

# *Planejamento em Inteligência Artificial*

*planning is the reasoning side of acting*

Leliane Nunes de Barros

# Livro

- M. Ghallab, D. Nau, and P. Traverso  
*Automated Planning: Theory and Practice*  
Morgan Kaufmann Publishers, 2004

# Pré-requisitos

- Algoritmos de busca
  - ◆ *Depth-first, breadth-first, best-first search*
  - ◆ A\*, heurísticas admissíveis Versus não-admissíveis
- Lógica
  - ◆ Lógica Proposicional
  - ◆ Lógica de Primeira Ordem (predicados e quantificadores)
  - ◆ Cláusulas de Horn e Provador de Teoremas

# ***Plano (www.dictionary.reference.com)***

1. Um esquema, programa ou método contruído de antemão para realizar um objetivo (meta): *plano de ataque*.
2. Uma proposta ou um projeto (completo) tentativa de um curso de ações: *qual é o seu plano para essa noite?*
3. Uma disposição sistemática de elementos ou partes importantes; uma configuração ou esqueleto (*outline*): *plano de instalação; plano de uma estória*.
4. Um desenho ou diagrama feito em escala para mostrar a estrutura ou disposição de alguma coisa.
5. Em um desenho em perspectiva, um dos planos imaginários perpendiculares à linha de visão (cortes ou projeções) entre o observador e o objeto que está sendo observado.
6. Um programa ou política estipulando um serviço ou benefício: *plano de pensão; plano de saúde* ou *plano de governo*.

Sinônimos: design, projeto, esquema, estratégia

# Plano ([www.dictionary.reference.com](http://www.dictionary.reference.com))

1. Um esquema, programa ou método contruído de antemão para realizar um objetivo (meta): *plano de ataque*.
2. Uma proposta ou um projeto (completo) tentativa de um curso de ações: *qual é o seu plano para essa noite?*
3. Uma disposição sistemática de elementos ou partes importantes; uma configuração ou esqueleto (*outline*): *plano de instalação; plano de uma estória*.
4. Um desenho ou diagrama feito em escala para mostrar a estrutura ou disposição de alguma coisa.
5. Em um desenho em perspectiva, um dos planos imaginários perpendiculares à linha de visão (cortes ou projeções) entre o observador e o objeto que está sendo observado.
6. Um programa ou política estipulando um serviço ou benefício: *plano de pensão; plano de saúde* ou *plano de governo*.

Sinônimos: escala, design, projeto, esquema, estratégia

# Planos e Planejamento

- **Plano:**

- ◆ *Uma coleção de ações para desempenhar alguma tarefa ou atingir algum objetivo.*
- ◆ *[uma representação ] de comportamento futuro ... normalmente um conjunto de ações, com restrições temporais e outros tipos de restrições, para serem executadas por um agente (ou agentes) - Austin Tate [MIT Encyclopedia of the Cognitive Sciences, 1999]*

- **Planejamento**

- ◆ *Planejamento é o processo de escolha e organização de ações através da antecipação (previsão) de seus efeitos. Esse processo de raciocínio tem o objetivo de satisfazer (através da execução de ações), alguns objetivos previamente estabelecidas.*
- ◆ *Planejamento automático é a sub-área da IA que estuda esse processo de raciocínio, usando o computador. Aplicação: sistemas que exigem comportamento autônomo e deliberativo.*

# Planejamento

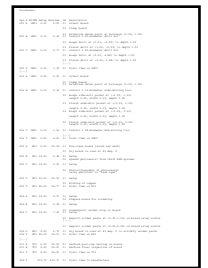
Existem vários programas para ajudar planejadores humanos

- ◆ Gerenciamento de Projeto, armazenamento/recuperação de planos, geração automática de escalonamento com iniciativa mixta



Geração automática de planos é muito difícil!

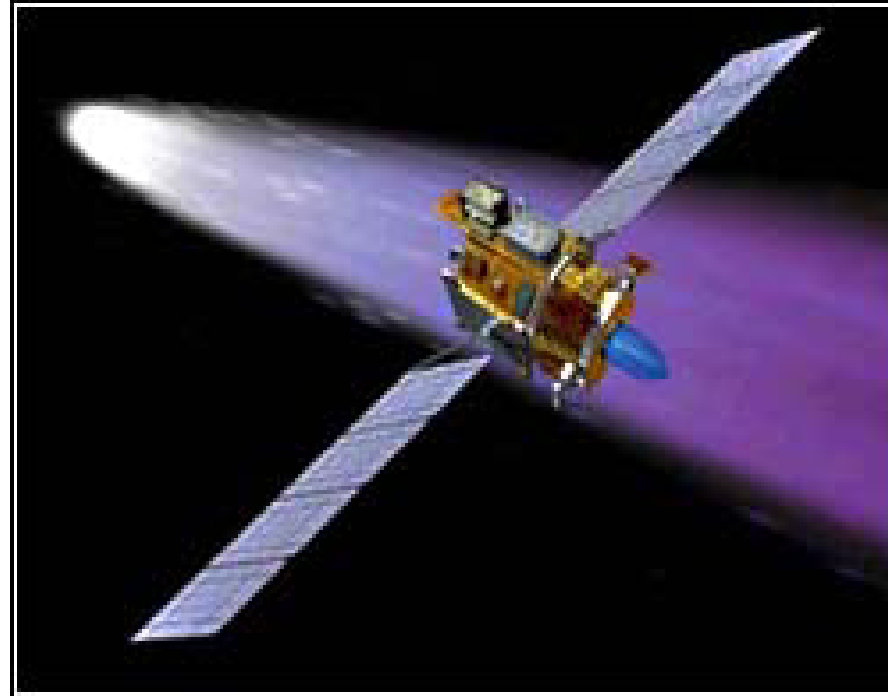
- ◆ Existem muitos protótipos de pesquisa, poucos sistemas práticos (usados em aplicações reais)
- ◆ Início da área de pesquisa: ~1970
- ◆ Pesquisa começa a dar retorno (~1995):
  - » Exemplos de sucesso em problemas práticos difíceis



# NASA Unmanned Spacecraft

## *Remote Agent eXperiment (RAX)*

- ◆ Software autônomo de planejamento/controlado de IA
- ◆ Usado na espaçonave DS1 em Maio de 1998
- ◆ A espaçonave foi controlada por vários minutos pelo RAX



## Veículo de exploração (*rover*) em Marte

- ◆ Guiado por um software autônomo de planejamento/controlado de Inteligência Artificial



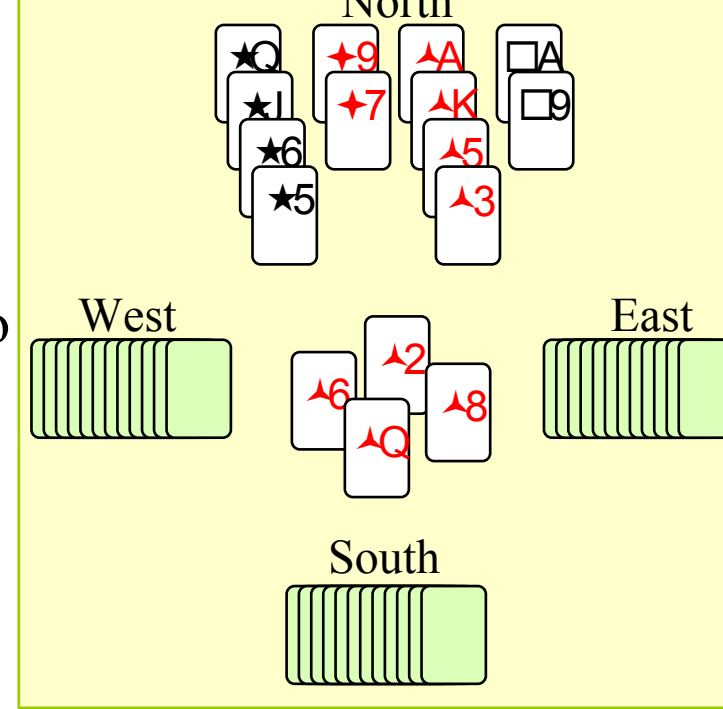
# Outros Exemplos

- *Computer bridge: Bridge Baron*

- ◆ Usou Planejamento em IA para ganhar o campeonato mundial de 1997 de bridge
- ◆ Software comercial: vendeu milhares de cópias

- Planejamento de processo de manufatura

- ◆ É usado para planejar operações de estamparia (*bending*) na indústria automotiva



# Formas conhecidas de Planejamento

## Planejamento de caminho e movimentação:

- ◆ definição de uma tarefa geométrica de uma posição inicial à uma posição meta + o controle de um sistema móvel (robôs móveis, veículos, braços mecânicos, agente virtual). Deve levar em conta o modelo do ambiente bem como a dinâmica e cinemática do sistema móvel

## Planejamento de percepção:

- ◆ geração de planos de ações envolvendo ações de sensoriamento, por exemplo, na modelagem ou identificação de um ambiente, um objeto e na localização de um sistema móvel. Essa forma de planejamento tenta responder:
  - » Que informação é necessária? Quando ela é necessária? Qual é o sensor mais adequado para uma dada tarefa? Como usar a informação?

