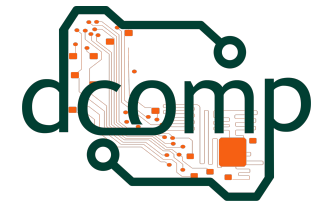




Universidade Federal do Espírito Santo
Centro de Ciências Agrárias – CCA UFES
Departamento de Computação



Paradigmas de Representação de Conhecimento

Inteligência Artificial

Site: <http://jeiks.net>

E-mail: jacsonrcsilva@gmail.com

Representação de Conhecimento

- Principais paradigmas de representação de conhecimento:
 - Sistemas ou Regras de Produção;
 - Redes Semânticas;
 - Quadros (*Frames*).

Sistemas de Produção

- Proposto pelo matemático Emil Post em 1943:
 - Demonstrou que um procedimento computável poderia ser modelado como um sistema de produção.
- Pode ser utilizado para:
 - implementar métodos de busca; e
 - para modelar a solução humana de problemas.
- Fornece um controle guiado por padrão de um processo de solução de problemas.
- Consiste em:
 - Um conjunto de *regras de produção*,
 - E uma *memória de trabalho*; e
 - Um ciclo de controle do tipo *reconhece-atua*.

Conjunto de regras de produção

- São chamadas simplesmente de *produções*;
- Cada *produção* define uma porção de conhecimento para a solução de um problema.
- A *produção* é um par condição-ação, onde:
 - A *condição* da regra é um padrão que determina quando a regra pode ser aplicada para um caso do problema;
 - A *ação* define o passo da solução do problema associado.

Memória de trabalho

- Contém uma descrição do *estado atual do mundo* num processo de raciocínio.
- A descrição é um padrão comparado com a condição para selecionar ações apropriadas para resolver o problema.
- Quando o elemento da condição de uma regra casa com o conteúdo da memória de trabalho, a ação associada com esta condição pode ser realizada.
- As ações das regras de produção são projetadas especificamente para alterar o conteúdo da memória de trabalho.

O ciclo reconhece-atua.

Estrutura de controle do sistema

- A *memória de trabalho* é inicializada com a descrição inicial do problema;
- O estado atual da solução do problema é mantido como um *conjunto de padrões* na memória de trabalho;
- Estes padrões são comparados com as condições das regras de produção,
 - É produzido o *conjunto de conflito*, que é um subconjunto de regras de produção.
 - As produções do *conjunto de conflito* estão *habilitadas* neste momento.
- Uma destas soluções no conjunto de conflito é selecionada, ou seja, a solução é *disparada*. Esta ação modifica a memória de trabalho.
- O ciclo de controle então se repete utilizando a memória de trabalho modificada.
- O processo termina quando o conteúdo da memória de trabalho não casa mais com nenhuma condição.

Conflitos

- A *resolução de conflito* escolhe uma regra para ser disparada no *conjunto de conflitos*.
 - Pode ser a escolha da primeira regra;
 - Ou a utilização de heurísticas para a seleção de regras.
 - Aqui é o local do sistema de produção que permite-se adicionar um controle heurístico a um algoritmo de busca.
- O modelo *puro* de sistema de produção finaliza no momento que nenhuma produção é mais encontrada.
 - Porém, muitas implementações práticas permitem regredir até um estado anterior da memória de trabalho.

Exemplo

Conjunto de produção:

