Agentes baseados em Conhecimento

Lógica de Primeira Ordem

Sumário

Agentes baseados em Conhecimento:

◇ O que é um agente baseado em conhecimento (ABC)

◇ Bases de conhecimento

◇ O mundo "Wumpus"

◇ Lógica em geral

◇ Revisão de lógica proposicional

Lógica de Primeira Ordem

◇ Sintaxe e semântica

◇ Manipulação de sentenças

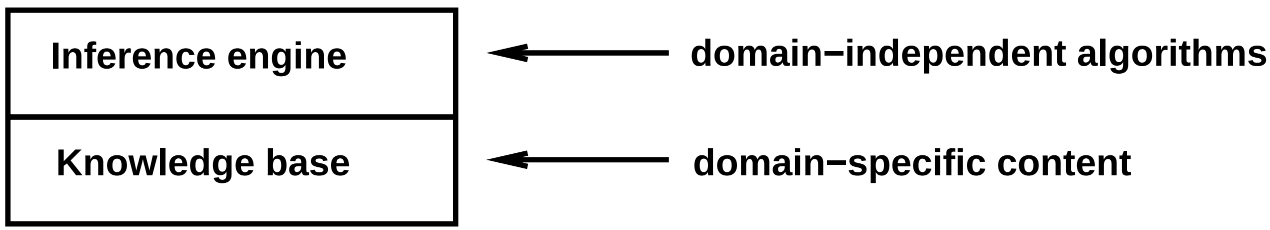
O que é um ABC

Até agora, ignoramos uma metodologia para descrever estados e ações: agentes apenas realizavam uma busca em um espaço de estados e ações pré-determinados.

Vamos agora estudar agentes com a capacidade de adaptar um modelo consistente do mundo à medida que se observam novos estados, e de se deduzir um curso de ações com base nesse modelo.

ABC = Base de conhecimento + mecanismo de inferência

Bases de conhecimento



Base de conhecimento (BC) = conjunto de sentenças em uma linguagem formal (linguagem de representação do conhecimento).

Linguagem formal = um conjunto de *strings* de símbolos de um *alfabeto* (Teoria da Computação)

Abordagem declarativa para construir um agente (ou outro sistema):

Informe o que se precisa saber; e depois **pergunte** o que fazer — respostas vem da BC.

ABCs: Níveis de Entendimento

Agentes podem ser entendidos ao nível de conhecimento
  i.e., o que eles sabem, não interessando **como** foi implementado;

Ou ao nível lógico
  i.e., como o conhecimento é codificado em sentenças e manipulado;

Ou ao nível de implementação
  i.e., **estruturas** de dados na BC e **algoritmos** que os manipulam

