

Planeadores

PLANEADORES

Lineares

2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeadores

-Planeamento envolve pesquisa em um Espaço de Estados:

- Como guiar a pesquisa?
- Como representar o Espaço de Estados
- Como avaliar as soluções?

-A pesquisa é feita com retrocesso nos pontos não deterministas

- Escolha de acções
- Instanciação de variáveis
- Ordenação de acções
- Escolha de sub-objectivos para resolver

2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeadores

-Direcção da pesquisa:

-Encadeamento Directo:

- 1 Escolha de acção cujas pré-condições são satisfeitas
- 2 Aplique e Altere descrição do Estado corrente
- 3 Até Objectivo estar satisfeito regresse a 1

-Encadeamento Inverso:

- 1 Escolha acção cujo efeito esteja de acordo com um (sub) objectivo
- 2 Junte pré-condições do operador ao conjunto de sub-objectivos ainda não atingidos (se fôr o caso)
- 3 Até o conjunto dos sub-objectivos não atingidos sêr vazio, regresse a 1

Planeadores

-Direcção da pesquisa (Vantagens / Desvantagens)

-Encadeamento Directo:

- Vantagem: Simples
- Desvantagem:
 - Factor de Ramificação grande
 - Pesquisa não focalizada

-Encadeamento Inverso:

- Vantagens:
 - Focalizada na resolução dos Objectivos
 - Normalmente mais eficiente
- Desvantagens:
 - Necessidade de raciocinar acerca das acções
 - Geralmente o encadeamento inverso (regressão) é incompleto

Planeadores

-Planeamento Linear:

- Pesquisa usando PILHA dos Objectivos ainda não atingidos
- Resolução de um Objectivo de cada vez
- Vantagens: Pesquisa simples e eficiente se objectivos independentes
- Desvantagens: Planos podem ser sub-óptimos. Incompleto

-Planeamento Não-Linear:

- Pesquisa usando um Conjunto de Objectivos
- Resolução concorrente de sub-objectivos
- Vantagens: Completo e pode produzir planos mais curtos
- Desvantagens: Espaço de Estados maior (objectivos podem ser dependentes)

Planeadores

-Plano resultante:

-Plano Totalmente Ordenado: o Plano é sempre uma sequência estrita de acções

-Vantagens: Algoritmo mais simples

-Desvantagem: Não há planos concorrentes e pode forçar a fazer decisões desnecessárias.

-Plano Parcialmente Ordenado: o Plano é uma ordenação parcial que pode ser linearizado antes da execução

-Vantagens: Compromissos adiados o mais possível e fácil de lidar com planos concorrentes

-Desvantagens: Difícil saber quais os objectivos a executar em cada instante

Planeadores

-Descrição do Estado corrente:

-Usando Base de Factos inicial e actualizando-a através das listas Junte e Apague dos Operadores do Plano

-Problemas: tendo de haver retrocesso na computação do Plano pode ser impossível devido a efeitos colaterais.

-Espaço de Estados:

- Nodos: Estados.
- Arcos: Acções.
- Plano: Caminho nesse Espaço

Planeadores

-Espaço de Planos:

-Espaço de estados é conjunto de planos. Arcos são operadores de plano (junte, ordene,...). A ordem de pesquisa é diferente da ordem de execução do plano

-Encontrar o Plano solução é como provar de axiomas e da descrição do estado inicial, e dos estados entretanto derivados, o Objectivo final.

-A Lógica de Predicados é um formalismo apropriado para este algoritmo

Planeadores

-Planeamento Parcialmente Ordenado (PPO)

-No Cálculo situacional um problema de planeamento é representado por fórmulas lógicas descrevendo:

Estado Inicial, Estado Final, Operadores

A Linguagem dos Planeadores "tipo-STRIPS" usa Operadores descrevendo Acções suas pré-condições e efeitos.

