Propositional Logic
Computer exercises

Mario Alviano

University of Calabria, Italy

A.Y. 2018/2019

Outline

SAT solvers input: DIMACS format

- Standard input format for SAT solvers
- Representation of CNF formulas stemming from a challenge run by DIMACS (Center for Discrete Mathematics and Theoretical Computer Science) in 1992

Example

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

DIMACS encoding

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

DIMACS format in BNF grammar

BNF grammar

<input> ::= <preamble> <formula> EOF

<preamble> ::= [<commentlines>] <problemline>
<commentlines> ::= <commentline> <commentlines> | <commentline>
<commentline> ::= c <text> EOL
<problemline> ::= p cnf <pnum> <pnum> EOL

<formula> ::= <clauselist>
<clauselist> ::= <clause> <clauselist> | <clause>
<clause> ::= <literal> <clause> | <literal> 0
<literal> ::= <num>

<text> ::= A sequence of non-special ASCII characters
<pnum> ::= A signed integer greater than 0
<num> ::= A signed integer different from 0

Setup

1. (I suggest to) Boot Linux
2. Download one of the solvers at
   `http://minisat.se/MiniSat.html`
   - MiniSat 2.2.0: only sources
   - MiniSat 1.14: sources and binary (if you are lazy)
3. Build and have a look at the help

```
$ minisat --help
```

Simple test (1)

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

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

Simple test (2)

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

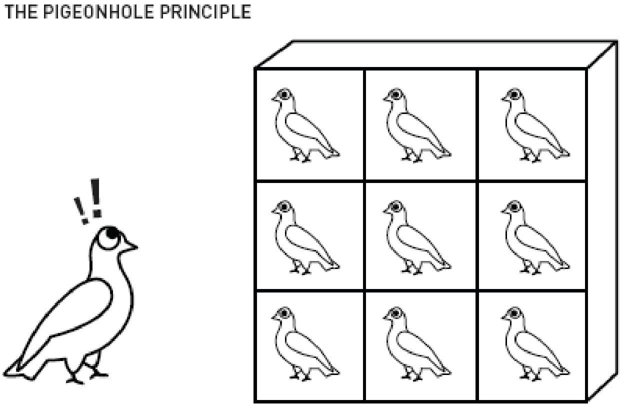
```
p cnf 2 4
 1  2  0
 1 -2  0
-1  2  0
-1 -2  0
```

More tests

Example

1. $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2) \wedge (x_4 \vee \neg x_3)$
2. $(x_1 \vee x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_2) \wedge (x_4 \vee \neg x_3)$
3. $\{x_1 \vee x_2 \vee x_3,\ x_1 \vee x_2 \vee \neg x_3,\ x_1 \vee \neg x_2 \vee x_3,\ x_1 \vee \neg x_2 \vee \neg x_3,$
   $\neg x_1 \vee x_4,\ x_1 \vee \neg x_4 \vee \neg x_5 \vee x_6,\ \neg x_1 \vee x_7\}$

Pigeons and Holes



Pigeons in their Holes

Pigeon Hole Problem

Pigeon Hole Problem — PHP

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

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 $i \leq m$ and $j \leq n$
- $x_{i,j}$ means that pigeon i is put into hole j

The formula should express:

1. Each pigeon is in some hole
2. Each pigeon is in at most one hole
3. In each hole there is at most one pigeon

Example: PHP(3,2)

