

# Inteligência Artificial

## *Representação de conhecimento*

# Introdução

- Para resolver os problemas mais complexos encontrados na inteligência artificial, é necessário uma grande quantidade de conhecimento e mecanismos para manipulá-lo.
  - Para a resolução de problemas, é preciso contar com uma forma adequada de representação do conhecimento.
- 
-

# Bases de um sistema de IA:

- Uma estrutura de representação de conhecimento
- Métodos de solução de problemas e de inferência

# Exemplo de problema

O Mundo do Wumpus envolve uma série de conhecimentos de natureza diversa envolvidos na formulação do problema

- Ambiente: agente, Wumpus, cavernas, buracos, ouro
- Estado inicial: agente na caverna (1,1) com apenas uma flecha, Wumpus e buracos em cavernas quaisquer
- Objetivos: pegar a barra de ouro e voltar à caverna (1,1) com vida
- Percepções: fedor, brisa, luz, choque (contra a parede da caverna) e grito do Wumpus
- Ações: avançar para próxima caverna, girar 90 graus à direita ou à esquerda

4	Stench		Breeze	PIT
3	Wumpus	Breeze Stench Gold	PIT	Breeze
2	Stench		Breeze	
1	START	Breeze	PIT	Breeze

# Conhecimento e sua representação

- Conhecimento é informação armazenada ou modelos possíveis de serem usados por pessoas ou máquinas para interpretar, predizer e responder apropriadamente ao mundo exterior.
- Forma  $\neq$  conteúdo
- Representação  $\neq$  objeto real
- Semântica envolve uma função de mapeamento entre o objeto do conhecimento e sua representação
- Conhecimento  $\neq$  Informação  $\neq$  Dados  $\neq$  Sabedoria

# O Que é Dado

- Dado é a estrutura fundamental sobre a qual um sistema de informação é construído. Precisão é um critério importante na avaliação da validade de um dado.
  - Fatos, imagens ou sons que podem ou não ser úteis para uma determinada tarefa.
  - Elementos de um Dado:
    - Campo - Características representando atributos específicos.
    - Registro - Uma coleção de campos interligados, relacionados a um objeto comum.
    - Arquivo - Uma coleção de registros interligados relatando um tópico comum
- 
-

# O que é informação

- A transformação de dados em informação é frequentemente realizada através da apresentação dos dados em uma forma compreensível para o usuário. Parte do processo de criar informação é generalizada a partir da base de dados.
  - Informação é uma declaração em relação à estrutura de uma entidade que permita a uma pessoa tomar uma decisão.
  - Informação é qualquer dado estruturado que tem significado dentro de um contexto.
  - Algumas vezes a estrutura e a apresentação do dado em um formato compreensível transforma o dado em informação.
- 
-

# O que é conhecimento

- “No sentido mais amplo, atributo geral que têm os seres vivos de reagir ativamente ao mundo circundante, na medida de sua organização biológica e no sentido de sua sobrevivência.” (Dicionário Aurélio, 1986)
  - Conhecimento é uma combinação de instintos, idéias, regras e procedimentos que guiam as ações e decisões. O conhecimento é uma apresentação concisa da experiência prévia de uma pessoa.
  - São as “regras práticas” que usamos quando fazemos algo. O conhecimento fornece a capacidade de resolver problemas, inovar e aprender baseado em experiências prévias .
- 
-



# O que é sabedoria

- Sabedoria é a capacidade de tomarmos decisões acertadas, de aplicarmos bem o conhecimento que possuímos, dentro das limitações de tempo e espaço a que estamos sujeitos no momento da nossa ação.
  - O conhecimento pode ser adquirido nos livros, na escola. A sabedoria, no entanto, vem da prática, da experiência adquirida à medida que vivemos e somos testados.
  - Dado não é Informação
  - Informação não é Conhecimento
  - Conhecimento não é Inteligência
  - Inteligência não é Sabedoria
- 
-

# Representação de Conhecimento

- O Conhecimento e o Raciocínio são importantes para agentes artificiais, pois permitem comportamentos bem-sucedidos que seriam muito difíceis de alcançar de outra forma.
- A manipulação do conhecimento exige, antes, formas de representação. Esta representação deve ser suficientemente rica e completa para evitar falhas evidentes de entendimento pelo motor de inferência, como por exemplo:
  - Gato – é um: ser vivo + mamífero + raças + etc.
  - Logo: leão, rato, cão, etc é gato.

# Funções básicas de um sistema de RC

- Dedução de novos resultados
- Consulta
- Organização e reorganização
- Interpretação

# Representação de Conhecimento

- Para que uma representação lógica seja adequada, duas condições são necessárias:
  1. Existência de uma correspondência um para um entre certas classes de símbolos da representação e conjuntos de objetos de interesse no mundo externo;
  2. Existência, para cada relação simples no mundo externo, de uma relação na representação, de tal maneira que a relação entre dois símbolos da representação seja válida se, e somente se, a relação correspondente for válida entre os objetos correspondentes do mundo externo.
- Sem essas condições temos o **Conhecimento Incerto**.

# Elementos da RC

- Quando discutimos representação do conhecimento lidamos com dois tipos diferentes de entidades:
  - **Fatos**: Verdades em algum mundo relevante, ou seja, tudo aquilo que queremos representar.
  - **Representação de fatos**: com algum formalismo escolhido, ou , aquilo que efetivamente seremos capazes de representar.
    - representações dos objetos no nível do conhecimento são definidas em termos de símbolos que podem ser manipulados por programas.

# Representação de Conhecimento

- É importante distinguir: FORMA e CONTEÚDO.
  - Exemplo: Um texto que usa a linguagem natural como recurso de representação pode ter, também seu conteúdo sintetizado através de outros recursos, como, por exemplo a lógica de predicados.
  - O conteúdo é o mesmo, as formas de representação diferentes facilitam a manipulação por diferentes agentes (computador, ser humano)

# Principais formas de representação

- Sistemas de produção
- Redes semânticas
- Quadros
- Scripts
- Lógica
  - Lógica proposicional
  - Lógica de primeira ordem
  - Lógica descritiva
  - Outras Lógicas usadas em IA
    - Lógica temporal, lógica fuzzy, lógica modal, crenças, etc.
- Representações computacionalmente computáveis são equivalentes, só que algumas representações são mais convenientes.