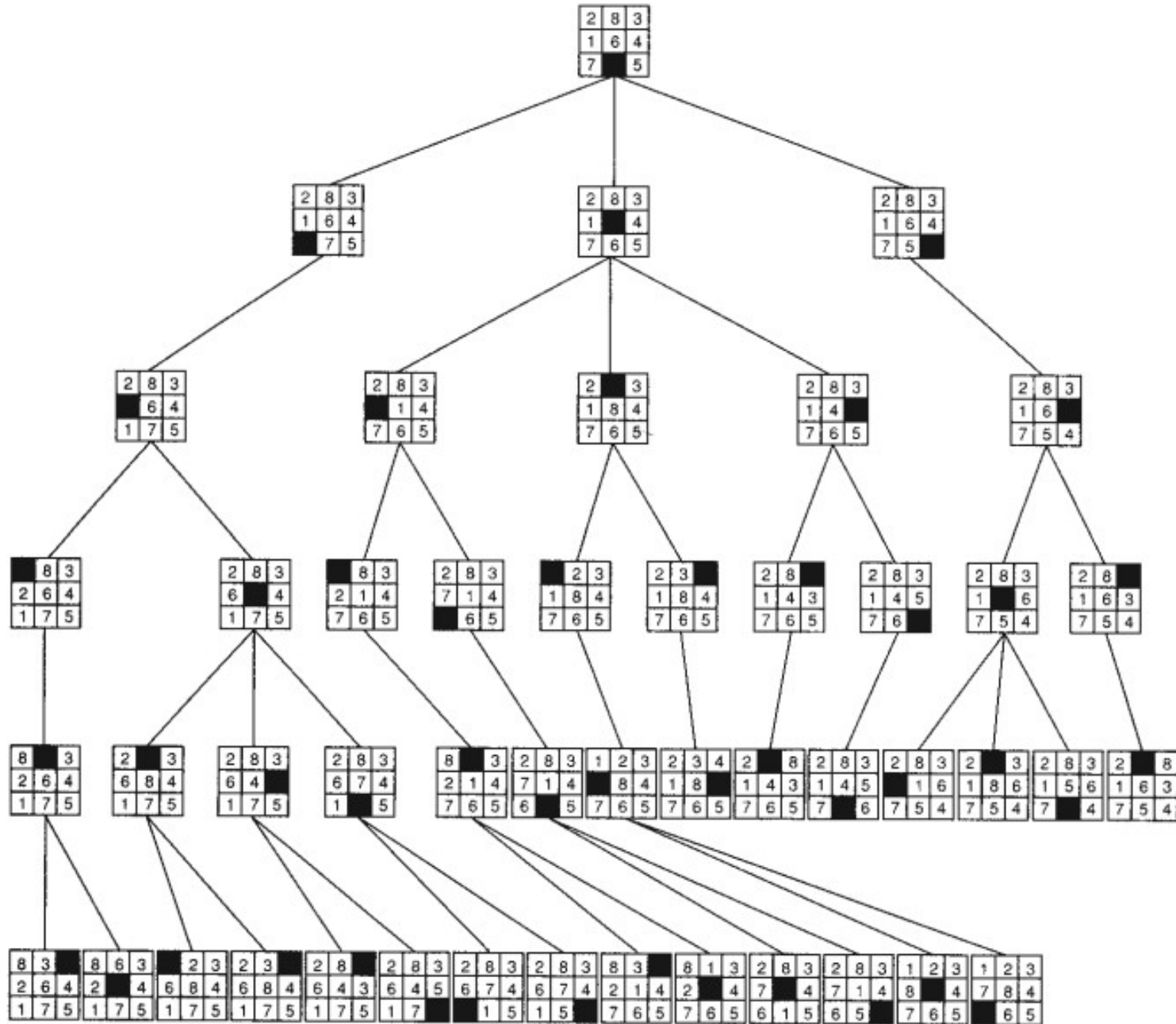
1 $ba \rightarrow ab$

2 $ca \rightarrow ac$

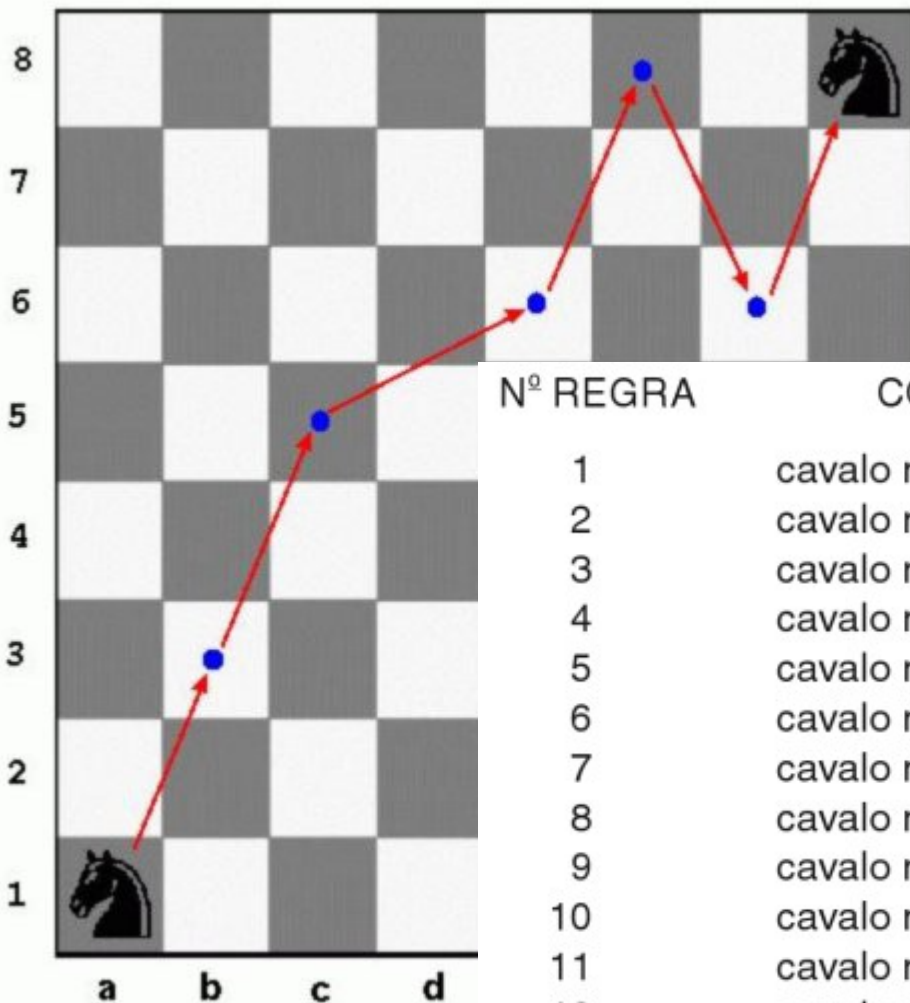
3 $cb \rightarrow bc$

N ^o Iteração	Memória de trabalho	Conjunto de conflito	Regra disparada
0	cbaca	1, 2, 3	1
1	cabca	2	2
2	acbca	2, 3	2
3	acbac	1, 3	1
4	acabc	2	2
5	aacbc	3	3
6	aabcc	∅	Parar

