## Cap. 11: PLANNING

- Uma vez tendo estabelecido um conjunto de estados e construído um modelo completo e correto, precisamos traçar um *plano de ação* de forma que o agente possa seguir as ações e atingir os objetivos.

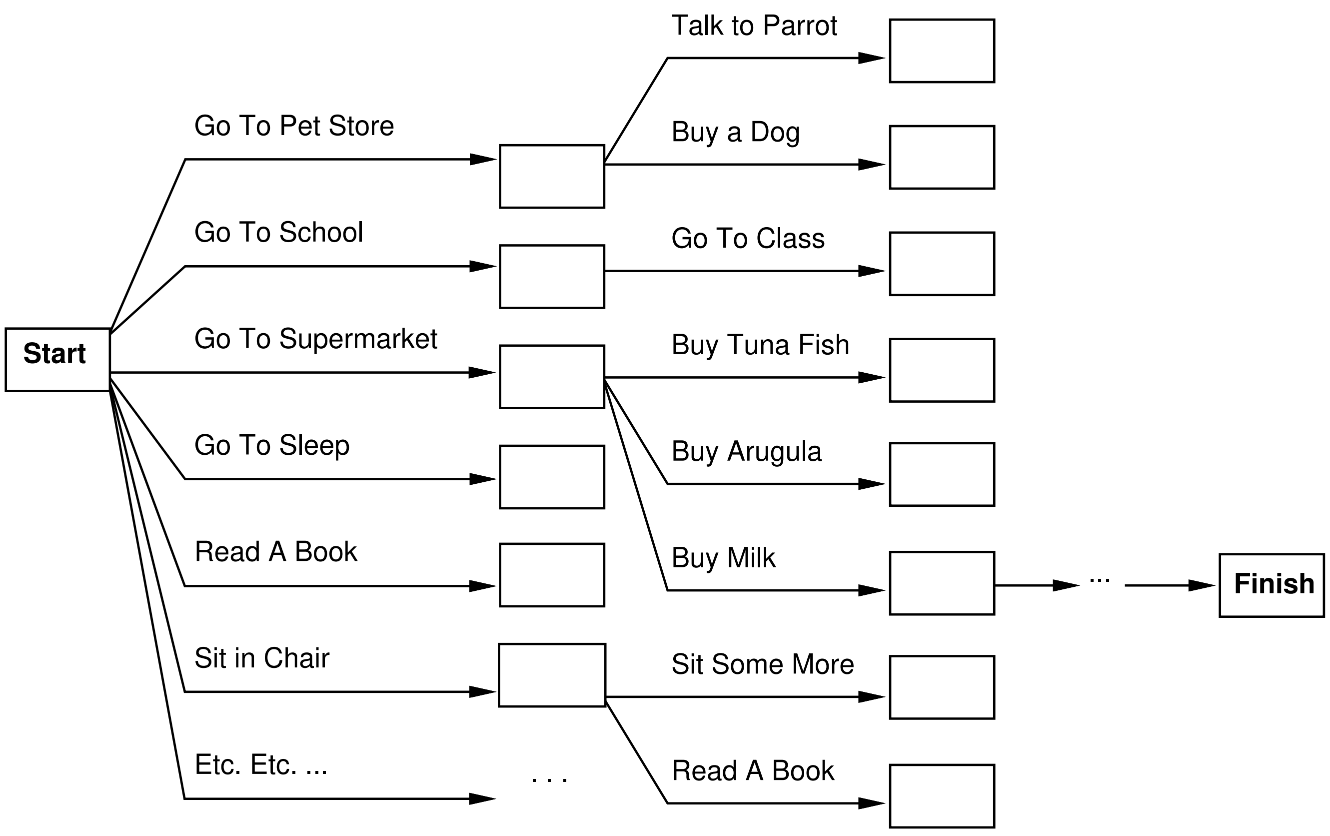
- Algoritmo básico:

```
function SIMPLE-PLANNING-AGENT(percept) returns an action
   static: KB, a knowledge base (includes action descriptions)
           p, a plan, initially NoPlan
           t, a counter, initially 0, indicating time
   local variables: G, a goal
                    current, a current state description

   TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
   current ← STATE-DESCRIPTION(KB, t)
   if p = NoPlan then
      G ← ASK(KB, MAKE-GOAL-QUERY(t))
      p ← IDEAL-PLANNER(current, G, KB)
   if p = NoPlan or p is empty then action ← NoOp
   else
      action ← FIRST(p)
      p ← REST(p)
   TELL(KB, MAKE-ACTION-SENTENCE(action, t))
   t ← t + 1
   return action
```

## From Problem Solving to Planning

- diferenças: representações de ações, estados, objetivos e planos.
- exemplo simples: comprar 1 l de leite, uma dúzia de maçãs e uma lanterna.
- problem solving:
  - estado inicial: agente está em casa, mas sem todas as coisas necessárias.
  - operadores: tudo que o agente pode fazer para obter os objetos.
  - opcional: heurística (ex, número de coisas ainda não compradas).