# Raciocínio e Inferência

- O raciocínio é uma operação lógica que utiliza uma ou mais proposições, para concluir, através de mecanismos de comparações e abstrações, quais são os dados que levam às respostas verdadeiras, falsas ou prováveis.
- o raciocínio pode ser considerado também um dos integrantes dos mecanismos dos processos cognitivos superiores da formação de conceitos e da solução de problemas, sendo parte do pensamento
- O processo de raciocínio é importante, pois permite a explicitação de uma solução adequada, para um problema em particular.
- Este processo dever ser também capaz de gerar novos conhecimentos a partir de conhecimentos previamente armazenados (inferência).
- Se a informação não estiver explicitamente na base, a inferência é necessária. Existe um compromisso entre a quantidade de conhecimento armazenada explicitamente na base de conhecimento e a atividade de inferência.

Capacidade Cognitiva

=

base de conhecimento + motor de inferência



# Raciocínio e Inferência

- Inferência, em Lógica, é o ato ou processo de derivar conclusões lógicas de premissas conhecidas ou decididamente verdadeiras
- Inferência: o raciocínio formal é utilizado em geral, nas representações baseadas em lógica. Sendo possível três tipos:
  - **Inferência Dedutiva**: a partir de elementos de conhecimento representados em forma lógica, utiliza-se uma regra de inferência válida para inferir um novo elemento. Algumas regras de inferência utilizadas em lógica são: Modus Ponens, Modus Tollens, Silogismo Hipotético, etc.
  - **Inferência Abdutiva**: a partir de um conhecimento geral da forma  $\forall x P(x) \rightarrow Q(x)$ , e tendo por objetivo provar  $Q(a)$ , toma-se por hipótese que a razão pela qual  $Q(a)$  se verifica é a validade de  $P(a)$ .
  - **Inferência Indutiva**: a partir de fatos experimentais que comprovam que a cada vez que a validade de  $P(a)$  é verificada, verifica-se a validade de  $Q(a)$  (mas não o contrário), para diferentes elementos  $a$ , pode-se inferir por indução que é válido.

# Raciocínio e Inferência

## - Inferência Dedutiva

João é engenheiro

Todo engenheiro é bom em cálculo

---

Logo (passo dedutivo): João é bom em cálculo

## - Inferência Indutiva

João é engenheiro

João é bom em cálculo

---

Logo (passo indutivo): Todo engenheiro é bom em cálculo

## - Inferência Abdutiva

João é bom em cálculo

Todo engenheiro é bom em cálculo

---

Logo (passo abdutivo): João é engenheiro

# Engenharia do Conhecimento

- Conhecimento: repositório de procedimentos, heurísticas, dados, etc., que compõe o conhecimento. Assemelha-se a um banco de dados no sentido de que exige manutenção (atualizações, inserções e deleções) mas o acesso a uma informação é mais elaborado.
- Engenheiro do Conhecimento: o profissional de ciência da computação responsável pela implantação da base de conhecimento. É um profissional com sólidos conhecimentos em técnicas de IA.
- Especialista do Domínio: é um profissional altamente capacitado no domínio para o qual estamos desenvolvendo a aplicação. Supre o engenheiro do conhecimento com os procedimentos (formais e heurísticas) necessários à construção da base de conhecimento.

# Características Essenciais

- Consistência: Não armazena informações conflitantes.
- Completude: Não apresenta lacunas no conhecimento armazenado. Todo o conhecimento necessário para a resolução do problema está explicitamente armazenado ou pode ser determinado via inferência.
- Coerência: Não existem ilhas isoladas de conhecimento não se relacionam com o restante do conhecimento armazenado.
- Não Redundância: A mesma unidade de conhecimento armazenada de forma duplicada.

# Características Desejáveis

- Boas representações explicitam as coisas importantes.
- Revelam restrições naturais, facilitando algumas classes de computações.
- São concisas, necessitando apenas de recursos mínimos e sendo ao mesmo tempo ainda eficientes quando efetuam inferências.
- Podem ser rapidamente recuperadas e armazenadas.
- Informações raramente usadas são abordadas e recuperadas apenas quando necessárias.
- Permitem uma aquisição fácil e são legíveis pelo especialista, quando for o caso.
- Permitem a aplicação dos mecanismos de inferência necessários.

# Sistemas ou Regras de Produção

- Concebidas por Emil Post (1943) quando demonstrou que um procedimento computável pode ser modelado como um sistema de produção.
- Muito utilizada nas décadas de 50 e 60. É o formalismo mais difundido de representação de conhecimento.
- Consiste em transformar o problema em um grafo de estados. Este grafo deve possuir um estado inicial e deve-se ter uma forma de identificar um estado final quando algum for atingido.
- Ou seja, consiste em:
  - Regras de Produção + Memória de Trabalho + ciclo de controle (tipo reconhece-atua)

# Sistemas de Produção

- Um Sistema de Produção é definido como uma tupla  $SP = \langle R, E, e_0, F \rangle$ , onde  $R$  é um conjunto de regras,  $E$  é um conjunto de estados,  $e_0$  é o estado inicial e  $F$  é o conjunto de estados finais.
- Uma Regra de Produção é constituída por um par  $\langle p, f \rangle$ , onde  $p: E \rightarrow \{V, F\}$  e  $f: E \rightarrow E$ . O elemento  $p$  é o padrão da regra, e  $f$  constitui a operação. Gera normalmente estruturas do tipo:
- SE <estado> ENTÃO <ação>  
onde:
  - <estado> ou <condição>: estabelece um teste cujo resultado depende do estado atual da base de conhecimento. Tipicamente o teste verifica a presença ou não de certas informações na base.
  - <ação>: altera o estado atual da base de conhecimento, adicionando, modificando ou removendo unidades de conhecimento presentes na base. Pode acarretar também efeitos externos à base, como por exemplo a escrita de uma mensagem no vídeo.

# Sistemas de Produção

- EXEMPLO: Problema dos Dois Baldes de Água
  - Enunciado: Você recebe dois baldes de água, um de quatro litros e outro de três litros. Nenhum deles possui qualquer marcação de medida. Há uma torneira que pode ser utilizada para encher os baldes de água. Como colocar exatamente dois litros d'água dentro do balde de quatro litros?
  - Conjunto de Estados: O espaço de estados para este problema pode ser modelado como o conjunto de pares ordenados de números naturais  $(x,y)$  tal que  $x = 0, 1, 2, 3$  ou  $4$  e  $y = 0, 1, 2$  ou  $3$ , onde  $x$  representa a quantidade de água no balde de 4 litros, e  $y$  representa a quantidade de água no balde de 3 litros.
  - Estado Inicial: Ambos os baldes estão vazios:  $(0,0)$ .
  - Estado Final: Constituído por todos os estados onde a qtde de água no primeiro balde é 2, ou seja:  $(2,n)$ , onde  $n = 0, 1, 2$  ou  $3$ .



