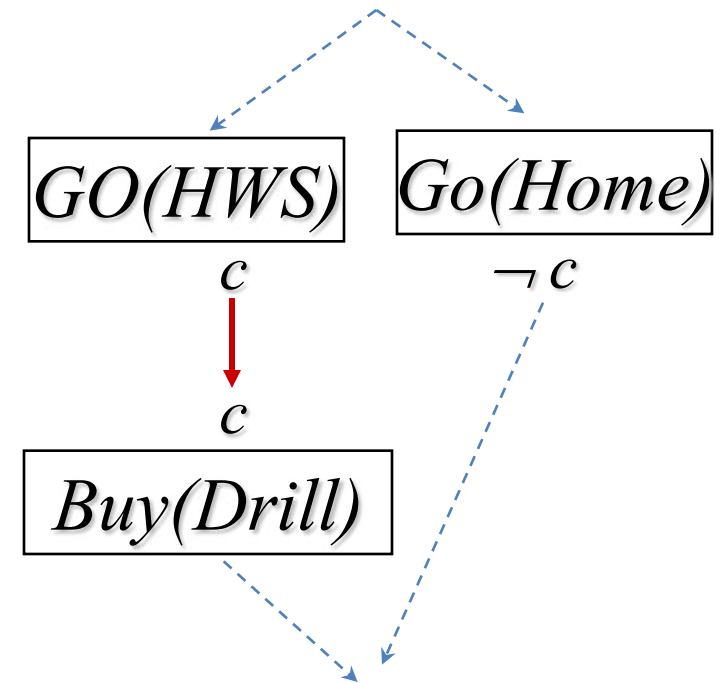
POP: Exemplo das Compras



POP: Problema da Ameaça

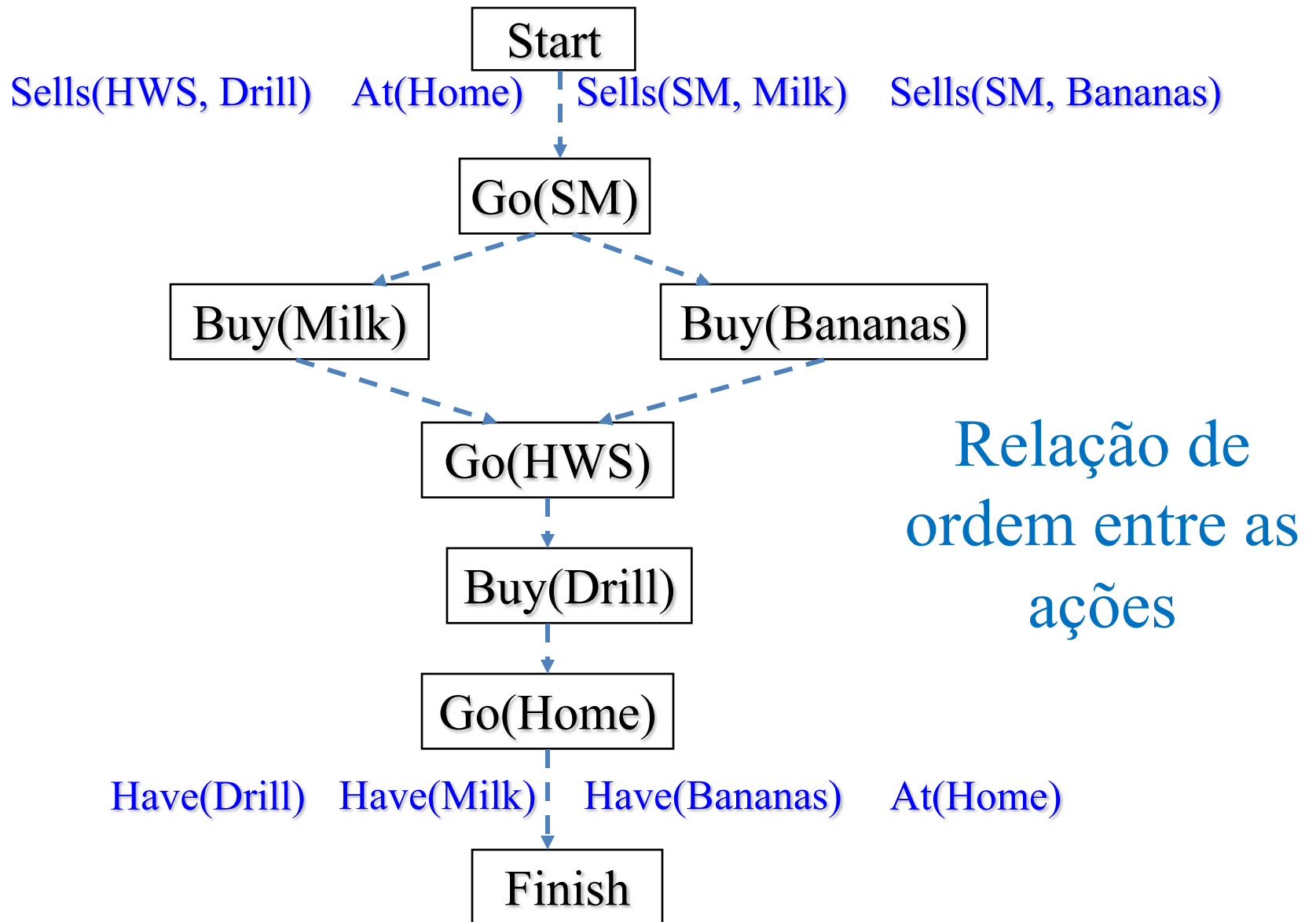
- Ameaça
 - Solução:
 - Demoção
- Não pode!**
- Promoção

