



# Planejamento e ação no mundo real

---



# Tempo e Recursos em STRIPS

---

- STRIPS -> O que a ação faz
- Qual a duração de uma ação? Quando ela ocorre?
  - Ex: Entrega de cargas
    - O avião chegará quando concluir o vôo. Mas quando?
- Importante para aplicações de **escalonamento de linha de produção**
  - Tarefas, seqüência de ações, duração das ações, recursos
- Problema:
  - Determinar um escalonamento que minimize o tempo total exigido para completar todos os trabalhos respeitando as restrições de recursos



# Exemplo: Montagem de Automóvel

---

**Início** (Chassi(C1)  $\wedge$  Chassi(C2)  $\wedge$  Motor(E1, C1, 30)  $\wedge$  Motor(E2, C2, 60)  $\wedge$  Rodas(W1, C1, 30)  $\wedge$  Rodas(W2, C2, 15))

**Objetivo** (Terminado(C1)  $\wedge$  Terminado(C2))

**Ação** (AdicionarMotor(e, c, m),  
PRECOND: Motor(e, c, d)  $\wedge$  Chassi(c)  $\wedge$   $\sim$ MotorColocado(c),  
EFFECT: MotorColocado(c)  $\wedge$  Duração(d))

**Ação** (AdicionarRodas(w, c),  
PRECOND: Rodas(w, c, d)  $\wedge$  Chassi(c)  
EFFECT: RodasMontadas(c)  $\wedge$  Duração(d))

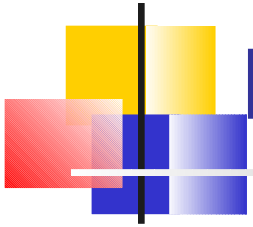
**Ação** (Inspeccionar(c),  
PRECOND: MotorColocado(c)  $\wedge$  RodasMontadas(c)  $\wedge$  Chassi(c)  
EFFECT: Terminado(c)  $\wedge$  Duração(10))



