

FUNDAMENTOS DE INTELIGÊNCIA ARTIFICIAL

Aula 8:

Knowledge - Conhecimento

Prof. Dr. Rodrigo Xavier de Almeida Leão
Cientista de Dados e Big Data



Knowledge-based agents

- Agentes que são capazes de raciocinar e agir representando o conhecimento internamente.
 1. Se não choveu, Mario surfou hoje.
 2. Mario surfou ou correu hoje, mas não ambos.
 3. Mario correu hoje.

Proposition Symbols

P

Q

R

Logical Connectives

\neg

not

\wedge

and

\vee

or

\rightarrow

implication

\leftrightarrow

biconditional

Not (\neg)

P	$\neg P$
false	true
true	false

Or (\vee)

P	Q	$P \vee Q$
false	false	false
false	true	true
true	false	true
true	true	true

Biconditional (\leftrightarrow)

P	Q	$P \leftrightarrow Q$
false	false	true
false	true	false
true	false	false
true	true	true

And (\wedge)

P	Q	$P \wedge Q$
false	false	false
false	true	false
true	false	false
true	true	true

Implication (\rightarrow)

P	Q	$P \rightarrow Q$
false	false	true
false	true	true
true	false	false
true	true	true

Modelo

- Inferir valores *booleanos* para cada símbolo proposicional

P: Está chovendo.

Q: É quarta-feira.

P = True

Q = False

Entailment

- $\text{Alpha} \models \text{Beta}$

Em qualquer modelo que Alpha seja verdadeiro então Beta também é verdadeiro.

Alpha = é terça feira de janeiro

Beta = é janeiro

Inferência

- Inferência é o processo de derivar novas sentenças de antigas.

Base de conhecimento (KB)

A base de conhecimento é um conjunto de sentenças conhecidas por um agente baseado em conhecimento.

Inferência

P: Hoje é terça

Q: Está chovendo

R: Mario vai surfar

KB: $(P \wedge \neg Q) \rightarrow R$ P $\neg Q$

Inferência: R

Algoritmo de Inferência

Busca a resposta para:

$KB \models \text{Alpha} \quad ?$

Model Checking

Para determinar se $KB \models \text{Alpha}$

- Verificar todos os modelos possíveis (estados possíveis)
- Se em todos os modelos em que KB é True, Alpha é True, então $KB \models \text{Alpha}$

KB: $(P \wedge \neg Q) \rightarrow R$ P $\neg Q$

Query: R

P	Q	R	KB
false	false	false	
false	false	true	
false	true	false	
false	true	true	
true	false	false	
true	false	true	
true	true	false	
true	true	true	

KB: $(P \wedge \neg Q) \rightarrow R$ P $\neg Q$

Query: R

P	Q	R	KB
false	false	false	false
false	false	true	false
false	true	false	false
false	true	true	false
true	false	false	false
true	false	true	true
true	true	false	false
true	true	true	false

check.py

```
from logic import *  
  
# Create new classes, each having a name, or a symbol, representing each proposition  
rain = Symbol("rain") # It is raining.  
hagrid = Symbol("hagrid") # Harry visited Hagrid
```

```
dumbledore = Symbol("dumbledore") # Harry visited Dumbledore
```

```
sentence = And(rain, hagrid)
```

```
print(sentence.formula())
```

check.py

```
from logic import *

# Create new classes, each having a name, or a symbol, representing each proposition
rain = Symbol("rain") # It is raining.
hagrid = Symbol("hagrid") # Harry visited Hagrid
```

```
dumbledore = Symbol("dumbledore") # Harry visited Dumbledore
```

```
knowledge = And(
    Implication(Not(rain), hagrid),
    Or(hagrid, dumbledore),
    Not(And(hagrid, dumbledore)),
    dumbledore
)
```

```
print(knowledge.formula())
```

```

from logic import *

# Create new classes, each having a name, or a symbol, representing each proposition
rain = Symbol("rain") # It is raining.
hagrid = Symbol("hagrid") # Harry visited Hagrid

dumbledore = Symbol("dumbledore") # Harry visited Dumbledore

# Save sentences into the KB
knowledge = And( # Starting from the "And" logical connective, because each proposition
    Implication(Not(rain), hagrid), #  $\neg(\text{It is raining}) \rightarrow (\text{Harry visited Hagrid})$ 
    Or(hagrid, dumbledore), #  $(\text{Harry visited Hagrid}) \vee (\text{Harry visited Dumbledore})$ .
    Not(And(hagrid, dumbledore)), #  $\neg(\text{Harry visited Hagrid} \wedge \text{Harry visited Dumbledore})$ 
    dumbledore # Harry visited Dumbledore. Note that while previous propositions c
)

