INF623

2024/1

UFV

Inteligência Artificial

A11: Representação do conhecimento I

Plano de aula

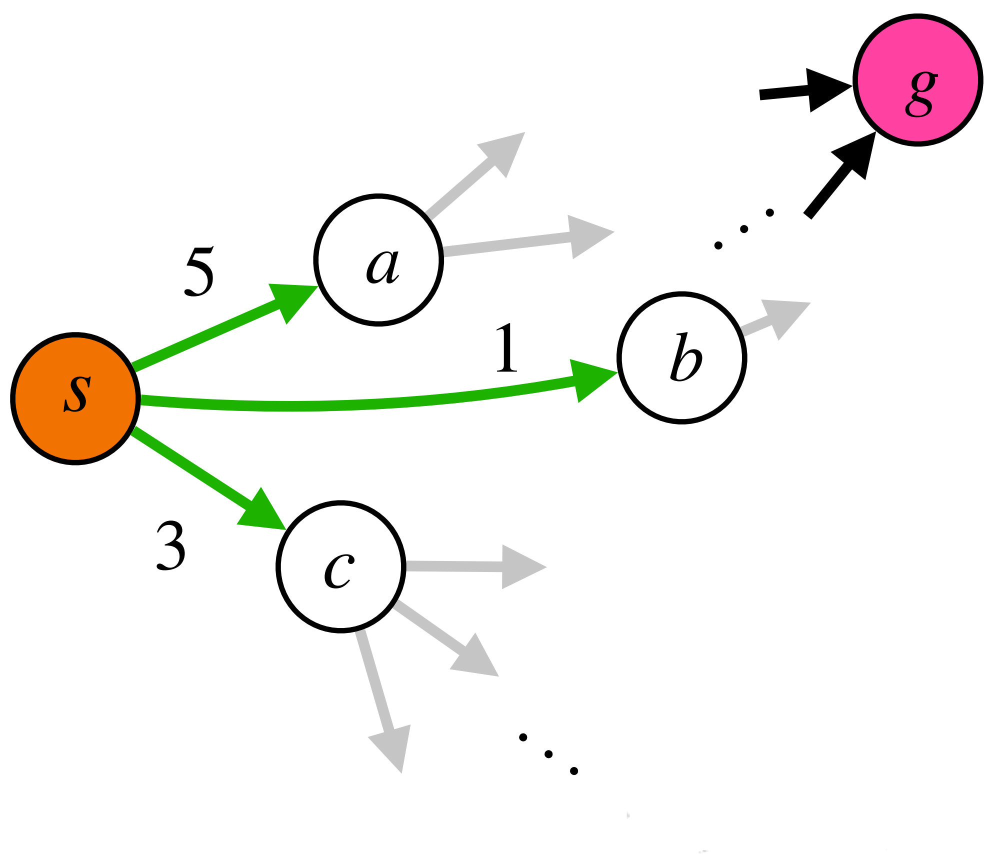
- Agentes baseados em conhecimento
- Lógica proposicional
  - Símbolos e conectivos lógicos
  - Modelo
  - Base de conhecimento
  - Consequência lógica
- Inferência
  - Verificação de modelos