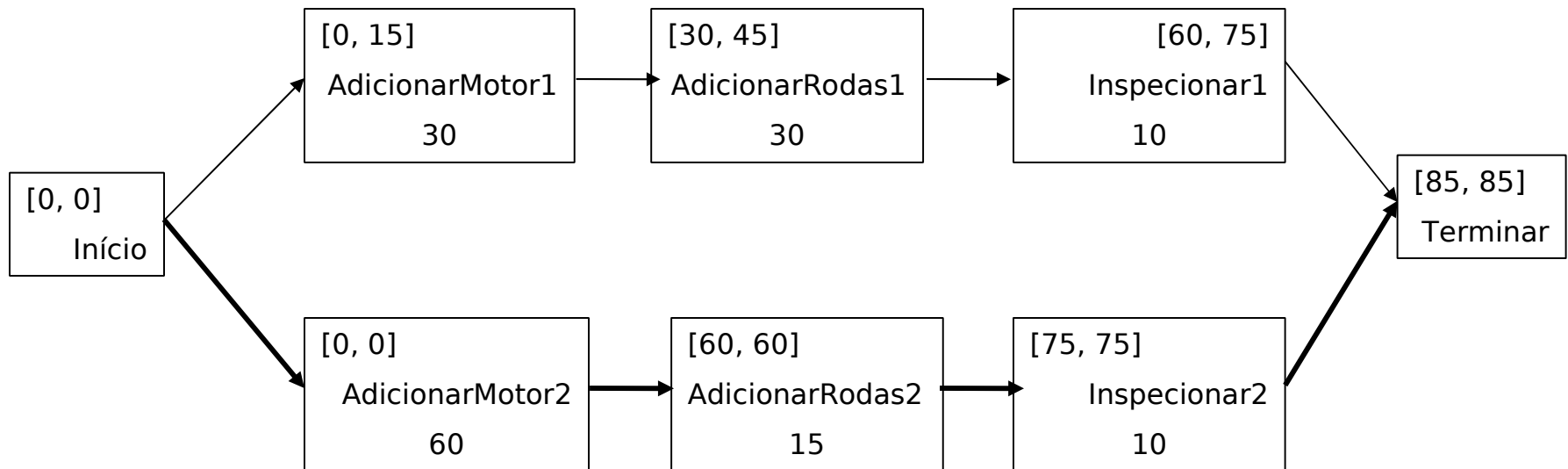
# Planejamento e Escalonamento

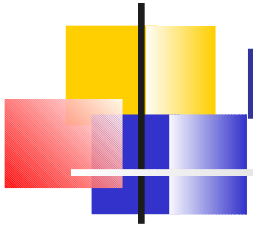
---

- Fase 1: Planejamento
  - Selecionar ações parcialmente ordenadas para satisfazer os objetivos do problema
  
- Fase 2: Escalonar
  - Adicionar informações temporais ao plano e assegurar que ele atenderá as restrições de recurso e prazo
  - Determinar os horários de início e término de cada ação
    - Primeiro horário possível e último horário possível
    - Método do caminho crítico

