

# Knights, Knaves, and Logical Reasoning

## Mechanising the Laws of Thought

Fabio Papacchini<sup>1</sup>

15 July 2015

---

<sup>1</sup>Special thanks to Francis Southern

# Introduction

Thinking



Formalising



Modelling



Computing

# Thinking

# A Puzzle

You are on a strange island where people are divided into

- Knights – always saying the truth
- Knaves – always saying lies

You meet two natives of the island Alice and Bob, and ask them

*“Are you knights or knaves?”*

Alice answers *“At least one of us is a knave”*

What are Alice and Bob?

Alice: *"At least one of us is a knave"*

Alice



Bob



Alice



Bob



Alice



Bob



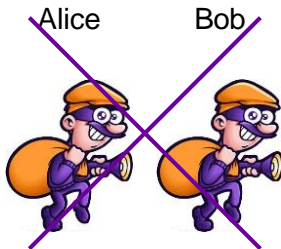
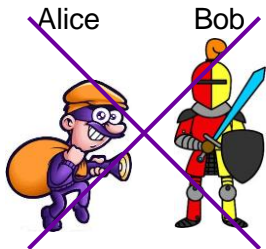
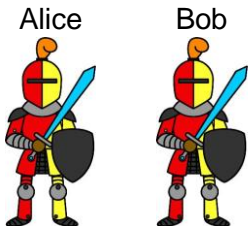
Alice



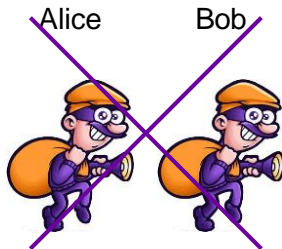
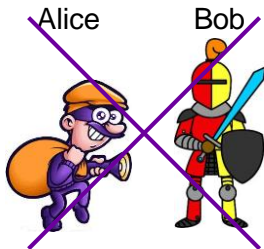
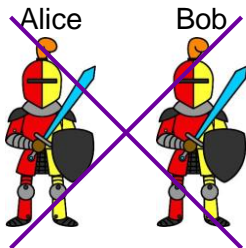
Bob



Alice: *"At least one of us is a knave"*



Alice: "At least one of us is a knave"



# Formalising



# Formalising Correct Reasoning

A: Socrates is a man

B: All men are mortal

C: All men are Socrates

C: Socrates is mortal

# Formalising Correct Reasoning

A: Socrates is a man

B: All men are mortal

C: All men are Socrates

C: Socrates is mortal

Woody Allen - Love and Death

Aristotle

# Formalising Correct Reasoning

A: Socrates is a man

B: All men are mortal

C: All men are Socrates

C: Socrates is mortal

Woody Allen - Love and Death

Aristotle

Linguistic, philosophical, or mathematical approaches to formalisation

Today: Propositional Logic

# Propositions

An expression which is either true or false.

# Propositions

An expression which is either true or false.

Proposition test: Is it true that...?

- $2 + 2 = 5$
- Manchester
- Grass is green
- We're in Manchester
- What's your name?
- It's raining

# Not – $\neg$ , And – &

## Not

$p$	$\neg p$
F	T
T	F

It's *not* raining

Grass is *not* green.

# Not – $\neg$ , And – $\&$

## Not

$p$	$\neg p$
F	T
T	F

It's *not* raining

Grass is *not* green.

## And

$p$	$q$	$p \& q$
F	F	F
F	T	F
T	F	F
T	T	T

Grass is green *and* it's raining.

We're in Manchester *and* we're in France.

## Or – $\vee$ , Implication (If, then) – $\rightarrow$

Or

$p$	$q$	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

Take an aspirin *or* lie down.

You can have milk *or* sugar in your tea.



## Or – $\vee$ , Implication (If, then) – $\rightarrow$

### Or

$p$	$q$	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

Take an aspirin *or* lie down.

You can have milk *or* sugar in your tea.

### Implication

$p$	$q$	$p \rightarrow q$
F	F	T
F	T	T
T	F	F
T	T	T

*If* you get 90% on this assignment, *then* you'll pass the course.