Outra utilização



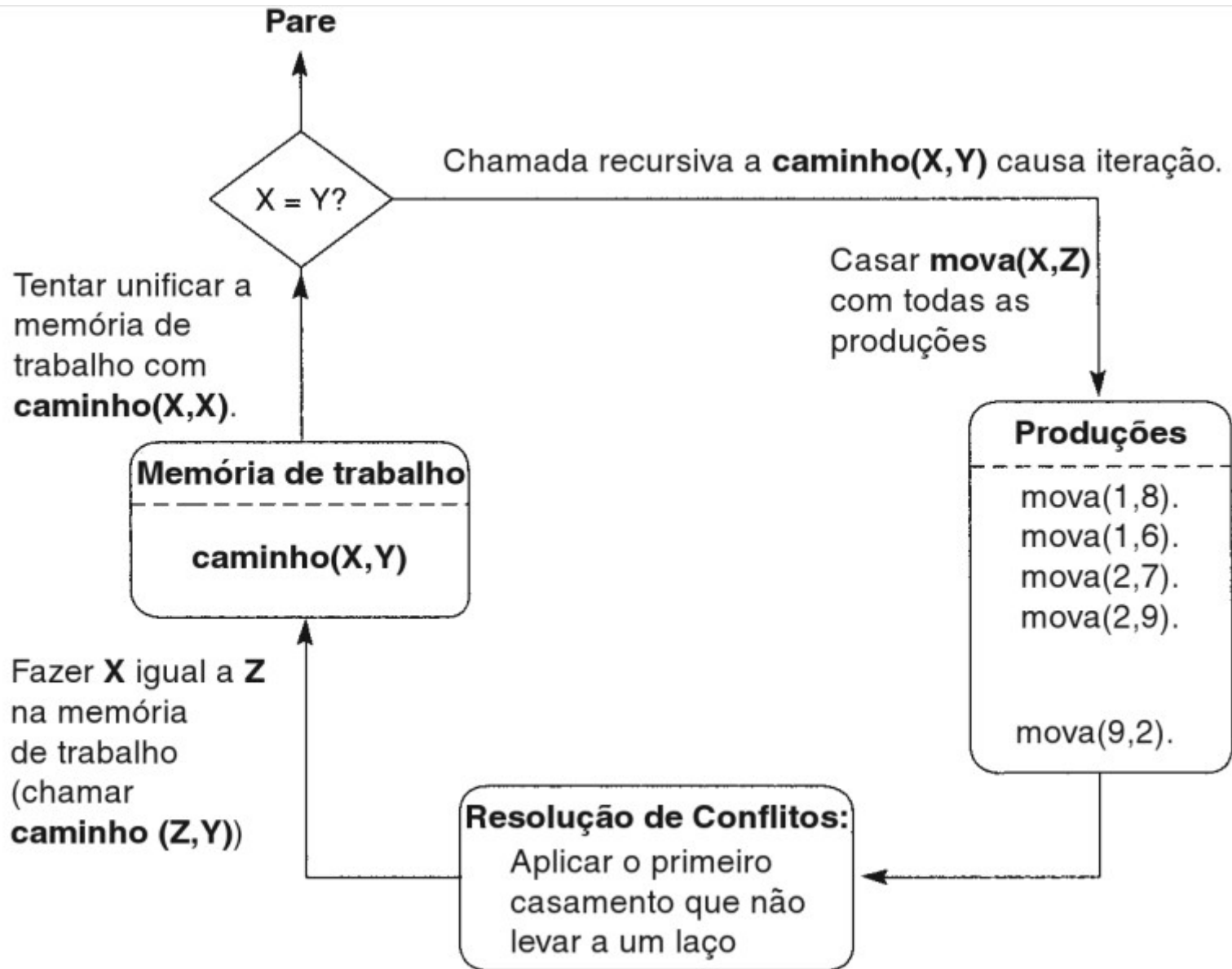
Problema do percurso do cavalo



Mova o cavalo da posição 1
até a posição 2
em um tabuleiro 3x3

Nº REGRA	CONDIÇÃO	AÇÃO
1	cavalo no quadrado 1	→ mova cavalo para quadrado 8
2	cavalo no quadrado 1	→ mova cavalo para quadrado 6
3	cavalo no quadrado 2	→ mova cavalo para quadrado 9
4	cavalo no quadrado 2	→ mova cavalo para quadrado 7
5	cavalo no quadrado 3	→ mova cavalo para quadrado 4
6	cavalo no quadrado 3	→ mova cavalo para quadrado 8
7	cavalo no quadrado 4	→ mova cavalo para quadrado 9
8	cavalo no quadrado 4	→ mova cavalo para quadrado 3
9	cavalo no quadrado 6	→ mova cavalo para quadrado 1
10	cavalo no quadrado 6	→ mova cavalo para quadrado 7
11	cavalo no quadrado 7	→ mova cavalo para quadrado 2
12	cavalo no quadrado 7	→ mova cavalo para quadrado 6
13	cavalo no quadrado 8	→ mova cavalo para quadrado 3
14	cavalo no quadrado 8	→ mova cavalo para quadrado 1
15	cavalo no quadrado 9	→ mova cavalo para quadrado 2
16	cavalo no quadrado 9	→ mova cavalo para quadrado 4

