

INE5633 Sistemas Inteligentes

Prof. A. G. Silva

20 de agosto de 2014

Decidibilidade

- **Objetivo:** determinar se teoremas são completos e corretos, e se são demonstráveis ou não
- É necessário um processo determinístico que afirme quando uma palavra dada é um teorema ou não, em um número finito de passos
- É possível enumerar os teoremas, pois são obtidos aplicando as regras com os axiomas e outras palavras verdadeiras
- Porém, não há nenhum método para enumerar os que não são teoremas. Ou seja, não há nenhuma garantia que uma palavra pertença ou não ao sistema

Decidibilidade II

- Pode-se determinar se uma palavra é ou não um teorema com busca exaustiva. Porém, se não há decisão após um número finito de passos, não há como afirmar se a palavra não é teorema ou se o sistema é decidível ou não.
- Desta forma, a decidibilidade de uma palavra w em um sistema está relacionada ao fato da mesma ser alcançada ou demonstrada após um número finito de passos ou derivações no sistema

Interpretação I

- Os sistemas formais são construídos como modelos baseados em propriedades matemáticas
- Uma **interpretação** de um sistema formal é uma relação existente entre as palavras do sistema e uma semântica do mundo externo, a qual atribui um significado a todo símbolo do sistema, como uma correspondência 1 para 1 entre cada símbolo e algum objeto neste mundo externo
- O objeto pode então ser entendido no senso comum e, como tal, pode ser julgado como **verdadeiro** e **falso**

Interpretação II

- Baseado no estudo e construção de uma representação abstrata do mundo real (sistema formal)
- Tem se provado vários teoremas neste sistema, trabalhando somente com as propriedades abstratas
- Um sistema poderia ser um modelo para qualquer número de situações reais
- É possível voltar ao ponto de partida, interpretando estes teoremas abstratos em termos de objetos e propriedades do mundo real

Comentários I

- Não há nenhum motivo *a priori* que justifique por que uma declaração que é verdade no senso comum nem sempre pode ser provada como verdadeira em sua representação correspondente em um determinado sistema formal
- Isto é, o fato de uma declaração ser verdadeira não garante que pode ser provada
- Um determinada palavra pode ser um teorema ou não e sua interpretação pode ser julgada como sendo verdadeira ou falsa

Comentários II

- Relações possíveis (T é teorema e NT, não):

		Interpretação {V,F}	
		1	2
Decidibilidade {T,NT}	1	TeV	TeF
	3	NTeV	NTeF

- No caso 1, a palavra é provável e sua interpretação é verdadeira
- No caso 2, a palavra é provável, mas sua interpretação, sendo falsa, é claramente sem interesse e este caso pode ser eliminado

Comentários III

- Para qualquer sistema formal, é necessário manter somente as interpretações para as quais todos os teoremas são verdadeiros e demonstráveis
- É demonstrável se, após um número finito de operações, houver conclusão sobre a pertinência ou não ao sistema (com a mesma conotação de decidibilidade)

Problemas

- Envolvam ações típicas de um ser humano como raciocinar, aprender, resolver problemas, entender conceitos, ... (processo cognitivo)
- Naturalmente é confuso distinguir os limites de processos humanos e computacionais em alguns casos. Exemplo: uma máquina de calcular faz tarefas típicas de seres humanos, mas é um problema de IA?
- Alguns pontos que caracterizam um problema típico em IA:
 - ▶ Não é *algoritmizável*, ou seja, não há formulação sob uma sequência rígida de passos a ser seguida
 - ▶ É tipicamente simbólico
 - ▶ Necessita de ferramentas apropriadas (facilidade em relação a linguagens de programação convencionais)
 - ▶ Uma solução pode ser alcançada por vários caminhos distintos. Pode-se ter um conjunto de soluções
 - ▶ Um grande número de estados gerados por uma complexidade de ordem exponencial $O(x^n)$ ($x > 2$ e $n > 2$)

