

Planeamento em Robótica

PLANEAMENTO AUTOMÁTICO

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

- Em Inteligência Artificial, “**Geração Automática de Planos**” estuda a resolução de um problema original através da decomposição em sub-problemas mais elementares, propondo uma resolução para estas atendendo à sua possível interacção.

-É característica do Planeamento a computação dos vários passos da resolução **ANTES** da execução. O Plano resultante deve resolver o problema (a tarefa) original apresentada.

-Planear é importante sobretudo quando os passos necessários à resolução de um problema não podem ser desfeitos ou ignorados (ex: jogo de xadrez). Nos casos em que possam ser desfeitos (ex: puzzles) ou ignorados (demonstração de teoremas) não é tão importante.

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

- Domínios para Planeamento:
 - Planeamento das acções de um Robô
 - Planeamento na Produção
 - Planeamento no processamento de Imagens
 - Planeamento na Logística do transporte
 - Gestão de Crises
 - ...

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

- **Dificuldades** inerentes ao Planeamento:
 - Representação do ambiente:
 - Incerteza, "**problema do enquadramento**"
 - Impredictibilidade do ambiente onde outros agentes actuam
 - Incerteza nos efeitos das acções
 - Efeitos co-laterais das acções
 - Limitação de recursos incluindo o tempo.
- Quando ambiente não totalmente predizível o plano pode falhar:
 - Emendar parte do plano. Não esquecer as interações entre as partes
 - Retrocesso dirigido pelas dependências
- Planeadores são normalmente "dirigidos por objectivos" (facilita determinar as dependências e o factor de ramificação é inferior)

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

- Algoritmo de Planeamento é **Correcto** se todas as soluções forem planos legais.
- Algoritmo de Planeamento é **Completo** se, sempre que existam soluções, pelo menos uma é encontrada.
- Algoritmo de Planeamento é **Estritamente Completo** se todas as soluções podem ser encontradas
- Algoritmo de Planeamento é **Ótimo** se a ordem em que as soluções são encontradas são consistentes com uma medida de qualidade

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

- Simplificar os algoritmos de pesquisa do Plano:
 - Representação explícita de Objectivos, Estados, Acções e Planos
 - Decomposição dos objectivos: "Dividir para Conquistar": assume-se que os sub-objectivos são (quase-) independentes
 - Ordem de pesquisa para encontrar a solução diferente da ordem de execução do plano
- Assume-se Plano para agente simples sem efeitos co-laterais
- Exemplificação no "**Mundo dos Blocos**":
 - Blocos iguais; posição na mesa indiferente
 - Não mais que um Bloco directamente em cima de outro
 - Manipulador só pega em um bloco de cada vez.
 - Estados (situações) representados por literais instanciados ($em(A,B)$)
 - Acções através de operadores ($pega(x)$)
 - Axiomas: manipulador livre e Bloco x limpo para pegar(x)

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

- Acções típicas:
 - Colocar(x,y) Retirar(x,y)
 - Pegar(x) Largar(x)

- Situações típicas:
 - Em(x,y) Na_mesa(x)
 - Manipulador_vazio Segurar(x)
 - Limpo(x)

- Ciclo básico de um Planeador:
 1. Escolher a melhor acção
 2. Aplicá-la e computar a nova situação
 3. Detectar Objectivo ou impossibilidade E FIM
 SENÃO Voltar para 1

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

- Algoritmos e Técnicas para geração Automática de Planos
 - Análise Meios-Fins
 - GPS
 - Operadores
 - Tabela das Diferenças
 - Planeadores usando "Pilhas de Objectivos"
 - "tipo- STRIPS"
 - Descrição dos Operadores e Listas
 - Planeadores "Não-Lineares"
 - Uso de "Conjuntos de Objectivos"
 - Planeadores Hierárquicos
 - "Tipo-NOAH"
 - Ciclo "expansão e crítica"
 - Planeadores envolvendo restrições Geométricas
 - CIARC

Generalização de Planos

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

-Algoritmos e Técnicas para geração Automática de Planos

-Análise Meios-Fins

Computa as diferenças entre Estados Actual e Final

Procura operações que reduzam essas diferenças

Operadores denotam a realização de acções

Exemplo:

$Na_mesa(z,e1) \rightarrow Na_mesa(z, Fazer(Retirar(x,y),E2))$

o operador Fazer mapeia o resultado de acções em estados seguintes

Operadores (regras) só tratam com partes da descrição do problema

Ex: efeito do Operador Retirar(x,y)

$Limpo(x,e) \wedge Em(x,y,e) \rightarrow Suspensao(x, Fazer(Retirar(x,y),e1)) \wedge$

$Limpo(y, Fazer(Retirar(x,y),e1))$

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

E o resto do Estado ?

Deve usar-se Axiomas de Enquadramento.

Indicam o que deve ser assumido mesmo se não explicitamente declarado.

Por exemplo, o que não muda.