## Planejamento para recuperação de informação:

- ◆ o ambiente é um Banco de Dados ou a WWW

## Planejamento de navegação:

- ◆ combinação de planejamento de movimentação com percepção (movimentação evitando obstáculos, seguir um caminho até encontrar um marcação)

## Planejamento de manipulação:

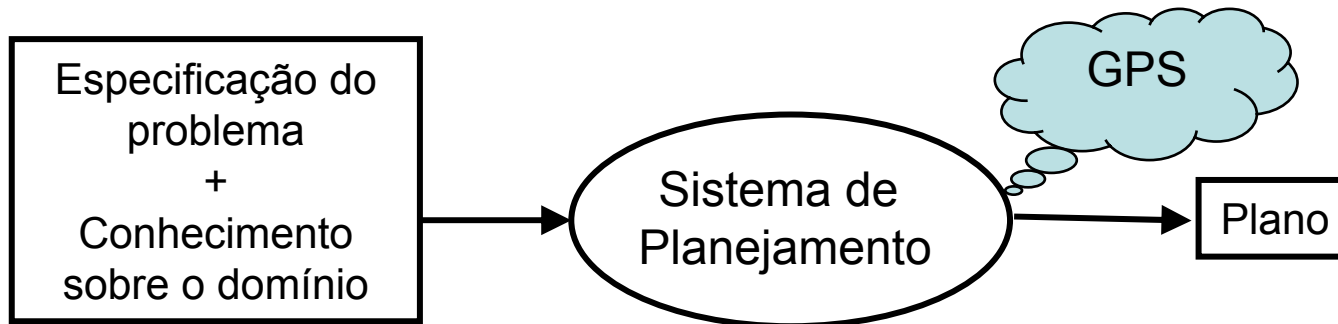
- ◆ problemas de manipulação de objetos para construção e montagem.

## Planejamento de comunicação:

- ◆ construção de diálogos em problemas de cooperação entre vários agentes, humanos ou artificiais. Ex: planejamento instrucional em Sistemas Tutores Inteligentes, serviço de tele-atendimento.

# Soluções dedicadas

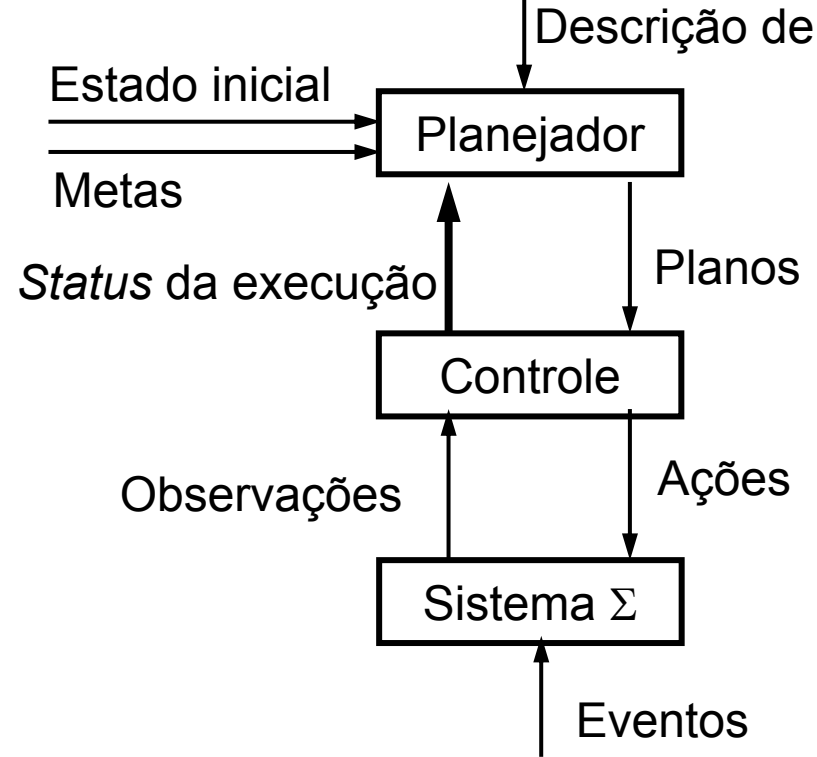
- Uso de modelos específicos e técnicas adequadas:
  - ◆ Geometria, cinemática e dinâmica. Programação matemática e técnicas de otimização.
- Limitações das abordagens dependentes do domínio:
  - ◆ Não são tratados aspectos comuns a todas essas formas de planejamento. Estudos sobre esses aspectos ajudam a melhorar estratégias dependentes de domínio
  - ◆ Custo maior para tratar cada de problema de planejamento como um novo problema ao invés de adaptar um ferramentas de propósito geral
  - ◆ Para a construção de agentes autônomos e inteligentes, as abordagens específicas não têm apresentado o sucesso desejado.



# Tópicos da aula

- Modelo conceitual de planejamento
- Suposições restritivas
- Planejamento classico
- Relaxando suposições
- Example: Robôs de cais de porto (carga e descarga de navios)

# Modelo Conceitual



- **Ingredientes:**

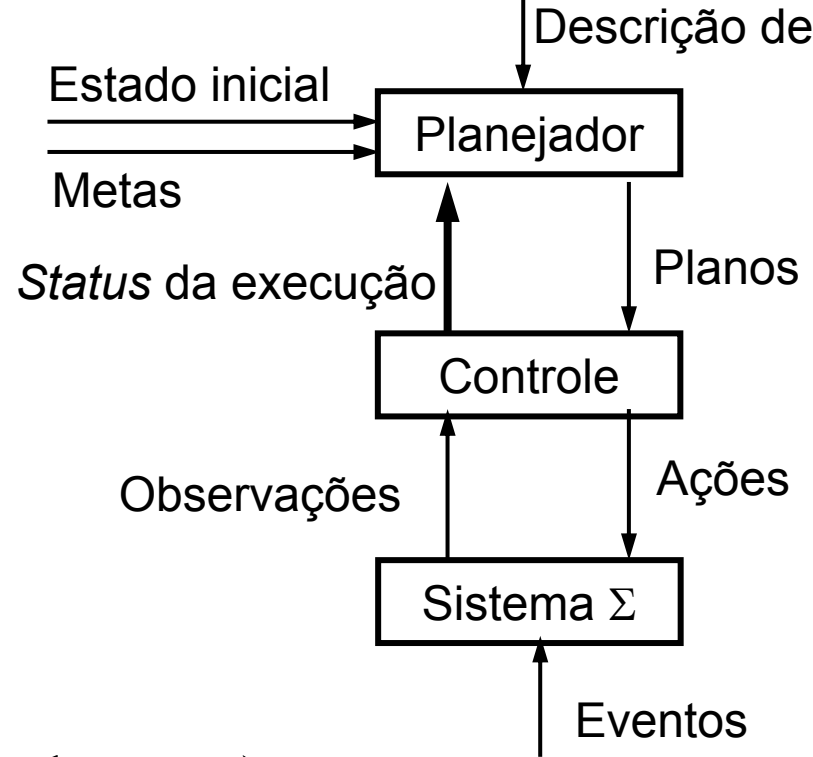
- ◆ Modelo do ambiente: *estados possíveis*
- ◆ Modelo de como o ambiente pode mudar: *efeitos de ações*
- ◆ Especificação de *condições iniciais* e *metas*
- ◆ *Planos de ações* que são gerados pelo planejador
- ◆ Um modelo de execução de um plano no ambiente
- ◆ Um modelo de observação do ambiente