Go(Home) ocorre depois de
Buy(Drill) → insere uma relação de
ordem entre Go(Home) e Buy(Drill)

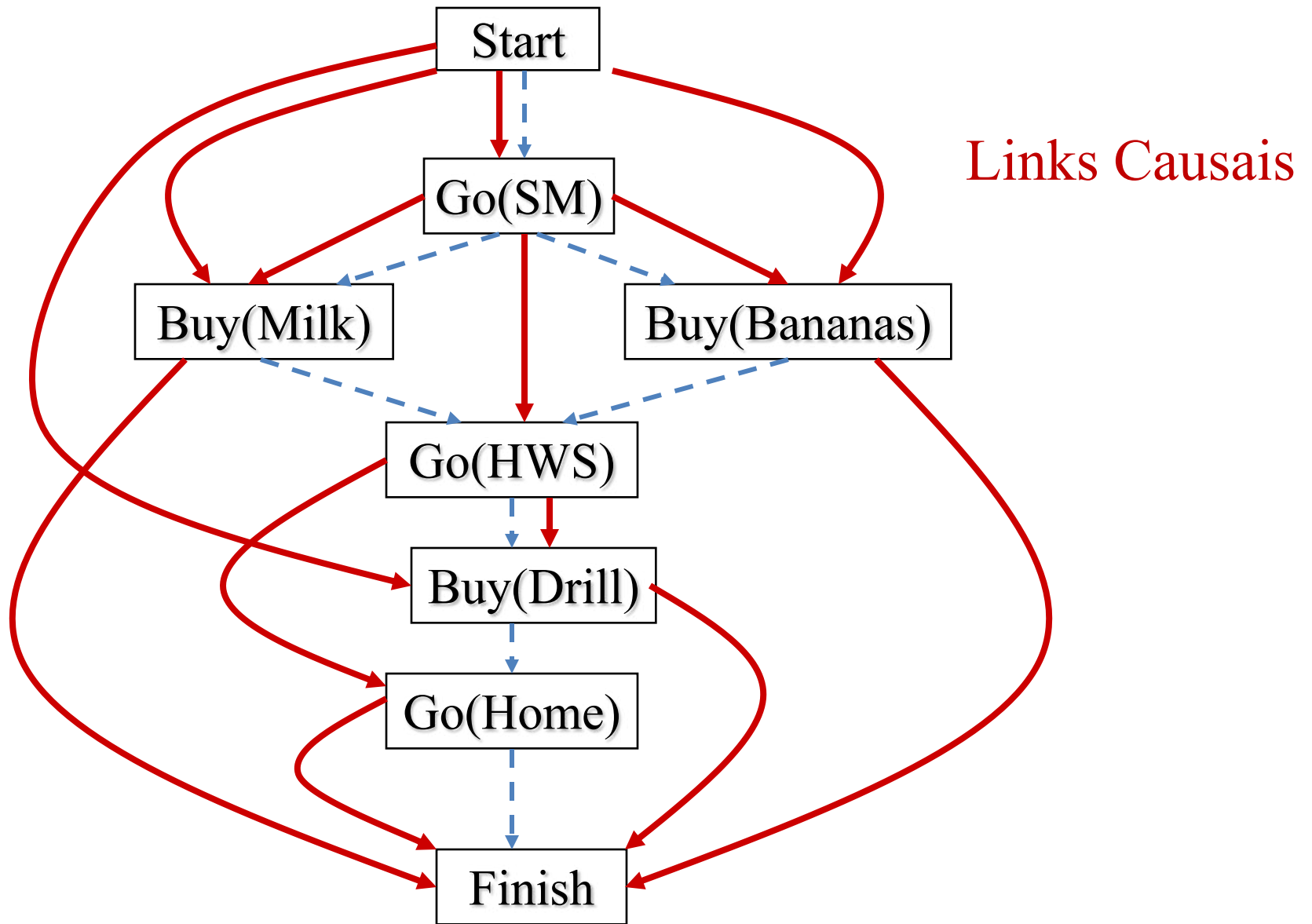


S3 ameaça a condição c estabelecida por S1 e protegida pelo link causal S1 para S2

POP: Solução do Exemplo das Compras



POP: Solução do Exemplo das Compras

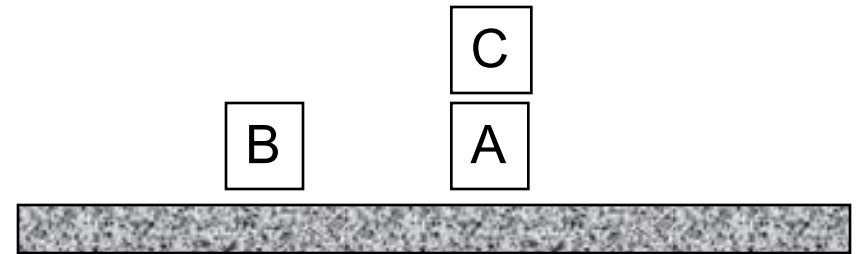


Planejamento: Engenharia de Conhecimento

1. Decidir sobre o que falar
2. Decidir sobre um vocabulário de condições, operadores e objetos
3. Codificar os operadores para o domínio
4. Codificar uma descrição da instância do problema
5. Colocar o problema para o planejador existente e obter os planos

Mundo dos Blocos: Estados e Ações

- O que falar (1)