1. Each pigeon is in some hole

   $x_{1,1} \vee x_{1,2},\ x_{2,1} \vee x_{2,2},\ x_{3,1} \vee x_{3,2}$

2. Each pigeon is in at most one hole

   $x_{1,1} \rightarrow \neg x_{1,2}$

   $x_{1,2} \rightarrow \neg x_{1,1}$

   $x_{2,1} \rightarrow \neg x_{2,2}$

   $x_{2,2} \rightarrow \neg x_{2,1}$

   $x_{3,1} \rightarrow \neg x_{3,2}$

   $x_{3,2} \rightarrow \neg x_{3,1}$

3. In each hole there is at most one pigeon

$$\begin{array}{l} x_{1,1} \rightarrow \neg x_{2,1} \wedge \neg x_{3,1} \\ x_{2,1} \rightarrow \neg x_{1,1} \wedge \neg x_{3,1} \\ x_{3,1} \rightarrow \neg x_{1,1} \wedge \neg x_{2,1} \\ x_{1,2} \rightarrow \neg x_{2,2} \wedge \neg x_{3,2} \\ x_{2,2} \rightarrow \neg x_{1,2} \wedge \neg x_{3,2} \\ x_{3,2} \rightarrow \neg x_{1,2} \wedge \neg x_{2,2} \end{array} \equiv \begin{array}{l} x_{1,1} \rightarrow \neg x_{2,1},\ x_{1,1} \rightarrow \neg x_{3,1} \\ x_{2,1} \rightarrow \neg x_{1,1},\ x_{2,1} \rightarrow \neg x_{3,1} \\ x_{3,1} \rightarrow \neg x_{1,1},\ x_{3,1} \rightarrow \neg x_{2,1} \\ x_{1,2} \rightarrow \neg x_{2,2},\ x_{1,2} \rightarrow \neg x_{3,2} \\ x_{2,2} \rightarrow \neg x_{1,2},\ x_{2,2} \rightarrow \neg x_{3,2} \\ x_{3,2} \rightarrow \neg x_{1,2},\ x_{3,2} \rightarrow \neg x_{2,2} \end{array}$$

PHP(m,n) — Formula

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

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

CNF Formula

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

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

Example: PHP(3,2) in CNF

1. Each pigeon is in some hole

   $x_{1,1} \vee x_{1,2},\ x_{2,1} \vee x_{2,2},\ x_{3,1} \vee x_{3,2}$

2. Each pigeon is in at most one hole

   $\neg x_{1,1} \vee \neg x_{1,2}$

   $\neg x_{1,2} \vee \neg x_{1,1}$

   $\neg x_{2,1} \vee \neg x_{2,2}$

   $\neg x_{2,2} \vee \neg x_{2,1}$

   $\neg x_{3,1} \vee \neg x_{3,2}$

   $\neg x_{3,2} \vee \neg x_{3,1}$

3. In each hole there is at most one pigeon

   $\neg x_{1,1} \vee \neg x_{2,1},\ \neg x_{1,1} \vee \neg x_{3,1}$

   $\neg x_{2,1} \vee \neg x_{1,1},\ \neg x_{2,1} \vee \neg x_{3,1}$

   $\neg x_{3,1} \vee \neg x_{1,1},\ \neg x_{3,1} \vee \neg x_{2,1}$

   $\neg x_{1,2} \vee \neg x_{2,2},\ \neg x_{1,2} \vee \neg x_{3,2}$

   $\neg x_{2,2} \vee \neg x_{1,2},\ \neg x_{2,2} \vee \neg x_{3,2}$

   $\neg x_{3,2} \vee \neg x_{1,2},\ \neg x_{3,2} \vee \neg x_{2,2}$

PHP in DIMACS

We must convert the variables to positive integers

$$
\begin{array}{rcc}
x_{1,1} & \mapsto & 1 \\
 & \vdots & \\
x_{m,1} & \mapsto & m \\
x_{1,2} & \mapsto & m+1 \\
 & \vdots & \\
x_{m,2} & \mapsto & 2 \times m \\
 & \vdots & \\
x_{1,n} & \mapsto & (n-1) \times m + 1 \\
 & \vdots & \\
x_{m,n} & \mapsto & n \times m
\end{array}
$$