# Modelo Conceitual

- Sistema de transição de estado

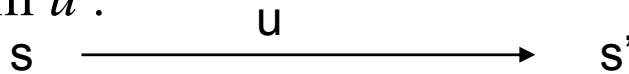
$$\Sigma = (S, A, E, \gamma)$$

- ◆  $S = \{s_1, s_2, \dots\} = \{\text{estados}\}$
- ◆  $A = \{a_1, a_2, \dots\} = \{\text{ações}\}$  (controladas pelo agente)
- ◆  $E = \{e_1, e_2, \dots\} = \{\text{eventos exógenos}\}$  (não controladas pelo agente)
- ◆ Função de transição de estado  
 $\gamma: S \times (A \cup E) \rightarrow 2^S$



# Modelo Conceitual

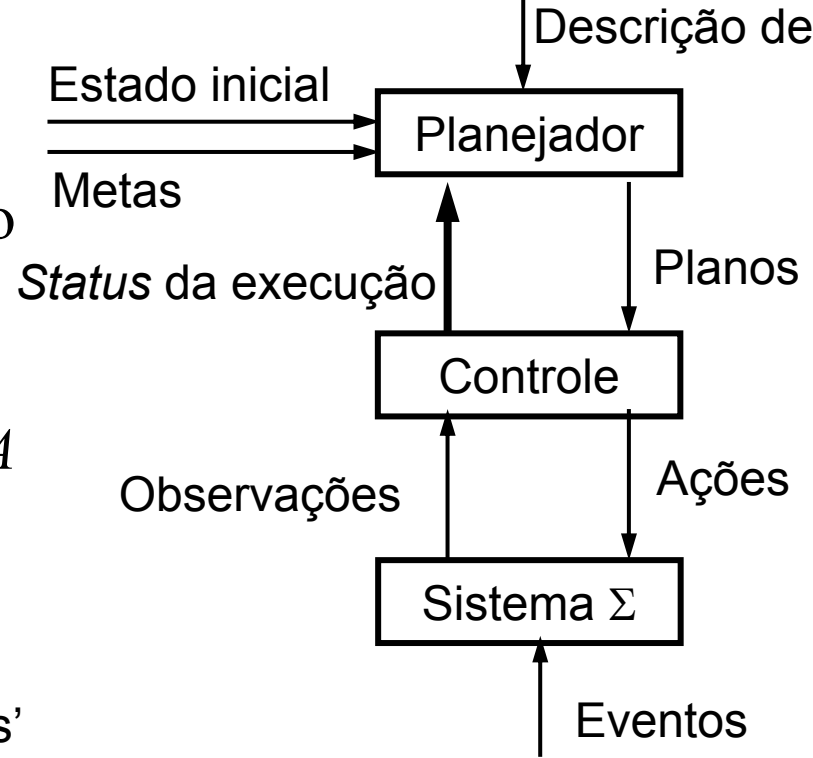
Um sistema  $\Sigma$  do tipo estado-transição pode ser representado por um grafo dirigido cujos nós são estados em  $S$ . Se  $s' \in \gamma(s, u)$ , onde  $u$  é o par  $(a, e)$ ;  $a \in A$  e  $e \in E$ , então o grafo contém um arco (chamado de transição de estado) de  $s$  a  $s'$ , rotulado com  $u$ :



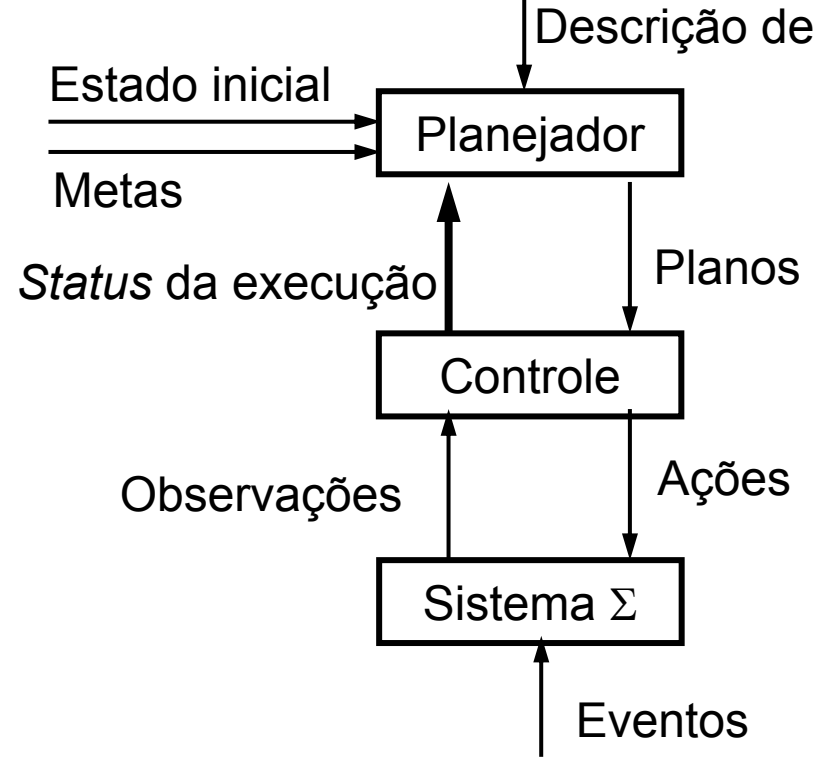
Se  $a$  é uma ação aplicável no estado  $s$ , aplicá-la em  $s$  leva a um outro estado  $\gamma(s, a)$ . O sistema evolui através dos eventos e ações.

$\epsilon$  é um evento neutro  $\Rightarrow \gamma(s, a, \epsilon) = \gamma(s, a)$

no-op é uma ação neutra  $\Rightarrow \gamma(s, no-op, e) = \gamma(s, e)$



# Modelo Conceitual



- Função de observação  $h: S \rightarrow O$ 
  - ◆ produz observação  $o$  sobre o estado atual  $s$
- Controle: dada a observação  $o \in O$ , produz ação  $a \in A$
- Planejador:
  - ◆ entrada: descrição de  $\Sigma$ , estado inicial  $s_0 \in S$  e alguma meta
  - ◆ saída: produz um plano para guiar o controle



# Modelo Conceitual

Metas possíveis:

- ◆ Um conjunto de estados meta  $S_g$

*Encontre uma seqüência de transição de estados terminando em um estado meta*

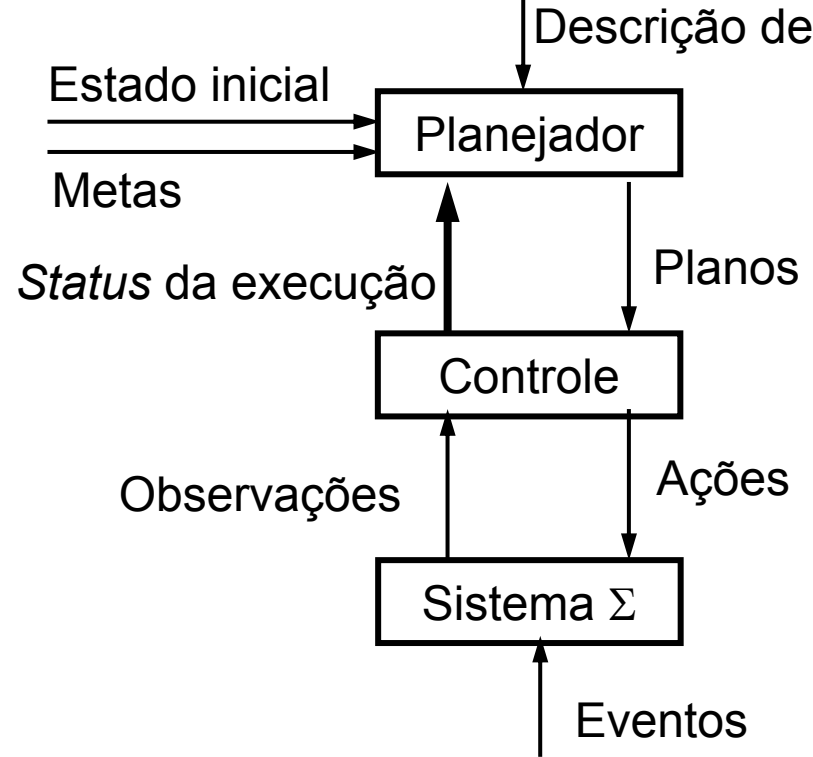
- ◆ Alguma condição sobre o conjunto de estados percorridos pelo sistema (metas extendidas)

*Atinja  $S_g$ , passando por estados que valha  $r$ , e permaneça nele*

- ◆ Função utilidade (ou recompensa) relacionada aos estados

*Otimize alguma função utilidade*

- ◆ Tarefas para executar, especificadas recursivamente como conjuntos de sub-tarefas e ações



# Esse modelo funciona?

- Suposição: o sistema de controle é robusto
- Um modelo mais realista: intercala planejamento com execução de ações, com a adição de um sistema de supervisão/revisão de planos (replanejamento).