Algoritmo recursivo de caminho



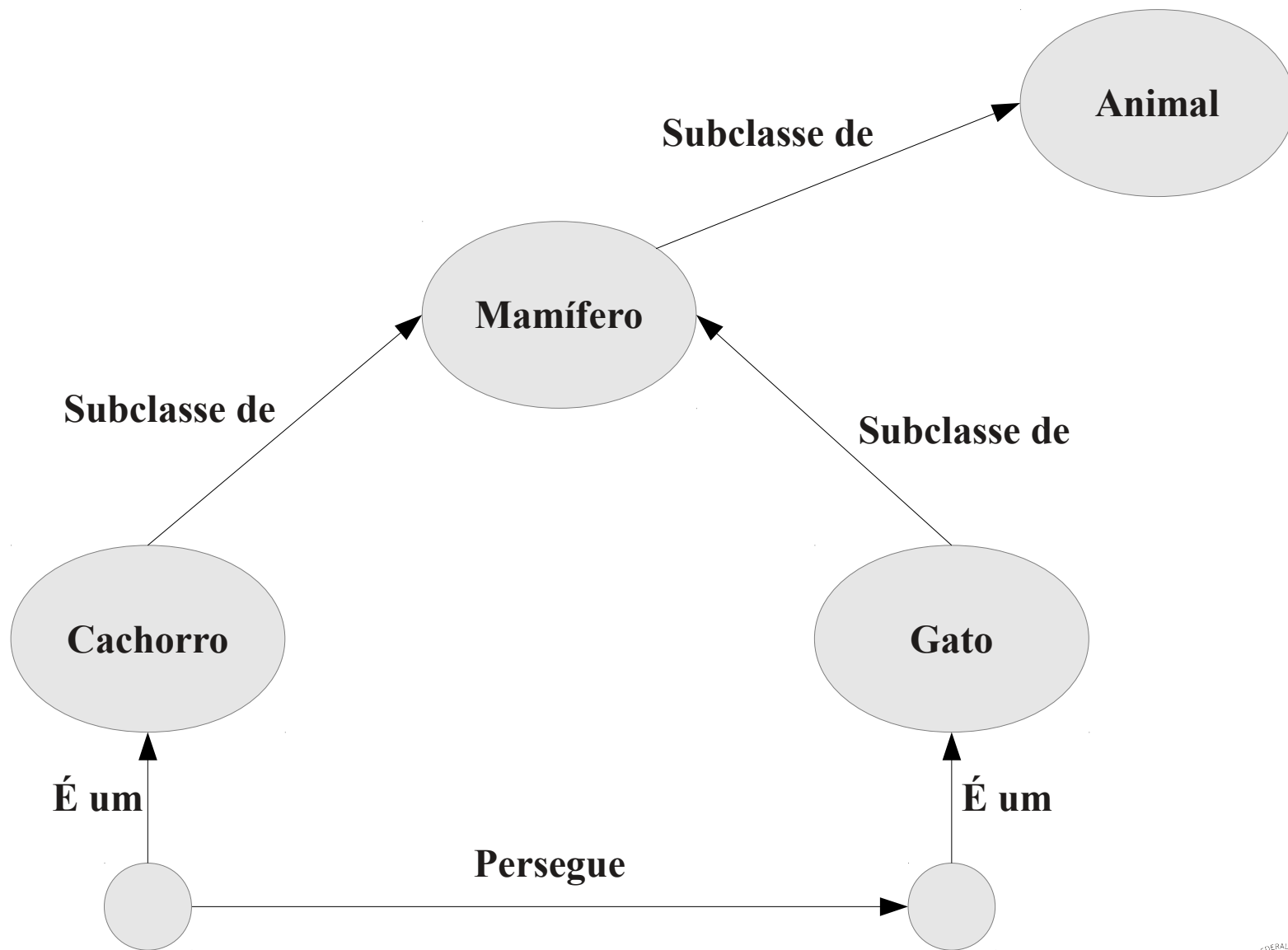
Redes Semânticas

- Consiste em um conjunto de nodos conectados por um conjunto de arcos.
 - Os nodos, em geral, representam objetos.
 - Os arcos representam relações binárias entre os objetos.
- Originalmente foram usadas para suporte a linguagem natural. Em 1968 Ross Quillian as usou para representar modelos psicológicos de memória humana chamado memórias semânticas.
- Quillian desenvolveu um programa que define palavras em inglês de forma similar a dicionários.
 - Em vez de definir palavras formalmente, cada definição simplesmente conduz a outras definições em uma forma desestruturada e, possivelmente circular.
 - Ao procurar uma palavra, percorremos a “rede” até que estejamos satisfeitos com o que compreendemos da palavra original.