Definição de um problema I

- Na existência de um problema, é possível identificar os seguintes componentes:
 - ▶ **Um objetivo** ou meta a ser alcançada (*goal*)
 - ▶ **Um conjunto de ações** possíveis que impliquem em resultados alternativos e diferenciáveis para eventualmente produzir ou encontrar a meta desejada
 - ▶ **Uma situação inicial** ou ponto de partida para que ações sejam executadas com o objetivo de alcançar uma meta
- Formalmente um problema pode ser representado pela seguinte tupla:

$$\textit{Problema} = (I , O , C)$$

- ▶ *I*: conjunto de expressões que representam o *estado da situação inicial* no domínio do problema
- ▶ *O*: *conjunto de todas operações* ou *transformações* que possam ser efetuadas sobre um dado estado qualquer, derivando um novo estado
- ▶ *C*: é a condição que deve satisfazer a expressão de término

Definição de um problema II

- Um encaminhamento de solução é o encadeamento de um estado de partida $\#i$, ou uma situação inicial até um estado final ou objetivo
- Caso um sequência de operadores seja aplicada, a partir do estado inicial, e conduza a um estado terminal, então esta *sequência bem sucedida* de operadores conduz à uma *solução do problema*
- Considerando que os problemas pertinentes à IA não apresentam soluções algorítmicas ou, se existir, sua complexidade é impraticável em termos computacionais, uma alternativa é a de realizar *buscas* sobre os estados do problema
- Adicionalmente, alternativa *heurísticas* podem ser apresentadas como um esquema acelerador sobre os estados do problema

Definição de um problema II

- As heurísticas reportam um conhecimento explícito que visa reduzir drasticamente a quantidade de estados do problema e, como isso, viabilizar uma implementação computacional
- De modo sistemático, com funções heurísticas ou não, realizar uma exploração sobre as instâncias que vão surgindo durante um processo de busca é um dos focos da IA
- Estudo de métodos que realizem *busca cegas* ou *heurísticas* sobre as situações dos problemas que levem a uma ou mais soluções
- Estas *situações* ou *instâncias* são comumente chamadas de *estados*
- **A solução de um problema é uma caminho entre os estados inicial e um dos finais, sob um grafo de busca que é inerente ao problema solução**

Conceitos de estado – novo enfoque I

- **O estado inicial:** pelo qual geralmente o problema não é solucionado e no qual é possível agir para satisfazer o objetivo requerido
- **Os operadores:** definem como passar de um estado a outro
- **Teste de sucesso:** determina se o estado atual corresponde ao objetivo
- **Função de custo:** especifica o custo para cada passo em um caminho
- O estado inicial e os operadores definem o espaço da busca na árvore de conhecimento com a finalidade de se encontrar a solução desejada

Conceitos de estado – novo enfoque II

- Exemplo: quebra-cabeça de 8 pedras

- ▶ Estado inicial:

5	4	
6	1	8
7	3	2

- ▶ Operadores: exprimem os estados que podem ser assumidos pela célula vazia
- ▶ Estado final do problema: objetivo a ser alcançado

1	2	3
8		4
7	6	5

- ▶ Teste de sucesso: verifica se o estado atual corresponde ao estado final
- ▶ Custo: para cada passo dado na resolução do problema o custo é 1

Conceitos de estado – novo enfoque III

- Uma solução para um problema é um caminho desde o estado inicial até um estado objetivo
- A qualidade da solução é medida pela função de custo de caminho
- Uma solução ótima tem o menor custo de caminho entre todas as soluções
- Para que um problema possa ser resolvido, estes componentes devem ser expressos em uma terminologia claramente entendida para um potencial “solucionador” do problema (computador)
- Esta necessidade de descrever um problema em termos precisos (formalmente) possui um efeito colateral importante:
 - ▶ Forçar o esboço do problema de forma clara e sem ambiguidades

Estado e espaço de estados I

- O *estado* enfatiza as condições dos valores de parâmetros nas quais o sistema se encontra
- Por exemplo, as peças sobre um tabuleiro de xadrez descrevem um estado sobre o andamento de uma partida
 - ▶ Um estado em particular é o inicial, quando as peças de ambos jogadores estão alinhadas em posições regularmente definidas
 - ▶ Um outro estado seria um caso em que um rei estivesse submetido a um ataque do tipo *xequê*
 - ▶ O cheque, diferentemente do estado inicial, apresenta uma quantidade significativa de estados possíveis, não necessariamente representando o fim do jogo