```

```
knowledge = And(  
    Implication(Not(rain), hagrid),  
    Or(hagrid, dumbledore),  
    Not(And(hagrid, dumbledore)),  
    dumbledore  
)  
  
print(model_check(knowledge, rain))
```


Engenharia de Conhecimento

- Transformar conhecimento para uma forma que o computador entenda.
- Quais símbolos são necessários para solucionar o problema com base em dedução lógica.
- Como codificar o problema lógico.

Detetive

People

Col. Mustard

Prof. Plum

Ms. Scarlet

Rooms

Ballroom

Kitchen

Library

Weapons

Knife

Revolver

Wrench

People



Rooms



Weapons



Símbolos Proposicionais

- Variáveis que podem ser verdadeiras ou falsas.

People

Col. Mustard

Prof. Plum

Ms. Scarlet

Rooms

Ballroom

Kitchen

Library

Weapons

Knife

Revolver

Wrench

Clue

(mustard \vee plum \vee scarlet)

(ballroom \vee kitchen \vee library)

(knife \vee revolver \vee wrench)

Ter a carta

$\neg plum$

Palpite

$\neg mustard \vee \neg library \vee \neg revolver$

Logic Puzzle

- Gilderoy, Minerva, Pomona and Horace each belong to a different one of the four houses: Gryffindor, Hufflepuff, Ravenclaw, and Slytherin House.
- Gilderoy belongs to Gryffindor or Ravenclaw.
- Pomona does not belong in Slytherin.
- Minerva belongs to Gryffindor.

Premissa do Problema

Cada pessoa está associada a uma casa diferente.

Propositional Symbols

GilderoyGryffindor
GilderoyHufflepuff
GilderoyRavenclaw
GilderoySlytherin

PomonaGryffindor
PomonaHufflepuff
PomonaRavenclaw
PomonaSlytherin

MinervaGryffindor
MinervaHufflepuff
MinervaRavenclaw
MinervaSlytherin

HoraceGryffindor
HoraceHufflepuff
HoraceRavenclaw
HoraceSlytherin

Cada pessoa está apenas em uma casa

(PomonaSlytherin \rightarrow \neg PomonaHufflepuff)

Cada casa possui apenas uma pessoa

(MinervaRavenclaw \rightarrow \neg GilderoyRavenclaw)

