Logica Proposizionale: Esercizi al Computer
Propositional Logic: Computer Exercises

Wolfgang Faber

University of Calabria, Italy

2007

1. Setup and Preparation
2. Simple Problems
3. Pigeon Hole Problem
4. Chess Piece Independence
5. Hardware Correctness

Setup – MiniSat

1. Boot Linux, open a browser
2. Go to `https://www.mat.unical.it/informatica/Ragionamento_Automatico`
3. Follow the link to MiniSat (`http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/MiniSat.html`)
4. Download `MiniSat_v1.14_linux`
5. Open a terminal window (ad esempio “Konsole”)
6. Add permission to execute: `chmod 755 MiniSat_v1.14_linux`
7. Try to execute `./MiniSat_v1.14_linux -h`

Setup – picosat

1. Go to `https://www.mat.unical.it/informatica/Ragionamento_Automatico`
2. Follow the link to PicoSat (`http://fmv.jku.at/picosat/`)
3. Download `http://fmv.jku.at/picosat/picosat-632.tar.gz`
4. Open a terminal window
5. Extract files (creates a directory `picosat-632`): `tar xvfz picosat-632.tar.gz`
6. Build PicoSat: `cd picosat-632; ./configure && make; cd ..`
7. Try to execute `picosat-632/picosat -h`

Simple Test

$$(x_1 \lor x_2 \lor \neg x_3) \land (\neg x_2) \land (x_4 \lor \neg x_3)$$

Example

```
c This is a CNF in DIMACS
c
p cnf 4 3
1 2 -3 0
-2 0
4 -3 0
```

Simple Test

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2) \wedge (x_4 \vee \neg x_3)$$

Example

```
c This is a CNF in DIMACS
c
p cnf 4 3
1 2 -3 0
-2 0
4 -3 0
```

More Tests

Example

$\Gamma = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2) \wedge (x_4 \vee \neg x_3)$

Example

$\Gamma = (x_1 \vee x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_2) \wedge (x_4 \vee \neg x_3)$

Example

$\{x_1 \vee x_2 \vee x_3, x_1 \vee x_2 \vee \neg x_3, x_1 \vee \neg x_2 \vee x_3, x_1 \vee \neg x_2 \vee \neg x_3, \neg x_1 \vee x_4, x_1 \vee \neg x_4 \vee \neg x_5 \vee x_6, \neg x_1 \vee x_7\}$

Pigeons and Holes

Pigeons in their Holes

Pigeon Hole Problem

Pigeon Hole Problem (PHP)

The problem is whether m pigeons can enter into n pigeon holes – $PHP(m, n)$.

Task

Find a family of formulas which are satisfiable when $PHP(m, n)$ is.

PHP - Modellation

Modellation

$m \times n$ propositional variables: $x_{i,j}$ where $i \leq m, j \leq n$
$x_{i,j}$ means that pigeon i is put into hole j.

The formula should express:

- Each pigeon is in some hole, and
- each pigeon is in at most one hole, and
- in each hole there is at most one pigeon.

Example: PHP(3,2)

$$\begin{aligned}
(x_{1,1} \vee x_{1,2}) \wedge (x_{2,1} \vee x_{2,2}) \wedge (x_{3,1} \vee x_{3,2})\wedge \\
(x_{1,1} \rightarrow (\neg x_{1,2})\wedge \\
(x_{2,1} \rightarrow (\neg x_{2,2})\wedge \\
(x_{3,1} \rightarrow (\neg x_{3,2})\wedge \\
(x_{1,2} \rightarrow (\neg x_{1,1})\wedge \\
(x_{2,2} \rightarrow (\neg x_{2,1})\wedge \\
(x_{3,2} \rightarrow (\neg x_{3,1}) \wedge \ldots
\end{aligned}$$