Um agente simples baseado em conhecimento

```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

O agente deve ser capaz de:

Representar estados, ações, etc.

- Incorporar novas observações
- Atualizar representações internas do mundo
- Deduzir propriedades ocultas do mundo
- Deduzir ações apropriadas

Deve ser capaz $\neq$ sempre seremos capazes de projetar um capaz ...

Descrição PAGE do mundo “Wumpus”

Percepções Brisa, Brilho, Cheiro

Ações Giro à esquerda, Giro à direita,Adiante, Agarre, Largue, Atire

Objetivos Levar ouro para o ponto inicial sem cair em buracos ou no quadrado do temível “Wumpus”

Ambiente

Quadrados adjacentes ao “Wumpus” têm mau cheiro
Quadrados adjacente a buracos têm brisa
Brilho é observado se e só se o ouro estiver no mesmo quadrado
Tiros matam o “Wumpus” se você estiver cara a cara com ele
Apenas uma bala
Ouro só pode ser agarrado se estiver no mesmo quadrado
O ouro largado cai no mesmo quadrado

Características do mundo Wumpus:

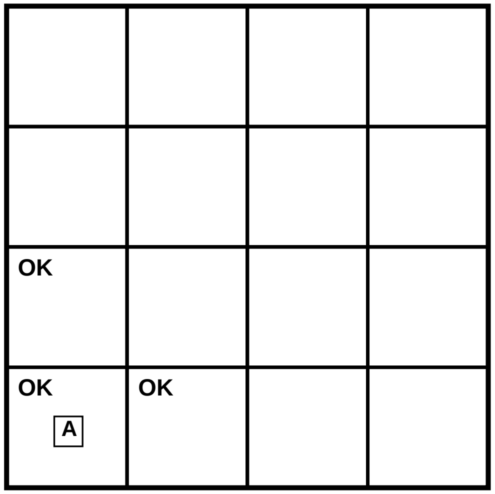
Mundo é determinístico?? Sim—resultados de ações são especificados com exatidão

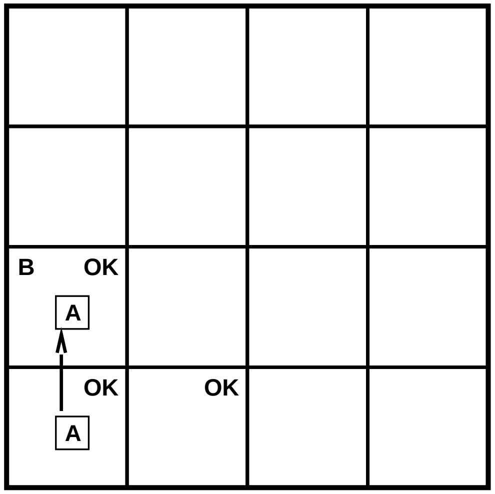
Mundo é completamente acessível?? Não—percepção é local

Mundo é estático?? Sim—“Wumpus” e buracos não se movem

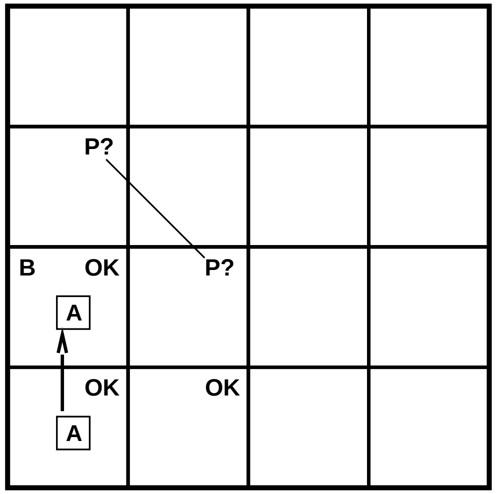
Mundo é discreto?? Sim

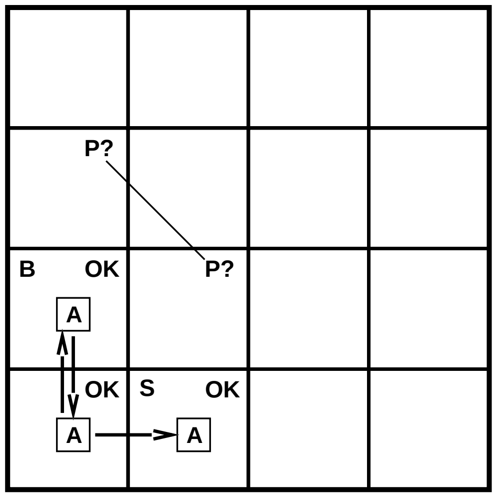
Explorando o mundo "Wumpus"



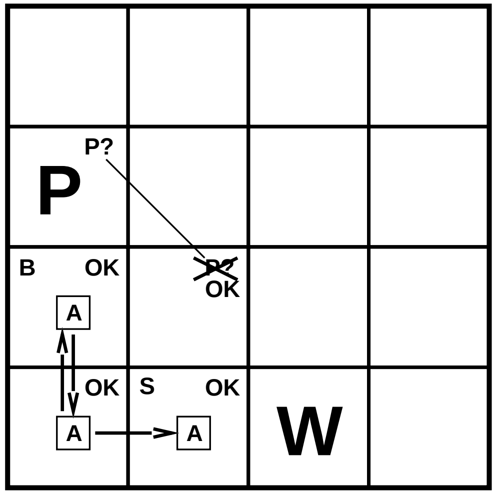


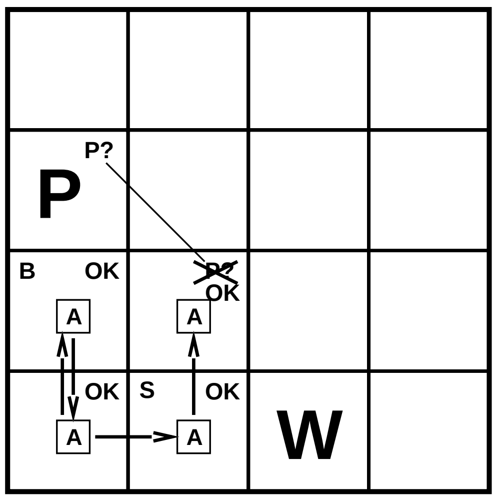
Explorando o mundo "Wumpus"



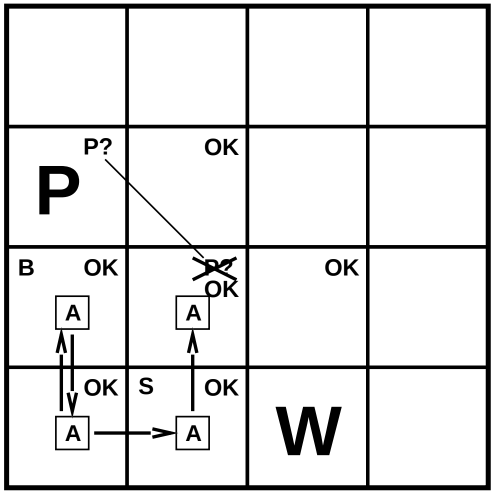


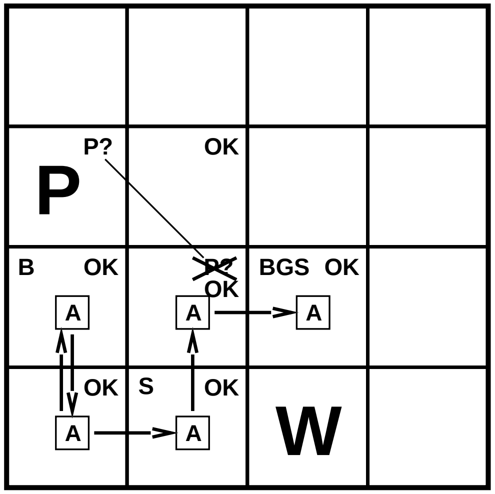
Explorando o mundo “Wumpus”



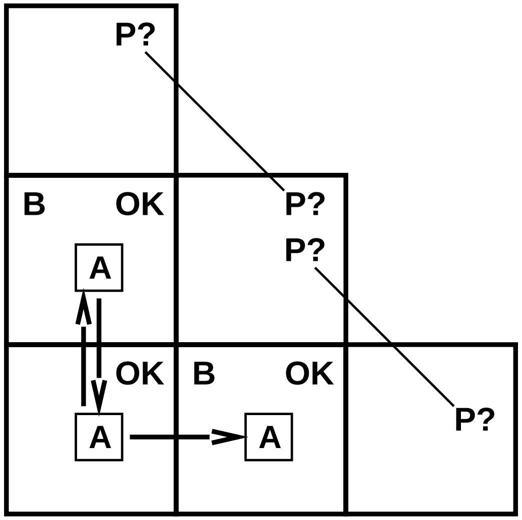


Explorando o mundo "Wumpus"



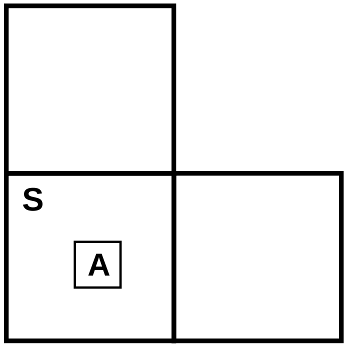


Outros pontos críticos



Brisa em (1,2) e (2,1)
$\Rightarrow$nenhuma ação segura

Assumindo buracos distribuídos uniformemente,
(2,2) é provavelmente um buraco



Cheiro em (1,1)
$\Rightarrow$ não pode se mover
Possa usar uma estratégia <u>coercitiva</u>:
atiro para a frente
"Wumpus" estava lá $\Rightarrow$ morto $\Rightarrow$ seguro
"Wumpus" não estava lá $\Rightarrow$ seguro

Lógica em geral

Uma **lógica** é uma linguagem formal para representar informação a partir da qual podem ser tiradas conclusões.

A <u>sintaxe</u> define as sentenças da linguagem.

A <u>semântica</u> define o "significado" das sentenças; i.e., define a <u>veracidade</u> de uma sentença em um mundo

Por exemplo, na linguagem aritmética usual

$x + 2 \geq y$ é uma sentença; $x2 + y >$ não é uma sentença

$x + 2 \geq y$ é verdadeiro se o número $x + 2$ não for menor que o número y

$x + 2 \geq y$ é verdadeiro em um mundo onde $x = 7,\ y = 1$
$x + 2 \geq y$ é falso em um mundo onde $x = 0,\ y = 6$