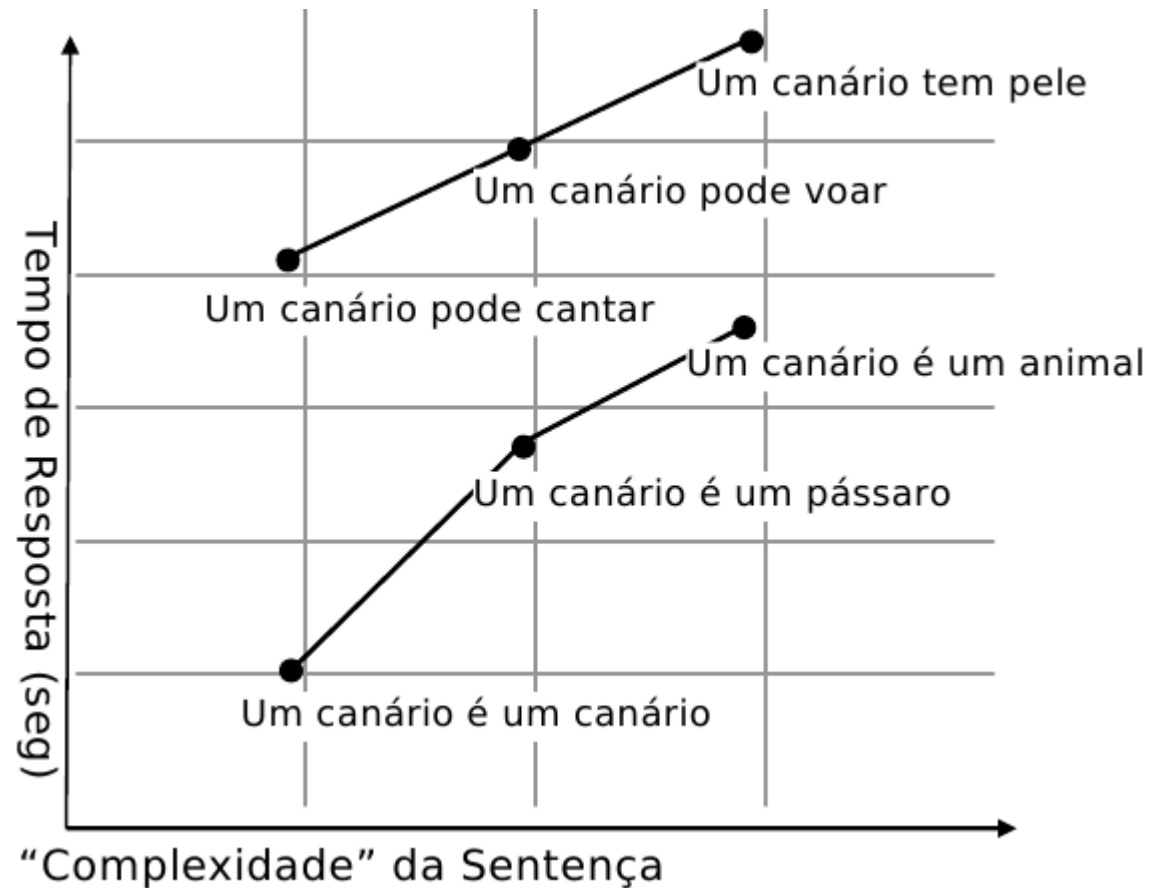
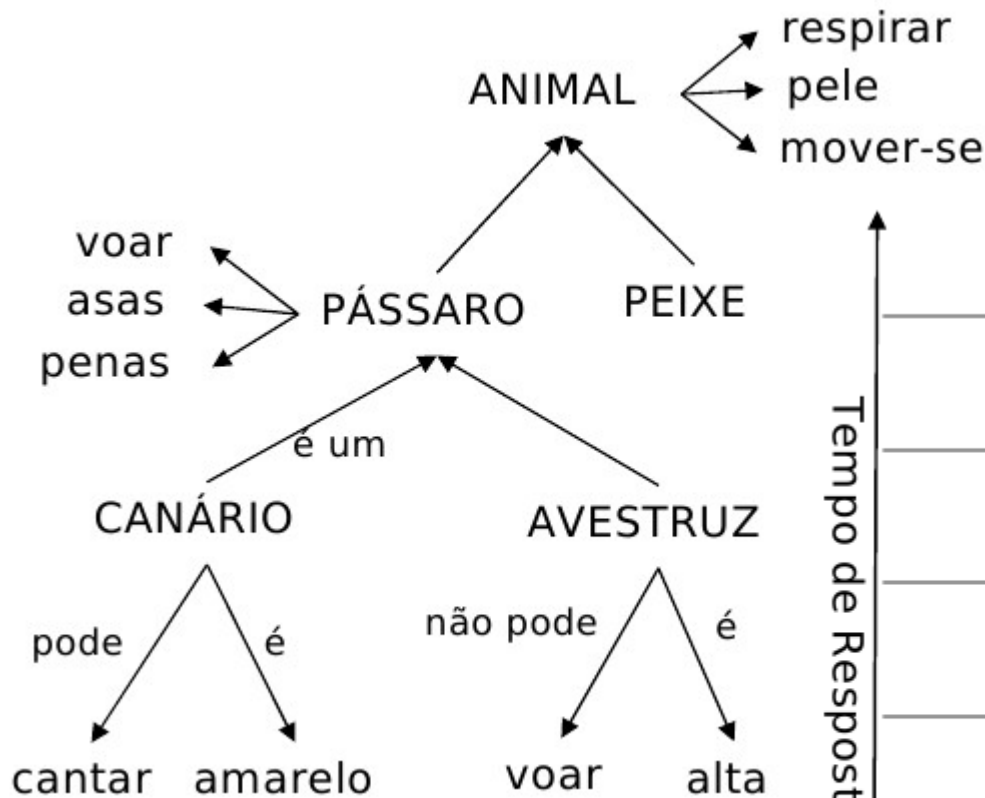
Redes Semânticas