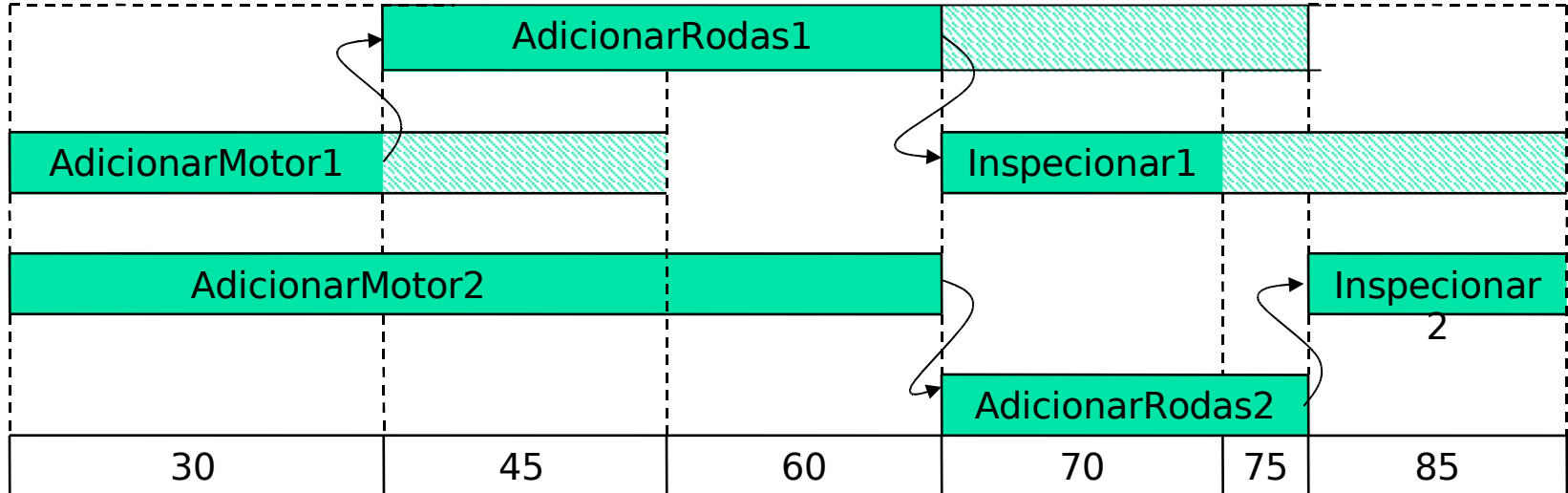


# Plano de ordem parcial





# Plano em função do tempo





# Recursos

---

- Exemplo:
  - Adição de um motor a um carro exige um guincho de motor
  - Se existe apenas um guincho então não podemos adicionar dois motores a dois carros ao mesmo tempo
- Recurso reutilizável
  - Ocupado somente durante a ação
- Recurso consumível
  - É um recurso (parcialmente) utilizado por uma ação
  - Pode ser manipulado como a representação do tempo
- Representação: RESOURCE: R(K)