Tipos de lógica

Lógicas são caracterizadas pelo que elas definem como "primitivas"

Compromisso **ontológico**: o que existe—fatos? objetos? tempo? crenças? Ontologia é o que se assume como parte do mundo do agente.

Compromisso **epistemológico**: que estados de conhecimento? Epistemologia é o que se assume como possíveis estados de conhecimento do agente.

Language	Ontological Commitment	Epistemological Commitment
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $0 \ldots 1$
Fuzzy logic	degree of truth	degree of belief $0 \ldots 1$

Implicação Lógica

$BC \models \alpha$

A base de conhecimento BC implica ou vincula (*entails*) uma sentença α se e só se α é verdadeira em todos os mundos onde BC é verdadeira.

Diz-se então que α é uma implicação de (ou está vinculada à — *entailed by*) BC.

Por exemplo, a BC que contém "o Flamengo venceu" e "o Vasco empatou" implica "O Flamengo venceu ou o Vasco empatou"

Modelos

Estudiosos de lógica pensam tipicamente em termos de <u>modelos</u>, que são mundos formalmente estruturados, com respeito aos quais veracidade pode ser avaliada

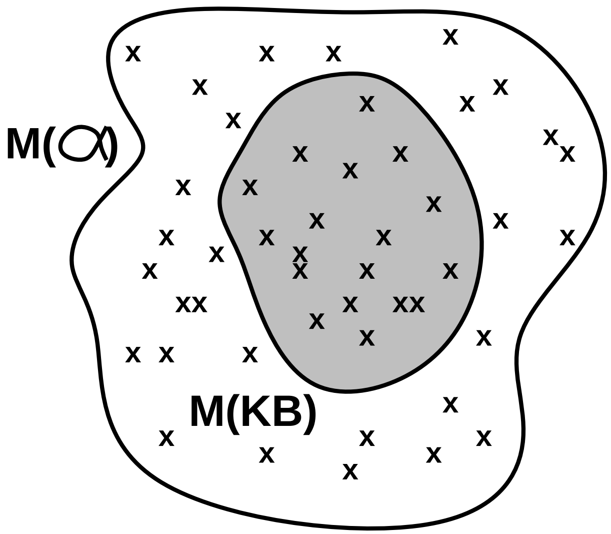
Dizemos que m é um <u>modelo</u> de uma sentença α se α é verdade em m

$M(\alpha)$ é o conjunto de todos os modelos de α

Então $BC \models \alpha$ se e só se $M(BC) \subseteq M(\alpha)$

Por exemplo BC = Palmeiras venceu e Corinthians venceu

α = Palmeiras venceu



Inferência

$BC \vdash_i \alpha =$ sentença α pode ser derivada de BC por um procedimento i

Correção: i é correto ($sound$) se
sempre que $BC \vdash_i \alpha$, também for verdade que $BC \models \alpha$

Um procedimento correto não **inventa** coisas.

Completeza: i é completo se
sempre que $BC \models \alpha$, também for verdade que $BC \vdash_i \alpha$

Um procedimento completo **prova** qualquer sentença implicadada pela BC.

Definiremos depois uma lógica (lógica de primeiro ordem) que é expressiva o bastante para dizer quase qualquer coisa de interesse, e para a qual existe um procedimento de inferência **correto e completo**.

O procedimento responderá a qualquer pergunta cuja resposta segue do que é conhecido pelo BC.

Lógica proposicional: Sintaxe

A lógica proposicional é a lógica mais simples—ilustra idéias básicas

Os símbolos proposicionais P_1, P_2 etc são sentenças

Se S é uma sentença, $\neg S$ é uma sentença

Se S_1 e S_2 são sentenças, $S_1 \wedge S_2$ é uma sentença

Se S_1 e S_2 são sentenças, $S_1 \vee S_2$ é uma sentença

Se S_1 e S_2 são sentenças, $S_1 \Rightarrow S_2$ é uma sentença

Se S_1 e S_2 são sentenças, $S_1 \Leftrightarrow S_2$ é uma sentença

Lógica proposicional: Semântica

Cada modelo especifica verdadeiro/falso para cada símbolo proposicional