# Sistemas de Produção

- ♦ EXEMPLO: Problema dos Dois Baldes de Água
  - Um possível conjunto de regras para este problema seria:
    - $r_1 (x,y|x<4) \rightarrow (4,y)$  Encher o balde de 4 litros
    - $r_2 (x,y|y<3) \rightarrow (x,3)$  Encher o balde de 3 litros
    - $r_3 (x,y|x>0) \rightarrow (0,y)$  Esvaziar o balde de 4 litros no chão
    - $r_4 (x,y|y>0) \rightarrow (x,0)$  Esvaziar o balde de 3 litros no chão
    - $r_5 (x,y|x+y>4) \rightarrow (4,y-(4-x))$  Despejar água do balde de 3 litros dentro do balde de 4 litros até que este esteja cheio
    - $r_6 (x,y|x+y>3) \rightarrow (x-(3-y),3)$  Despejar água do balde de 4 litros dentro do balde de 3 litros até que este esteja cheio
    - $r_7 (x,y|x+y \leq 4 \text{ e } y>0) \rightarrow (x+y,0)$  Despejar toda a água do balde de 3 litros dentro do balde de 4 litros
    - $r_8 (x,y|x+y \leq 3 \text{ e } x>0) \rightarrow (0,x+y)$  Despejar toda a água do balde de 4 litros dentro do balde de três litros

# Sistemas de Produção

- ♦ EXEMPLO: Problema dos Dois Baldes de Água
  - Uma solução possível para o problema seria aplicar em seqüência as regras r2, r9, r2, r7, r5 e r9 (r2, r5, r2, r5, r3 e r5).
  - Esta solução não é a única POSSÍVEL. Além disso, não foi mostrado como a solução foi encontrada.
  - Este é exatamente o ponto onde entram os ALGORITMOS DE BUSCA no Espaço de Estados.
- ♦ A modelagem de um problema como um sistema de produção consiste apenas em definir o espaço de estados e as regras. Este processo dificilmente poderia ser feito automaticamente. Como na programação tradicional, trata-se de um processo de modelagem de uma realidade perceptível utilizando uma ferramenta definida. Mas, UMA VEZ ESTABELECIDO O MODELO, O PROCESSO PODE SER LIBERADO PARA A MÁQUINA E ESTA ENCONTRAR SOZINHA A SOLUÇÃO.

# Sistemas de Produção

- O problema das Três Jarras
  - Há três jarras de vinho com capacidade para oito, cinco e três litros. A jarra maior maior está cheia de vinho e as outras estão completamente vazias. Queremos dividir o vinho em porções iguais entre as duas primeiras jarras de modo que cada uma fique com 4 litros.
  - Exercício: Ache as regras deste sistema de produção.

# Sistemas de Produção

## ♦ Outros Exemplos:

### 1. Problema dos Missionários e Canibais

- O estado inicial é (3,3,0,0,0) e o único estado final é (0,0,3,3,1).
- As regras são todas de movimentação de no máximo 2 pessoas de uma margem a outra do rio. Assim, pode-se movimentar dois canibais, dois missionários, um canibal e um missionário, apenas um canibal ou apenas um missionário, tanto da margem original para a margem oposta quanto vice-versa.
- Tem-se assim 10 regras possíveis, das quais uma é mostrada a seguir:  

$$R_{MM} \rightarrow \text{Se}(m_1, c_1, m_2, c_2, 0 | (m_1 \geq 2) \text{ e } [m_1 - 2 \geq c_1 \text{ ou } (m_1 - 2 = 0)] \text{ e } (m_2 + 2 \geq c_2)) \text{ Então } (m_1 - 2, c_1, m_2 + 2, c_2, 1)$$
- Exercício: Ache as outras regras.

# Missionarios e canibais

## Outra formulação:

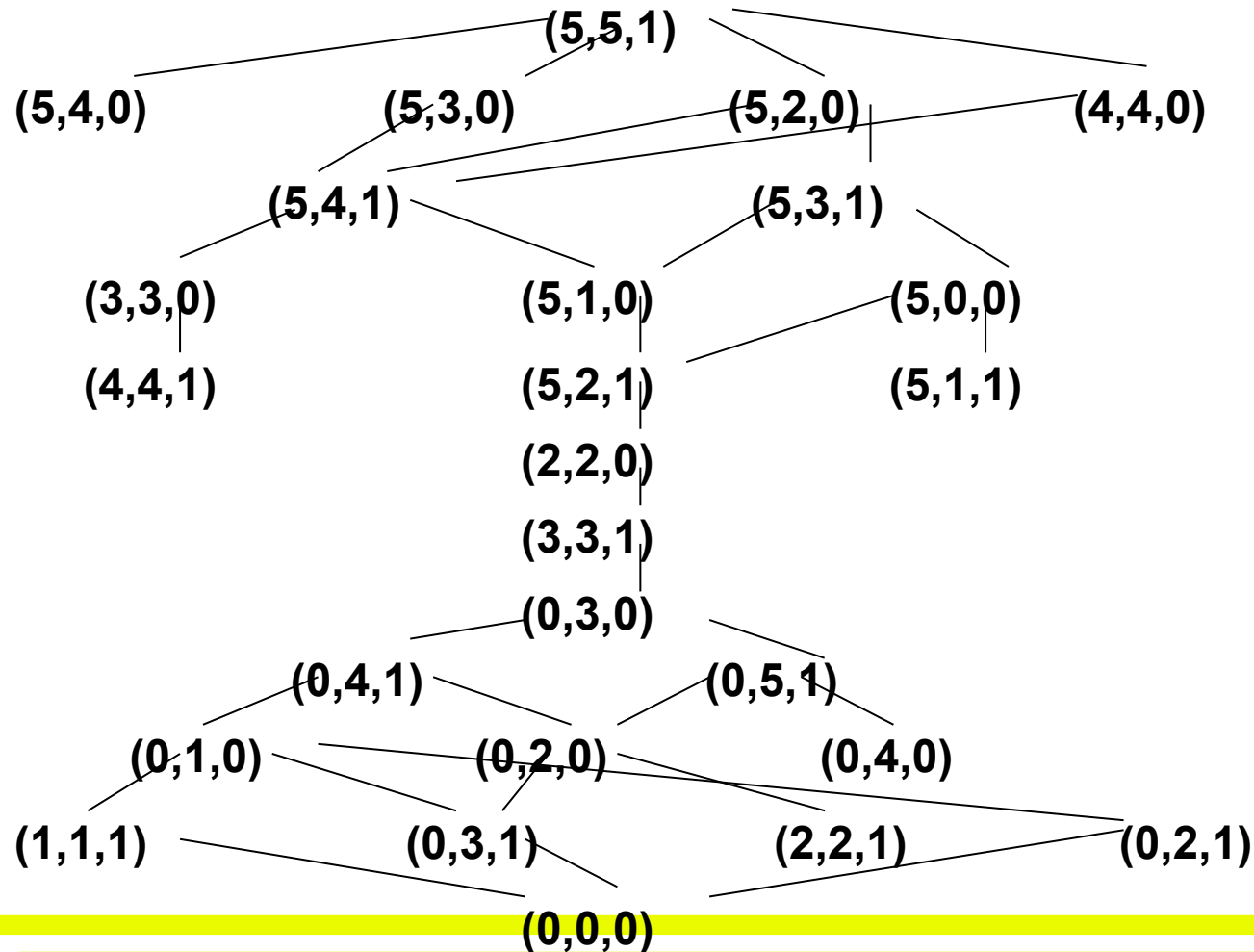
- Na margem esquerda de um rio há 5 missionários e 5 canibais, que querem atravessá-lo para a outra margem . Há um bote que pode transportar no máximo 3 pessoas ao mesmo tempo. Os missionários não podem ficar em menor número que os canibais em qualquer margem, caso contrário serão devorados. Como fazer para que todos cheguem a salvo na outra margem do rio?