 - um conjunto de blocos sobre uma mesa a serem empilhados numa certa ordem
 - só se pode mover um bloco se não houver nada em cima dele



- Vocabulário (2):

- Estado:

- On(x, y): bloco x está em cima de y
 - Clear(x): bloco x está livre

- Ações:

- PutOn(x, y): mover x para cima de y
 - PutOnTable(x): mover x para a mesa

Mundo dos Blocos: Ações / Operadores (3)

Op(ACTION: PutOn(x,y),

PRECOND: On(x,z), Clear(x), Clear(y)

EFFECT: ADD: On(x,y), Clear(z)

DEL: \neg On(x,z), \neg Clear(y))

On(x,z), Clear(x), Clear(y)

PutOn(x,y)

On(x,y), \neg Clear(y), Clear(z), \neg On(x,z)

Op(ACTION: PutOnTable(x),

PRECOND: On(x,z), Clear(x)

EFFECT: ADD: On(x,Table), Clear(z)

DEL: \neg On(x,z))

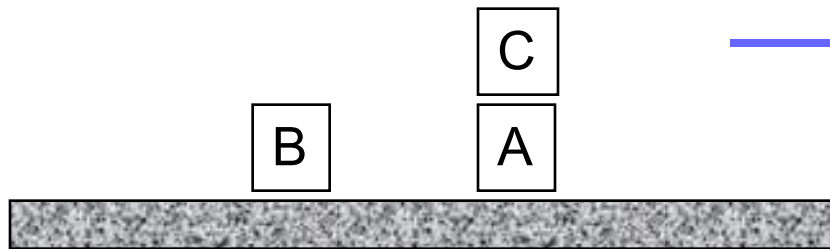
On(x,z), Clear(x)