- Redes Semânticas Elementares
 - Usa-se nodos para representar substantivos, adjetivos, pronomes e nomes próprios.
 - Os arcos são reservados basicamente para representar verbos transitivos e preposições.
 - Relações de inclusão entre classes são representadas por relações “subclasse-de”.
 - Os nodos rotulados representam classes genéricas, enquanto que os nodos anônimos representam indivíduos específicos.
 - Para saber se um nodo representa uma instância, é só observar se ele está na origem de algum arco do tipo “é-um”.

Exemplo de rede semântica



Redes Semânticas



Além da habilidade de associar conceitos, os humanos também organizam hierarquicamente o seu conhecimento. A informação é armazenada em níveis apropriados mais altos da taxonomia.

Quadros (*Frames*)

- Os Quadros ou Cenários (*Frames*), e sua variação, os roteiros (*Scripts*)
 - introduzidos para permitir a expressão das estruturas internas dos objetos,
 - mantendo a possibilidade de representar herança de propriedades.
- As pessoas, ao enfrentarem uma nova situação, guardam o repertório do comportamento para situações similares.
 - Ex.: alguém que já assistiu alguma vez a um júri popular sabe que tipo de “quadro” irá encontrar se for a outro.
- As ideias fundamentais foram introduzidas por Marvin Minsky em 1975, no artigo “A framework to represent knowledge”.
- Origina-se nas mesmas ideias das linguagens de programação orientadas a objetos.

Quadros (*Frames*)

- Segundo Minsky (1975):
 - “Quando alguém encontra uma nova situação (ou modifica substancialmente o seu entendimento sobre um problema), recupera da memória uma estrutura chamada ‘frame’. Esta estrutura é um arcabouço memorizado que deve ser adaptado para se adequar à realidade, alterando detalhes, conforme a necessidade”.
- Um quadro consiste em um conjunto de atributos (*slots*) que através de seus valores, descrevem as características do objeto representado pelo quadro.
- Os valores atribuídos aos atributos podem ser
 - valores do objeto em particular,
 - valores *default*,
 - ponteiros para outros quadros (que criam redes de dependências) e
 - conjuntos de regras de procedimento que podem ser implementados.
- Os conjuntos de procedimentos indicam que procedimento deve ser executado quando certas condições forem satisfeitas.

Quadros (*Frames*)

- Os quadros também são organizados em uma hierarquia de especialização, criando outra dimensão de dependência entre eles (herança).
 - Permite assim especificar propriedades de uma classe de objetos através da declaração de herança desta classe à outra.
- O processo de herança e instanciação favorece a reutilização de código.
- São úteis para domínio de problemas onde a forma e o conteúdo do dado desempenham um papel importante na solução do problema.

Exemplo

- Quadro: Cadeira
 - Slot: número de pernas - inteiro (default: 4);
 - Slot: tipo-de-encosto - curvo, reto, não-tem (default: curvo);
 - Slot: tipo-de-assento - redondo, anatômico, reto (default: anatômico);
 - Slot: número-de-braços - 2,1,0 (default: 0);
 - Slot: cor - preta, branca, incolor, azul (default: incolor);