# Exemplo – Representação Recursos

---

**Início** (Chassi(C1)  $\wedge$  Chassi(C2)  $\wedge$  Motor(E1, C1, 30)  $\wedge$  Motor(E2, C2, 60)  $\wedge$  Rodas(W1, C1, 30)  $\wedge$  Rodas(W2, C2, 15)  $\wedge$  GuinchosMotor(1)  $\wedge$  EstaçãoRodas(1)  $\wedge$  Inspetores(2))

**Objetivo** (Terminado(C1)  $\wedge$  Terminado(C2))

**Ação** (AdicionarMotor(e, c, m),  
PRECOND: Motor(e, c, d)  $\wedge$  Chassi(c)  $\wedge$   $\sim$ MotorColocado(c),  
EFFECT: MotorColocado(c)  $\wedge$  Duração(\_d),  
RESOURCE: GuinchosMotor(1))

**Ação** (AdicionarRodas(w, c),  
PRECOND: Rodas(w, c, d)  $\wedge$  Chassi(c)  
EFFECT: RodasMontadas(c)  $\wedge$  Duração(d),  
RESOURCE: EstaçãoRodas(1))

**Ação** (Inspeccionar(c),  
PRECOND: MotorColocado(c)  $\wedge$  RodasMontadas(c)  $\wedge$  Chassi(c)  
EFFECT: Terminado(c)  $\wedge$  Duração(10),  
RESOURCE: Inspetores(1))



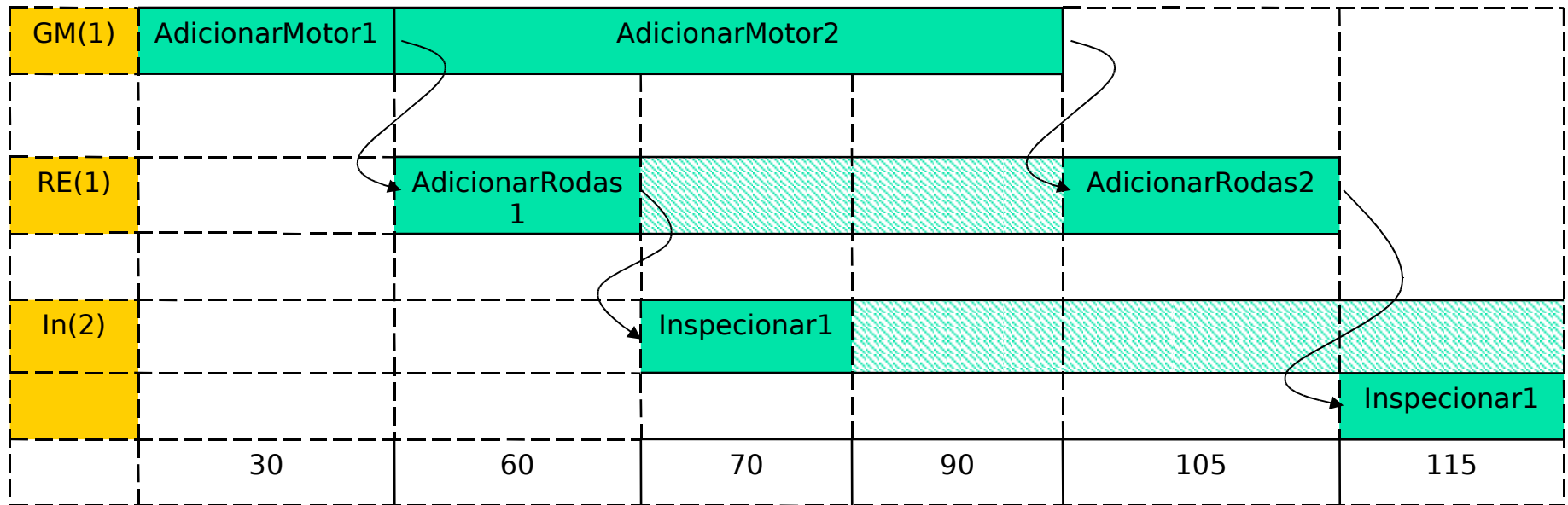
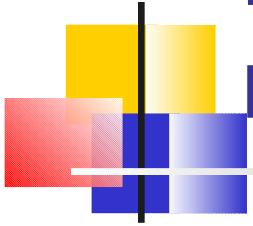