PutOnTable(x)

On(x,Table), Clear(z), \neg On(x,z)

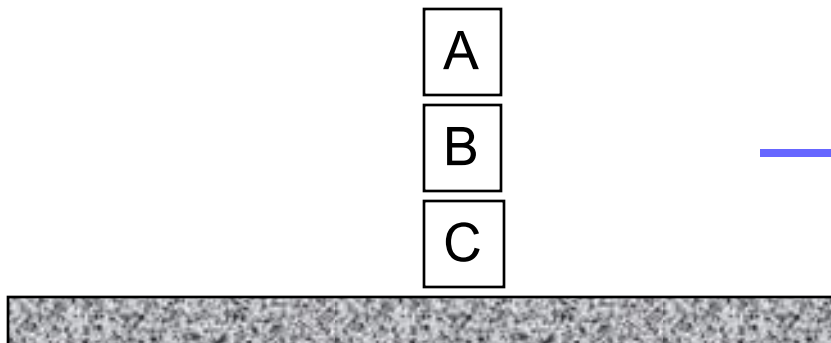
Mundo dos Blocos: Estados (4)

Estado inicial



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

Estado final

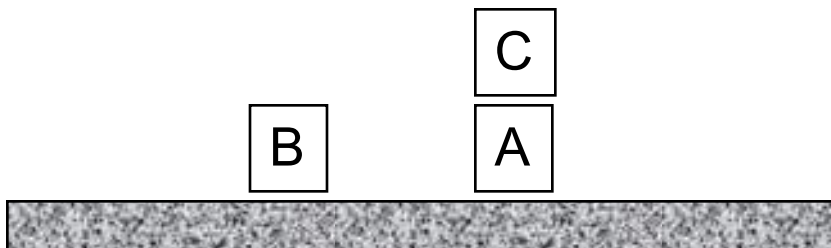


On(A, B)
On(B, C)
On(C, Table)
Clear(A)

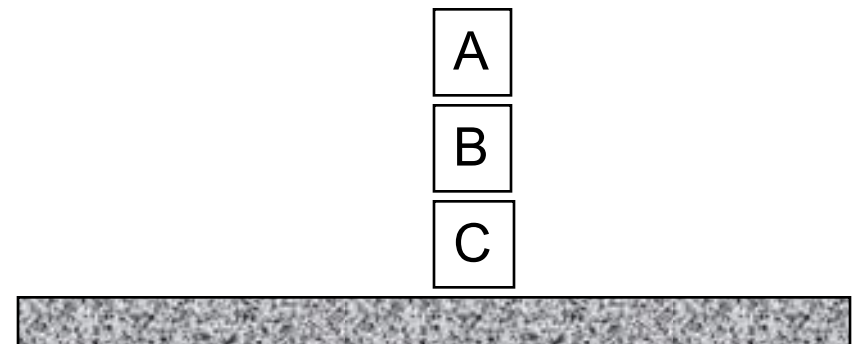
Mundo dos Blocos: Estados

- Com isto é possível resolver problemas do mundo dos blocos... Enviar toda esta descrição a um planejador (5)