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

Example: PHP(3,2) in DIMACS

1. $x_{1,1} \vee x_{1,2}$, $x_{2,1} \vee x_{2,2}$, $x_{3,1} \vee x_{3,2}$
2. $\neg x_{1,1} \vee \neg x_{1,2}$
   $\neg x_{1,2} \vee \neg x_{1,1}$
   $\neg x_{2,1} \vee \neg x_{2,2}$
   $\neg x_{2,2} \vee \neg x_{2,1}$
   $\neg x_{3,1} \vee \neg x_{3,2}$
   $\neg x_{3,2} \vee \neg x_{3,1}$
3. $\neg x_{1,1} \vee \neg x_{2,1}$, $\neg x_{1,1} \vee \neg x_{3,1}$
   $\neg x_{2,1} \vee \neg x_{1,1}$, $\neg x_{2,1} \vee \neg x_{3,1}$
   $\neg x_{3,1} \vee \neg x_{1,1}$, $\neg x_{3,1} \vee \neg x_{2,1}$
   $\neg x_{1,2} \vee \neg x_{2,2}$, $\neg x_{1,2} \vee \neg x_{3,2}$
   $\neg x_{2,2} \vee \neg x_{1,2}$, $\neg x_{2,2} \vee \neg x_{3,2}$
   $\neg x_{3,2} \vee \neg x_{1,2}$, $\neg x_{3,2} \vee \neg x_{2,2}$

```
p cnf 6 21
 1  4  0          -1 -2  0
 2  5  0          -1 -3  0
 3  6  0          -4 -5  0
-1 -4  0          -4 -6  0
-4 -1  0          -2 -1  0
-2 -5  0          -2 -3  0
-5 -2  0          -5 -4  0
-3 -6  0          -5 -6  0
-6 -3  0          -3 -1  0
-3 -2  0
-6 -4  0
-6 -5  0
```

PHP — Why?

Hard problem for SAT solvers

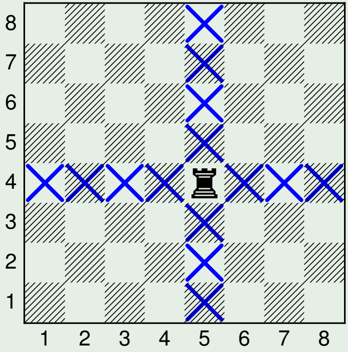
- $PHP(m, n)$ is unsatisfiable if and only if $m > n$
- SAT solvers have an exponential behavior
- Symmetries tend to mess up backtracking algorithms
- Try it with the PHP(m,n) formula generator!

Rook Independence Problem

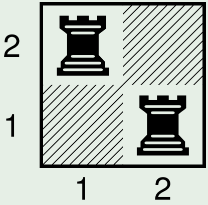
Rook Independence Problem — RIP(m,n)

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

Reminder: Rook moves



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:

1. Each rook is placed on some field
2. Each rook is placed on at most one field
3. Each field holds at most one rook
4. No rook threatens another rook

Example: RIP(3,2) (1)

1. Each rook is placed on some field

$x_{1,1,1} \vee x_{1,2,1} \vee x_{2,1,1} \vee x_{2,2,1}$

$x_{1,1,2} \vee x_{1,2,2} \vee x_{2,1,2} \vee x_{2,2,2}$

$x_{1,1,3} \vee x_{1,2,3} \vee x_{2,1,3} \vee x_{2,2,3}$

2. Each rook is placed on at most one field

$x_{1,1,1} \rightarrow \neg x_{1,2,1} \wedge \neg x_{2,1,1} \wedge \neg x_{2,2,1}$

$x_{1,2,1} \rightarrow \neg x_{1,1,1} \wedge \neg x_{2,1,1} \wedge \neg x_{2,2,1}$

$x_{2,1,1} \rightarrow \neg x_{1,1,1} \wedge \neg x_{1,2,1} \wedge \neg x_{2,2,1}$

$x_{2,2,1} \rightarrow \neg x_{1,1,1} \wedge \neg x_{1,2,1} \wedge \neg x_{2,1,1}$

$\vdots$

$x_{1,1,3} \rightarrow \neg x_{1,2,3} \wedge \neg x_{2,1,3} \wedge \neg x_{2,2,3}$

$x_{1,2,3} \rightarrow \neg x_{1,1,3} \wedge \neg x_{2,1,3} \wedge \neg x_{2,2,3}$

$x_{2,1,3} \rightarrow \neg x_{1,1,3} \wedge \neg x_{1,2,3} \wedge \neg x_{2,2,3}$

$x_{2,2,3} \rightarrow \neg x_{1,1,3} \wedge \neg x_{1,2,3} \wedge \neg x_{2,1,3}$

Example: RIP(3,2) (2)