UFV

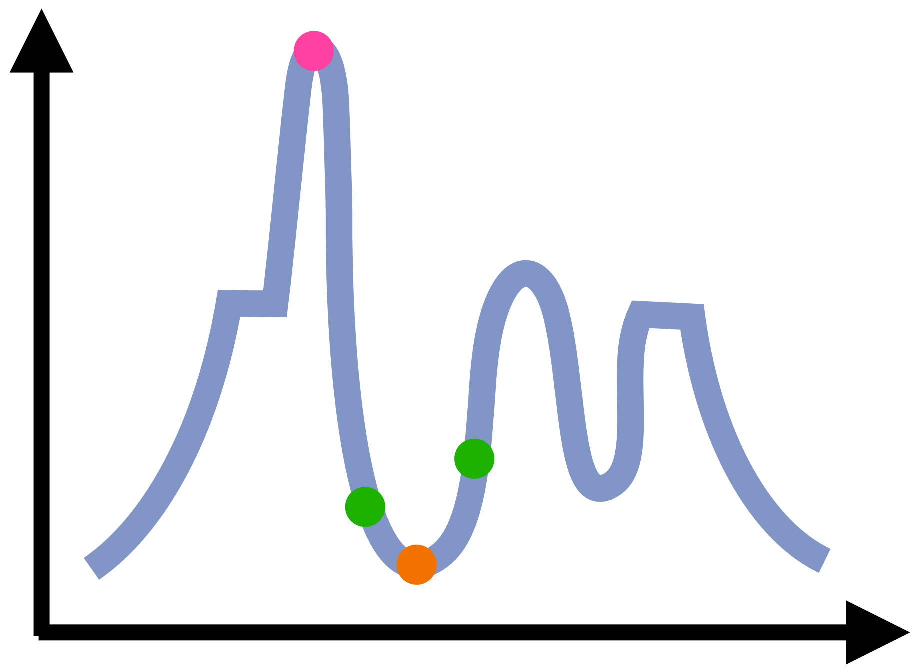
Resolução de problemas com busca

Os agentes baseados em busca tomam decisões considerando apenas as ações possíveis ou sua vizinhança.