Estado inicial



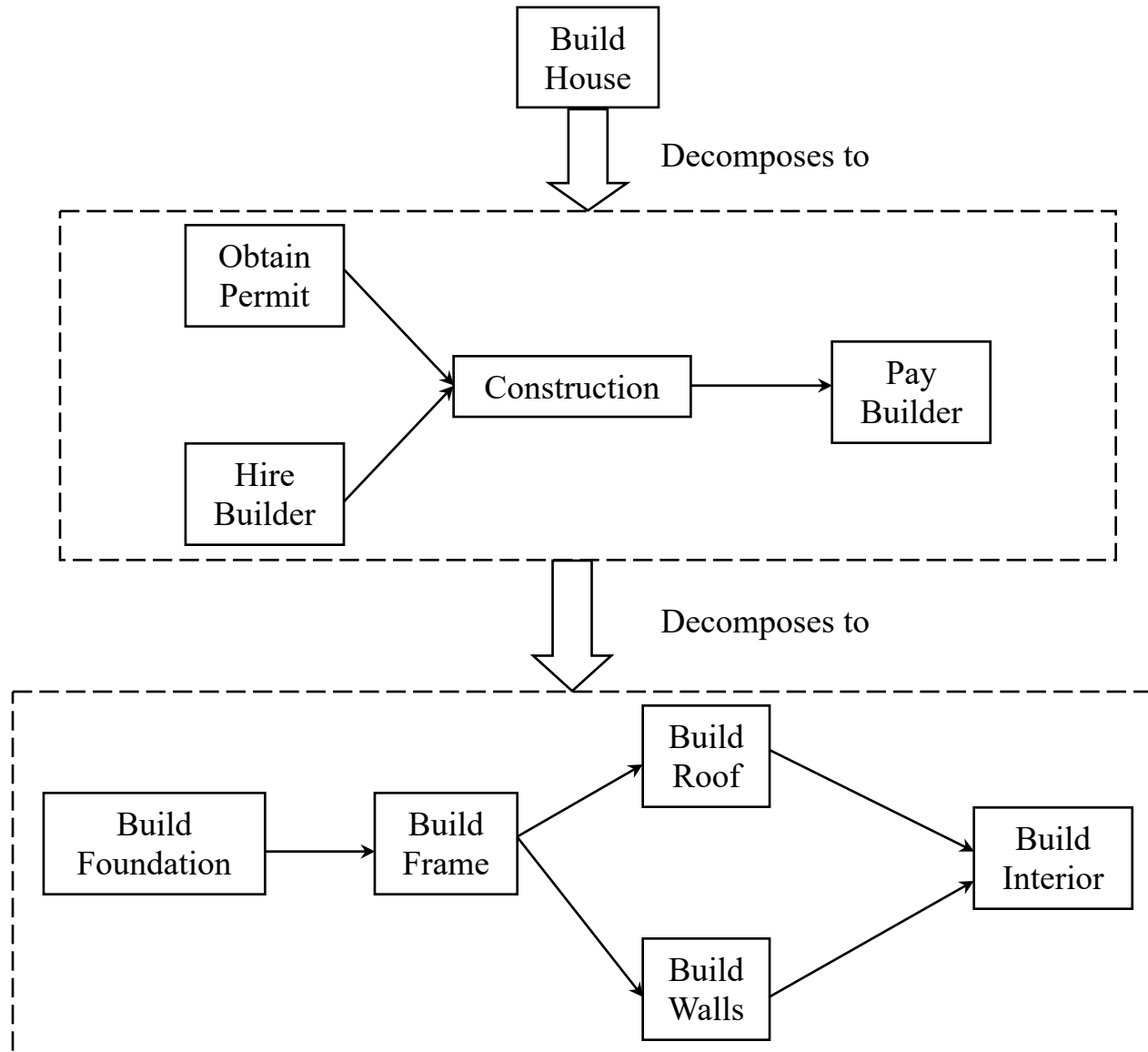
Estado final



Limitações de POP-STRIPS

- Planejamento em um único nível de granularidade
- Precondições e efeitos não contextuais (inclui tudo em um único contexto / “mundo”)
- Não possui representação do tempo
- Não tem representação de recursos limitados (ações consomem recursos)

Exemplo de Planejamento Hierárquico



Aplicações de Planejamento

- Construção de prédios:
 - SIPE
- Escalonamento de tarefas industriais
 - TOSCA (Hitachi)
 - ISIS (Whestinghouse)
- Construção, integração e verificação de espaçonaves:
 - Optimum-AIV (Agência Espacial Européia)
- Planejamento para Missões Espaciais
 - Voyager, Telescópio espacial Hubble (NASA)
 - ERS-1 (Agência Espacial Europeia)
- Robótica, logística, manufatura, etc...

Referências Bibliográficas

- S. Russel and P. Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River, USA. 2nd. Edition, 2003. Chapter 11 and 12.