Planeadores

Plano:

Conjunto de passos (Operadores)

Conjunto de Restrições de Ordenação dos passos

$(S_i < S_j)$

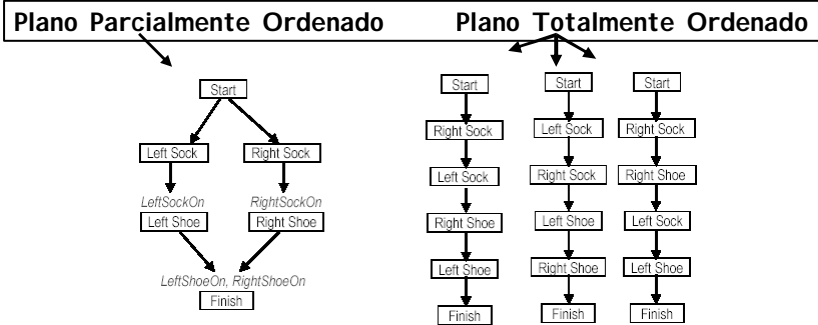
Conjunto de ligações das Variáveis

Conjunto das ligações causais $(S_i \rightarrow^c S_j)$

i.e.: S_i realiza a pré-condição c para S_j

Robótica Inteligente

Planos Parcial e Totalmente Ordenados



2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeadores

Algoritmo de Planeamento Parcialmente Ordenado (PPO)

{ PPO começa com plano parcial mínimo e estende cada passo pela realização de cada pré-condição desse passo}

FUNÇÃO **PPO**(início,objectivo,operadores) RETORNA plano

plano ← Faz_Plano_Mínimo(início, objectivo) {aplica o 1ºop}

REPETIR

SE solução(plano) RETORNAR(plano)

{ $S_{i,c}$ pré-condição $S_{i,c}$ ← Seleccionar_Sub-objectivo (plano)

não satisfeita Seleccionar_Operador(plano,Operadores, $S_{i,c}$)

de S_i } Resolver_inconsistências (plano)

FIM

2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeadores

Algoritmo de Planeamento Parcialmente Ordenado (PPO)
FUNÇÃO seleccionar_Sub-objectivo(plano) RETORNAR($S_{i,c}$)
 Escolher_passo(S_i , plano)
 RETORNAR pré-condições não satisfeita c de S_i , $S_{i,c}$
FUNÇÃO Seleccionar_Operador(plano, Operadores, $S_{i,c}$)
 Escolher passo S_j (operador do plano) tendo c como efeito
 SE \neg existe S_j ENTÃO FALHA
 SENÃO Juntar ligação_causal $S_j \rightarrow c$ a Ligações(plano)
 Juntar Restrições de Ordem $S_j < S_i$ a
 Ordenação(plano)
 SE S_j novo passo ENTÃO
 Juntar S_j a Passos(plano)
 Juntar início $< S_j <$ Objectivo a Ordenação(plano)

2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeadores

Algoritmo de Planeamento Parcialmente Ordenado (PPO)
PROCEDIMENTO resolver_inconsistências(plano)
 PARA CADA
 S_{inc} ameaçando ligação causal $S_a \rightarrow c$ S_b em Ligações(plano)
 FAZER
 {escolher promover} Junte $S_{inc} < S_a$ a Ordenação(plano)
 {ou despromover} Junte $S_b < S_{inc}$ a Ordenação(plano)
 SE NÃO Consistente(plano)
 ENTÃO FALHA

FIM

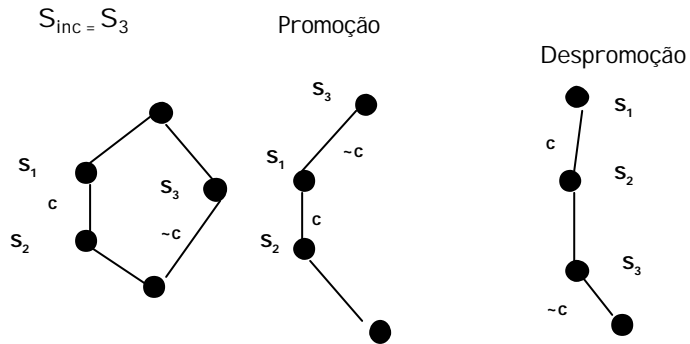
2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeadores

Algoritmo de Planeamento Parcialmente Ordenado (PPO)



2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeador tipo- "STRIPS"

- Representação das acções simplificada relativamente a representação em Lógica:

- Consideram-se só conjunções de pré-condições ou efeitos
- Não há condicionais
- Não há quantificações

