Extensão da linguagem para geração de Planos

- Além de passos bicondicionais: *planos parametrizados.*
- Ex: Cor(Cadeira,c) $\wedge$ Cor(Mesa,c)
- Suposição: a cadeira não está pintada e temos disponível tintas e pincel.
- Plano sugerido: [PercebeCor(Mesa),Sabe("Cor(Mesa,c)"), SelecionaTinta(c), Pinta(Cadeira,c)]
- Últimos passos parametrizados, até executar o plano agente não sabe qual cor de tinta utilizar.
- Variável c: *variável de execução.*
- Em STRIPS:

  Op(ACAO: PercebeCor(x),
  EFEITO: Sabe("Cor(x,c)"))

Extensão da linguagem para geração de Planos

- Em cálculo de situações:
  $\forall x, s \exists c Sabe(``Cor(x,c)'', Result(PercebeCor(x), s))$
- Variáveis x e c têm tratamento diferente logicamente.
- Variável c deve ser tratada como uma função de Skolem do objeto percebido.
- Na execução do plano, variáveis dee execução do tipo de c são unificadas apenas com outras variáveis do plano e não com ctes ou com elas mesmas (função de Skolem).
- Objetivo ter Cadeira e Mesa da mesma cor pode ser alcançado pintando as duas da mesma cor independentemente da cor da Mesa.
- Para prevenir estas situações no gerador de planos:
  Mantem(Cor(Mesa,x))

Extensão da linguagem para geração de Planos

- Outras extensões:
  While(Sabe("NaoComplPintada(Cadeira)"),[Pinta(Cadeira,c), VerificaCor(Cadeira)])
- Planos condicionais análogos a *programação automática*: geração de programas através de especificações.

Replanejamento

- Mesmo tendo planos condicionais, o q fazer qdo alguma coisa dá errada ou não estava prevista no plano original?
- *Monitoração de execução*: verifica se todas as pré-condições já foram alcançadas e se o objetivo já foi atingido através de anotação a cada passo do plano.
- Anotações: para cada passo cumprido, pré-condições necessárias para satisfazer os passos restantes.
- Utilização de estrutura de dados especial para representação eficiente de planos anotados: *tabela triangular*.

Replanejamento

- *Monitoração de Ações*: verificação de pré-condições a medida em que cada ação é executada.
- Mais simples, anotações não necessárias, mais próximo do mundo real, não gasta tempo executando partes do plano, no caso de outro agente ter modificado algum estado.
- Desvantagem: não percebe passos a frente.

Replanejamento

- Quando um plano falha? Causas classificadas em:
- *Não determinismo limitado*: ações podem ter efeitos inesperados, mas os possíveis efeitos podem ser enumerados e descritos de forma axiomática como parte das ações do plano.
- Combinação de CPOP e parte disjuntiva de POP-DUNC (cap 12), pode-se gerar planos condicionais que tenham ações com efeitos disjuntivos.
- *Não determinismo ilimitado*: conjunto de efeitos possíveis não enumerável. Exs: dirigir, planejamento econômico e estratégia militar.
- Solução: planejar com conjuntos enumeráveis de efeitos e **replanejar**.

```
function REPLANNING-AGENT(percept) returns an action
  static: KB, a knowledge base (includes action descriptions)
          p, an annotated plan, initially NoPlan
          q, an annotated plan, initially NoPlan
          G, a goal

  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
  current ← STATE-DESCRIPTION(KB, t)
  if p = NoPlan then
     p ← PLANNER(current, G, KB)
     q ← p
     if p = NoPlan or p is empty then return NoOp
  if PRECONDITIONS(p) not currently true in KB then
     p′ ← CHOOSE-BEST-CONTINUATION(current, q)
     p ← APPEND(PLANNER(current, PRECONDITIONS(p′), KB), p′)
     q ← p
  action ← FIRST(p)
  p ← REST(p)
  return action
```

Replanejamento com Monitoração de execução

- Pintar Cadeira:
  - Hip 1: Motor está com defeito e deixa algumas partes da cadeira sem pintar. No final do plano (não restam mais ações), agente testa se pré-condição do objetivo foi atingida. Olhando para o plano original, agente detecta q estado atual igual ao estado antes de pintar, então retorna ao passo anterior.
  - Hip 2: Tinta termina antes de acabar de pintar. Agente volta ao passo do plano antes do passo Pinta (SelecionaTinta).

Replanejamento com Monitoração de execução

- Loops: podem ocorrer por causa do plano ou por causa da interação entre o agente, ações e meio externo.
- Diferença entre looping plans e replanejamento com monitoração de execução: tempo em que objetivo vai ser atingido.
- Agentes podem "aprender" na falta de pré-condições explícitas (cap. 21).

Integração Planejamento-Execução

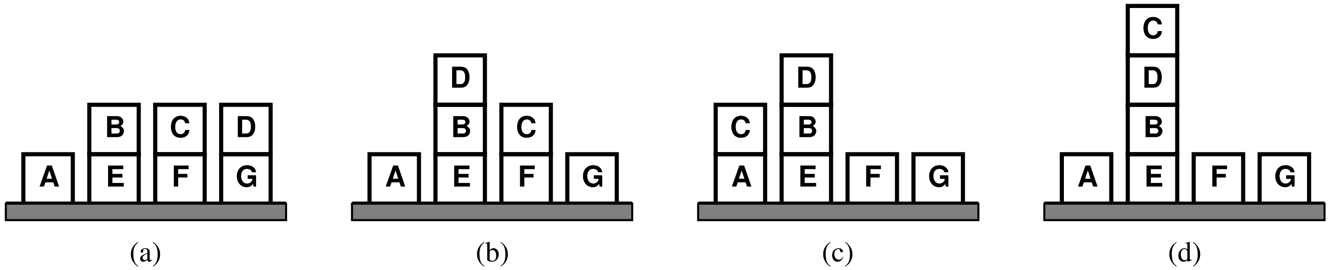
- Gerador de planos e executor integrados: *situated planning agent*.
- Agente está **sempre** planejando.
- Atividades: executa algumas ações do plano que estão prontas, refina o plano para resolver deficiências, refina o plano dada alguma info adicional obtida durante a execução, modifica o plano dadas modifs no ambiente etc.