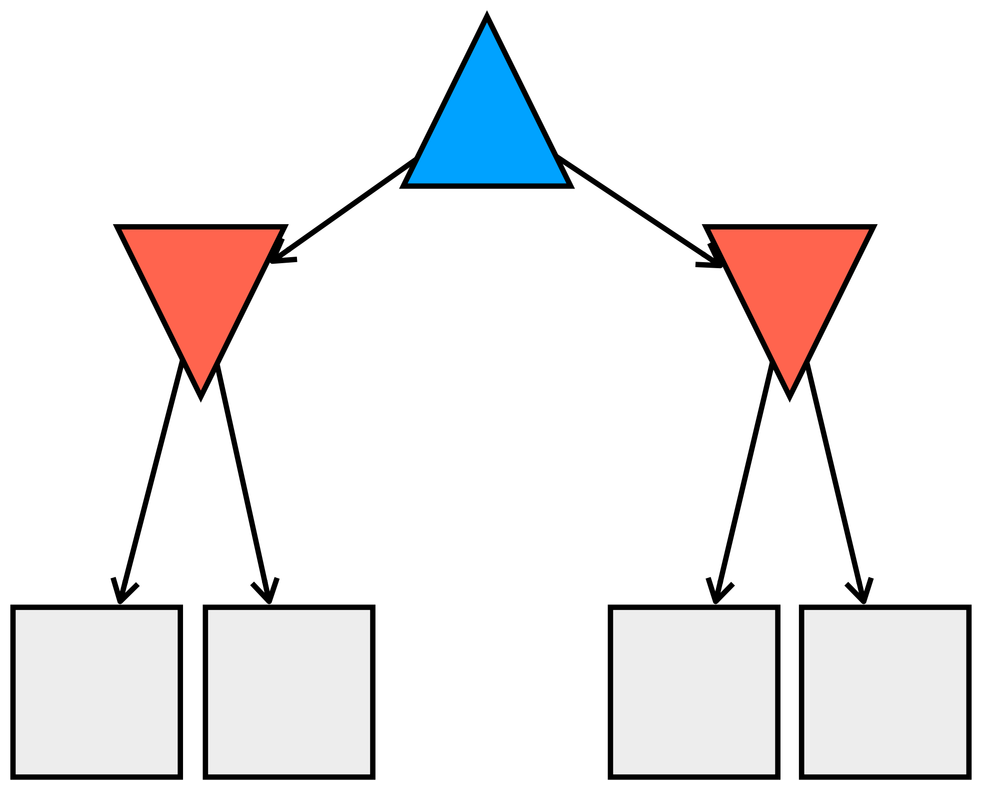
Busca no espaço de estados



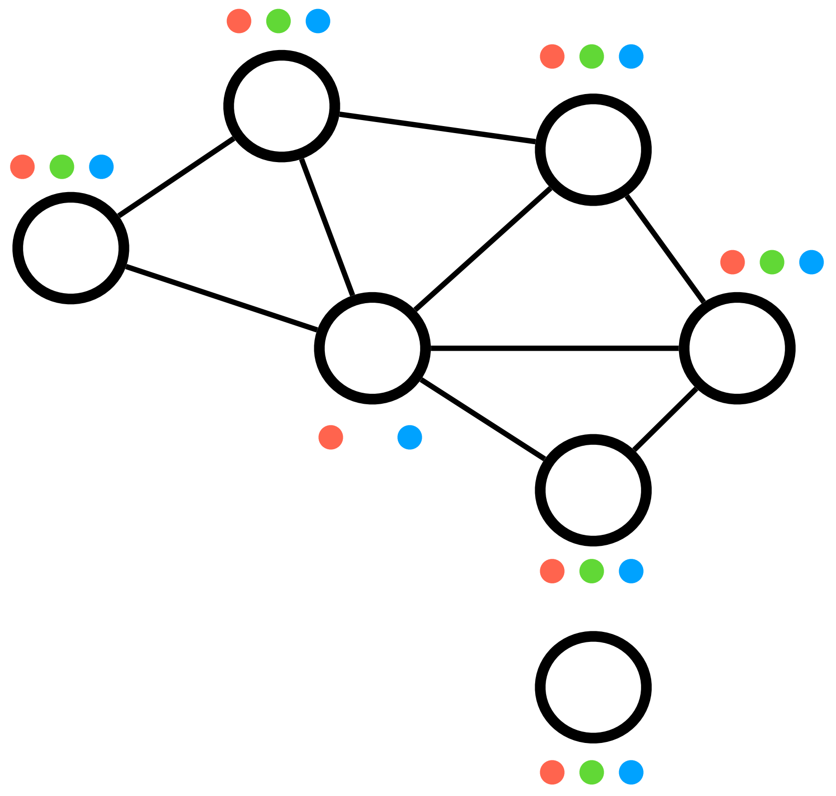
Busca local

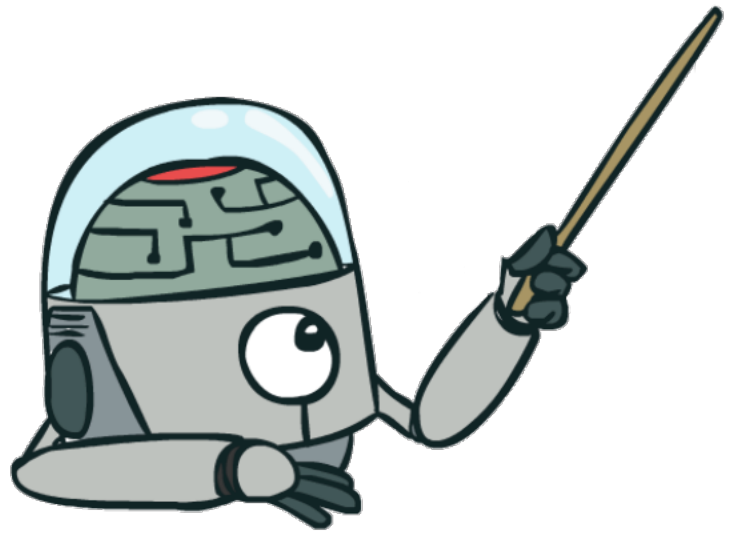


Busca competitiva



PSR





Eles não conhecem fatos sobre o mundo.