Talk to Parrot
Go To Pet Store
Buy a Dog
Go To School
Go To Class
Start
Go To Supermarket
Buy Tuna Fish
Go To Sleep
Buy Arugula
Read A Book
Buy Milk
...
Finish
Sit in Chair
Sit Some More
Etc. Etc. ...
. . .
Read A Book

## From Problem Solving to Planning

- três idéias principais: transparência de ações e estados, liberdade para adicionar ações ao plano de ações a qualquer momento (sem ser incremental), e independência das partes.
- Planning utilizando cálculo de situações:
  - estado inicial: $Em(Casa, S_0) \wedge \neg Tem(Leite, S_0) \wedge \neg Tem(Macas, S_0) \wedge \neg Tem(Lanterna)$
  - estado final: $\exists s Em(Casa, s) \wedge Tem(Leite, s) \wedge Tem(Macas, s) \wedge Tem(Lanterna, s)$
  - operadores: ex, axioma do estado sucessor para comprar leite:
    $\forall a, s Tem(Leite, Result(a, s)) \Leftrightarrow ((a = Compra(Leite) \wedge Em(Supermercado, s) \vee Tem(Leite, s) \wedge a \neq Largou(Leite)))$

## From Problem Solving to Planning

- Em "planning" representamos *sequências de ações*, não somente ações únicas. Em cálculo de situações representamos Result:
  - $\forall s Result'([], s) = s$
  - $\forall a, p, s Result'([a \mid p], s) = Result'(p, Result(a, s))$
- Uma solução para o problema das compras é um plano $p$ que qdo aplicado ao estado inicial $S_0$ produz a solução:
  - $Em(Casa, Result'(p, S_0)) \wedge Tem(Leite, Result'(p, S_0)) \wedge Tem(Macas, Result'(p, S_0)) \wedge Tem(Lanterna, Result'(p, S_0))$
- ASK daria uma solução do tipo:
  $p = [Va(Supermercado), Compre(Leite), Compre(Macas), Va(CasaEVideo), Compre(Lanterna), Va(Casa)]$

## From Problem Solving to Planning

- soluções teóricas nem sempre são *eficientes* se usarmos regras de inferência não guiadas.
- para soluções práticas:
  - Restringir a linguagem de definição do problema.
  - Utilizar algoritmos de "planning" particulares (*planner* para cada problema, invés de algoritmos gerais para provas de teoremas).

## Representações básicas para Planning

- Maioria dos "planners" utilizados descreve estados e operadores em STRIPS ou extensões (linguagem baseada em cálculo de situações).
- Estados: representados por conjunções de literais "ground" (completamente instanciados), sem funções (predicados aplicados a símbolos constantes). Ex: $Em(Casa) \wedge \neg Tem(Leite) \wedge \neg Tem(Macas) \wedge \neg Tem(Lanterna) \wedge \ldots$
- Descrição do estado não precisa ser completa.
- Maior parte dos "planners" assume que se nada é dito sobre um literal, ele é falso (negação por falha em programação lógica).

## Representações básicas para Planning

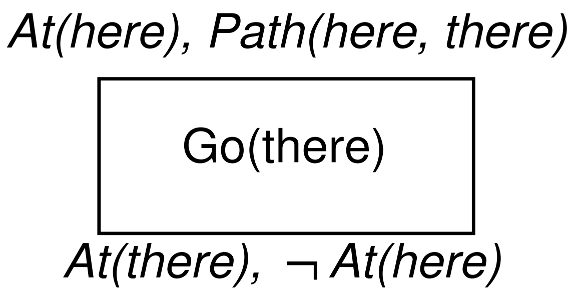
- Objetivos: tb representados por conjunções de literais. Ex: $Em(Casa) \wedge Tem(Leite) \wedge Tem(Macas) \wedge Tem(Lanterna)$.
- Podem conter variáveis. Ex, estar numa loja que venda leite: $Em(x) \wedge Vende(x, Leite)$.
- Como em provadores automáticos de teoremas, assume-se que todas as variáveis do objetivo estão quantificadas existencialmente.
- Diferenças entre objetivos num "planner" e num provador de teoremas: objetivo do "planner" pergunta por uma sequência de ações que permitam validar o objetivo se este for executado. Objetivo no provador de teoremas é diretamente provado verdadeiro.

## Representações básicas para Planning

- Representação de ações: operadores STRIPS consistem de três componentes:
  - Descrição da ação: para o "planner" serve apenas como um nome para a ação.
  - Precondição: conjunção de átomos (literais positivos) que devem ser verdadeiras antes do operador ser aplicado.
  - Efeito (ou poscondição): conjunção de literais (positivos ou negativos) que descrevem como a situação é modificada quando o operador é aplicado.