Example: PHP(3,2)

$$
\begin{aligned}
I(x_{1,1} \vee x_{1,2}) \wedge (x_{2,1} \vee x_{2,2}) \wedge (x_{3,1} \vee x_{3,2}) \wedge \\
(x_{1,1} \rightarrow (\neg x_{2,1} \wedge \neg x_{3,1}) \wedge \\
(x_{2,1} \rightarrow (\neg x_{1,1} \wedge \neg x_{3,1}) \wedge \\
(x_{3,1} \rightarrow (\neg x_{1,1} \wedge \neg x_{2,1}) \wedge \\
(x_{1,2} \rightarrow (\neg x_{2,2} \wedge \neg x_{3,2}) \wedge \\
(x_{2,2} \rightarrow (\neg x_{1,2} \wedge \neg x_{3,2}) \wedge \\
(x_{3,2} \rightarrow (\neg x_{1,2} \wedge \neg x_{2,2})
\end{aligned}
$$

PHP - Formula

$$\bigwedge_{i=1}^{m} \bigvee_{j=1}^{n} x_{i,j}$$

$$\wedge \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{n} (x_{i,j} \rightarrow \bigwedge_{\substack{k=1\\k\neq j}}^{n} \neg x_{i,k}) \wedge \qquad \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{n} (x_{i,j} \rightarrow \bigwedge_{\substack{k=1\\k\neq i}}^{m} \neg x_{k,j})$$

PHP - CNF Formula

$$\bigwedge_{i=1}^{m} \bigvee_{j=1}^{n} x_{i,j}$$

$$\wedge \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{n} \bigwedge_{\substack{k=1\\k\neq j}}^{n} (\neg x_{i,j} \vee \neg x_{i,k}) \wedge \qquad \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{n} \bigwedge_{\substack{k=1\\k\neq i}}^{m} (\neg x_{i,j} \vee \neg x_{k,j})$$

Example: PHP(3,2) in CNF

$$
\begin{aligned}
& (x_{1,1} \vee x_{1,2}) \wedge (x_{2,1} \vee x_{2,2}) \wedge (x_{3,1} \vee x_{3,2}) \\
& \wedge(\neg x_{1,1} \vee \neg x_{1,2})\wedge \\
& \wedge(\neg x_{2,1} \vee \neg x_{2,2})\wedge \\
& \wedge(\neg x_{3,1} \vee \neg x_{3,2})\wedge \\
& \wedge(\neg x_{1,2} \vee \neg x_{1,1})\wedge \\
& \wedge(\neg x_{2,2} \vee \neg x_{2,1})\wedge \\
& \wedge(\neg x_{3,2} \vee \neg x_{3,1}) \\
\wedge \ldots &
\end{aligned}
$$

Example: PHP(3,2) in CNF

$$
\begin{aligned}
&\dots\\
&\wedge(\neg x_{1,1} \vee \neg x_{2,1})\\
&\wedge(\neg x_{1,1} \vee \neg x_{3,1})\\
&\wedge(\neg x_{2,1} \vee \neg x_{1,1})\\
&\wedge(\neg x_{2,1} \vee \neg x_{3,1})\\
&\wedge(\neg x_{3,1} \vee \neg x_{1,1})\\
&\wedge(\neg x_{3,1} \vee \neg x_{2,1})\\
&\wedge(\neg x_{1,2} \vee \neg x_{2,2})\\
&\wedge(\neg x_{1,2} \vee \neg x_{3,2})\\
&\wedge(\neg x_{2,2} \vee \neg x_{1,2})\\
&\wedge(\neg x_{2,2} \vee \neg x_{3,2})
\end{aligned}
$$

PHP in Dimacs

We must convert the variables to positive integers.

$x_{1,1}$	1
$\dots$	
$x_{m,1}$	m
$x_{1,2}$	$m+1$
$\dots$	
$x_{m,2}$	$2 \times m$
$\dots$	
$x_{1,n}$	$(n-1) \times m + 1$
$\dots$	
$x_{m,n}$	$n \times m$

Therefore, $x_{i,j}$ will be represented by $(j-1) \times m + i$.