Por exemplo	A	B	C
	$Verdadeiro$	$Falso$	$Falso$

Regras para avaliar a verdade com respeito a um modelo m:

$\neg S$	é verdadeiro iff	S	é falso		
$S_1 \wedge S_2$	é verdadeiro iff	S_1	é verdadeiro <u>e</u>	S_2	é verdadeiro
$S_1 \vee S_2$	é verdadeiro iff	S_1	é verdadeiro <u>ou</u>	S_2	é verdadeiro
$S_1 \Rightarrow S_2$	é verdadeiro iff	S_1	é falso <u>ou</u>	S_2	é verdadeiro
i.e.,	é falso iff	S_1	é verdadeiro <u>e</u>	S_2	é falso
$S_1 \Leftrightarrow S_2$	é verdadeiro iff	$S_1 \Rightarrow S_2$	é verdadeiro <u>e</u>	$S_2 \Rightarrow S_1$	é verdadeiro

Inferência proposicional: Método da enumeração

Seja $\alpha = A \vee B$ e $BC = (A \vee C) \wedge (B \vee \neg C)$

É o caso que $BC \models \alpha$?
Confira todos os possíveis modelos— α será verdadeira quando BC for verdadeira

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α
$False$	$False$	$False$				
$False$	$False$	$True$				
$False$	$True$	$False$				
$False$	$True$	$True$				
$True$	$False$	$False$				
$True$	$False$	$True$				
$True$	$True$	$False$				
$True$	$True$	$True$				

Formas normais

Outros enfoques para inferência usam operações sintáticas em sentenças, frequentemente expressas em formas padronizadas

Forma Normal Conjuntiva (CNF—universal)

$$\textit{conjunção} \text{ de } \underbrace{\textit{disjunções} \text{ de } \textit{literais}}_{\textit{cláusulas}}$$

Por exemplo, $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Forma Normal Disjuntiva (DNF—universal)

$$\textit{disjunção} \text{ de } \underbrace{\textit{disjunções} \text{ de } \textit{literais}}_{\textit{cláusulas}}$$

Por exemplo, $(A \wedge B) \vee (A \wedge \neg C) \vee (A \wedge \neg D) \vee (\neg B \wedge \neg C) \vee (\neg B \wedge \neg D)$

Forma de Horn (restrita)

conjunção de *cláusulas Horn* (cláusulas com ≤ 1 literais positivos)

Por exemplo, $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

Comumente escrito como um conjunto de implicações:

$B \Rightarrow A$ e $(C \wedge D) \Rightarrow B$

Validez e Satisfatibilidade

Uma sentença é <u>válida</u> se for verdadeira em <u>todos</u> os modelos

Por exemplo, $A \lor \neg A$, $\quad A \Rightarrow A$, $\quad (A \land (A \Rightarrow B)) \Rightarrow B$

Validez relaciona-se com inferência pelo <u>Teorema da Dedução</u>:

$BC \models \alpha$ se e só se $(BC \Rightarrow \alpha)$ é válido

Uma sentença é <u>satisfatível</u> se for verdadeira em <u>algum</u> modelo

Por exemplo, $A \lor B$, $\quad C$

Uma sentença é <u>não-satisfatível</u> se não for verdadeira em <u>nenhum</u> modelo

Por exemplo, $A \land \neg A$

Satisfatibilidade relaciona-se com inferência por:

$BC \models \alpha$ se e só se $(BC \land \neg\alpha)$ é não-satisfatível