- Resolução:
  - Consideraremos a seguinte descrição para o conjunto de estados possíveis utilizados para representar o problema  
 $(Me, Ce, Bte)$  onde,
    - $Me$  = número de missionários na margem esquerda
    - $Ce$  = número de canibais na margem esquerda
    - $Bte = 1$  se o bote está na margem esquerda, 0 na direita
  - Logo, o estado inicial é  $(5, 5, 1)$  e o final  $(0, 0, 0)$

# Missionários e canibais

- Representação do Espaço de Estados do Problema



# Sistemas de Produção

- MYCIN
  - IF
    1. A infecção é principalmente por bactérias, e
    2. O local da cultura é um dos locais esterelizados, e
    3. O local suspeito de entrada do organismo é o trato gastro-intestinal
  - Then
    - Existe uma evidência sugestiva (0.7) de bacteróide.
- DENDRAL
  - IF
    - O espectro para as moléculas apresenta dois picos de massas X1 e X2, tal que:
      1.  $X1 + X2 = M + 28$ , e
      2.  $X1 - 28$  é um pico alto, e
      3.  $X2 - 28$  —é um pico alto, e
      4. Pelo menos X1 ou X2 são altos,
  - Then
    - A molécula contém um grupo cetona.



# Sistemas de Produção

## ♦ VANTAGENS

- Modularidade: podem ser considerados como peças independentes. Novas regras podem ser acrescentadas ao conjunto já existente sem maiores preocupações.
- Naturalidade: pode ser considerada uma forma natural de pensar a descrição de conhecimentos.
- Uniformidade: todas as regras são escritas seguindo o mesmo padrão. Permite que pessoas não familiarizadas com o sistema possam também analisar seu conteúdo.

## ♦ DESVANTAGENS

- Opacidade: é difícil verificar a completeza destes sistemas, bem como verificar os possíveis fluxos de processamento.
- Ineficiência: resulta particularmente do número de regras a combinar e também do esforço do matching necessário ao suporte de execução das regras. (matching entende-se como a verificação das regras que se aplicam ao estado do problema)
- Não raciocinam em vários níveis.
- Não sabem como e quando violam suas próprias regras.
- Não tem acesso ao raciocínio que está por trás das regras.

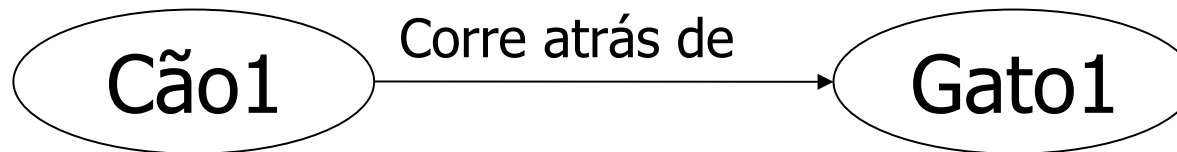
# Redes Semânticas

- ♦ Redes Semânticas são uma tentativa de se formalizar como nosso conhecimento é organizado na memória
- ♦ 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

- Redes Semânticas explicitam o relacionamento entre objetos e propriedades
- Usa-se nodos para representar substantivos, adjetivos, pronomes e nomes próprios.
- Os arcos são reservados basicamente para representar verbos transitivos e preposições.
- Exemplo: “O cão corre atrás do gato”

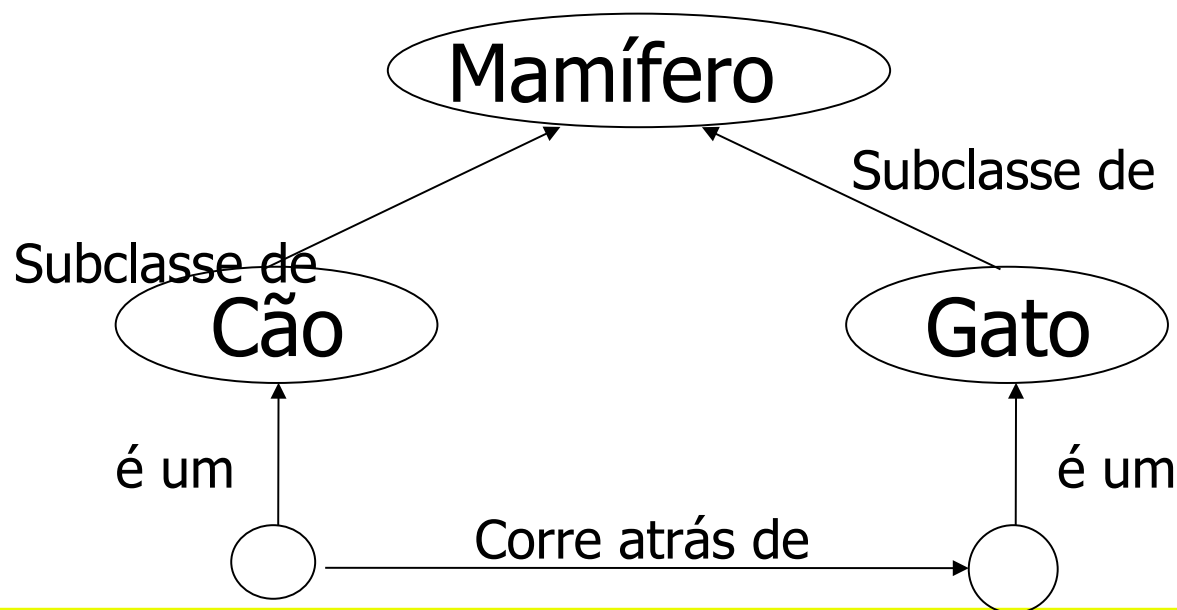


- Pode-se generalizar também a relação entre eles, representando os indivíduos específicos com nodos anônimos e indicando a inclusão destes indivíduos em sua respectiva classe, através da relação “é-um”.