Exemplo: Está chovendo hoje

UFV

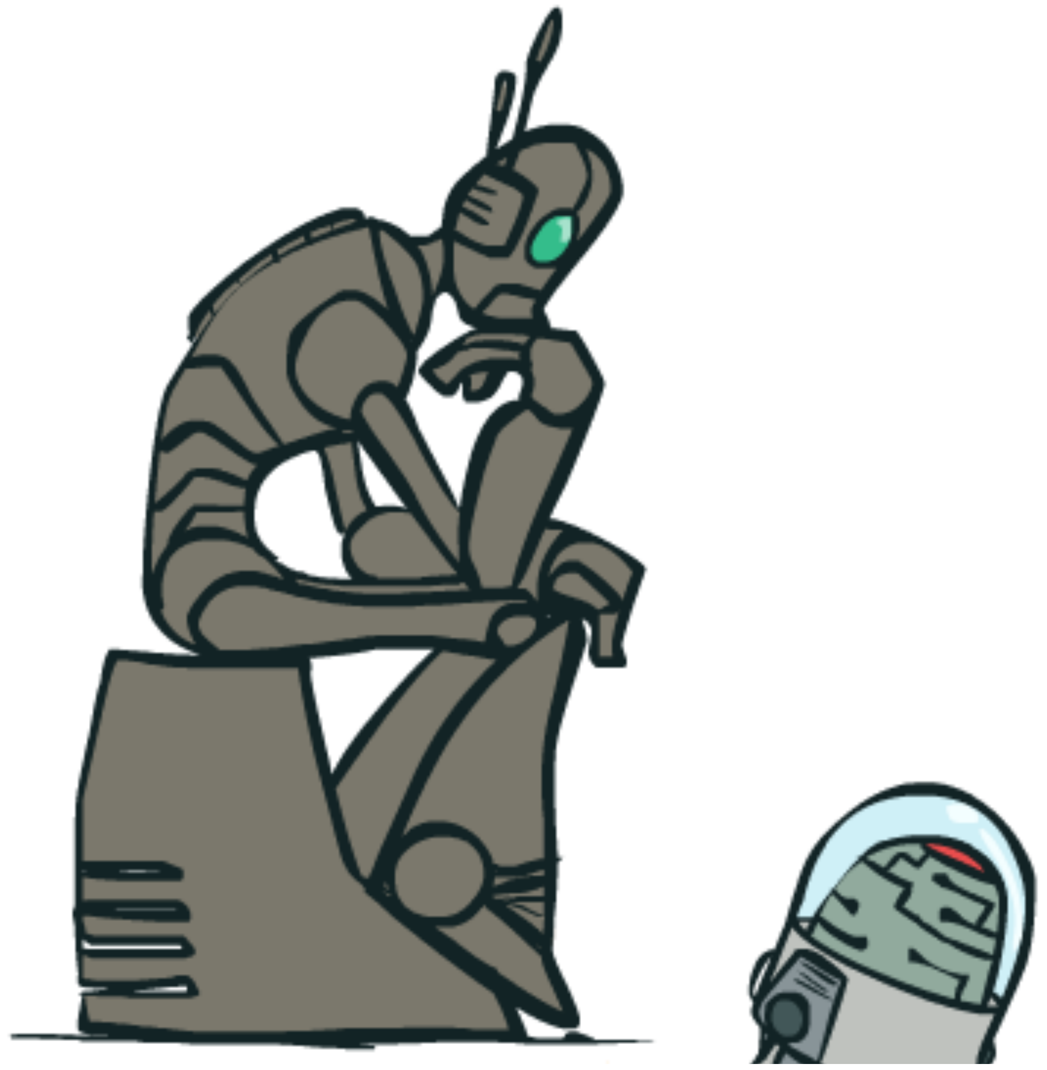
Exemplo 1: raciocínio lógico

- Considere as seguinte afirmações:
  - A: Se não houve greve, então Lucas foi à aula.
  - B: Lucas foi à aula ou para casa, mas não aos dois.
  - C: Lucas foi para casa.
- Houve greve? 🤔
  - C e B → Lucas não foi a aula.
  - A → Houve greve!

UFV

Agentes baseados em conhecimento

Agentes baseados em conhecimento representam o mundo por uma **base de conhecimento** e derivam conclusões a partir de **consultas** utilizando **inferência** lógica.