# Modelo Conceitual: Exemplo dos Robôs de Porto

Sistema de transição de estado

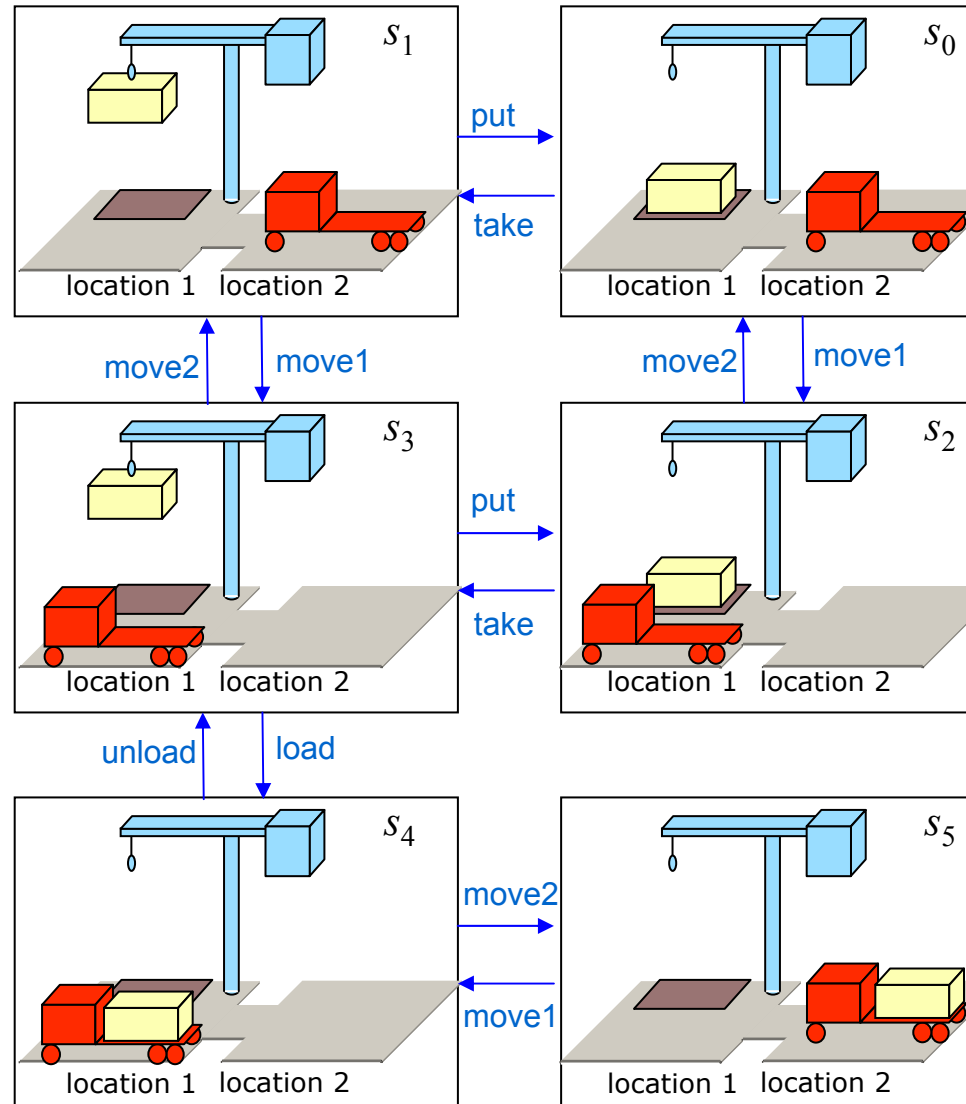
$$\Sigma = (S, A, E, \gamma)$$

- ◆  $S = \{s_0, \dots, s_5\}$
- ◆  $A = \{\text{move1, move2, put, take, load, unload}\}$
- ◆  $E = \{\}$
- ◆  $\gamma$ : como ilustrado na figura

$h(s) = s$  para todo  $s$

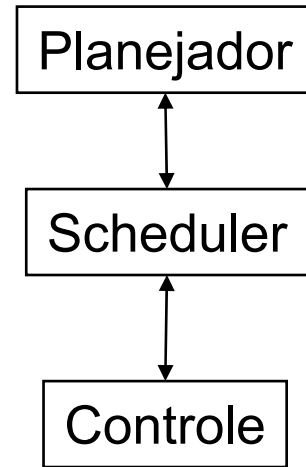
Entrada do planejador:

- ◆ sistema  $\Sigma$
- ◆ Estado inicial  $s_0$
- ◆ Estado meta  $s_5$



# Planejamento Versus Escalonamento (*Scheduling*)

- Escalonamento
  - ◆ Decide **como** executar um conjunto dado de ações usando um número limitado de recursos em um intervalo de tempo limitado
  - ◆ É tipicamente NP-completo
- Planejamento
  - ◆ Decide **quais** ações usar para atingir um conjunto de metas
  - ◆ Pode ser muito pior que NP-completo
    - » Na maioria dos casos, é não-decidível
    - » Muitas pesquisas assumem conjuntos de restrições para garantir a decidibilidade
    - » Vamos ver algumas dessas restrições = == =====>



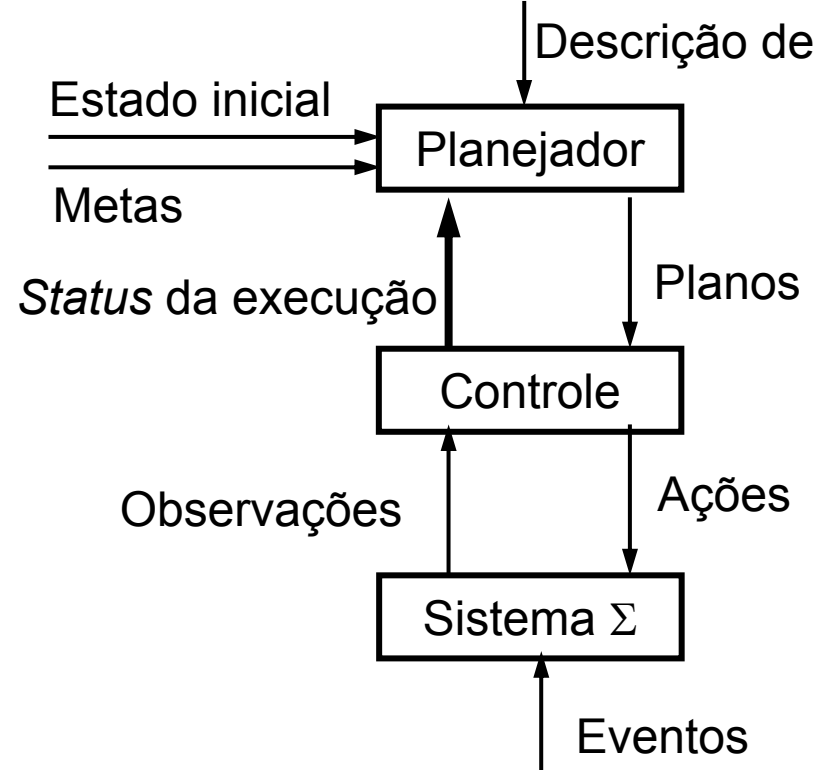
# Suposições restritivas

## A0 ( $\Sigma$ finito):

- ◆ O espaço de estados  $S$  é finito
- ◆  $S = \{s_0, s_1, s_2, \dots, s_k\}$  para algum  $k$