## Representações básicas para Planning

- Ex: $Op(ACAO : Va(la), PRECON : Em(aqui) \wedge Caminho(aqui, la), EFEITO : Em(la) \wedge \neg Em(aqui))$



## Representações básicas para Planning

- Um operador pode conter variáveis: *esquema de operador*.
- Corresponde a uma família de ações.
- Somente operadores completamente instanciados podem ser executados.
- Linguagem de descrição de precondições e efeitos bastante restrita. Pode ser menos restrita.

## Representações básicas para Planning

- Operador é dito *aplicável* num estado $s$, se há alguma forma de instanciar variáveis do operador tal que a precondição seja verdadeira para esta instância. No estado resultante, todos os literais positivos da poscondição também devem ser verdadeiros, exceto os negados.

- Ex: situação inicial, $Em(casa), Caminho(Casa, Supermercado)\ldots$, ação Va(Supermercado) é aplicável, e a situação resultante contém $\neg Em(Casa), Em(Supermercado)$, $Caminho(Casa, Supermercado)\ldots$

## Espaço de Situações e Espaço de Planos

- Dada uma definição do problema na linguagem STRIPS, algoritmos de busca podem ser utilizados para resolver o problema de "planning" do estado inicial para o objetivo.
- Chamados geradores de planos sobre o *espaço de situações*, e *progressivos*.
- Problema com esta abordagem: fator de ramificação!
- Alternativa: gerador de planos *regressivo*, vai do objetivo para o estado inicial de forma a cortar o espaço de busca.
- Algoritmo original utilizado por STRIPS era de um gerador de planos sobre o espaço de situações regressivo, mas era incompleto, pois não sabia lidar com o fato de gerar uma conjunção.
- Ineficiente produzir algoritmo completo utilizando esta

abordagem de geração de planos por situações.

## Espaço de Situações e Espaço de Planos

- Alternativa: busca pelo *espaço de planos*.
- Começamos com um plano simples, incompleto: *plano parcial*.
- Aplicamos os operadores (de plano) e geramos outros planos até chegar a um que nos leve à solução.
- Possíveis operadores de plano: adicionar um novo passo ao plano, imposição de ordem aos passos do plano, instanciação de uma variável etc.
- Solução é o plano final. Caminho para chegar a este plano é irrelevante (eficiência...)

## Representações para Planos

- Operações sobre planos:
  - Operadores de refinamento: adiciona restrições a um plano parcial. Elimina alguns planos do conjunto possível de planos.
  - Operadores de modificação: qualquer outro operador que não seja de refinamento.
- Ex: Plano para calçar um par de sapatos.
- Objetivo:
  $ColocadoSapatoPeDireito \wedge ColocadoSapatoPeEsquerdo$.
- Estado inicial: não necessário.

## Representações para Planos

- Operadores:
  - $Op(ACAO : ColocarSapatoPeDireito, PRECOND : ColocadaMeiaPeDireito, EFEITO : ColocadoSapatoPeDireito)$
  - $Op(ACAO : ColocarMeiaPeDireito, EFEITO : ColocadaMeiaPeDireito)$
  - $Op(ACAO : ColocarSapatoPeEsquerdo, PRECOND : ColocadaMeiaPeEsquerdo, EFEITO : ColocadoSapatoPeEsquerdo)$
  - $Op(ACAO : ColocarMeiaPeEsquerdo, EFEITO : ColocadaMeiaPeEsquerdo)$

## Representações para Planos

- Plano parcial deve ter dois passos: ColocarSapatoPeDireito, ColocarSapatoPeEsquerdo. Quem vem primeiro?
- *Menos "comprometedor"*: somente escolher planos com informações relevantes no momento. Deixar outras escolhas para serem feitas mais tarde.
- *Plano de ordem parcial*: alguns passos do plano são ordenados, outros não.
- *Plano de ordem total*: todos os passos são ordenados.
- *Linearização* de planos: geração de um plano de ordem total após adição de restrições ao plano anterior.
- Restrições no valor das variáveis a serem instanciadas.

## Representações para Planos

- Um plano é formalmente definido como uma estrutura de dados contendo os seguintes 4 componentes:
  - conjunto de passos. Cada passo é um dos operadores do problema.
  - conjunto de restrições de ordenação dos passos.
  - conjunto de instâncias de variáveis para cada passo.
  - conjunto de ligações causais. Ex: $S_i \longrightarrow^c S_j$, com $c$ sendo precondição de $S_j$.
- Plano inicial descreve o problema ainda não resolvido. Consiste de dois passos: início e fim com relação de precedência Inicio $\prec$ Fim.

## Representações para Planos

- Passo Início não tem precondições e o efeito é adicionar todas as proposições verdadeiras ao estado inicial.
- Passo Fim tem o objetivo como precondição, e nenhum efeito.
- Ex:

Plan(STEPS:{ $S_1$: Op(ACAO:Inicio),
$S_2$: Op(ACAO: Fim,
PRECOND: $ColocadoSapatoPeDireito \wedge$
$ColocadoSapatoPeEsquerdo$)},
ORDERINGS: { $S_1 \prec S_2$ },
BINDINGS: {},
LINKS: {})