- **Base de conhecimento** é composta por **sentenças**;
- **Sentenças** são afirmações sobre o mundo representadas em uma linguagem de representação do conhecimento;
- **Consultas** são sentenças que queremos verificar se são verdadeiras ou não.

Linguagem Natural

Sentenças em linguagem natural podem ser pouco precisas para inferência lógica:

- Sentenças:
  - 1 real é melhor do que nada
  - Nada é melhor do que a paz mundial
- Conclusões
  - 1 real é melhor do que a paz mundial ???

Precisamos de uma linguagem formal para representação e inferência!

Lógica proposicional

Linguagem formal para representação de conhecimento em forma de sentenças proposicionais:

- **Proposições** são afirmação à qual se pode associar uma valor **Verdadeiro** ou **Falso**:

  "A UFV fica em Viçosa" - proposição Verdadeira

  "Viçosa fica no estado de São Paulo" - proposição Falsa

  Proposições são representadas por **símbolos proposicionais**, tradicionalmente escritos com letras do alfabeto latino (P, Q e R)

- **Sentenças** são formadas combinando proposições com **conectivos lógicos:**

  $\neg$ (não) , $\wedge$ (e) , $\vee$ (ou), $\Rightarrow$ (implica), $\Leftrightarrow$ (se e somente se)

UFV

Conectivos lógico ¬

P	$\neg P$
F	V
V	F

"Não P", exemplo:

- P: "Está chovendo"
- $\neg P$: "Não está chovendo"

UFV

Conectivos lógico $\wedge$

P	Q	$P \wedge Q$
F	F	F
F	V	F
V	F	F
V	V	V

"P e Q" , exemplo:

- P: "Lucas vai estudar"
- Q: "Lucas vai trabalhar"
- $P \wedge Q$: "Lucas vai estudar e trabalhar"

UFV

Conectivos lógico $\vee$

P	Q	$P \vee Q$
F	F	F
F	V	V
V	F	V
V	V	V

"P ou Q", exemplo:

- P: "Lucas vai estudar"
- Q: "Lucas vai trabalhar"
- $P \vee Q$: "Lucas vai estudar ou trabalhar"

Esse conectivo é chamado de OU inclusivo, mas existe também o OU exclusivo:

- OU inclusivo ($\vee$) – $\{P = V, Q = V\}$ é Verdadeiro
- OU exclusivo ($\oplus$) – $\{P = V, Q = V\}$ é Falso

UFV

Conectivos lógicos: Implicação $\Rightarrow$

P	Q	$P \Rightarrow Q$
F	F	V
F	V	V
V	F	F
V	V	V

"Se P, então Q", exemplo:

- P: "Está chovendo"
- Q: "Lucas vai estudar"
- $P \Rightarrow Q$:

"Se está chovendo, então Lucas vai estudar"

Se é Verdade que "*está chovendo*" e "*Lucas vai estudar*", então a afirmação "*Se está chovendo, então Lucas vai estudar*" é **Verdadeira**

Se é Verdade que "*está chovendo*", mas "Lucas não vai estudar", então a afirmação "Se está chovendo, então Lucas vai estudar" é **Falsa**

Se "não está chovendo", Lucas pode ir estudar ou não, portanto a afirmação "Se está chovendo, então Lucas vai estudar" é **Trivialmente Verdadeira**

Conectivos lógicos: bicondicional $\Leftrightarrow$

P	Q	$P \Leftrightarrow Q$
F	F	V
F	V	F
V	F	F
V	V	V

"Se e somente se P, então Q" , exemplo:

- P: "Está chovendo"
- Q: "Lucas vai estudar"
- $P \Leftrightarrow Q$

"Se e somente se estiver chovendo, Lucas vai estudar"

O "se e somente se" é uma implicação nas duas direções:

- $P \Leftrightarrow Q$ é igual a $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$

No exemplo acima, $P \Leftrightarrow Q$ significa que:

- Se "está chovendo", então "Lucas vai estudar";
- Se "Lucas vai estudar", então "está chovendo"

UFV

Exemplo 1: greve

- Se não houve greve, então Lucas foi à aula.
- Lucas foi à aula ou para casa, mas não aos dois.
- Lucas foi para casa.