Example: PHP(3,2) in Dimacs

```
p cnf 6 15
1 4 0
2 5 0
3 6 0
-1 -4 0
-4 -1 0
-2 -5 0
-5 -2 0
-3 -6 0
-6 -3 0
-1 -2 0
-1 -3 0
-4 -5 0
-4 -6 0
-2 -1 0
-2 -3 0
-5 -4 0
-5 -6 0
-3 -1 0
-3 -2 0
-6 -4 0
-6 -5 0
```

PHP - Why?

Hard problem for SAT solvers

If $m > n$, $PHP(m, n)$ is unsatisfiable, and any system based on resolution has an exponential behavior.

Try it with the PHP(m,n) formula generator!

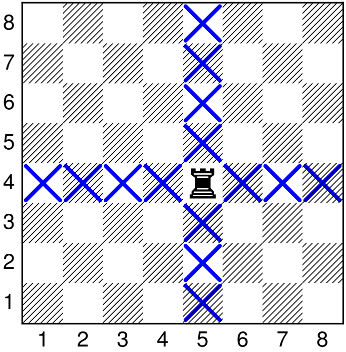
Rook Independence Problem

Rook Independence Problem – RIP(m,n)

Place m rooks on an $n \times n$ chessboard so that they do not threaten each other.

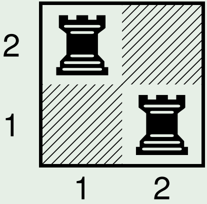
Rook Independence Problem

Reminder: Rook Moves



Rook Independence Problem

Example (RIP(2,2))



RIP - Modellation

Modellation

$m \times n \times n$ propositional variables: $x_{i,j,k}$ where
$i \leq n, j \leq n, k \leq m$
$x_{i,j,k}$ means that rook k is put onto field (i, j).

The formula should express:

- Each rook is placed on some field, and
- each rook is placed on at most one field, and
- each field holds at most one rook, and
- no rook threatens another rook.

Example: RIP(3,2)

$$
\begin{aligned}
&(x_{1,1,1} \vee x_{1,2,1} \vee x_{2,1,1} \vee x_{2,2,1}) \\
\wedge&(x_{1,1,2} \vee x_{1,2,2} \vee x_{2,1,2} \vee x_{2,2,2}) \\
\wedge&(x_{1,1,3} \vee x_{1,2,3} \vee x_{2,1,3} \vee x_{2,2,3}) \\
\wedge&\ldots
\end{aligned}
$$

Example: RIP(3,2) (ctd)

$$\begin{aligned}
\ldots \wedge & (x_{1,1,1} \rightarrow \neg x_{1,2,1} \wedge \neg x_{2,1,1} \wedge \neg x_{2,2,1}) \\
\wedge & (x_{1,2,1} \rightarrow \neg x_{1,1,1} \wedge \neg x_{2,1,1} \wedge \neg x_{2,2,1}) \\
\wedge & (x_{2,1,1} \rightarrow \neg x_{1,1,1} \wedge \neg x_{1,2,1} \wedge \neg x_{2,2,1}) \\
\wedge & (x_{2,2,1} \rightarrow \neg x_{1,1,1} \wedge \neg x_{1,2,1} \wedge \neg x_{2,1,1}) \\
\wedge & \ldots
\end{aligned}$$