Estado e espaço de estados II

- Em contextos mais simples (menor quantidade de células e número de peças) como o jogo da velha, é possível descrever sequência de operações sobre os estados que conduzam à vitória ou ao empate
- Ao conjunto de todos os estados que podem ser alcançados a partir do estado inicial, pela aplicação de operadores, denomina-se **espaço de busca** ou **espaço de estados**

Estado e espaço de estados III

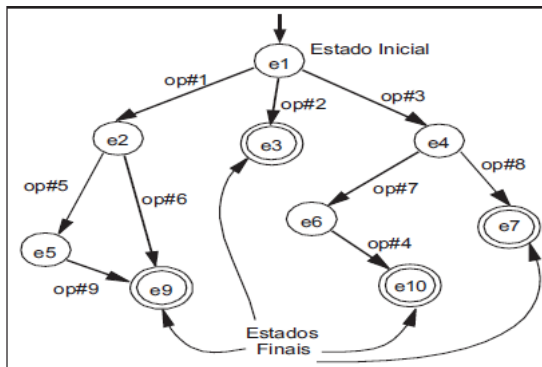
- Uma solução do problema é descrita pelo caminho estabelecido entre uma sequência de estados, de um inicial a outro final, e seus operadores correntes. Ou seja:

$$\begin{array}{ccc} e_i & \xrightarrow{\#i} & e_{i+1} \\ e_{i+1} & \xrightarrow{\#j} & e_{i+2} \\ \dots & \dots & \dots \\ e_{i+n} & \xrightarrow{\#k} & e_f \end{array}$$

Onde e_i e e_f são os estados inicial e final, e op_1, op_2, \dots, op_n , a sequência de operadores utilizada nesta transformação/transição entre estados

Estado e espaço de estados IV

- Um espaço de estados é determinado por uma sequência finita de operadores sob um *árvore de busca*



- Correspondência entre o problema codificado e a alcançabilidade de estados
- Interessante na análise do comportamento dinâmico

Árvores de buscas

- Apesar de um problema normalmente estabelecer um *grafo de busca*, utiliza-se, muitas vezes, na literatura, o termo *árvore de busca*
- Um novo estado (um nó) deve ser explorado sistematicamente e métodos de buscas em árvores são mais comuns que em grafos
- A manutenção de nós visitados e explorados, que se ramificam, é mais visível e tratável sob uma estrutura em árvore
- Todo grafo pode ser reduzido em uma árvore
- Na busca de solução de problemas, um estado em particular pode ser gerado a partir de vários outros e este gerar estados já produzidos anteriormente
- Ou seja, a presença de *ciclos* ou *caminhos circulares* na exploração é uma situação comum na resolução de problemas, porém indesejáveis:
 - ▶ Não expressam uma evolução à solução do problema
 - ▶ Ocupam memória desnecessariamente
 - ▶ Não necessariamente expressam uma atividade recursiva, muitas vezes necessária à solução

Problemas em IA

- Problemas sem solução algorítmica conhecida são de IA
- Entende-se por algoritmo um sequência precisa, ordenada e finita de operações, executadas em um tempo aceitável
- Apesar do jogo de xadrez envolver uma quantidade finita de possíveis situações, examinar todas as estratégias de jogo é trabalho que pode levar anos (logo, não há algoritmo)
- Sob esta visão de limites de recursos, as características marcantes em problemas da IA simbólica são:
 - ▶ **Conhecimento exposto em símbolos:** em contraste com o processamento numérico tradicional associado a computadores, na resolução de problemas em IA, os símbolos são expressos por meio de letras, palavras, sinais, desenhos e até mesmo, números
 - ▶ **Múltiplas escolhas:** há ausência de um algoritmo que trate a situação ou estado corrente, levando a escolha/alternativas de muitos caminhos a uma ação correta. Estas escolhas, a partir de um estado, são análogas a um autômato não-determinístico.