- P: houve greve
- Q: Lucas foi à aula
- R: Lucas foi para casa

- $\neg P \Rightarrow Q$
- $(Q \wedge \neg R) \vee (\neg Q \wedge R)$ - ou exclusivo (XOR)
- R

Exercício

Formalize o seguinte argumento a seguir usando lógica proposicional

- Se o time joga bem, ganha o campeonato.
- Se o time não joga bem, o técnico é culpado.
- Se o time ganha o campeonato, os torcedores ficam contentes.
- Os torcedores não estão contentes.

Exercício

Formalize o seguinte argumento a seguir usando lógica proposicional

- Se o time joga bem, ganha o campeonato.
- Se o time não joga bem, o técnico é culpado.
- Se o time ganha o campeonato, os torcedores ficam contentes.
- Os torcedores não estão contentes.

- P: O time joga bem
- Q: O time ganha o campeonato
- R: O técnico é culpado
- S: Os torcedores ficam contentes

- $P \Rightarrow Q$
- $\neg P \Rightarrow R$
- $Q \Rightarrow S$
- $\neg S$

UFV

Modelo

Um **modelo** é uma atribuição de um valor-verdade a cada proposição.

- P: houve greve
- Q: Lucas foi à aula
- R: Lucas foi para casa

Um modelo pode ser pensado como uma configuração possível para o mundo.

$\{P = \text{Verdadeiro}, Q = \text{Falso}, R = \text{Verdadeiro}\}$

Modelo

Existem 2^N modelos possíveis para um mundo com N proposições. No exemplo abaixo, temos $2^3 = 8$ modelos pois o mundo contém $N = 3$ proposições.

P : Houve greve	Q : Lucas foi à aula	R: Lucas foi para casa
F	F	F
F	F	V
F	V	F
F	V	V
V	F	F
V	F	V
V	V	F
V	V	V

UFV

Base de conhecimento

Uma **base de conhecimento** (BC) é um conjunto de **sentenças** em lógica proposicional conhecidas por um agente:

Símbolos:

- P**:** Houve greve.
- Q**:** Lucas foi à aula.
- R**:** Lucas foi para casa.

Base de conhecimento (BC**):**

- (Se não houve greve, então Lucas foi à aula) – $\neg P \Rightarrow Q$
- (Lucas foi à aula ou para casa, mas não aos dois) – $(Q \land \neg R) \lor (\neg Q \land R)$
- (Lucas foi para casa) – R

Base de conhecimento

Adicionar uma sentença à **base de conhecimento restringe o número de modelos possíveis**:

BC:
(Vazia)

P: Houve greve Q: Lucas foi à aula R: Lucas foi para casa

F	F	F	V
F	F	V	V
F	V	F	V
F	V	V	V
V	F	F	V
V	F	V	V
V	V	F	V
V	V	V	V

UFV

Base de conhecimento

Adicionar uma sentença à **base de conhecimento restringe o número de modelos possíveis**:

BC:

- $\neg P \Rightarrow Q$

P: Houve greve Q: Lucas foi à aula R: Lucas foi para casa

F	F	F	F
F	F	V	F
F	V	F	V
F	V	V	V
V	F	F	V
V	F	V	V
V	V	F	V
V	V	V	V

UFV

Base de conhecimento

Adicionar uma sentença à **base de conhecimento restringe o número de modelos possíveis**:

BC:

- $\neg P \Rightarrow Q$
- $(Q \wedge \neg R) \vee (\neg Q \wedge R)$

P: Houve greve Q: Lucas foi à aula R: Lucas foi para casa

F	F	F	F
F	F	V	F
F	V	F	V
F	V	V	F
V	F	F	F
V	F	V	V
V	V	F	V
V	V	V	F

UFV

Base de conhecimento

Adicionar uma sentença à **base de conhecimento restringe o número de modelos possíveis**:

BC:

- $\neg P \Rightarrow Q$
- $(Q \wedge \neg R) \vee (\neg Q \wedge R)$
- R

P: Houve greve

Q: Lucas foi à aula

R: Lucas foi para casa

F	F	F	F
F	F	V	F
F	V	F	F
F	V	V	F
V	F	F	F
V	F	V	V
V	V	F	F
V	V	V	F

UFV

Consequência lógica

Dadas sentenças α e β, dizemos que β é uma **consequência lógica** de α se para todos os modelos em que α é Verdadeiro, β também é Verdadeiro.

$$\alpha \models \beta$$

Essa ideia que usaremos em nossos algoritmos de inferência!