# Redes Semânticas

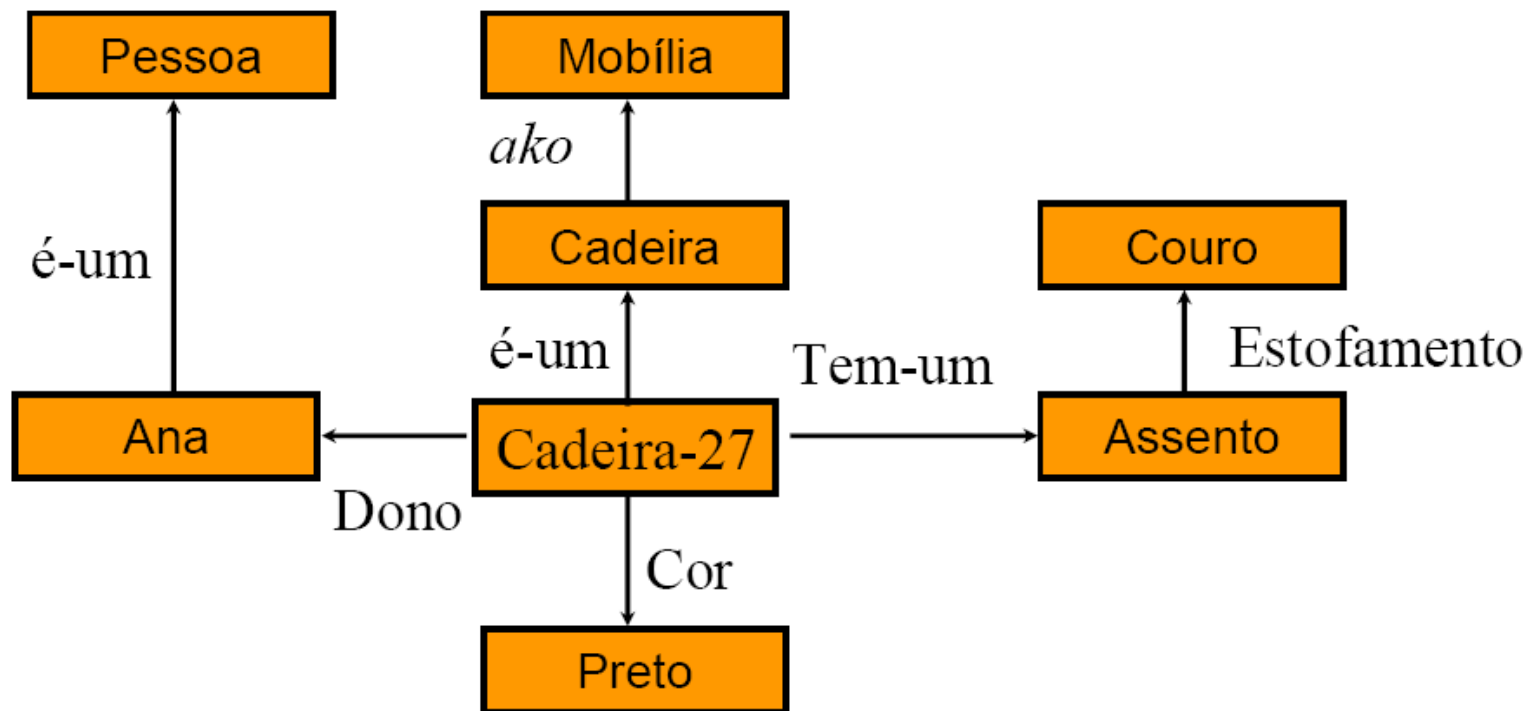
## ◆ Redes Semânticas Elementares

- 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”.



# Redes Semânticas

- Mais um exemplo



# Redes Semânticas

- Forma mais flexível e intuitiva de representar conhecimento
  - Suportam herança de propriedades
  - Relações entre classes e instâncias
    - ako (a-kind-of): relações entre classes
    - é-um (is-a): relações entre classes e instâncias
      - Uma classe pertence a uma classe mais alta ou a uma categoria de objetos
  - Relações todo-parte
    - tem-um (has-a): identifica características ou atributos das entidades
    - parte-de (part-of): identifica características ou atributos das entidades
  - Relações diversas
    - Variados: identifica características gerais

# Redes Semânticas

- Relações primitivas e derivadas
  - Conhecimento primitivo: definido explicitamente pelos fatos
  - Conhecimento derivado: definido pelas regras
    - Permite uma representação mais compacta
    - Permite que conclusões sejam tiradas das observações do domínio
      - Importante porque não observamos tudo diretamente do domínio
      - Muito do que sabemos do domínio é inferido pelas observações e pelo conhecimento mais geral
    - Associar as propriedades com as classes e não com os indivíduos
    - Permitir um atributo especial `é_um` entre um indivíduo e uma classe ou `a_kind_of` entre duas classes, permitindo herança de propriedades



# Herança Múltipla

- Um indivíduo normalmente é membro de mais de uma classe
  - Ex: a mesma pessoa pode ser uma mãe, uma professora, uma árbitra de futebol...
- Um indivíduo pode herdar propriedades de todas as classes – **herança múltipla**
- Se existem valores *default* para as propriedades
  - Podemos ter um problema quando um indivíduo herda valores *default* conflitantes de classes diferentes: **problema da herança múltipla**