*If* you're late, *then* you'll give me a fiver.

# Biimplication (If and only if) – $\leftrightarrow$

## Biimplication

$p$	$q$	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

I'll buy you a new wallet *if (and only if)* you need one.

He studies *if (and only if)* he can.

## An Example: $(p \ \& \ q) \rightarrow r$

$p$	$q$	$r$	$(p \ \& \ q)$	$(p \ \& \ q) \rightarrow r$
F	F	F		
F	F	T		
F	T	F		
F	T	T		
T	F	F		
T	F	T		
T	T	F		
T	T	T		

## An Example: $(p \ \& \ q) \rightarrow r$

$p$	$q$	$r$	$(p \ \& \ q)$	$(p \ \& \ q) \rightarrow r$
F	F	F	F	
F	F	T	F	
F	T	F	F	
F	T	T	F	
T	F	F	F	
T	F	T	F	
T	T	F		
T	T	T		

## An Example: $(p \ \& \ q) \rightarrow r$

$p$	$q$	$r$	$(p \ \& \ q)$	$(p \ \& \ q) \rightarrow r$
F	F	F	F	
F	F	T	F	
F	T	F	F	
F	T	T	F	
T	F	F	F	
T	F	T	F	
T	T	F	T	
T	T	T	T	

## An Example: $(p \ \& \ q) \rightarrow r$

$p$	$q$	$r$	$(p \ \& \ q)$	$(p \ \& \ q) \rightarrow r$
F	F	F	F	T
F	F	T	F	T
F	T	F	F	T
F	T	T	F	T
T	F	F	F	T
T	F	T	F	T
T	T	F	T	
T	T	T	T	

## An Example: $(p \ \& \ q) \rightarrow r$

$p$	$q$	$r$	$(p \ \& \ q)$	$(p \ \& \ q) \rightarrow r$
F	F	F	F	T
F	F	T	F	T
F	T	F	F	T
F	T	T	F	T
T	F	F	F	T
T	F	T	F	T
T	T	F	T	F
T	T	T	T	

## An Example: $(p \ \& \ q) \rightarrow r$

$p$	$q$	$r$	$(p \ \& \ q)$	$(p \ \& \ q) \rightarrow r$
F	F	F	F	T
F	F	T	F	T
F	T	F	F	T
F	T	T	F	T
T	F	F	F	T
T	F	T	F	T
T	T	F	T	F
T	T	T	T	T



# Modelling

# The Trick

$k_A$  = Alice is a knight       $\neg k_A$  = Alice is a knave

“Alice says X” is the same as  $k_A \leftrightarrow X$

# The Trick

$k_A$  = Alice is a knight       $\neg k_A$  = Alice is a knave

“Alice says X” is the same as  $k_A \leftrightarrow X$

Alice says

- “at least one of us is a knave”
- “I’m a knave or Bob is a knave”
- $\neg k_A \mid \neg k_B$

$$\Rightarrow k_A \leftrightarrow (\neg k_A \mid \neg k_B)$$

# The Trick

$k_A$  = Alice is a knight       $\neg k_A$  = Alice is a knave

“Alice says X” is the same as  $k_A \leftrightarrow X$

Alice says

- “at least one of us is a knave”
- “I’m a knave or Bob is a knave”
- $\neg k_A \mid \neg k_B$

$$\Rightarrow k_A \leftrightarrow (\neg k_A \mid \neg k_B)$$

$k_A$	$k_B$	$\neg k_A$	$\neg k_B$	$\neg k_A \mid \neg k_B$	$k_A \leftrightarrow (\neg k_A \mid \neg k_B)$
F	F	T	T	T	
F	T	T	F	T	
T	F	F	T	T	
T	T	F	F	F	