Example: RIP(3,2) (ctd)

$$
\begin{aligned}
\ldots \wedge (x_{1,1,1} &\rightarrow \neg x_{1,1,2} \wedge \neg x_{1,1,3}) \\
\wedge (x_{1,1,2} &\rightarrow \neg x_{1,1,1} \wedge \neg x_{1,1,3}) \\
\wedge (x_{1,1,3} &\rightarrow \neg x_{1,1,1} \wedge \neg x_{1,1,2}) \\
\wedge (x_{1,2,1} &\rightarrow \neg x_{1,2,2} \wedge \neg x_{1,2,3}) \\
\wedge (x_{1,2,2} &\rightarrow \neg x_{1,2,2} \wedge \neg x_{1,2,3}) \\
\wedge (x_{1,2,3} &\rightarrow \neg x_{1,2,1} \wedge \neg x_{1,2,2}) \\
\wedge (x_{2,1,1} &\rightarrow \neg x_{2,1,2} \wedge \neg x_{2,1,3}) \\
\wedge (x_{2,1,2} &\rightarrow \neg x_{2,1,1} \wedge \neg x_{2,1,3}) \\
\wedge (x_{2,1,3} &\rightarrow \neg x_{2,1,1} \wedge \neg x_{2,1,2}) \\
\wedge (x_{2,2,1} &\rightarrow \neg x_{2,2,2} \wedge \neg x_{2,2,3}) \\
\wedge (x_{2,2,2} &\rightarrow \neg x_{2,2,1} \wedge \neg x_{2,2,3}) \\
\wedge (x_{2,2,3} &\rightarrow \neg x_{2,2,1} \wedge \neg x_{2,2,2}) \\
\wedge \ldots &
\end{aligned}
$$

Example: RIP(3,2) (ctd)

$$
\begin{aligned}
&\ldots \\
&\wedge(x_{1,1,1} \rightarrow \neg x_{1,2,2} \wedge \neg x_{1,2,3} \wedge \neg x_{2,1,2} \wedge \neg x_{2,1,3}) \\
&\wedge(x_{1,1,2} \rightarrow \neg x_{1,2,1} \wedge \neg x_{1,2,3} \wedge \neg x_{2,1,1} \wedge \neg x_{2,1,3}) \\
&\wedge(x_{1,1,3} \rightarrow \neg x_{1,2,1} \wedge \neg x_{1,2,2} \wedge \neg x_{2,1,1} \wedge \neg x_{2,1,2}) \\
&\wedge(x_{2,1,1} \rightarrow \neg x_{2,2,2} \wedge \neg x_{2,2,3} \wedge \neg x_{1,1,2} \wedge \neg x_{1,1,3}) \\
&\wedge \cdots \\
&\wedge(x_{1,2,1} \rightarrow \neg x_{1,1,2} \wedge \neg x_{1,1,3} \wedge \neg x_{2,2,2} \wedge \neg x_{2,2,3}) \\
&\wedge \cdots \\
&\wedge(x_{2,2,1} \rightarrow \neg x_{2,1,2} \wedge \neg x_{2,1,3} \wedge \neg x_{1,2,2} \wedge \neg x_{1,2,3})
\end{aligned}
$$

RIP(m,n) - Formula

$$\bigwedge_{k=1}^{m}\bigvee_{i=1}^{n}\bigvee_{j=1}^{n} x_{i,j,k} \wedge \bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n}\bigwedge_{k=1}^{m}(x_{i,j,k} \longrightarrow \bigwedge_{\substack{l=1\\h=1\\(l,h)\neq(i,j)}}^{\substack{n\\n}} \neg x_{l,h,k})\wedge$$

$$\wedge \bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n}\bigwedge_{k=1}^{m}(x_{i,j,k} \longrightarrow \bigwedge_{\substack{l=1\\l\neq k}}^{m} \neg x_{i,j,l})\wedge$$

$$\bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n}\bigwedge_{k=1}^{m}(x_{i,j,}$$

RIP - CNF Formula

Your turn!

- Figure out CNF.
- Use variable enumeration as in the following slide.
- Copy $pigeonhole - fixed.pl$ to $rip.pl$ and modify accordingly.
- Try it with MiniSat and PicoSat for some m, n.