## A1 ( $\Sigma$ totalmente observável):

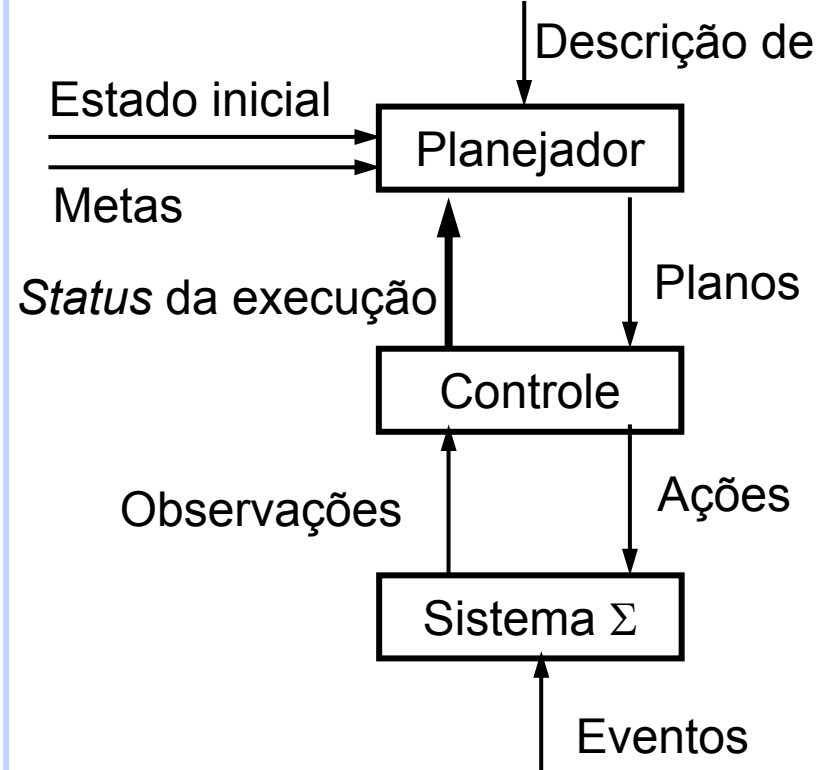
- ◆ A função de observação  $h: S \rightarrow O$  é a função identidade
- ◆ i.e., o controle sempre sabe em que estado ele está.



$$\Sigma = (S, A, E, \gamma)$$
$$S = \{\text{estados}\}$$
$$A = \{\text{ações}\}$$
$$E = \{\text{eventos}\}$$
$$\gamma: S \times (A \cup E) \rightarrow 2^S$$

# Suposições restritivas

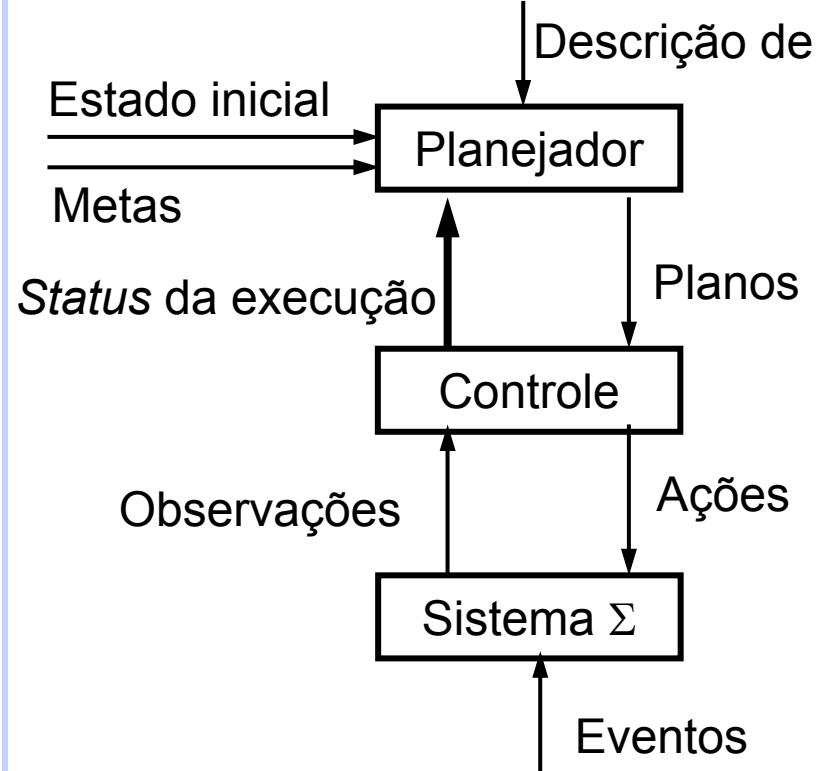
- **A2 ( $\Sigma$  determinístico):**
  - ◆ Para todo  $u$  em  $A \cup E$ ,  $|\gamma(s, u)| = 1$
  - ◆ Cada ação ou evento tem apenas um saída possível
- **A3 ( $\Sigma$  estástico):**
  - ◆  $E$  é vazio: nenhuma mudança ocorre exceto aquelas efetuadas pelo controle
- **A4 (satisfação de metas = *attainment goals*):**
  - ◆ Um estado meta  $s_g$  ou um conjunto de estados meta  $S_g$



$$\Sigma = (S, A, E, \gamma)$$
$$S = \{\text{estados}\}$$
$$A = \{\text{ações}\}$$
$$E = \{\text{eventos}\}$$
$$\gamma: S \times (A \cup E) \rightarrow 2^S$$

# Suposições restritivas

- **A5 (planos sequenciais):**
  - ◆ A solução é uma seqüência de ações totalmente ordenada  $(a_1, a_2, \dots a_n)$
- **A6 (tempo implícito):**
  - ◆ Transições de estados (ações) instantâneas, sem duração de tempo
- **A7 (planejamento *off-line*):**
  - ◆ Planejador não sabe o *status* da execução



$$\Sigma = (S, A, E, \gamma)$$
$$S = \{\text{estados}\}$$
$$A = \{\text{ações}\}$$
$$E = \{\text{eventos}\}$$
$$\gamma: S \times (A \cup E) \rightarrow 2^S$$

# Planejamento Clássico

- Planejamento clássico requer todas as 8 suposições
  - ◆ *Conhecimento completo sobre um sistema determinístico, estático, estado-finito com satisfação de metas e tempo implícito*
- Planejamento se reduz ao seguinte problema:
  - ◆ Dado  $(\Sigma, s_0, S_g)$ , encontre uma seqüência de ações  $(a_1, a_2, \dots, a_n)$  que produza uma seqüência de transições de estados

$$s_1 = \gamma(s_0, a_1),$$

$$s_2 = \gamma(s_1, a_2),$$

...