i.e., prova-se α por *reductio ad absurdum*

Métodos de prova

Os métodos de prova se dividem em dois tipos básicos:

Verificação de modelo

enumeração de tabela-verdade (correta e completa para lógica proposicional)

busca heurística em espaço de modelos (correto mas incompleto)

Por exemplo, o algoritmo GSAT (Ex. 6.15)

Aplicação de regras de inferência

Geração correta de sentenças novas a partir das velhas

<u>Prova</u> = uma sucessão de aplicações de regras de inferência

Possa usar regras de inferência como operadores em um algoritmo de busca

Regras de inferência para lógica proposicional

Resolução (para CNF): completa para lógica proposicional

$$\frac{\alpha \vee \beta, \qquad \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

Modus Ponens (para Forma de Horn): completa para BCs Horn

$$\frac{\alpha_1, \ldots, \alpha_n, \qquad \alpha_1 \wedge \cdots \wedge \alpha_n \;\Rightarrow\; \beta}{\beta}$$

Pode ser usado com forward chaining ou backward chaining

Outras regras de inferência . . .

Sumário

Agentes lógicos aplicam inferência a uma base de conhecimento para obter informação nova e tomar decisões

Conceitos básicos de lógica:
–sintaxe: estrutura formal das sentenças
–semântica: veracidade das sentenças com respeito a modelos
–implicação: verdade consequente de uma sentença dada uma outra
–inferência: sentenças derivadas de outras sentenças
–correção: derivações produzem apenas sentenças implicadas –completeza: derivações podem produzir todas as sentenças implicadas

O mundo "Wumpus" requer a habilidade para representar informação parcial ou oculta, argumentação através de casos, etc.

Lógica proposicional é suficiente para algumas destas tarefas

O método da tabela-verdade é correto e completo para lógica proposicional

Sintaxe de LPO: Elementos básicos

Constantes	$KingJohn,\ 2,\ UCB, \ldots$
Predicados	$Brother, \rangle, \ldots$
Funções	$Sqrt,\ LeftLegOf, \ldots$
Variáveis	$x,\ y,\ a,\ b, \ldots$
Conectivos	$\wedge \quad \vee \quad \neg \quad \Rightarrow \quad \Leftrightarrow$
Igualdade	$=$
Quantificadores	$\forall\ \exists$

Termos e Sentenças Atômicas

Um objeto é uma entidade no mundo. Um termo é uma expressão lógica que se refere a um objeto. Uma sentença atômica expressa fatos.

$$\begin{aligned} \text{Sentença atômica} &= predicate(term_1, \ldots, term_n) \\ &\quad \text{ou } term_1 = term_2 \end{aligned}$$

$$\begin{aligned} \text{Termo} &= function(term_1, \ldots, term_n) \\ &\quad \text{ou } constant \text{ ou } variable \end{aligned}$$

Por exemplo, $Brother(KingJohn, RichardTheLionheart)$
$(Length(LeftLegOf(Richard)))$

Sentenças atômicas podem ser combinadas à conectivos, formando sentenças complexas:

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \;\Rightarrow\; S_2, \quad S_1 \;\Leftrightarrow\; S_2$$

Por exemplo $Sibling(KingJohn, Richard) \;\Rightarrow\; Sibling(Richard, KingJohn)$
$>(1,2) \vee \leq(1,2)$

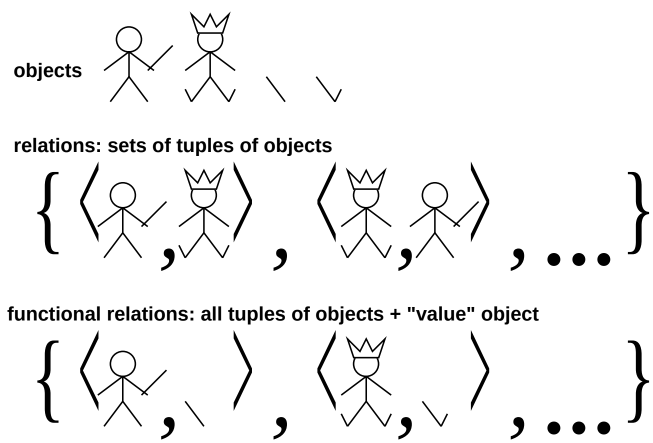
Verdade em lógica de primeiro-ordem

Sentenças são verdadeiras com respeito a um modelo e uma interpretação. Um modelo contém objetos e relações entre eles

Uma interpretação especifica referências para
símbolos constantes →objetos
símbolos de predicados →relações
símbolos funcionais →relações funcionais

Uma sentença atômica $predicate(term_1, \ldots, term_n)$ é verdadeira se os objetos referidos por $term_1, \ldots, term_n$ estão na relação referida por *predicate*

Exemplo de modelo:



Quantificação universal

$\forall \langle variaveis \rangle \ \langle sentenca \rangle$

Todo mundo na Unicamp é inteligente:
$\forall x \ \ At(x, Unicamp) \Rightarrow Clever(x)$

$\forall x \ \ P$ é equivalente à <u>conjunção</u> de <u>instâncias</u> de P

$$
\begin{aligned}
& At(KingJohn, Unicamp) \Rightarrow Clever(KingJohn) \\
\wedge \ & At(Richard, Unicamp) \Rightarrow Clever(Richard) \\
\wedge \ & At(Unicamp, Unicamp) \Rightarrow Clever(Unicamp) \\
\wedge \ & \ldots
\end{aligned}
$$

Tipicamente, $\Rightarrow$ é o conectivo principal a se usar com $\forall$.
Engano comum: usar $\wedge$ como conectivo principal para $\forall$:

$$\forall x \ \ At(x, Unicamp) \wedge Clever(x)$$

significa " Todo mundo está na Unicamp e todo mundo é inteligente"

Quantificação existencial

$\exists \langle variaveis \rangle \ \ \langle sentenca \rangle$

Alguém em Stanford é inteligente:
$\exists x \ \ At(x, Stanford) \land Clever(x)$

$\exists x \ \ P$ é equivalente a uma <u>disjunção</u> de <u>instâncias</u> de P

$$
\begin{array}{ll}
 & At(KingJohn, Stanford) \land Clever(KingJohn) \\
\lor & At(Richard, Stanford) \land Clever(Richard) \\
\lor & At(Stanford, Stanford) \land Clever(Stanford) \\
\lor & \ldots
\end{array}
$$

Tipicamente, $\land$ é o conectivo principal a se usar com $\exists$.
Engano comum: usar $\Rightarrow$ como conectivo principal para $\exists$:

$$\exists x \ \ At(x, Stanford) \ \Rightarrow \ Clever(x)$$

é verdade se existir alguém que não está em Stanford (expanda como uma disjunção e verifique)!

Propriedades de quantificadores

$\forall x \ \forall y$ é igual a $\forall y \ \forall x$ (Por que??)

$\exists x \ \exists y$ é igual a $\exists y \ \exists x$ (Por que??)

$\exists x \ \forall y$ não é igual a $\forall y \ \exists x$

$\exists x \ \forall y \ Loves(x, y)$
"Há uma pessoa que ama todo o mundo"

$\forall y \ \exists x \ Loves(x, y)$
"Todo mundo é amado por pelo menos uma pessoa"