RIP - Variable Enumeration

$x_{1,1,1}$	1
$\dots$	
$x_{n,1,1}$	n
$x_{1,2,1}$	$n+1$
$\dots$	
$x_{n,2,1}$	$2 \times n$
$\dots$	
$x_{1,n,1}$	$(n-1) \times n + 1$
$\dots$	
$x_{n,n,1}$	$n \times n$
$x_{1,1,2}$	$n \times n + 1$
$\dots$	
$x_{1,n,2}$	$n \times n + (n-1) \times n + 1$
$\dots$	
$x_{n,n,2}$	$2 \times n \times n$
$\dots$	
$x_{n,n,m}$	$m \times n \times n$

$$x_{i,j,k} \sim (k-1) \times n \times n + (j-1) \times n + i$$

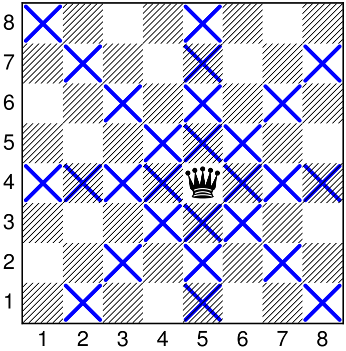
Queen Independence Problem

Queen Independence Problem – QIP(m,n)

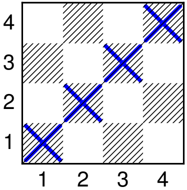
Place m queens on an $n \times n$ chessboard so that they do not threaten each other.

Queen Independence Problem

Reminder: Queen Moves

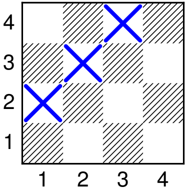


Queen Independence Problem: Diagonals



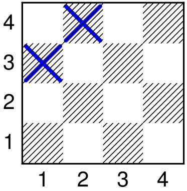
1,1
2,2
3,3
4,4

Queen Independence Problem: Diagonals



1,2	1,1
2,3	2,2
3,4	3,3
	4,4

Queen Independence Problem: Diagonals

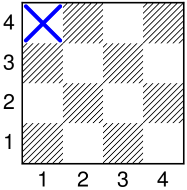


1,3
2,4

1,2
2,3
3,4

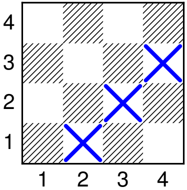
1,1
2,2
3,3
4,4

Queen Independence Problem: Diagonals



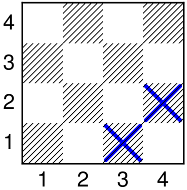
1,4	1,3	1,2	1,1
	2,4	2,3	2,2
		3,4	3,3
			4,4

Queen Independence Problem: Diagonals



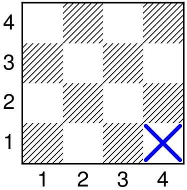
1,4	1,3	1,2	1,1	2,1
	2,4	2,3	2,2	3,2
		3,4	3,3	4,3
			4,4	

Queen Independence Problem: Diagonals



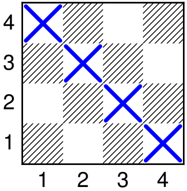
1,4	1,3	1,2	1,1	2,1	3,1
	2,4	2,3	2,2	3,2	4,2
		3,4	3,3	4,3	
			4,4		

Queen Independence Problem: Diagonals



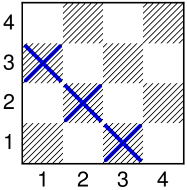
1,4	1,3	1,2	1,1	2,1	3,1	4,1
	2,4	2,3	2,2	3,2	4,2	
		3,4	3,3	4,3		
			4,4			