# The Trick

$k_A$  = Alice is a knight       $\neg k_A$  = Alice is a knave

“Alice says X” is the same as  $k_A \leftrightarrow X$

Alice says

- “at least one of us is a knave”
- “I’m a knave or Bob is a knave”
- $\neg k_A \mid \neg k_B$

$$\Rightarrow k_A \leftrightarrow (\neg k_A \mid \neg k_B)$$

$k_A$	$k_B$	$\neg k_A$	$\neg k_B$	$\neg k_A \mid \neg k_B$	$k_A \leftrightarrow (\neg k_A \mid \neg k_B)$
F	F	T	T	T	F
F	T	T	F	T	
T	F	F	T	T	
T	T	F	F	F	

# The Trick

$k_A$  = Alice is a knight       $\neg k_A$  = Alice is a knave

“Alice says X” is the same as  $k_A \leftrightarrow X$

Alice says

- “at least one of us is a knave”
- “I’m a knave or Bob is a knave”
- $\neg k_A \mid \neg k_B$

$$\Rightarrow k_A \leftrightarrow (\neg k_A \mid \neg k_B)$$

$k_A$	$k_B$	$\neg k_A$	$\neg k_B$	$\neg k_A \mid \neg k_B$	$k_A \leftrightarrow (\neg k_A \mid \neg k_B)$
F	F	T	T	T	F
F	T	T	F	T	F
T	F	F	T	T	
T	T	F	F	F	

# The Trick

$k_A$  = Alice is a knight       $\neg k_A$  = Alice is a knave

“Alice says X” is the same as  $k_A \leftrightarrow X$

Alice says

- “at least one of us is a knave”
- “I’m a knave or Bob is a knave”
- $\neg k_A \mid \neg k_B$

$$\Rightarrow k_A \leftrightarrow (\neg k_A \mid \neg k_B)$$

$k_A$	$k_B$	$\neg k_A$	$\neg k_B$	$\neg k_A \mid \neg k_B$	$k_A \leftrightarrow (\neg k_A \mid \neg k_B)$
F	F	T	T	T	F
F	T	T	F	T	F
T	F	F	T	T	T
T	T	F	F	F	

# The Trick

$k_A$  = Alice is a knight       $\neg k_A$  = Alice is a knave

“Alice says X” is the same as  $k_A \leftrightarrow X$

Alice says

- “at least one of us is a knave”
- “I’m a knave or Bob is a knave”
- $\neg k_A \mid \neg k_B$

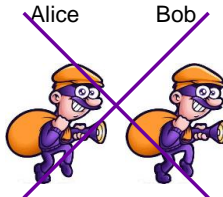
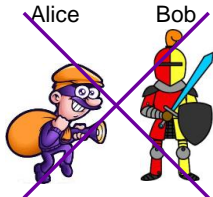
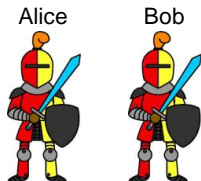
$$\Rightarrow k_A \leftrightarrow (\neg k_A \mid \neg k_B)$$

$k_A$	$k_B$	$\neg k_A$	$\neg k_B$	$\neg k_A \mid \neg k_B$	$k_A \leftrightarrow (\neg k_A \mid \neg k_B)$
F	F	T	T	T	F
F	T	T	F	T	F
T	F	F	T	T	T
T	T	F	F	F	F



# From Solving to Modelling

Alice: "At least one of us is a knave"



# From Solving to Modelling

Alice: “At least one of us is a knave”

$k_A$  = Alice is a knight

The trick: “Alice says X” is the same as  $k_A \leftrightarrow X$

“At least one of us is a knave” =  $\neg k_A \mid \neg k_B$

Alice says “At least one of us is a knave” =  $k_A \leftrightarrow (\neg k_A \mid \neg k_B)$



# From Solving to Modelling

Alice: *“At least one of us is a knave”*

$k_A$  = Alice is a knight

The trick: “Alice says X” is the same as  $k_A \leftrightarrow X$

“At least one of us is a knave” =  $\neg k_A \mid \neg k_B$

Alice says “At least one of us is a knave” =  $k_A \leftrightarrow (\neg k_A \mid \neg k_B)$

It can be (really) hard, but you only have to do it once!