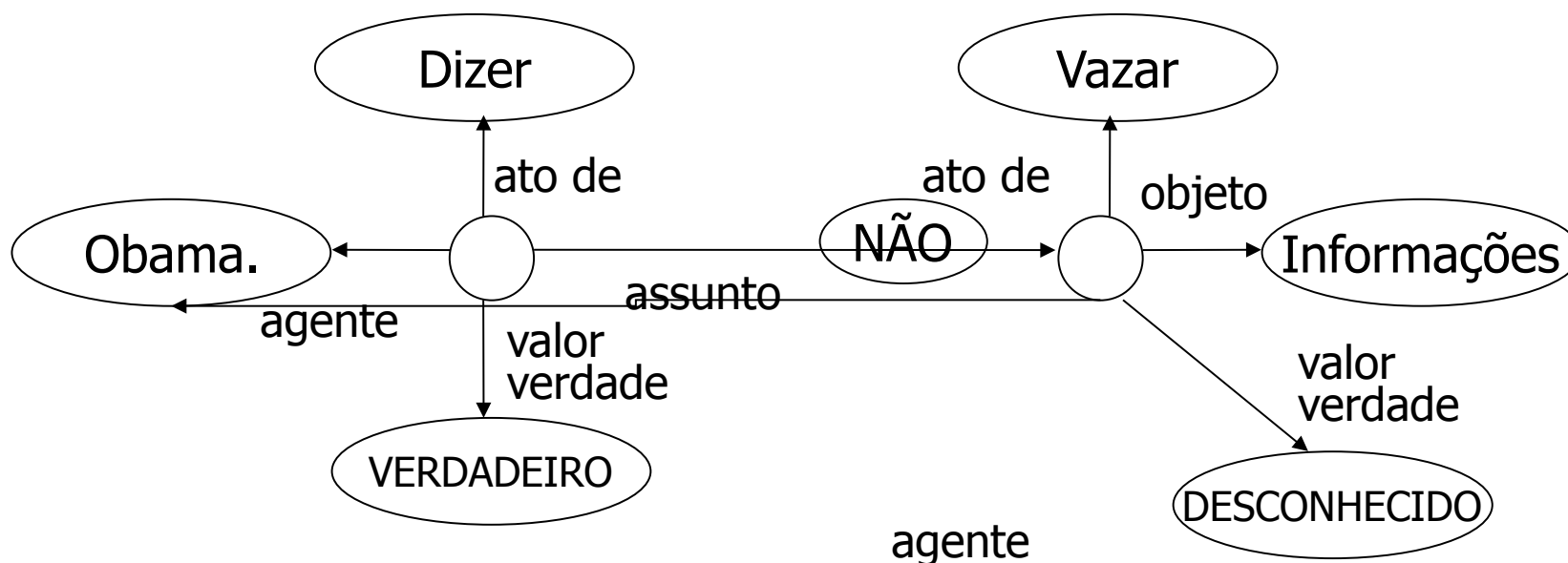
# Sistemas de Redes Semânticas

- Base de conhecimento
  - Formada por nós e *links* da rede
- Máquina de inferência
  - Faz busca e casamento de padrões
  - A busca se dá para frente e para trás através dos *links*
- A busca pode ser usada de várias maneiras para se extrair informações
  - Como uma ferramenta explicativa
  - Para explorar exaustivamente um tópico
  - Para encontrar o relacionamento entre dois objetos

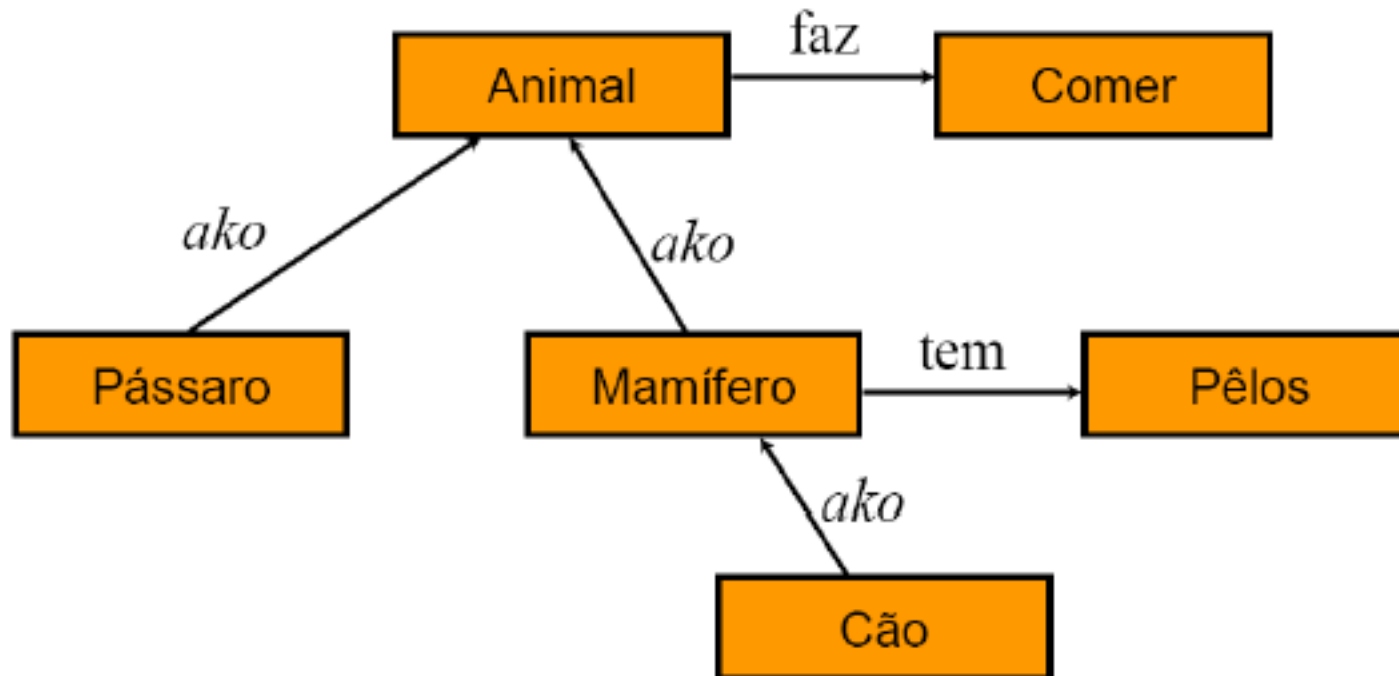
# Redes Semânticas

## ♦ Redes Semânticas com Valores de Verdade

- Pode ser necessário representar fatos dos quais não se conhece o valor de verdade, ou mesmo, fatos que sabemos serem falsos. Uma maneira de fazê-lo é rotulando cada nodo-predicado com um valor VERDADEIRO, FALSO ou DESCONHECIDO.
- “Obama disse que não vazou informações privilegiadas.”