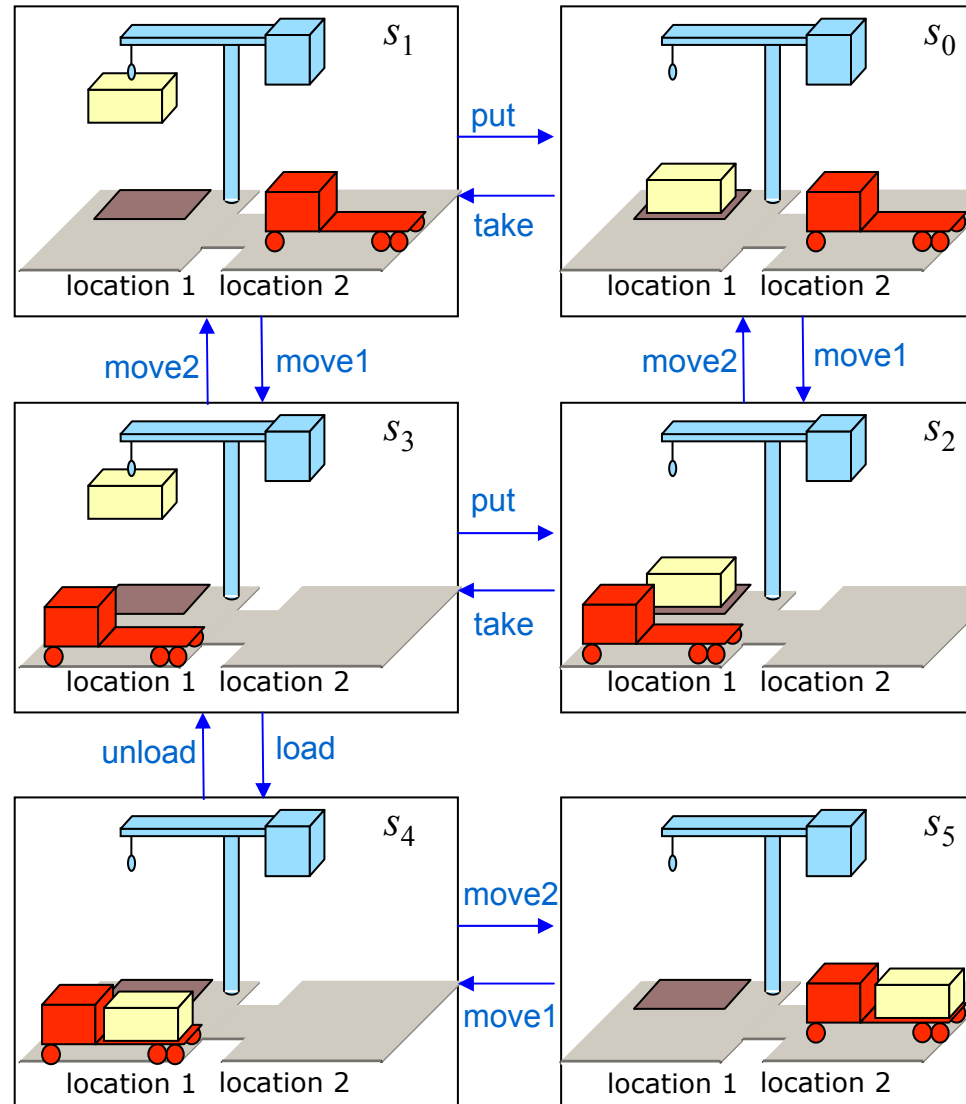
$$s_n = \gamma(s_{n-1}, a_n)$$

tal que  $s_n$  pertença à  $S_g$ .



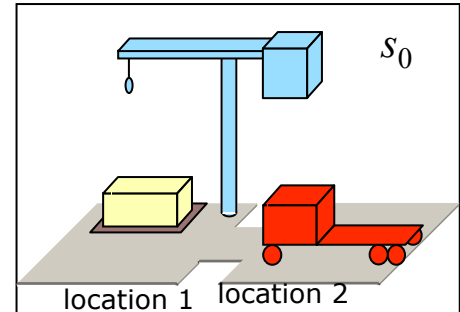
# Planejamento Clássico: Exemplo

- Exemplo dos Robôs de Porto:
  - ◆ sistema finito, determinístico, estático
  - ◆ conhecimento completo
  - ◆ metas de satisfação
  - ◆ tempo implícito
  - ◆ planejamento *offline*
- Planejamento clássico é basicamente uma busca de caminho em um grafo
  - ◆ estados são nós
  - ◆ ações são arestas
- Esse é um problema trivial?



# Planejamento Clássico

- Computacionalmente muito difícil
  - ◆ generalização do exemplo dos Robôs de Porto:
    - » 5 localizações, 3 pilhas, 3 robôs, 100 *containers*
  - ◆ isso implica em  $10^{277}$  estados
    - » mais do que  $10^{190}$  vezes o número de partículas no universo!
- A grande maioria das pesquisas de IA (!!)  
tem sido sobre planejamento clássico
- Abordagem muito restritiva para tratar a maioria dos problemas de interesse prático
  - ◆ No entanto, muitas das idéias de soluções do planejamento clássico têm se mostrado úteis na resolução de problemas práticos



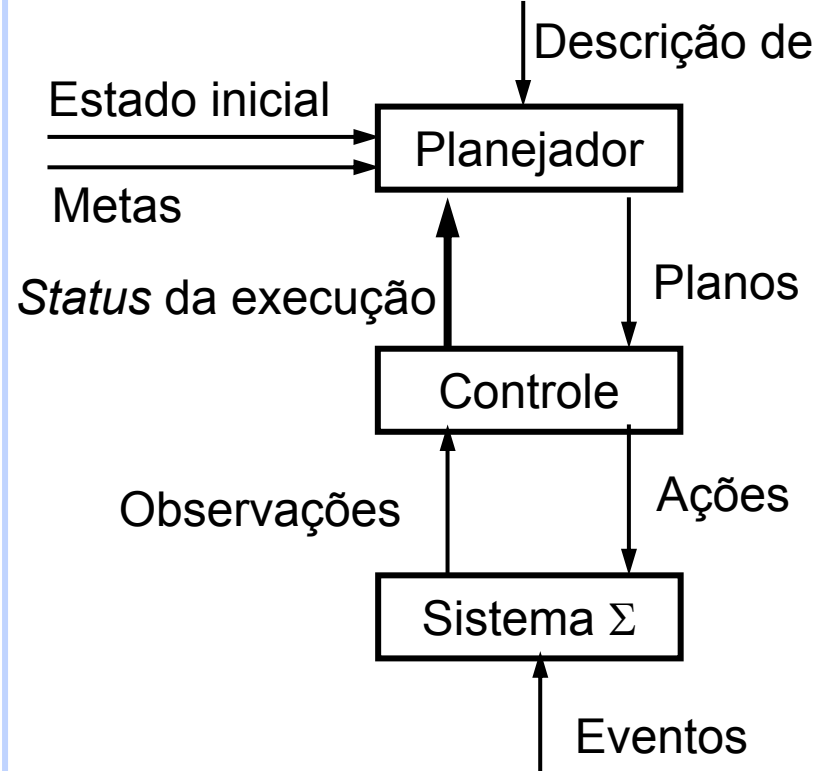
# Relaxando as Suposições

Relaxar A0 ( $\Sigma$  finito):

- ◆ Discreto, e.g. lógica de 1<sup>a</sup>. ordem
- ◆ Contínuo, e.g. ações com variáveis numéricas
- ◆ Seções:
  - » 2.4 (extensões de planeamento clássico)
  - » 10.5 (planejadores com regras de controle)
  - » 11.7 (planejamento HTN)
- ◆ Estudo de caso: Capítulo 21 (análise de manufaturabilidade)

Relaxar A1 ( $\Sigma$  totalmente observável):

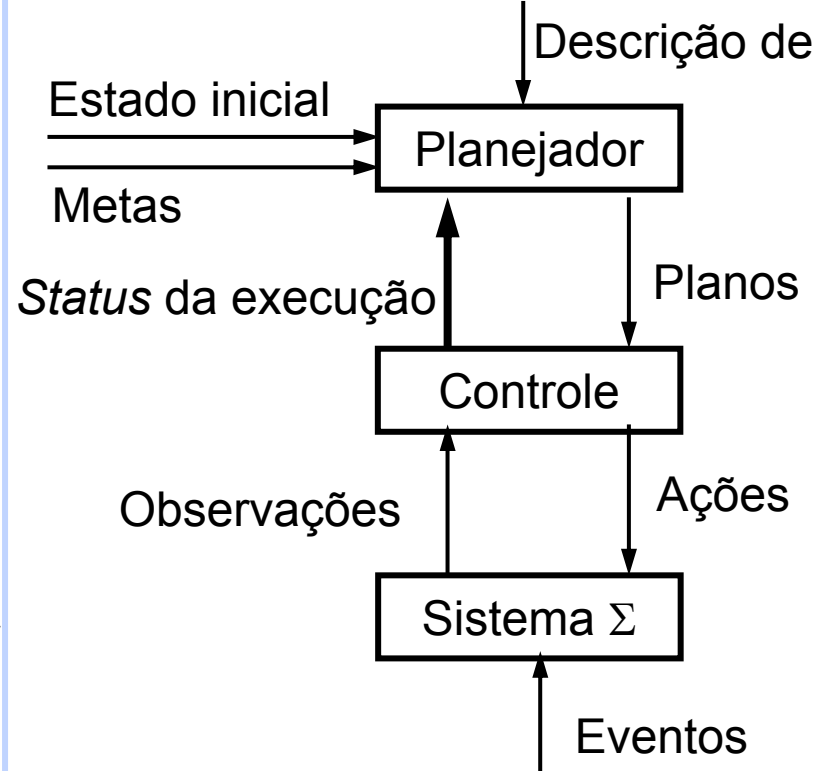
- ◆ Se não relaxarmos nenhuma outra restrição, então a única incerteza é sobre  $s_0$



$$\begin{aligned}\Sigma &= (S, A, E, \gamma) \\ S &= \{\text{estados}\} \\ A &= \{\text{ações}\} \\ E &= \{\text{eventos}\} \\ \gamma &: S \times (A \cup E) \rightarrow 2^S\end{aligned}$$