- Quadro: Cadeira-do-Renato

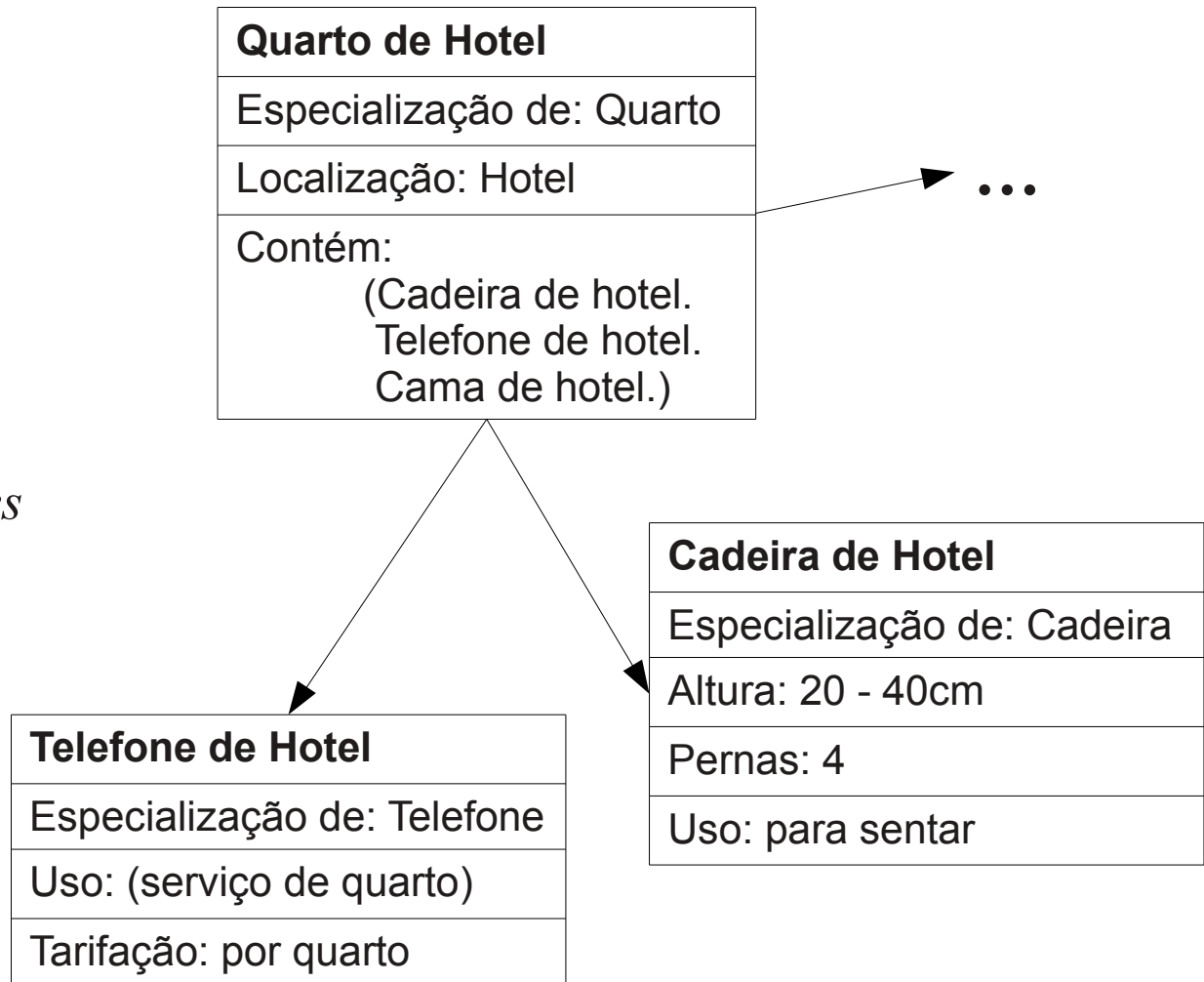
É-UM Cadeira

- Slot: número de pernas – 4;
- Slot: tipo-de-encosto – (default: curvo);
- Slot: tipo-de-assento – redondo;
- Slot: número-de-braços – 0;
- Slot: cor – (default: incolor);

Exemplo

- Parte de uma descrição por *frame* de um quarto de hotel.
- Cada *frame* individual pode ser visto como uma estrutura de dados.

- *Slots* do *frame* contém:
 - Identificação *frame*
 - Relação com outros *frames*
 - Descritores de requisitos (altura do acento)
 - Informação sobre uso
 - Informação *default* (cadeira tem 4 pernas)



Quadros (*Frames*)

- Quadros superam o poder das redes semânticas pois permitem que objetos complexos sejam representados como um único *frame*, em vez de uma grande estrutura de rede.
- Os *frames* tornam mais fácil organizar o conhecimento hierarquicamente.