Ex: $Em(x,y,s) \wedge a \neq Pegar(x) \rightarrow Em(x,y,Fazer(a,s))$

$\sim Em(x,y,s) \wedge a \neq Colocar(x,y) \rightarrow \sim Em(x,y, Fazer(a,s))$

$Segurar(x,s) \wedge a \neq Colocar(x,y) \wedge a \neq Largar$

$\rightarrow Segurar(x,Fazer(a,s))$

Axiomas do domínio:

$\forall x, s \text{ Limpo}(x, s) \leftrightarrow \sim \exists y \text{ Em}(y, x, s)$

$\forall x, y, z : \text{Bloco } \text{Em}(y, x, s) \wedge \text{Em}(z, x, s) \leftrightarrow y = z$

$\forall x: \text{Bloco } \sim \text{Em}(x, x, s)$

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

Programar todos os **Axiomas de Enquadramento** pode ser entediante e ineficaz

As representações usadas nos planeadores habituais (tipo-"strips") resolvem implicitamente o Problema do Enquadramento ("**Frame Problem**").

- Estado é representado por um conjunto de factos (BD).
- Podem juntar-se variáveis de estado aos predicados:
 - Manipulador_livre(e); Limpo(x,e)
- Usa-se a AMF (CWA)
- Efeitos das acções garantidos por listas (Junte e Apague) de operações sobre a BD
 - Manipulador_livre \wedge Limpo(x,e) \wedge a=pegar(x) \rightarrow Segurar(x, Fazer(a,e)) \wedge $\forall y$ (Em(y,x,e) \rightarrow (\neg Em(y,x, Fazer(a,e)))
- Não há representação explícita do tempo
- Não há regras de Inferência

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

- Operadores típicos:

OPERADORES	PRÉ-CONDIÇÕES	RESULTADOS
Empurrar (obj, loc)	Junto(Rob, obj) E Grande(obj) Limpo(obj) E Manip_livre	Junto(obj.loc)E Junto(rob,loc)
Transportar (obj,loc)	Junto(Rob, obj) E Pequeno(obj) Limpo(obj) E Manip_livre	Junto(obj.loc)E Junto(rob,loc)
Mover(rob,loc)	-----	Junto(rob,loc)
Pegar(obj)	Junto(rob,obj) E Manip_livre	Segurar(obj)
Largar(obj)	Segurar(obj)	\sim Segurar(obj)E Na_mesa(obj)
Colocar(obj1,obj2)	Segurar(obj1) E Junto(obj2)	Em(obj1,obj2)

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

TABELA DAS DIFERENÇAS OPERADORES

ACÇÕES	Colocar	empurrar	transportar	ir	pegar	largar
Mover Obj		v	v			
Mover Rob				v		
Limpar Obj					v	
Empilhar Obj	v					
Libertar Manip	v					v
Segurar Obj					v	

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

-Exemplo: MOVER mesa com 2 livros de X para Y

-Acção a efectuar: MOVER objecto

-OPERADORES aplicáveis:

-EMPURRAR (objecto) ou

-TRANSPORTAR(objecto)

Consideremos o operador TRANSPORTAR(obj) :

Pré.Condições: Junto(Rob,obj)
Pequeno(obj)

A última pré-condição não se verifica pois o objecto MESA deverá estar caracterizado na Base de Conhecimentos do Planeador como Objecto grande.

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

Dado que as pré-condições do Operador EMPURRAR(obj) estão Satisfeitas, então devemos considerá-lo para o Plano de acções em Construção

Neste momento temos então o seguinte:

A.....B|-----|C.....|D
 Início EMPURRAR Objectivo

Ou seja anulou-se a diferença entre os pontos B e C que foi Considerada uma diferença importante. Há ainda que anular as diferenças entre os pontos A e B e ainda entre C e D.

Planeamento em Robótica

Continuando a encontrar as pré-condições para a aplicação do operador já considerado, temos:

Sendo Local (o,l)

Junto(Rob,obj) implica I R(l).

Sendo obj=Mesa e os dois livros o1 e o2:

Limpo(MESA) implica PEGAR(O1), LARGAR(O1), PEGAR(O2), LARGAR(O2)

Início PEGAR PEGAR EMPURRAR COLOCAR
 |-----|-----|-----|-----|-----|-----|-----|-----|
 A IR LARGAR LARGAR B C C1 D

O algoritmo continuaria até que todas as diferenças fossem anuladas!

Planeamento em Robótica

-Vantagem do Algoritmo:

- Espaço de pesquisa reduzido pois resolve um objectivo de cada vez.
- Bom só quando os objectivos são totalmente independentes.

-Dificuldades deste Algoritmo:

- Ordenação das diferenças a serem reduzidas
- Uso de Tabelas extensas
- Estratégias de inferência para ordenar os operadores aplicáveis

-Por isso o Algoritmo Análise Meios-Fins não é utilizado na sua forma mais pura, tal como foi descrito para o "General Problem Solver" de A.Newell.

-No entanto está na base de outros algoritmos que descreveremos.

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP

Planeamento em Robótica

-Algoritmo "Análise MEIOS-FINS"

AMF(EC,EO)

1 SE EC=EO RETORNAR

2 SENÃO

2a Seleccionar a Diferença mais importante e reduzi-la ou verificar impossibilidade

2b SE Lista de Operadores Lop=[] **FALHAR**

2c SENÃO

Seleccionar novo Operador aplicável, O

Aplicar O a EC:

Gerar descrição de O-start e O-result % Pré e pós condições de aplicação de O%

SE Anterior \leftarrow AMF(EC,O-start) **E**

Posterior \leftarrow AMF(O-result,EO)

ENTÃO RETORNAR ([Anterior, O, Posterior])

SENÃO VOLTAR a 2b

2003 / LEIC

Eugénio Oliveira

Eugénio Oliveira / FEUP