# Relaxando as Suposições

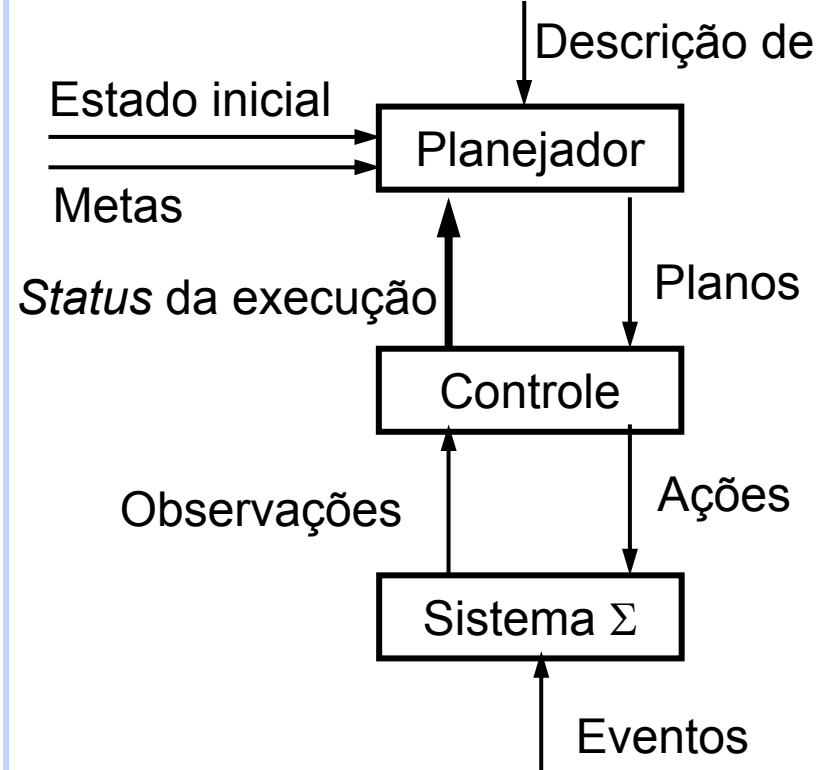
- Relaxar A2 ( $\Sigma$  determinístico):
  - ◆ Ações têm mais do que 1 saída (efeito) possível
  - ◆ Busca por políticas ou planos de contingência
  - ◆ Com probabilidades:
    - » *Discrete Markov Decision Processes* (MDPs)
    - » Capítulos 11
  - ◆ Sem probabilidades:
    - » Sistemas de transição não determinísticos => planejamento como verificação de modelos
    - » Capítulos 12, 18



$$\Sigma = (S, A, E, \gamma)$$
$$S = \{\text{estados}\}$$
$$A = \{\text{ações}\}$$
$$E = \{\text{eventos}\}$$
$$\gamma: S \times (A \cup E) \rightarrow 2^S$$

# Relaxando as Suposições

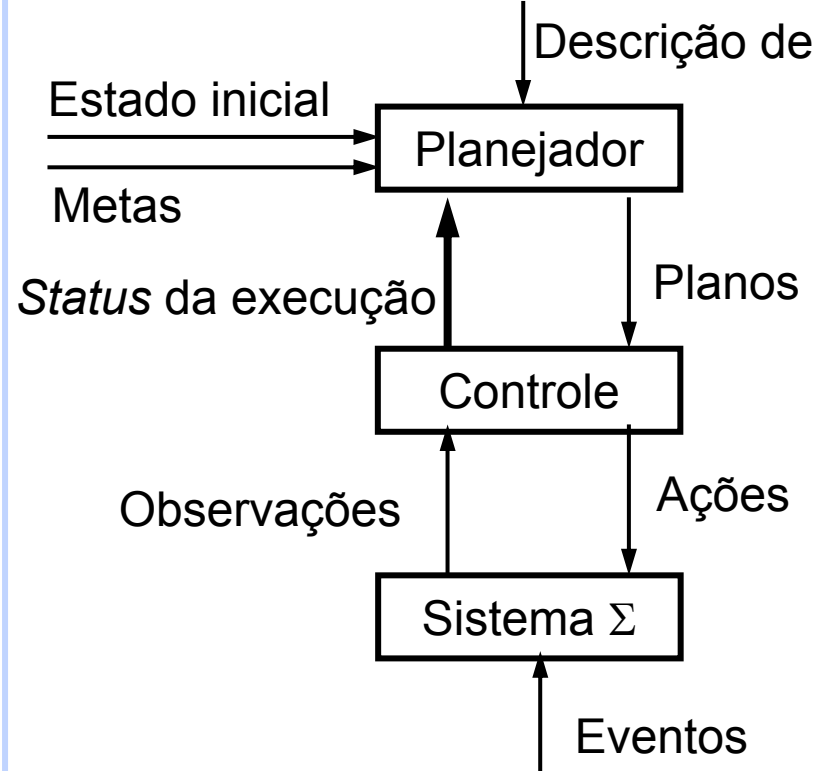
- Relaxar A1 e A2:
  - ◆ POMDPs finitos
    - » Planejar sobre *estados de crença*
    - » Tempo e espaço exponenciais
    - » Seção 16.3
- Relaxar A0 e A2:
  - ◆ MDPs contínuos or híbridos
    - » Teoria de controle da engenharia
- Relaxar A0, A1, e A2
  - ◆ POMDPs contínuos or híbridos
    - » Estudo de caso: Capítulo 20 (robótica)



$$\Sigma = (S, A, E, \gamma)$$
$$S = \{\text{estados}\}$$
$$A = \{\text{ações}\}$$
$$E = \{\text{eventos}\}$$
$$\gamma: S \times (A \cup E) \rightarrow 2^S$$

# Relaxando as Suposições

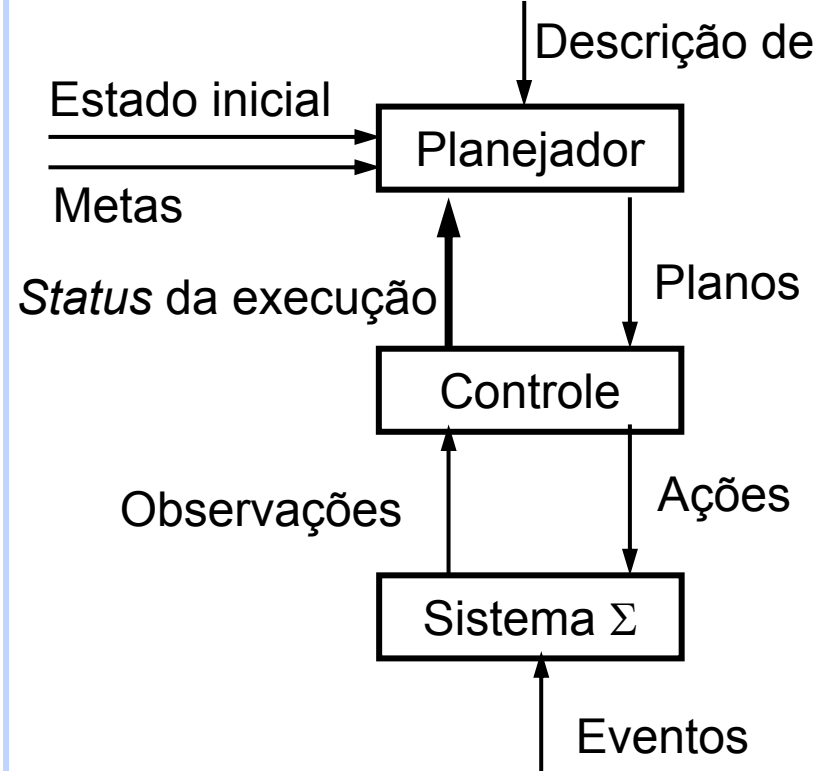
- Relaxar A3 ( $\Sigma$  estático):
  - ◆ Outros agentes ou ambientes dinâmicos
    - » Jogos finitos de soma-zero e informação perfeita (cursos introdutórios de IA)
  - ◆ Ambientes de comportamento aleatório
    - » Análise de decisão (business, pesquisa operacional)
    - » Algumas vezes pode ser mapeado em MDPs or POMDPs
  - ◆ Estudos de caso: Capítulos 19 (espaço), Capítulo 22 (evacuação de emergência)
- Relaxar A1 e A3
  - ◆ Jogos com informação imperfeita
  - ◆ Estudo de caso: Capítulo 23 (bridge)



$$\Sigma = (S, A, E, \gamma)$$
$$S = \{\text{estados}\}$$
$$A = \{\text{ações}\}$$
$$E = \{\text{eventos}\}$$
$$\gamma: S \times (A \cup E) \rightarrow 2^S$$