Queen Independence Problem: Diagonals



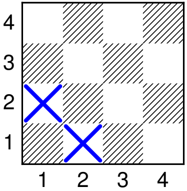
1,4
2,3
3,2
4,1

Queen Independence Problem: Diagonals



1,3	1,4
2,2	2,3
3,1	3,2
	4,1

Queen Independence Problem: Diagonals

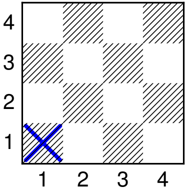


1,2
2,1

1,3
2,2
3,1

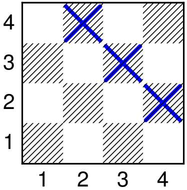
1,4
2,3
3,2
4,1

Queen Independence Problem: Diagonals



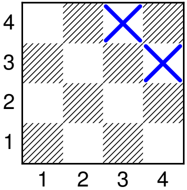
1,1	1,2	1,3	1,4
	2,1	2,2	2,3
		3,1	3,2
			4,1

Queen Independence Problem: Diagonals



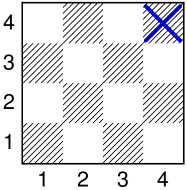
1,1	1,2	1,3	1,4	2,4
	2,1	2,2	2,3	3,3
		3,1	3,2	4,2
			4,1	

Queen Independence Problem: Diagonals



1,1	1,2	1,3	1,4	2,4	3,4
	2,1	2,2	2,3	3,3	4,3
		3,1	3,2	4,2	
			4,1		

Queen Independence Problem: Diagonals

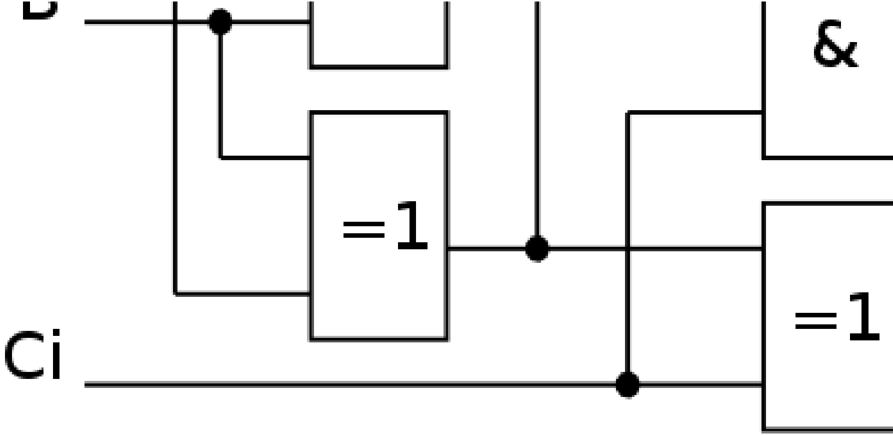


1,1	1,2	1,3	1,4	2,4	3,4	4,4
	2,1	2,2	2,3	3,3	4,3	
		3,1	3,2	4,2		
			4,1			

Homework

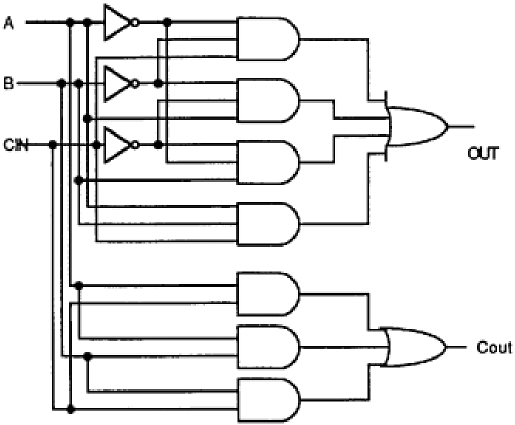
Solve these two latin square problems using a SAT solver!
`http://www.latinsquares.com/LSQ.1019T.pdf`

Full Adder



Standard Circuit

Full Adder



Alternative Circuit

Full Adder

Full Adder Equivalence Problem

Do these two circuits implement the same functionality?

Full Adder

Full Adder Equivalence Problem

Do these two circuits implement the same functionality?