Características I

- Problemas, em geral, apresentam características quanto a natureza de etapas em suas soluções. Por exemplo, problemas podem ser:
 - ▶ **Ignoráveis:** no encaminhamento da solução não há mérito de custo (exemplo: demonstração de teorema)
 - ▶ **Recuperáveis:** passos da busca podem ser desfeitos (exemplo: quebra-cabeça do 8); recuperar passos é voltar a trajetória no espaço de soluções ou retroceder (*backtrack*)
 - ▶ **Irrecuperáveis:** contrário do anterior (exemplo: jogo de xadrez)
 - ▶ **Interação com o ambiente:**
 - ★ Acesso completo: total conhecimento dos estados; determinação do próximo estado com exatidão; problema de estado único
 - ★ Acesso parcial: conhecimento total dos estados não é totalmente acessível; mecanismo que determine o melhor próximo estado a ser considerado, a fim de chegar ao estado final
 - ★ Ambiente previsível ou não: caso dos problemas contingentes (exemplo: jogos de cartas)

Características II

- **Avaliação da solução:** solução encontrada é **absoluta** se, ao iniciar a busca com condições iniciais diferentes, o resultado será o mesmo; caso contrário, ela é relativa.
- **Tipo de solução:** determinação do quão boa é uma solução; questões como se a solução percorreu o caminho ótimo, se o estado final encontrado satisfaz o conjunto de restrições, etc
- **Função do conhecimento:** quantidade de conhecimento requerida (exemplo 1: para o jogo da velha, não é necessário muito conhecimento)
- **Interação com o ser humano:** quantidade de informação para que seja entendida por um ser humano (exemplo: sistemas especialistas utilizados para auxílio a diagnósticos)

Metodologia proposta I

- A solução de um problema é análoga à busca por meio de um diagrama de espaço de estados
- A questão é como representar as transições
- Para um melhor entendimento, é preciso relembrar que os elementos básicos de um problema são:
 - ▶ Um *estado corrente*
 - ▶ As *condições* de contorno/restrições sobre os movimentos a partir de um estado qualquer. Reunem-se as pré-condições para que se possa legitimar um dado movimento
 - ▶ Um *novo estado, sucessor ou posterior*
 - ▶ Um conjunto de *ações* sobre a representação simbólica do problema

Metodologia proposta II

- Pode-se então formalizar a construção de uma regra que descreva um movimento entre dois estados
- A formalização por meio de um sistema de transcrição é poderosa e adaptada à uma representação típica de **regras de produção**

Nome da regra (parâmetros) ::

$$\text{Est. corrente} \mid \{\text{Condições}\} \implies \text{Est. sucessor} \mid \{\text{Ações}\}$$

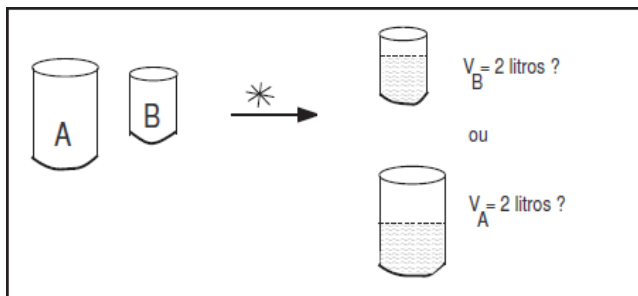
Leitura da proposição: *dado um estado corrente, que segue um conjunto de condições, este alcança um novo estado sucessor, executando um conjunto de ações*

Método de resolução de problemas

- Após a formulação do problema como um sistema de produção, representando os operadores necessários, é necessário realizar testes da seguinte forma
 - 1 Gerar uma solução possível
 - 2 Testar se é realmente uma solução
 - 3 Se não for uma solução, voltar à etapa 1
- Esta estratégia é conhecida como **algoritmo de busca gerar-e-testar** (mais simples estratégia de busca existente)
- É aplicável se e somente se for possível limitar o espaço de busca necessário para se encontrar uma solução
- É um procedimento de busca em profundidade (soluções completas precisam ser geradas antes de serem testadas)
- Em sua forma mais sistemática, é simplesmente uma busca exaustiva do espaço do problema
- No caso de problemas simples, o método pode ser razoável