# Modelling a Sudoku

			7			4	1	
		3		2				6
1		7	4			5	2	3
4		1	6				8	
	2	9		7		6	3	
	7				4	2		1
7	5	2			6	3		9
3				4		1		
	1	4			3			

What propositions do we need?

# Modelling a Sudoku

			7			4	1	
		3		2				6
1		7	4			5	2	3
4		1	6				8	
	2	9		7		6	3	
	7				4	2		1
7	5	2			6	3		9
3				4		1		
	1	4			3			

What propositions do we need?

Number  $n$  is in row  $i$  and column  $j$

- number 7 is in row 1 and column 4
- number 2 is in row 6 and column 7

$p_{1,1,1}, p_{1,1,2}, \dots, p_{9,9,8}, p_{9,9,9}$

729 propositions!

# Modelling a Sudoku

			7			4	1	
		3		2				6
1		7	4			5	2	3
4		1	6				8	
	2	9		7		6	3	
	7				4	2		1
7	5	2			6	3		9
3				4		1		
	1	4			3			

What propositions do we need?

Number  $n$  is in row  $i$  and column  $j$

- number 7 is in row 1 and column 4
- number 2 is in row 6 and column 7

$p_{1,1,1}, p_{1,1,2}, \dots, p_{9,9,8}, p_{9,9,9}$

729 propositions!

What do we have to model?

- at least one number per cell ( $p_{1,1,4} \vee \dots \vee p_{9,1,4}$ )
- at most one number per cell ( $p_{7,1,4} \rightarrow \neg p_{1,1,4}, p_{7,1,4} \rightarrow \neg p_{2,1,4}$ )

# Modelling a Sudoku

			7			4	1	
		3		2				6
1		7	4			5	2	3
4		1	6				8	
	2	9		7		6	3	
	7				4	2		1
7	5	2			6	3		9
3				4		1		
	1	4			3			

What propositions do we need?

Number  $n$  is in row  $i$  and column  $j$

- number 7 is in row 1 and column 4
- number 2 is in row 6 and column 7

$p_{1,1,1}, p_{1,1,2}, \dots, p_{9,9,8}, p_{9,9,9}$

729 propositions!

What do we have to model?

- at least one number per cell ( $p_{1,1,4} \vee \dots \vee p_{9,1,4}$ )
- at most one number per cell ( $p_{7,1,4} \rightarrow \neg p_{1,1,4}, p_{7,1,4} \rightarrow \neg p_{2,1,4}$ )
- no number can be repeated in a row

# Modelling a Sudoku

			7			4	1	
		3		2				6
1		7	4			5	2	3
4		1	6				8	
	2	9		7		6	3	
	7				4	2		1
7	5	2			6	3		9
3				4		1		
	1	4			3			

What propositions do we need?

Number  $n$  is in row  $i$  and column  $j$

- number 7 is in row 1 and column 4
- number 2 is in row 6 and column 7

$p_{1,1,1}, p_{1,1,2}, \dots, p_{9,9,8}, p_{9,9,9}$

729 propositions!

What do we have to model?

- at least one number per cell ( $p_{1,1,4} \vee \dots \vee p_{9,1,4}$ )
- at most one number per cell ( $p_{7,1,4} \rightarrow \neg p_{1,1,4}, p_{7,1,4} \rightarrow \neg p_{2,1,4}$ )
- no number can be repeated in a row/column



# Modelling a Sudoku

			7			4	1	
		3		2				6
1		7	4			5	2	3
4		1	6				8	
	2	9		7		6	3	
	7				4	2		1
7	5	2			6	3		9
3				4		1		
	1	4			3			

What propositions do we need?

Number  $n$  is in row  $i$  and column  $j$

- number 7 is in row 1 and column 4
- number 2 is in row 6 and column 7

$p_{1,1,1}, p_{1,1,2}, \dots, p_{9,9,8}, p_{9,9,9}$

729 propositions!