# Representação de Recursos

---

- A ação não poderá ser executada se o recurso não estiver disponível
- Efeito temporário de disponibilidade
- Agregação: agrupar objetos individuais em quantidade quando os objetos são indistinguíveis em relação ao propósito geral.
  - Exemplo: Inspetor(2) ao invés de Inspetor(I1) e Inspetor(I2)

# Plano em função do tempo e dos recursos





# Planejamento Hierárquico

---

- Problemas complexos = decomposição hierárquica
- Em cada nível da hierarquia uma tarefa é reduzida a um pequeno número de atividades no nível seguinte
- Quando soluções de alto nível sempre tem implementações satisfatórias de baixo nível
  - Algoritmos de planejamento mais eficientes
- Idéia => Combinar
  - Planejamento de Ordem Parcial (POP)
  - Redes Hierárquicas de tarefas (RHT)



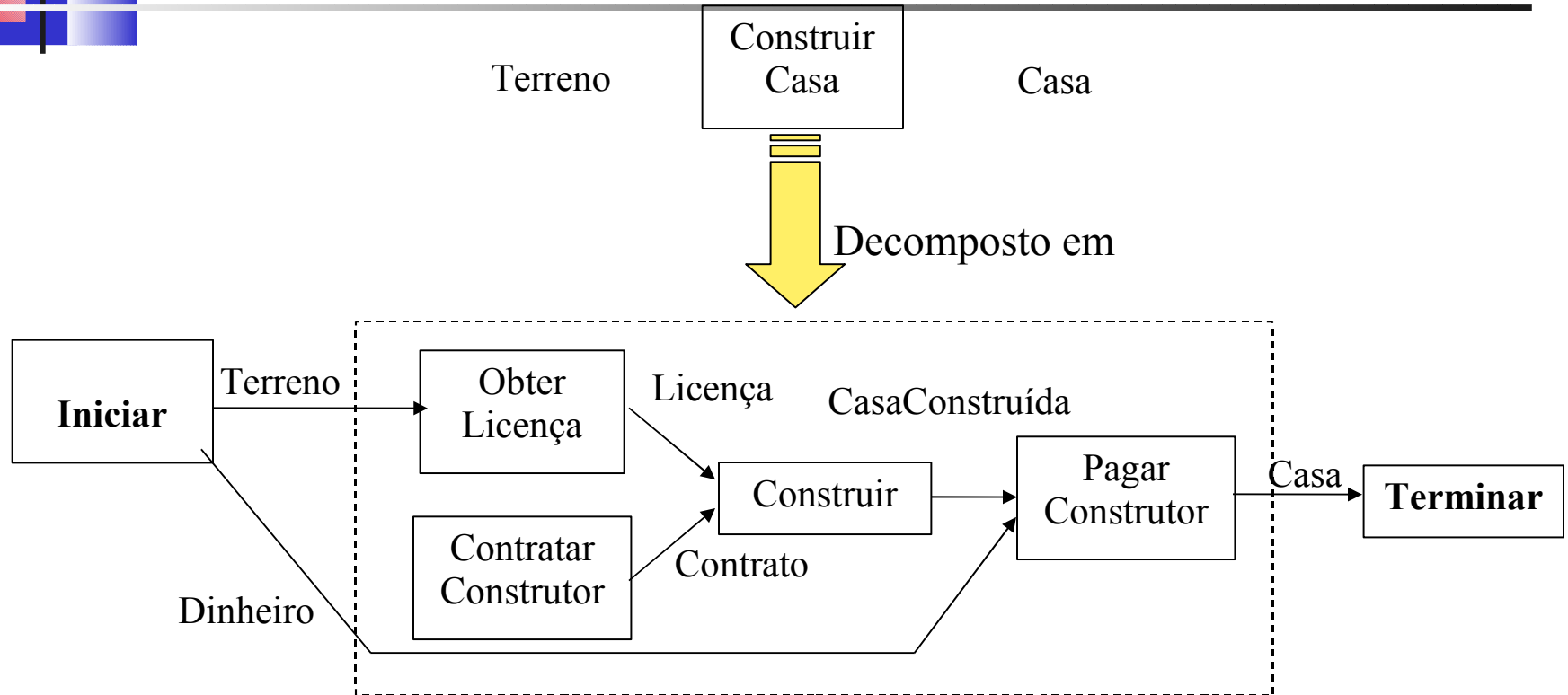
# Planejamento de RHT

---

- Plano inicial = descrição de nível muito alto do que deve ser feito
  - Ex: construir uma casa
- Planos são refinados pela aplicação de **decomposição de ações**
  - Que reduz uma ação de alto nível a um conjunto de ações de nível mais baixo
  - Construir casa pode se decompor em:
    - obter licença, contratar empreiteira, construir e pagar empreiteira
- O processo de decomposição continua até restarem somente ações primitivas no plano

# Decomposição

## Exemplo

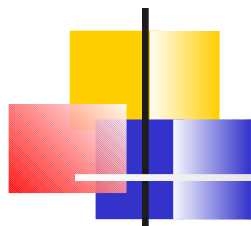




# Planejamento de RHT

---

- Uma ação de alto nível pode ter várias decomposições possíveis
  - Devemos ter uma biblioteca de métodos de decomposição de ações
  - Cada método pode ser uma expressão da forma:  
*Decompor(a, p)*
- As precondições e os efeitos de alto nível
  - São um subconjunto das precondições e efeitos da implementação primitiva (ações primitivas)
- Descrições de alto nível ignoram completamente todos os efeitos internos das decomposições
  - Reduz a complexidade, mas podem introduzir conflitos