- Estado é uma Base de Factos (literais instanciados)
- O que não está na Base de Factos não é verdadeiro
- Representam-se os efeitos do operador através de duas listas que actuam na Base de Factos (JUNTE e APAGUE)
- Não há representação explícita do tempo
- Não há regras de inferência explícitas

2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Robótica Inteligente

Robô Shakey (Stanford)

- Move from place to place:

Go(y):

PRECOND: At(Shakey,x)
In(x,r) \wedge In(y,r)
On(Shakey,Floor)

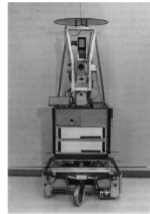
EFFECT: At(y)

- Push movable objects:

Push(b, x, y):

PRECOND: Pushable(b)
At(b,x)
At(Shakey,x)
In(x,r) \wedge In(y,r)
On(Shakey,Floor)

EFFECT: At(b,y)



- Climb onto rigid objects:

Climb(b):

PRECOND: Climbable(b)
At(Shakey,x) \wedge At(b,x)
On(Shakey,Floor)

EFFECT: On(Shakey,b)

2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeador tipo- "STRIPS"

- STRIPS (Stefike) é um **Planeador** usando:

- PILHA de OBJECTIVOS

-Técnica usada para resolver alguns dos problemas que incluem Objectivos compostos que não podem interagir.

-Utiliza-se uma só PILHA quer para Objectivos quer para Operadores usados para os satisfazer.

-São necessários ainda:

- uma BASE de FACTOS descrevendo a situação corrente

- Conjunto de OPERADORES com as respectivas LISTAS:

-Pré-Condições

-Junte

-Apague

2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeador tipo- "STRIPS"

- Representação das Acções no STRIPS (no "mundo dos blocos"):

-Esquemas de **Operadores**:

-PEGAR_da_mesa(a)

Pré-Condições: Bloco(a), Manip_livre, Limpo(a), Em(a,Mesa)

Junte: Segura(a)

Apague: Manip_livre, Em(a,Mesa)

-RETIRAR_de_outro_bloco(a)

Pré-Condições: Bloco(a), Manip_livre, Limpo(a),
Em(b,a),Bloco(b)

Junte: Segura(a), Limpo(b)

Apague: Manip_livre, Em(a,b)

Planeador tipo- "STRIPS"

-Esquemas de Operadores:

-LARGAR_na_mesa(a)

Pré-Condições: Bloco(a), Segura(a)

Junte: Manip_livre, Em(a,Mesa)

Apague: Segura(a)

-EMPILHAR(a,b)

Pré-Condições: Bloco(a), Segura(a)
Bloco(b), Limpo(b)

Junte: Manip_livre, Em(a,b)

Apague: Segura(a), Limpo(b)

Planeador tipo- "STRIPS"

- Técnicas para tornar mais eficiente o algoritmo:
- Detecção de "folhas mortas". Chegados a estados dos quais não se pode progredir para o objectivo, retrocede-se
- Cortar passos que se afastam da solução (em encadeamento directo). Por exemplo:
 - Se se juntar sub-objectivos incompatíveis com outros que se devem atingir;
 - Se para se resolver A é preciso resolver B, C e de novo A (ciclos).

Planeador tipo- "STRIPS"

- Mesmo que os problemas sejam "quase-decomponíveis" assume-se a decomponibilidade e resolve-se. Compara-se a solução intermédia com o objectivo e tenta-se anular a diferença (esperando que seja menor que a inicial).
- Usar o mais possível a técnica do "least commitment"
- Ex: adiar o mais possível a ordenação das operações mantendo-as em paralelo.

Planeador tipo- "STRIPS"

-STRIPS($e_{inicial}$, objectivos)

estado= $e_{inicial}$; plano=[]; pilha=[]

Colocar objectivos na pilha

Repetir até pilha vazia

Se topo da pilha=objectivo unificável com estado

Então retirar da pilha

Senão Se topo da pilha é objectivo conjuntivo

Então ordenar sub-objectivos e colocá-los na pilha

Senão Se topo da pilha objectivo simples

Então seleccione operador op cuja lista-junte unifica com objectivo

substitua objectivo pelo operador

coloque pré-condições do op na pilha

Senão Se topo da pilha é operador

retire operador

estado=aplicar($operador$,estado);plano=[plano, op]