A consequência lógica é diferente da implicação:

- A implicação ($\Rightarrow$) é um conectivo lógico entre duas proposições.
- A consequência lógica ($\models$) é uma relação entre conjuntos de modelos onde sentenças são verdadeiras – $M(\alpha) \subseteq M(\beta)$.
  - Se $\alpha \models \beta$, então α é uma afirmação *mais forte* que β: descarta *mais* mundos possíveis

Inferência

Inferência é o processo de derivar novas sentenças a partir de uma base de conhecimento. Em IA, o processo de inferência é iniciado por uma consulta:

Base de conhecimento (BC)**:**

- (Se não houve greve, então Lucas foi à aula) – $\neg P \Rightarrow Q$
- (Lucas foi à aula ou para casa, mas não aos dois) – $(Q \wedge \neg R) \vee (\neg Q \wedge R)$
- (Lucas foi para casa) – R

Consulta:

- Houve greve? – P é Verdadeiro?
- **Formalmente, em inferência, queremos saber se** $BC \models P$

UFV

Inferência por verificação de modelos

Dada uma base de conhecimento BC e uma consulta α, verificar se $BC \vDash \alpha$

Verificação de modelo:

- Para verificar se $BC \vDash \alpha$
  - Enumerar todos os modelos possíveis;
  - Se em todos modelos em que BC é Verdadeiro, α também é Verdadeiro, então $BC \vDash \alpha$
  - Caso contrário, α não é uma consequência lógica de BC

Inferência por verificação de modelos

Consulta: $\alpha = P$ (houve greve?)

- Enumerar todos os modelos possíveis;
- Se em todos modelos em que BC é V, α também é V, então $BC \vDash \alpha$

BC:

- $\neg P \Rightarrow Q$
- $(Q \wedge \neg R) \vee (\neg Q \wedge R)$
- R

P: Houve greve

Q: Lucas foi à aula

R: Lucas foi para casa

F	F	F	F
F	F	V	F
F	V	F	F
F	V	V	F
V	F	F	F
V	F	V	V
V	V	F	F
V	V	V	F

Nesse exemplo, só existe um **modelo** em que BC é Verdadeiro, e nesse modelo, P também é Verdadeiro, portanto $BC \vDash P$ e por isso "houve greve".

UFV

Inferência por verificação de modelos

```
def consequencia-logica(BC, alpha):
    simbolos = lista de símbolos proposicionais em BC e alpha
    return verificar-todos(BC, alpha, simbolos, {})

def verificar-todos(BC, alpha, simbolos, modelo):
    if len(simbolos) == 0: # Se o modelo tem um valor para cada símbolo
        if verdadeiro-em-modelo(BC, modelo): # BC é verdadeira no modelo?
            return verdadeiro-em-modelo(alpha, modelo) # Consulta é verdadeira no modelo?
        return True # Quando BC for falso, sempre retornar verdadeiro
    else:
        p = simbolos[:1]  # p é uma lista com o primeiro elemento de símbolos
        rest = simbolos[1:] # rest é uma lista com elementos restantes de símbolos
        return verificar-todos(BC, alpha, rest, dict(modelo, p=True)) and
               verificar-todos(BC, alpha, rest, dict(modelo, p=False))
```

Próxima aula

A12: Representação do conhecimento II

Engenharia de conhecimento, regras de inferência, prova de teoremas, forma normal conjuntiva, inferência por resolução, lógica de primeira ordem