# Relaxando as Suposições

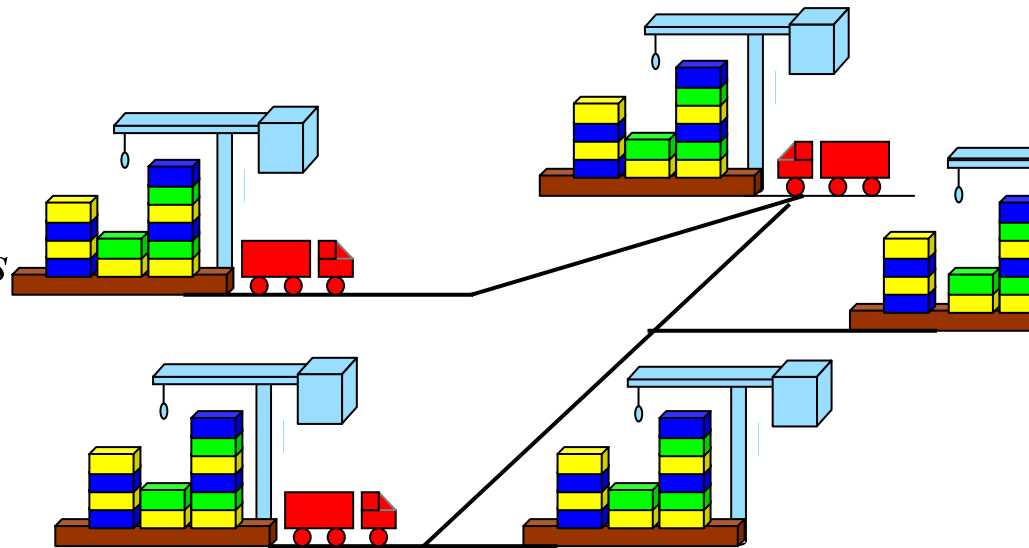
- Relaxar A5 (planos sequenciais) e A6 (tempo implícito):
  - ◆ Planejamento temporal
  - ◆ Capítulos 13, 14
- Relaxar A0, A5, A5
  - ◆ Planejamento e escalonamento de recursos
  - ◆ Capítulo 15
- Existem outras 247 combinações que não são discutidas no livro.



$$\Sigma = (S, A, E, \gamma)$$
$$S = \{\text{estados}\}$$
$$A = \{\text{ações}\}$$
$$E = \{\text{eventos}\}$$
$$\gamma: S \times (A \cup E) \rightarrow 2^S$$

# Dock Worker Robots

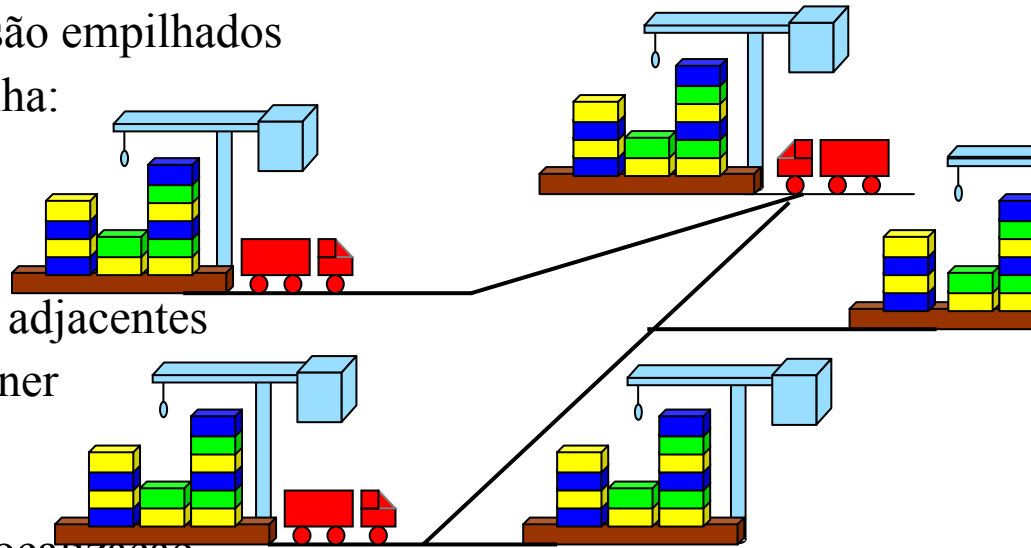
- Generalização do exemplo anterior
  - ◆ Um cais de porto com várias localizações
    - » e.g., docas, navios com docas, áreas de armazenagem, áreas de transferência de carga
  - ◆ *Containers*
    - » vão/vêm de navios
  - ◆ Carros Robôs
    - » Podem mover *containers*
  - ◆ Guindastes
    - » podem carregar ou descarregar *containers*





# Um exemplo de execução: Dock Worker Robots

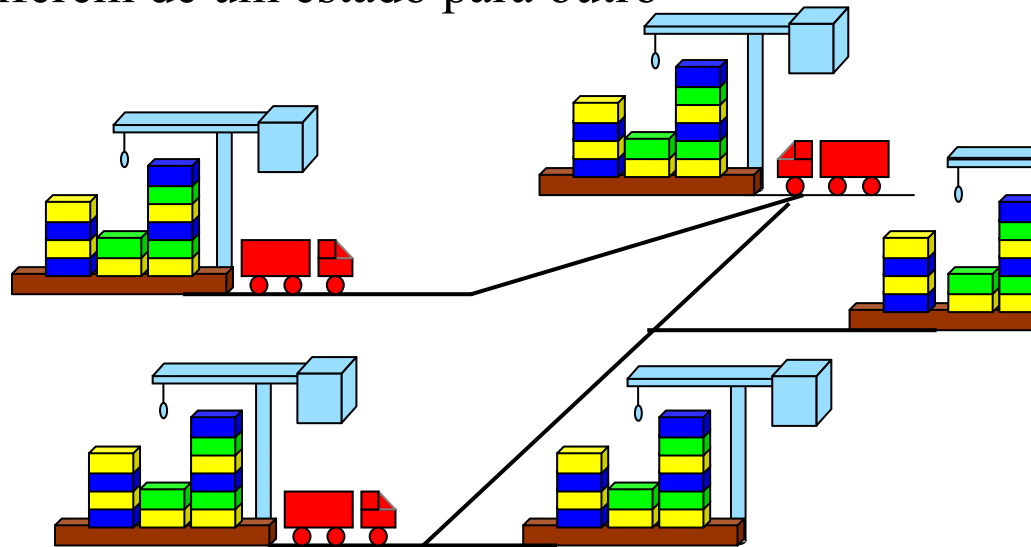
- **Localizações:**  $l_1, l_2, \dots$
- **Containers:**  $c_1, c_2, \dots$ 
  - ◆ Podem ser empilhados, carregados sobre os robôs, ou carregados pelos guindastes
- **Pilhas:**  $p_1, p_2, \dots$ 
  - ◆ Áreas fixas onde os containers são empilhados
  - ◆ Plataforma no fundo de cada pilha:  
*pallet*
- **Carros Robôs:**  $r_1, r_2, \dots$ 
  - ◆ Podem mover para localizações adjacentes
  - ◆ carregam no máximo um container
- **Guindaste:**  $k_1, k_2, \dots$ 
  - ◆ cada um pertence a uma única localização
  - ◆ carrega um container de uma pilha para um carro robô e vice-versa
  - ◆ Se há uma pilha em uma localização então deve haver também um guindaste na mesma localização



# A running example: Dock Worker Robots

- Relações fixas: é a mesma em todos os estados  
 $\text{adjacent}(l, l')$     $\text{attached}(p, l)$     $\text{belong}(k, l)$
- Relações dinâmicas (fluentes): diferem de um estado para outro

$\text{occupied}(l)$     $\text{at}(r, l)$   
 $\text{loaded}(r, c)$     $\text{unloaded}(r)$   
 $\text{holding}(k, c)$     $\text{empty}(k)$   
 $\text{in}(c, p)$     $\text{on}(c, c')$   
 $\text{top}(c, p)$     $\text{top}(\text{pallet}, p)$



- Ações:  
 $\text{take}(c, k, p)$     $\text{put}(c, k, p)$   
 $\text{load}(r, c, k)$     $\text{unload}(r)$     $\text{move}(r, l, l')$