Integração Planejamento-Execução

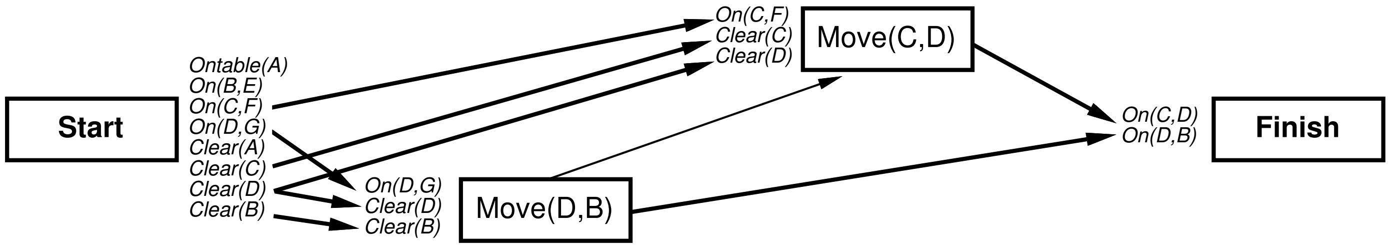
- Exemplo:
  - estado inicial: figura.
  - objetivo: Sobre(C,D) $\wedge$ Sobre(D,B).

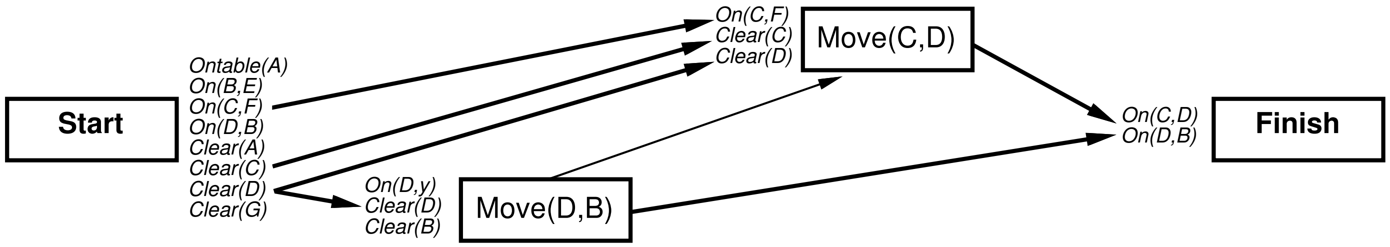
    Op(ACAO: Mover(x,y),
    PRECOND: TopoLimpo(x) $\wedge$ TopoLimpo(y) $\wedge$ Sobre(x,z),
    EFEITO: Sobre(x,y) $\wedge$ TopoLimpo(z) $\wedge$
    $\neg$ Sobre(x,z) $\wedge$ $\neg$ TopoLimpo(y))

Integração Planejamento-Execução

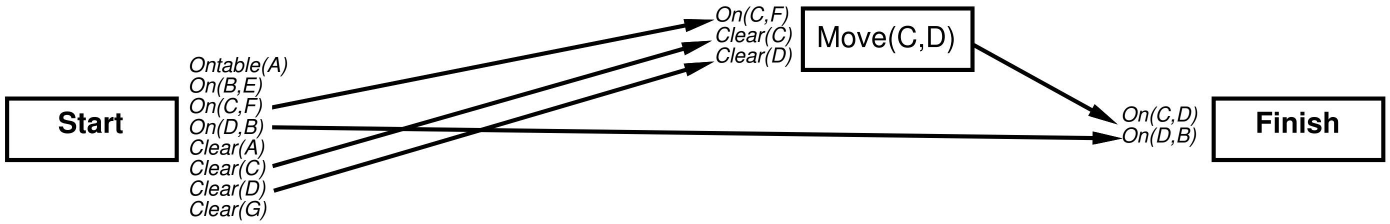


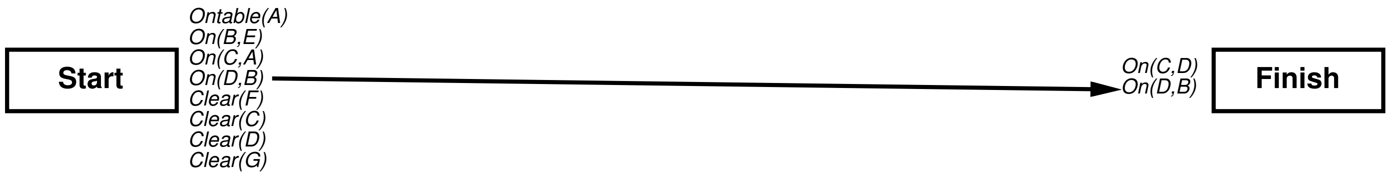
Integração Planejamento-Execução





Integração Planejamento-Execução





Integração Planejamento-Execução