Dualidade dos quantificadores: cada um pode ser expresso usando-se o outro

$\forall x \ Likes(x, IceCream) \qquad \neg\exists x \ \neg Likes(x, IceCream)$

$\exists x \ Likes(x, Broccoli) \qquad \neg\forall x \ \neg Likes(x, Broccoli)$

Manipulando orações

Irmãos são irmãos ou irmãs

$\forall\, x, y \;\; Brother(x, y) \;\Leftrightarrow\; Sibling(x, y).$

Fraternalidade é reflexiva

$\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x)$

A mãe da pessoa é o genitor fêmea da pessoa

$\forall\, x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \land Parent(x, y))$

Um primo em primeiro grau é um filho de irmão de um pai

$\forall\, x, y \;\; FirstCousin(x, y) \;\Leftrightarrow\; \exists\, p, ps \;\; Parent(p, x) \land Sibling(ps, p) \land Parent(ps, y)$

Igualdade

$term_1 = term_2$ é verdade sob uma determinada interpretação se e só se $term_1$ e $term_2$ se referem ao mesmo objeto

Por exemplo, $1 = 2$ e $\forall\, x \;\; \times(Sqrt(x), Sqrt(x)) = x$ são satisfatíveis
$2 = 2$ é válido

Por exemplo, definição correta de $Sibling$ em termos de $Parent$:

$$\forall\, x, y \;\; Sibling(x,y) \;\Leftrightarrow\; [\neg(x = y) \wedge \exists\, m, f \;\; \neg(m = f) \wedge Parent(m,x) \wedge Parent(f,x) \wedge Parent(m,y) \wedge Parent(f,y)]$$

Interagindo com BCs em LPO

Primeiro passo: definir percepções e ações.

Suponha um agente no mundo "Wumpus" usando uma BC baseada em LPO, que percebe um cheiro e uma brisa (mas nenhum brilho) em $t = 5$:

$\textsc{Tell}(KB, Percept([Smell, Breeze, None], 5))$
$\textsc{Ask}(KB, \exists\, a \;\; Action(a, 5))$

I.e., a BC requer alguma ação particular em $t = 5$?

Resposta: Sim, $\{a/Shoot\}$ $\leftarrow$ substituição (*binding list*)

Dado uma sentença S e uma substituição σ, $S\sigma$ denota o resultado de "ligar" σ a S; por exemplo,
$S = Cleverer(x, y)$
$\sigma = \{x/Hillary, y/Bill\}$
$S\sigma = Cleverer(Hillary, Bill)$

$\textsc{Ask}(\mathrm{KB}, \mathrm{S})$ devolve algum/todos σ tais que $KB \models S\sigma$

Um Agente Reflexo

<u>"Percepção"</u>
$\forall\, b, g, t \;\; Percept([Smell, b, g], t) \;\Rightarrow\; Smell(t)$
$\forall\, s, b, t \;\; Percept([s, b, Glitter], t) \;\Rightarrow\; AtGold(t)$

<u>Reflexo</u>: $\forall\, t \;\; AtGold(t) \;\Rightarrow\; Action(Grab, t)$

<u>Reflexo com estado interno</u>: já tenho o ouro?
$\forall\, t \;\; AtGold(t) \wedge \neg Holding(Gold, t) \;\Rightarrow\; Action(Grab, t)$

$Holding(Gold, t)$ não pode ser observado
$\Rightarrow$ manter rastro de mudança é <u>essencial</u>

Fatos são válidos em <u>situações</u>, e não eternamente!

Mantendo rastro de mudanças

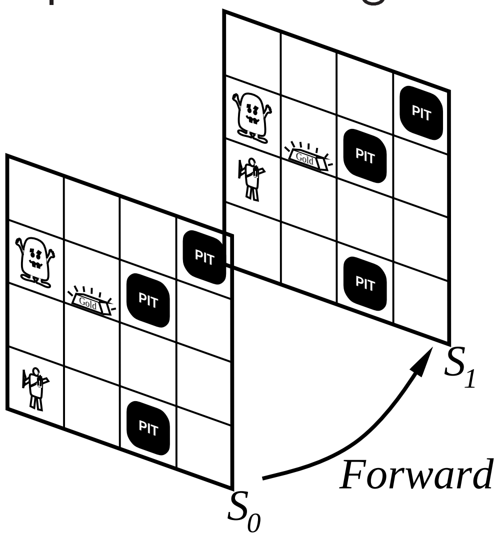
Cálculo de Situações é um modo de representar mudanças em LPO: Adiciona um "argumento de situação" para cada predicado temporário