Exemplo: problema dos vasos I

- Considere dois vasos, cuja capacidade de um deles é de 4 litros (vaso A) e do outro é de 3 litros (vaso B)
- Na ausência de um medidor externo ou de uma marcação nos vasos, o objetivo é alcançar a quantidade final de 2 litros de um líquido em um dos vasos (no vaso A ou no B)

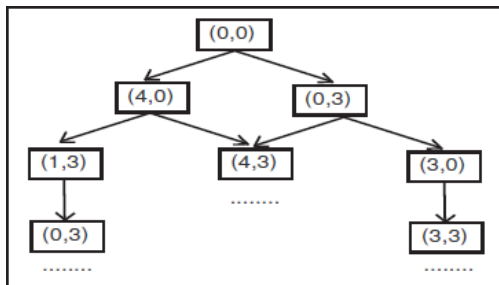


Exemplo: problema dos vasos II

- Ações primitivas e permitidas:
 - ▶ Encher, completamente, um dos vasos, A ou B (a presença de uma torneira é implícita)
 - ▶ Esvaziar ou despejar completamente para fora o líquido de um dos vasos, A ou B
 - ▶ Despejar o conteúdo de A em B, ou de B em A, podendo restar alguma quantidade de líquido em um deles ou não
- Formalização em tupla para definir a quantidade de líquido nos vasos A e B respectivamente
 - ▶ Espaço de estados: $(0,0)$, $(4,0)$, $(1,3)$, $(0,3)$, ...
 - ▶ Estado inicial: $(0,0)$ que é o nó de partida
 - ▶ Estado final: $(2,n)$ ou $(n,2)$ que são os nós finais

Exemplo: problema dos vasos III

- Espaço de estados:



Exemplo: problema dos vasos IV

- Alguma regras possíveis:

- 1 Encher o vaso A (4 litros) ::

$$[x, y] \mid \{x \leq 4\} \implies [4, y] \mid \{\}$$

- 2 Encher o vaso B (3 litros) ::

$$[x, y] \mid \{y \leq 3\} \implies [x, 3] \mid \{\}$$

- 3 Despejar fora alguma água do vaso A (4 litros) ::

$$[x, y] \mid \{x > 0\} \implies [w, y] \mid \{w = x - z\}$$

- 4 Despejar fora alguma água do vaso B (3 litros) ::

$$[x, y] \mid \{y > 0\} \implies [x, w] \mid \{w = y - z\}$$

Exemplo: problema dos vasos V

- Alguma regras possíveis:

- 5 Esvaziar o vaso A (4 litros) ::

$$[x, y] \mid \{x > 0\} \implies [0, y] \mid \{\}$$

- 6 Esvaziar o vaso B (3 litros) ::

$$[x, y] \mid \{y > 0\} \implies [x, 0] \mid \{\}$$

- 7 Encher o vaso A usando B, sobrando água em B ::

$$[x, y] \mid \{x + y > 4, y > 0\} \implies [4, w] \mid \{w = y - (4 - x)\}$$

- 8 Encher o vaso B usando A, sobrando água em A ::

$$[x, y] \mid \{x + y > 3, x > 0\} \implies [w, 3] \mid \{w = x - (3 - y)\}$$

Exemplo: problema dos vasos VI

- Algumas regras possíveis:

- 9 Despejar o vaso B em A, sem sobras ::

$$[x, y] \mid \{x + y \leq 4, y > 0\} \implies [w, 0] \mid \{w = x + y\}$$

- 10 Despejar o vaso A em B, sem sobras ::

$$[x, y] \mid \{x + y \leq 3, x > 0\} \implies [0, w] \mid \{w = x + y\}$$

- Observações

- ▶ Regras 3 e 4 são impossíveis de ocorrer (primitivas devem ser do tipo tudo-ou-nada, ou seja, não se pode echer ou esvaziar um “pouquinho”)
- ▶ Regras mais fortes, que também realizem ações de outras regras, podem ser definidas
- ▶ Algumas regras podem ser redundantes, contraditórias, desnecessárias, impossíveis