2003 / LEIC

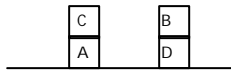
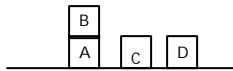
Eugénio Oliveira/FEUP

Robótica

Planeador tipo- "STRIPS"

Início

Objectivo



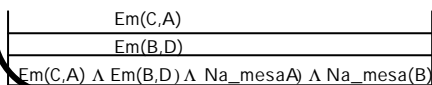
Modelo do Mundo:

$Em(B,A) \wedge Na_mesa(C) \wedge$

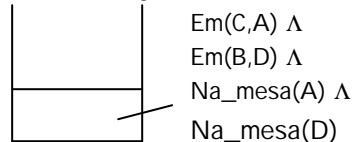
$Na_mesa(D) \wedge Na_mesa(A) \wedge$

$Manip_livre$

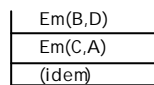
1ª hipótese:



Pilha dos Objectivos:



2ª hipótese (solução trivial)



2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeador tipo- "STRIPS"

Limpo(A)
Segura(C)
Limpo(A) \wedge Segura(C)
COLOCAR(C,A)
Em(B,D)
Objectivo

Em(C,A). \rightarrow COLOCAR(C,A)

Esta ordem porque uma heurística diz que Segura (x) deve ser a última acção da conjunção

Limpo(A) é verdadeiro? Não! Operador a usar: RETIRAR(x,A)

Em(x,A)
Limpo(x)
Manip_livre
Em(x,A) \wedge Limpo(x) \wedge Manip_livre
* RETIRAR(x,A)
Segura(C)
Limpo(A) \wedge Segura(C)
* COLOCAR(C,A)
Em(B,D)
Objectivo

Em(x,A) é Em(B,A) na base de dados
 Limpo(B) ? "Axioma de Enquadramento"
 se nada em B então Limpo(B)
 Manip_livre é verdade

Planeador tipo- "STRIPS"

A consideração de uma posição na PILHA com a conjunção de todas as condições que depois aparecem isoladamente noutras posições não é supérflua.

Devem testar-se as combinações dos sub-objectivos como precaução.

Poderia acontecer que para satisfazer uma das condições isoladamente se tivesse desfeito outro já entretanto satisfeito

No caso vertente não é o que se passa.

Planeador tipo- "STRIPS"

Estado Actual:

*RETIRAR(B,A)
Segura(C) \wedge Limpo(A)
*COLOCAR(C,A)
Em(B,D)
Objectivo

As pré-condições do operador são aplicáveis, logo cria-se novo **Modelo do Mundo** com as listas Junte e Apague do operador RETIRAR(B,A)

1º OPERADOR **RETIRAR(B,A)**

Modelo do Mundo

Segura(B) \wedge	Segura(C)
Limpo(A) \wedge	Segura(C) \wedge Limpo(A)
Na_mesa(A) \wedge	*COLOCAR(C,A)
Na_mesa(D) \wedge	Em(B,D)
Na_mesa(C)	Objectivo

para resolver Segura(C) duas hipóteses:
1ª PEGAR(C)
2ª RETIRAR(C,x)

Planeador tipo- "STRIPS"

Sem olhar à frente ("look ahead") criámos duas possíveis pilhas:

Na_mesa(C)
Limpo(C)
Manip_livre
M \wedge Limpo(C) \wedge Na_mesa(C)
PEGAR(C)
Segura(C) \wedge Limpo(A)
COLOCAR(C,A)
Em(B,D)
Objectivo

1ª hipótese

Em(C,x)
Limpo(C)
Manip_livre
M \wedge Em(C,x) \wedge Limpo(C)
RETIRAR(C,x)
Segura(C) \wedge Limpo(A)
COLOCAR(C,A)
Em(B,D)
Objectivo

2ª hipótese

Planeador tipo- "STRIPS"

Comparando os Objectivos com a Base de Factos, é preferível

A 1ª hipótese porque.

Na_mesa(C) é verdade

Optando pela 2ª hipótese, teríamos que garantir $Em(C,x)$ o que implicava COLOCAR(C,x).

Nas pré-condições deste Operador existe Segura(C) que é precisamente O que estamos a tentar resolver. Isto é entrávamos em ciclo.