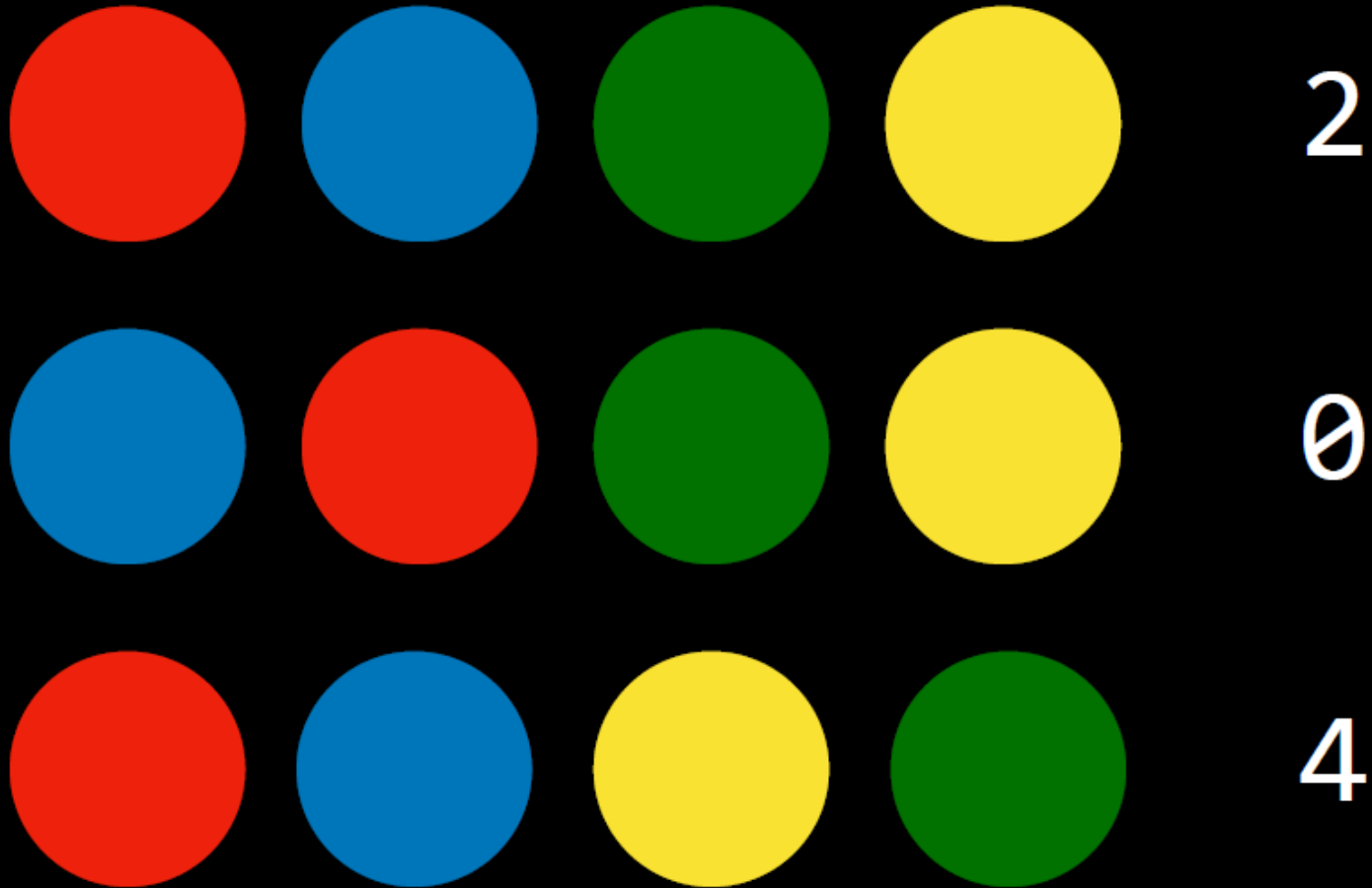
Informações do problema

(GilderoyGryffindor \vee GilderoyRavenclaw)

Logic Puzzle

- Entre Mario, Adão, José e Pedro existe um príncipe, um ladrão, um padre e um policial.
- Mario é príncipe ou padre.
- José não é policial.
- Adão é príncipe.

Mastermind



Model Check

2^n Modelos N = número de variáveis

Regras de Inferência

Modus Ponens

If it is raining, then Harry is inside.

It is raining.

Harry is inside.

Modus Ponens

$$\alpha \rightarrow \beta$$

$$\alpha$$

$$\beta$$

And Elimination

Harry is friends with Ron and Hermione.

Harry is friends with Hermione.

And Elimination

$$\alpha \wedge \beta$$

$$\alpha$$

Double Negation Elimination

It is not true that Harry did not pass the test.

Harry passed the test.

Double Negation Elimination

$$\neg(\neg\alpha)$$

$$\alpha$$

Implication Elimination

If it is raining, then Harry is inside.

It is not raining or Harry is inside.

Implication Elimination

$$\alpha \rightarrow \beta$$

$$\neg\alpha \vee \beta$$

Biconditional Elimination

It is raining if and only if Harry is inside.

If it is raining, then Harry is inside,
and if Harry is inside, then it is raining.

Biconditional Elimination

$$\alpha \leftrightarrow \beta$$

$$(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$$

De Morgan's Law

It is not true that both
Harry and Ron passed the test.

Harry did not pass the test
or Ron did not pass the test.

De Morgan's Law

$$\neg(\alpha \wedge \beta)$$

$$\neg\alpha \vee \neg\beta$$

De Morgan's Law

It is not true that
Harry or Ron passed the test.

Harry did not pass the test
and Ron did not pass the test.

De Morgan's Law

$$\neg(\alpha \vee \beta)$$

$$\neg\alpha \wedge \neg\beta$$

Distributive Property

$$(a \wedge (\beta \vee \gamma))$$

$$(a \wedge \beta) \vee (a \wedge \gamma)$$

Distributive Property

$$(a \vee (\beta \wedge \gamma))$$

$$(a \vee \beta) \wedge (a \vee \gamma)$$