Por exemplo, Now em $Holding(Gold, Now)$ denota uma situação.

Situações são ligadas pela função $Result$

$Result(a, s)$ é a situação que resulta ao se fazer a em s

Coisas que não mudam não precisam do argumento de situação.



Descrevendo ações (I)

"Axioma do Efeito" — descreve mudanças devido à ação
$\forall\, s \;\; AtGold(s) \;\Rightarrow\; Holding(Gold, Result(Grab, s))$

"Axioma do Frame" — descreve permanências devido à ação
$\forall\, s \;\; HaveArrow(s) \;\Rightarrow\; HaveArrow(Result(Grab, s))$

Problema do Frame: ache um modo elegante para lidar com permanências

Problema da Qualificação: descrições verdadeiras de ações reais requerem infinito detalhamento — e se ouro for escorregadio ou estiver colado no chão ou . . . É difícil garantir a efetividade das ações.

Problema de Ramificação: ações reais têm muitas consequências secundárias — poeira das brisas cai no ouro, luvas se desgastam, . . . É difícil considerar corretamente a importância das consequências.

Descrevendo ações (II)

Axiomas sucessor-estado diminuem o problema do frame representacional

Cada axioma é "sobre" um predicado (não uma ação *per se*):

$$\text{P verdadeiro depois} \;\Leftrightarrow\; [\text{uma ação fez P verdadeiro} \;\vee\; \text{P já é verdadeiro e nenhuma ação fez P falso}]$$

Para segurar o ouro:

$$\forall\, a, s \;\; Holding(Gold, Result(a, s)) \;\Leftrightarrow$$
$$[(a = Grab \wedge AtGold(s))$$
$$\vee\; (Holding(Gold, s) \wedge a \neq Release)]$$

Um único axioma para cada predicado "modificável".

Deduzindo propriedades escondidas

Propriedades das localizações :
$\forall\, l,t\;\; At(Agent,l,t) \land Smelt(t) \;\Rightarrow\; Smelly(l)$
$\forall\, l,t\;\; At(Agent,l,t) \land Breeze(t) \;\Rightarrow\; Breezy(l)$

Localizações próximas a buracos têm brisa:

Regra de diagnóstico — deduza causas de efeitos

$$\forall\, y\;\; Breezy(y) \;\Rightarrow\; \exists\, x\;\; Pit(x) \land Adjacent(x,y)$$

Regra causal — deduza efeitos de causas

$$\forall\, x,y\;\; Pit(x) \land Adjacent(x,y) \;\Rightarrow\; Breezy(y)$$

Nenhum destes é completo — por exemplo., a regra causal não diz se posições distantes de buracos podem ter brisa

Definição para o predicado $Breezy$:

$$\forall\, y\;\; Breezy(y) \;\Leftrightarrow\; [\exists\, x\;\; Pit(x) \land Adjacent(x,y)]$$

Resumo

Lógica de primeiro-ordem:

– objetos e relações são primitivas semânticas
– sintaxe: constantes, funções, predicados, igualdade, quantificadores

Poder expressivo aumentado: suficiente para definir mundo "Wumpus"

Cálculo de situações:

– convenções para descrever ações e mudança em LPO
– pode formular planejando como conclusão em uma BC com cálculo de situações