What do we have to model?

- at least one number per cell ( $p_{1,1,4} \vee \dots \vee p_{9,1,4}$ )
- at most one number per cell ( $p_{7,1,4} \rightarrow \neg p_{1,1,4}, p_{7,1,4} \rightarrow \neg p_{2,1,4}$ )
- no number can be repeated in a row/column/region

# Computing

# Automating the Process

## Truth table

- mechanical
- time consuming ( $2^n$  rows!)
- tedious

# Automating the Process

## Truth table

- mechanical
- time consuming ( $2^n$  rows!)
- tedious

Let a computer do it for you!

- ideal for mechanical tasks
- only needs an input formula
- more reliable than us
- much faster than us
- the output is easily customisable

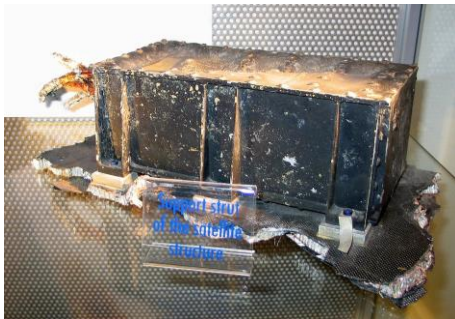
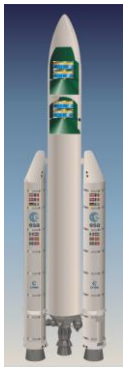
# Automated Reasoning

Much more than solving puzzles!

- software and hardware verification  
Intel and Microsoft
- information management  
biomedical ontologies, Semantic Web, databases
- combinatorial reasoning  
constraint satisfaction, planning, scheduling
- Internet security
- theorem proving in mathematics

# Where Could Have Been Used

Ariane 5 rocket failure due to a software bug, cost \$370 million.



# Where Has Been Used


To find and fix a bug in a widely used sorting algorithm!

## JDK / JDK-8072909 TimSort fails with ArrayIndexOutOfBoundsException on worst case long arrays

Agile Board More Actions ▾

Views ▾

Details

Type:  Bug

Priority:  P3

Affects Version/s: 7u76, 8, 9

Component/s: core-libs

Labels:  Bu60  jdk.port  regression  webbug

Subcomponent: java.util

Resolved in Build: b51

CPU: x86

OS: linux

Status:  Resolved


Resolution: Fixed

Fix Version/s: 9

People

Assignee: Lev Prima (Inactive)

Reporter: Webbug Group

 Vote (0)  Watch (4)

Dates

Created: 2015-02-06 08:22


Updated: 2015-06-09 15:26

Resolved: 2015-02-12 07:44

Agile

# Where Has Been Used

To find and fix a bug in a widely used sorting algorithm!

 JDK / JDK-8072909

## TimSort fails with ArrayIndexOutOfBoundsException on worst case long arrays

Agile Board More Actions ▾ Views ▾

### Details

Type:	Bug	Status:	Resolved
Priority:	P3	Resolution:	Fixed
Affects Version/s:	7u76, 8, 9	Fix Version/s:	9
Component/s:	core-libs		
Labels:	Bu60 jdk.port regression webbug		
Subcomponent:	java.util		
Resolved in Build:	b51		
CPU:	x86		
OS:	linux		

### People

Assignee:	Lev Prima (Inactive)
Reporter:	Webbug Group
Vote (0)	Watch (4)

### Dates

Created:	2015-02-06 08:22
Updated:	2015-06-09 15:26
Resolved:	2015-02-12 07:44

### Agile

Even Amazon and Facebook use automated reasoning techniques!



# Automated Reasoning Competitions

- The CADE ATP System Competition (CASC)
- OWL Reasoning Competition (ORE)
- SAT-Race



# Automated Reasoning Competitions

- The CADE ATP System Competition (CASC)
- OWL Reasoning Competition (ORE)
- SAT-Race



You can bet on the winner!

# Do You Want to Know More?

Look at the references on the handout!