start

a, b, c, d

a, b, c

Op 2

x, z

x, z, d

finish

Decomposto em

start

a, b, c, d

a, b

Op 2a

x, k

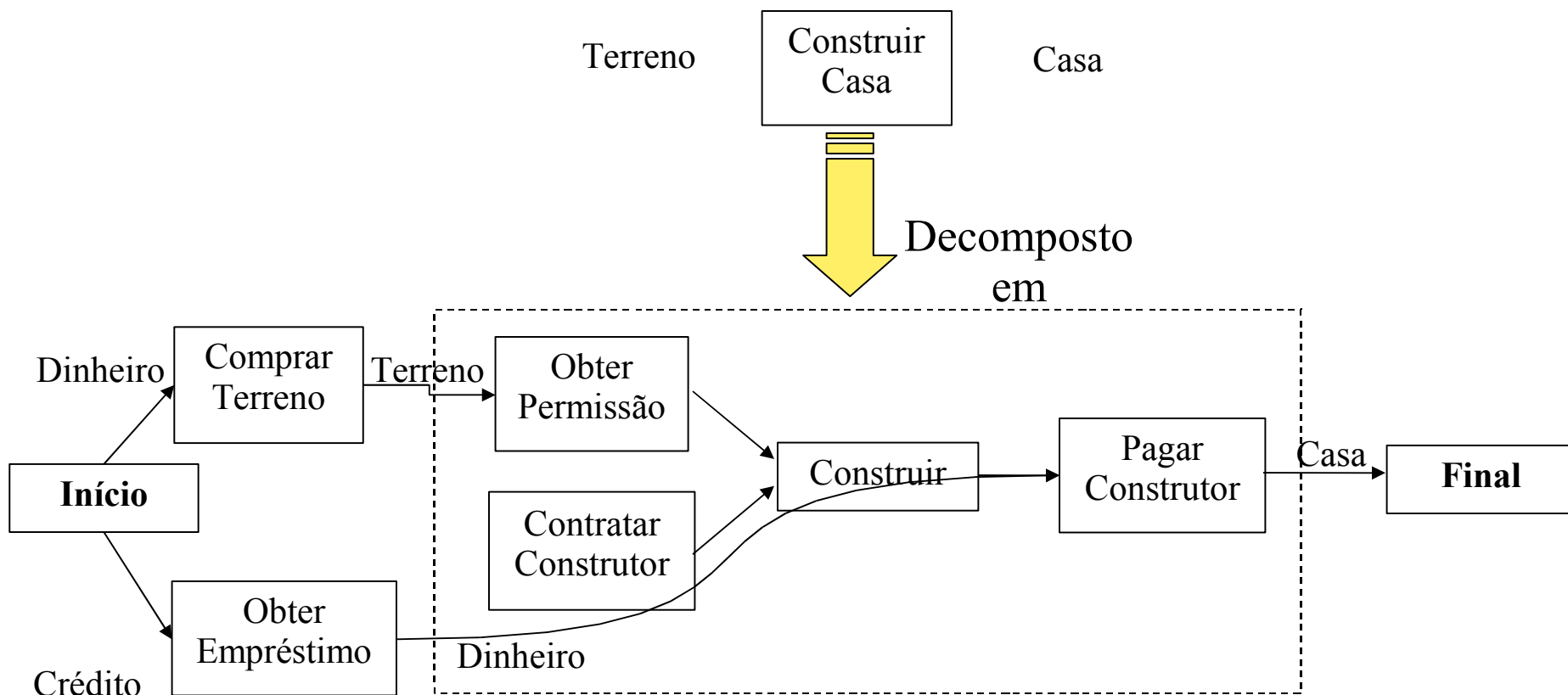
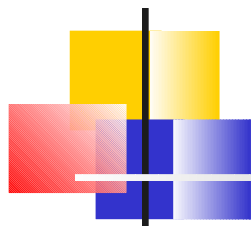
k, c

Op 2b

z

x, z, d

finish







# Problema

---

- Até agora... Domínios de planejamento clássico
  - Completamente observáveis, estáticos e determinísticos
  - Descrições das ações devem ser corretas e completas, descrevendo todas as conseqüências de forma exata.
- O mundo real
  - Parcialmente observável, não-determinístico, ou ambos
  - Descrição das ações são incorretas porque o mundo não necessariamente casa com o modelo do agente



# Indeterminismo

---

- Limitado

- Ações podem ter efeitos imprevisíveis, mas os efeitos possíveis podem se listados
- Ex: lançamento de uma moeda

- Ilimitado

- O conjunto de precondições ou efeitos possíveis é desconhecido ou grande demais para ser enumerado completamente
- Ex: domínio muito complexos ou dinâmicos, direção de veículos, planejamento econômico, estratégia militar



# Soluções – indeterminismo limitado

---

- Planejamento sem sensores
  - O algoritmo deve assegurar que o plano atingirá o objetivo em todas as circunstâncias possíveis, independente do estado inicial e dos resultados reais das ações
  - **Coerção** – o mundo pode ser forçado a entrar em um estado determinado
  - Nem sempre é aplicável
- Planejamento condicional (plano de contingência)
  - Constrói um plano condicional com diferentes ramificações para diferentes situações que podem surgir
  - Planejamento e depois execução