Na 1ª hipótese, $Na_mesa(C) \wedge Limpo(C)$ são verdadeiros. Manip_livre não Porque é verdadeiro Segura(B).

Podemos então usar dois operadores cujo resultado é Manip_livre:

COLOCAR(x,y)
POUSAR(x)

Planeador tipo- "STRIPS"

Olhando "à frente", queremos $Em(B,D)$, logo é mais eficiente seleccionar COLOCAR(x,y) com $x=B$ e $y=D$

Como o algoritmo vai descobrir qual seleccionar?

Compara as listas Junte dos dois operadores seleccionáveis com Os Objectivos na PI LHA e escolhe aquele que resolveria algum deles.

Isto é escolhe COLOCAR(B,D)

Planeador tipo- "STRIPS"

Nova situação da pilhas:

Limpo(D)
Segura(B)
Limpo(D) \wedge Segura (B)
*COLOCAR(B,D)
M_IALimpo(C) \wedge Na_mesa(C)
PEGAR(C)
Segura(C) \wedge Limpo(A)
COLOCAR(C,A)
Em(B,D)
Objectivo

Limpo(D) \wedge Segura(B)
verdadeiros
implica 2º OPERADOR:
COLOCAR(B,D)

Novo Modelo do mundo:

Na_mesa(A) \wedge
Na_mesa(C) \wedge
Na_mesa(D) \wedge
Em(B,D) \wedge
Manip_livre



2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeador tipo- "STRIPS"

Todas as Pré-condições de PEGAR(C) são verdadeiras.
Então: 3º OPERADOR PEGAR(C)

Todas as pré-condições de COLOCAR(C,A) são verdadeiras.
Então: 4º OPERADOR COLOCAR(C,A)

Agora o 2º Objectivo: Em(B,D) já tinha sido satisfeito.
Finalmente testar o objectivo conjunto.

O Gerador Automático de Planos retorna então o Plano:

RETIRAR(B,A)
COLOCAR(B,D)
PEGAR(C)
COLOCAR(C,A)

Usaram-se heurísticas para detectar passos incorrectos e interacção entre objectivos

Mas Infelizmente esta técnica não é eficaz em muitos casos...

2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

Planeador tipo- "STRIPS"

VANTAGENS dos Planeadores tipo-STRIPS:

Espaço de pesquisa reduzido (um objectivo de cada vez)
Bom se os objectivos são independentes
Planeamento Linear é um Algoritmo Correcto

DESVANTAGENS dos Planeadores tipo-STRIPS:

Pode produzir soluções sub-óptimas
Planeamento linear é Não Completo

Planeador tipo- "STRIPS"

Problemas com o Planeador STRIPS:

Pode propôr acções irreversíveis que inviabilizam o Plano

OPERADORES:

Carregar(c,a)

Descarregar(c,loc)

Mover(a,nloc)

Pré-cond:

$Em(a,loc) \wedge Em(c,loc)$

$Dentro(c,a) \wedge Em(a,loc)$

$Em(a,loc) \wedge Comb(a)$

Junte:

$Dentro(a,c)$

$Em(c,loc)$

$Em(a,nloc)$

Apague:

$Em(c,loc)$

$Dentro(c,a)$

$Em(a,loc) \wedge Comb(a)$

Ex: Inicio { $Em(Ob1,La) \wedge Em(Ob2,La) \wedge Em(A,La) \wedge Comb(A)$ }
Objectivo { $Em(Ob1,Lb) \wedge Em(Ob2,Lb)$ }

Planeador tipo- "STRIPS"

Problemas com o Planeador STRIPS:

Pode propôr acções irreversíveis que inviabilizam o Plano

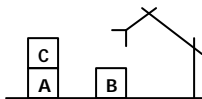
Utilizando o **Planeamento Linear** resolvia primeiro um sub-objectivo e depois tentaria resolver o outro sub-objectivo:

Mover(A,Lb), Descarregar(Ob1,Lb), ...

mas agora - Comb(A) podia ser verdadeiro e isso inviabilizaria a satisfação do Segundo Objectivo

Planeador tipo- "STRIPS"

Anomalia de "Sussman" mal resolvida por Planeadores Lineares:



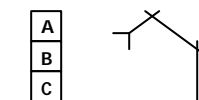
Início

Modelos do Mundo:

Em(C,A) \wedge
 Na_mesa(A) \wedge
 Na_mesa(B) \wedge
 Manip_livre

1ª hipótese

Em(A,B)
Em(B,C)
Em(A,B) \wedge Em(B,C)



Fim

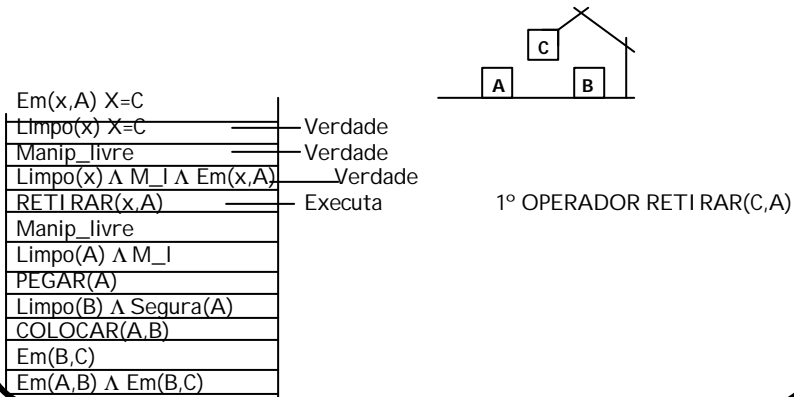
Em(A,B) \wedge
 Em(B,C) \wedge
 Na_mesa(C)

2ª hipótese

Em(B,C)
Em(A,B)
Em(B,C) \wedge Em(A,B)

Planeador tipo- "STRIPS"

Anomalia de "Sussman" mal resolvido por Planeadores Lineares:



2003 / LEIC

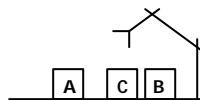
Eugénio Oliveira/FEUP

Robótica

Planeador tipo- "STRIPS"

Anomalia de "Sussman" mal resolvido por Planeadores Lineares:

Para tornar Verdadeiro Manip_livre
 Implica executar POU SAR(C)
 Então retira da PI LHA Manip_livre
 Até $Em(B,C)$



OPERADORES a executar:

- 1º RETI RAR(C,A)
- 2º POU SAR(C)
- 3º PEGAR(A)
- 4º COLOCAR(A,B)

Modelo do Mundo:

$Na_mesa(B) \wedge Em(A,B) \wedge$
 $Na_mesa(C) \wedge Manip_livre$

2003 / LEIC

Eugénio Oliveira/FEUP

Robótica

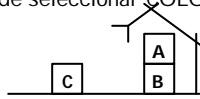
Planeador tipo- "STRIPS"

Anomalia de "Sussman" mal resolvido por Planeadores Lineares:

Mas agora, para solucionar Em(B,C) temos de seleccionar COLOCAR(B,C)
O que vai implicar:

OPERADORES a executar:

- 5º RETIRAR(A,B)
- 6º POUSAR(A)
- 7º PEGAR(B)
- 8º COLOCAR(B,C)
- 9º PEGAR(A)
- 10º COLOCAR(A,B)



Resolveu... Mas é muito ineficiente. Tal se verificaria igualmente com a ordem Contrária de resolução dos sub-objectivos.

Planeador tipo- "STRIPS"

Como resolver esta ineficiência?

Há duas hipóteses de solução:

1ª Hipótese:

Obter um Plano como anteriormente e depois retirar todas as acções do plano que imediatamente a seguir são desfeitas.

Para o caso concreto do exemplo deve cortar-se 4 com 5 e depois 3 com 6.

Resultava o seguinte Plano:

- 1º RETIRAR(C,A)
- 2º POUSAR(C)
- 3º PEGAR(B)
- 4º COLOCAR(B,C)
- 5º PEGAR(A)
- 6º COLOCAR(A,B)

Mas em tarefas complexas pode ser difícil computar um plano complexo
Que depois se desperdiça pois se deve simplificar!

Planeador tipo- "STRIPS"

2ª Hipótese:

Construir directamente um **Plano mais eficiente!**

Seguem-se agora algumas técnicas de Planeamento que tentam responder a estes Problemas.

Planeamento Não-Linear