# Exemplo de busca



# Exemplo de Busca

- Para provar a declaração “Cães comem”
  - Pode-se supor que cães comem, e usar busca sobre a rede para provar a hipótese
- Buscando a partir do nó “Cão, temos:
  - “Cão é um mamífero”
  - “Mamífero é um animal”
  - “Animal faz comer”
  - Isto é uma prova para “Cães comem”

# Exploração exaustiva

- Para derivar todo o conhecimento sobre “cães”, usa-se **busca em largura** a partir do nó “cão”
  - “Cães são mamíferos”
  - “Cães tem pêlos”
  - “Cães são animais”
  - “Cães comem”

# Relacionando dois tópicos

- Para verificar se “Cães” e “Pássaros” estão relacionados, pode-se executar, a partir de ambos os nós, uma **busca em largura**
- A interseção entre os nós visitados nos dá o relacionamento entre os nós iniciais
- Isto é chamado **ativação distribuída** ou **interseção de busca**

# Vantagens

- Representação visual fácil de entender
  - Os humanos entendem os relacionamentos com facilidade sem necessariamente saber a sintaxe
  - Ajuda os construtores da BC a estruturar seu conhecimento
- Flexibilidade na manipulação de nós e *links*
  - Adição, exclusão e modificação
- Economia escrita e tempo
  - Herança via relações “é-um” e “ako”
- “Capta senso-comum”
  - Semelhante ao armazenamento de informações no cérebro



# Limitações

- Busca em redes semânticas grandes pode ser muito ineficiente
- Não há homogeneidade na definição de nós e links
- Hereditariedade pode causar dificuldades no tratamento de exceções
- Pode haver conflito entre características herdadas
- É difícil representar conhecimento procedimental
- Sequenciamento e tempo não são explícitos

# Quadros (Frames)

- ♦ Os Quadros ou Cenários (“Frames”), e sua variação, os roteiros (“Scripts”), foram 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. Alguém que já assistiu alguma vez a um júri popular sabe que tipo de “quadro” irá encontrar se for a outro. (juiz, auxiliar de justiça, réu, advogado de defesa, promotor, etc.)
- ♦ Idéias fundamentais introduzidas por Marvin Minsky em 1975 (“A framework to represent knowledge”).
- ♦ Está na origem das idéias que levaram às linguagens de programação orientadas a objetos.

# Quadros (Frames)

- ♦ **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, além dos valores do objeto em particular, valores default, ponteiros para outros quadros, e conjuntos de regras de procedimento que podem ser implementados.
- ♦ Se os valores dos atributos forem apontadores para outros quadros, cria-se uma rede de dependências entre os quadros.
- ♦ Os conjuntos de procedimentos indicam que procedimento deve ser executado quando certas condições forem satisfeitas, por exemplo: ao ser criado o atributo, ao ser lido o valor do atributo, ao ser modificado o valor do atributo, ou ao ser destruído o valor do atributo.

# Quadros (Frames)

- ♦ Os quadros também são organizados em uma hierarquia de especialização, criando outra dimensão de dependência entre eles (herança). Permite a especificação de propriedades de uma classe de objetos através da declaração de que esta classe é uma subclasse de outra que goza da propriedade em questão.
- ♦ O processo de herança e instanciação favorece a reutilização de código evitando definições repetitivas e aproveitando funções de acesso definidas para as facetas se-lido, se-escrito, se-necessário, etc
- ♦ Deve-se notar que as estruturas de quadros são ativas, pois sua manipulação causa o disparo automático das facetas.
- ♦ São úteis para domínio de problemas onde a forma e o conteúdo do dado desempenham um papel importante na solução do problema.

# Quadros (Frames)

- ◆ 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)

# Frames

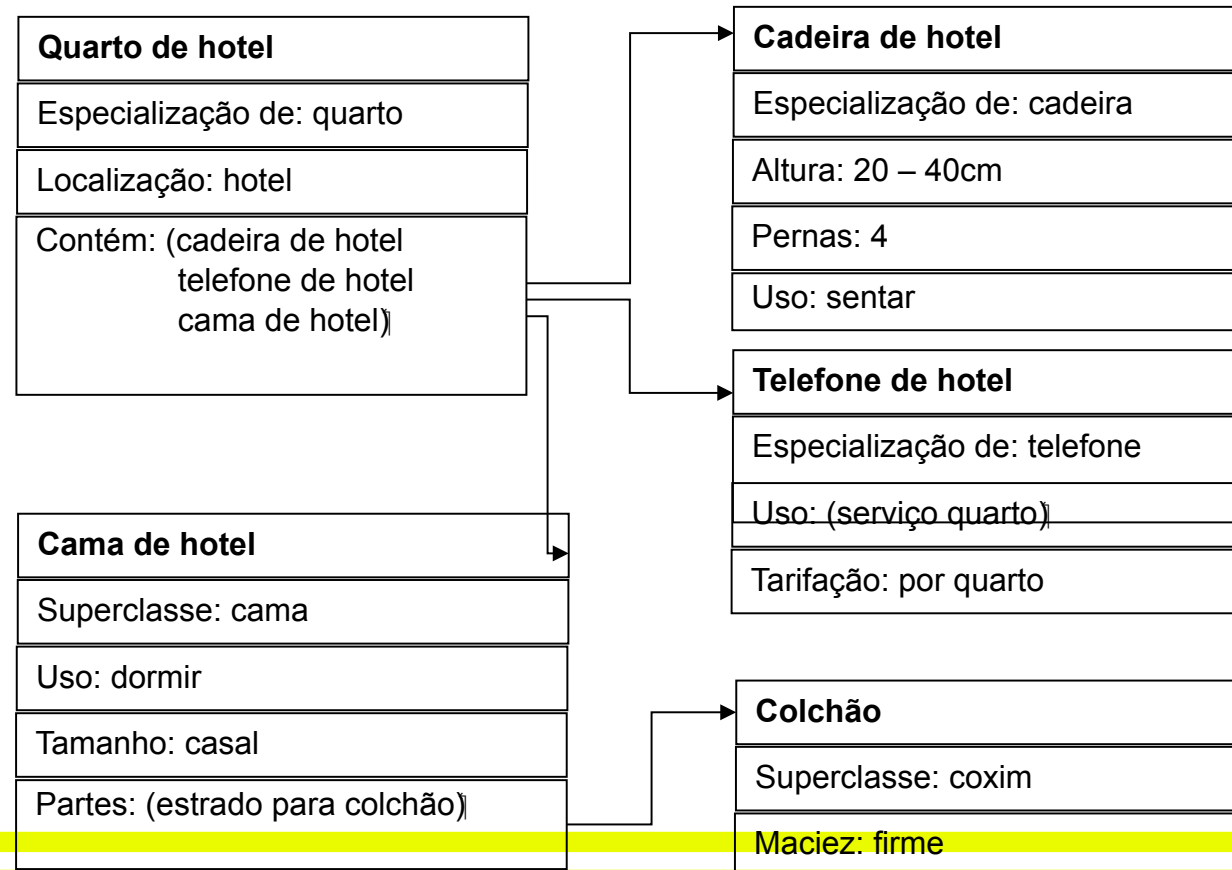
- *Frames*:
  - Possuem pelo menos dois atributos:
    - *Nome*
    - *ako* ou *is-a*
  - A fim de melhorar a estruturação (hierarquia), privilegiam dois tipos de relações:
    - *ako*: relação entre classe e sub-classe.
    - *is-a*: relação entre classe e instância.
- Cada atributo:
  - Aponta para um outro *frame* ou para um tipo primitivo, ex. *string*;
  - Consiste em um conjunto de **facet**as (atributos de atributos).