# Soluções – indeterminismo ilimitado

---

- Monitoramento de execução e replanejamento
  - Pode usar qq técnica de planejamento p/ construir o plano
  - Monitora a execução do plano através dos sensores
  - O replanejamento deve ser feito quando algo sai errado
- Planejamento contínuo
  - Persiste por todo o tempo
  - Pode manipular situações inesperadas, mesmo que ocorram enquanto o agente está construindo um plano
  - Podem formular novos objetivos



# Exemplo: Colorir cadeira e mesa da mesma cor

---

- Planejamento clássico
  - Estado inicial não está completamente especificado (qual cor?)
- Planejamento sem sensores
  - Abrir qq lata de tinta e pintar a cadeira e a mesa, forçando que ambas tenham a mesma cor
- Planejamento condicional
  - Perceber a cor da mesa e da cadeira, se forem iguais, pronto. Senão, examina o rótulo das latas de tinta, se houver alguma uma lata com a mesma cor de algum item, então pinta o outro item com a tinta, senão pinta ambos de qq cor.



# Exemplo: Colorir cadeira e mesa da mesma cor

---

- Replanejamento
  - Poderia gerar o mesmo plano do item anterior
  - Mas verificaria o resultado da ação.
  - Exemplo: se o agente tivesse deixado uma mancha na pintura, ele poderia replanejar e fazer a repetição da ação pintar
- Planejamento contínuo
  - Pode manipular eventos inesperados e revisar seus planos
  - Poderia acrescentar o objetivo de jantar usando a mobília, tendo que adiar o plano de pintar



# Planejamento condicional

---

- Ambientes completamente observáveis
  - Sempre conhece o estado atual do ambiente
  - Mas pode ser não-determinístico: o agente não pode prever o resultado das ações
  - Constrói um plano (tempo de planejamento) e verifica o estado do ambiente (tempo de execução) para decidir o que fazer
  - Para isso temos que estender STRIPS para que permita ações que tenham **efeitos disjuntivos**

**Ação** (Esquerda,  
PRECOND: EmDir,  
EFFECT: EmEsq **∨** EmDir)

Supondo que a  
movimentação para a  
esquerda às vezes falhe



# Planejamento condicional

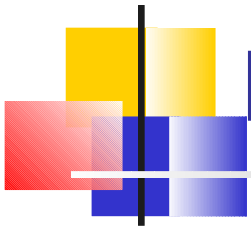
---

- STRIPS também deve permitir ações que tenham **efeitos condicionais**
- Sintaxe: “**when** <condição> : <efeito>”

**Ação** (Aspirar,  
PRECOND: ,  
EFFECT: (**when** EmEsq: LimpoEsq) ^ (**when** EmDir: LimpoDir))

**Ação** (Esquerda,      Se o quadrado de destino está limpo o  
PRECOND: EmDir,      agente pode depositar sujeira nele ao se  
EFFECT: EmEsq v (EmEsq <sup>mover</sup> ^ **when** LimpoEsqr: ~LimpoEsq))





# Planejamento condicional

---

- STRIPS também deve permitir **passos condicionais**
- Sintaxe: “**if** <teste> **then** <plano\_A> **else** <plano\_B>”
- Exemplo:  
*If EmEsq  $\wedge$  LimpoEsq then Direita else Aspirar*
- O espaço de busca é representado como um **grafo e-ou**



# Planejamento condicional

---

- Ambientes parcialmente observáveis
  - O estado inicial pertence a um conjunto de estados (estado de crença)
  - Ex: Problema do aspirador de pó sem sensores
  - O espaço de estados também pode ser representados por um grafo e-ou
  - Mas como os estados de crença são representados?
    - Conjuntos de descrições de estados completos
    - Sentenças lógicas que capturam exatamente o conjunto dos mundo possíveis no estado de crença
    - Proposições de conhecimento (hipótese do mundo fechado)