# Quadros (Frames)

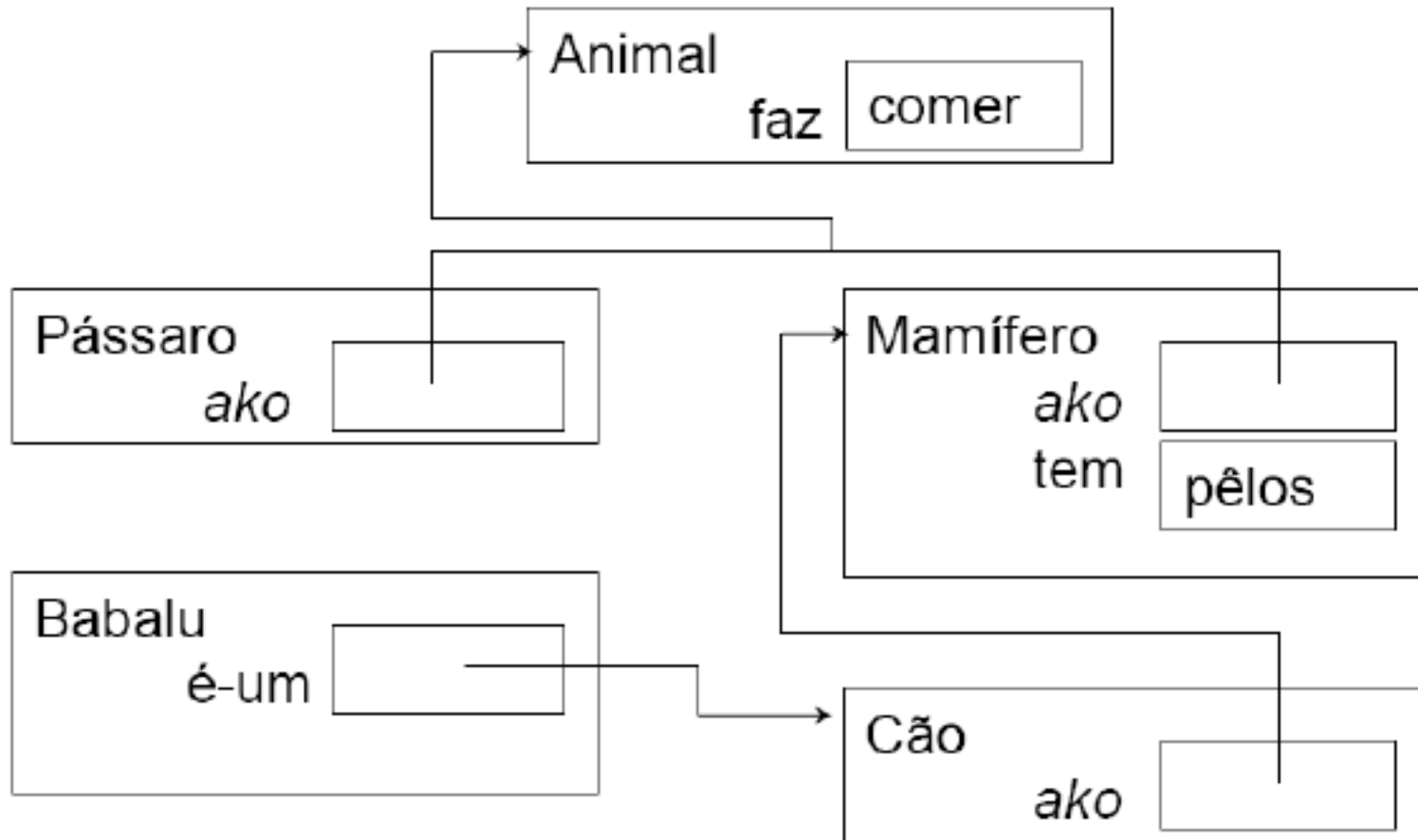
- Parte de uma descrição por frame de um quarto de hotel.
- Cada frame individual pode ser visto como uma estrutura de dados.

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)



# Outro exemplo





# 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.

# Roteiros (Scripts)

- ♦ Originalmente concebido por Schank e seu grupo de pesquisa (1977).
- ♦ Pesquisa em roteiros examina a organização do conhecimento na memória e o papel que esta organização desempenha no raciocínio.
- ♦ São estruturas semelhantes aos quadros que descrevem seqüências estereotipadas de eventos em contextos particulares.
- ♦ A principal diferença que se pode estabelecer em relação aos quadros fica a nível das estruturas adotadas.
- ♦ Os eventos descritos formam uma cadeia causal.

# Roteiros (Scripts)

- ♦ O início da cadeia é o conjunto de condições de entrada que permite a possibilidade de ocorrência do primeiro evento do roteiro.
- ♦ O fim da cadeia é o conjunto de resultados que permitirá a ocorrência de eventos posteriores.
- ♦ O raciocínio com roteiros serve especialmente para verificar se determinado evento ocorreu e também para verificar a relação entre os eventos, o que pode ser conseguido pelo exame da cadeia causal.

# Roteiros (Scripts)

Roteiro: comer-em-restaurante

Apoio: (restaurantes, dinheiro, alimento, menu, mesas, cadeiras)

Funções: (pessoas com fome, encontro de pessoas)

Ponto-de-vista: (pessoas com fome)

Tempo-de-ocorrência: (tempo-de-operação do restaurante)

Lugar-de-ocorrência: (localização do restaurante)

Seqüência-de-eventos

primeiro : Inicie o roteiro entrada-no-restaurante

então: Se (há-convite-para-sentar ou reservas) então siga roteiro orientação-do-garçom

então: siga roteiro aguarde-sentado

então: siga roteiro solicite-comida

então: siga roteiro comer, a menos que haja uma longa espera, caso em que seguirá o roteiro sai-do-restaurante-furioso.

- ♦ Outros Roteiros: refeição rápida (fast-food).