3. Each field holds at most one rook

$x_{1,1,1} \rightarrow \neg x_{1,1,2} \wedge \neg x_{1,1,3}$
$x_{1,1,2} \rightarrow \neg x_{1,1,1} \wedge \neg x_{1,1,3}$
$x_{1,1,3} \rightarrow \neg x_{1,1,1} \wedge \neg x_{1,1,2}$

$x_{1,2,1} \rightarrow \neg x_{1,2,2} \wedge \neg x_{1,2,3}$
$x_{1,2,2} \rightarrow \neg x_{1,2,2} \wedge \neg x_{1,2,3}$
$x_{1,2,3} \rightarrow \neg x_{1,2,1} \wedge \neg x_{1,2,2}$

$x_{2,1,1} \rightarrow \neg x_{2,1,2} \wedge \neg x_{2,1,3}$
$x_{2,1,2} \rightarrow \neg x_{2,1,1} \wedge \neg x_{2,1,3}$
$x_{2,1,3} \rightarrow \neg x_{2,1,1} \wedge \neg x_{2,1,2}$

$x_{2,2,1} \rightarrow \neg x_{2,2,2} \wedge \neg x_{2,2,3}$
$x_{2,2,2} \rightarrow \neg x_{2,2,1} \wedge \neg x_{2,2,3}$
$x_{2,2,3} \rightarrow \neg x_{2,2,1} \wedge \neg x_{2,2,2}$

Example: RIP(3,2) (3)

4. No rook threatens another rook

$$x_{1,1,1} \rightarrow \neg x_{1,2,2} \wedge \neg x_{1,2,3} \wedge \neg x_{2,1,2} \wedge \neg x_{2,1,3}$$
$$x_{1,1,2} \rightarrow \neg x_{1,2,1} \wedge \neg x_{1,2,3} \wedge \neg x_{2,1,1} \wedge \neg x_{2,1,3}$$
$$x_{1,1,3} \rightarrow \neg x_{1,2,1} \wedge \neg x_{1,2,2} \wedge \neg x_{2,1,1} \wedge \neg x_{2,1,2}$$

$$x_{2,1,1} \rightarrow \neg x_{2,2,2} \wedge \neg x_{2,2,3} \wedge \neg x_{1,1,2} \wedge \neg x_{1,1,3}$$
$$\vdots$$
$$x_{1,2,1} \rightarrow \neg x_{1,1,2} \wedge \neg x_{1,1,3} \wedge \neg x_{2,2,2} \wedge \neg x_{2,2,3}$$
$$\vdots$$
$$x_{2,2,1} \rightarrow \neg x_{2,1,2} \wedge \neg x_{2,1,3} \wedge \neg x_{1,2,2} \wedge \neg x_{1,2,3}$$
$$\vdots$$

RIP(m,n) — Formula

1. $$\bigwedge_{k=1}^{m} \bigvee_{i=1}^{n} \bigvee_{j=1}^{n} x_{i,j,k}$$
2. $$\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \bigwedge_{k=1}^{m} \left(x_{i,j,k} \rightarrow \bigwedge_{\substack{l=1 \\ h=1 \\ (l,h)\neq(i,j)}}^{\substack{n \\ n}} \neg x_{l,h,k} \right)$$
3. $$\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \bigwedge_{k=1}^{m} \left(x_{i,j,k} \rightarrow \bigwedge_{\substack{l=1 \\ l\neq k}}^{m} \neg x_{i,j,l} \right)$$
4. $$\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \bigwedge_{k=1}^{m} \left(x_{i,j,k} \rightarrow \bigwedge_{\substack{l=1 \\ h=1 \\ (l,h)\neq(i,k)}}^{\substack{m \\ n}} \neg x_{l,j,h} \wedge \bigwedge_{\substack{l=1 \\ h=1 \\ (l,h)\neq(j,k)}}^{\substack{m \\ n}} \neg x_{i,l,h} \right)$$

RIP — CNF Formula

Your turn!

1. Figure out CNF
2. Use variable enumeration as in the following slide
   - Or trick with some hashmap, associative array, dictionary, ...
3. Copy `php.pl` to `rip.pl` and modify accordingly
   - Or start with `php.py` if you like pythonic solutions!
4. Try it with MiniSat for some m, n

RIP — Variable enumeration

$$\begin{array}{lcc}
x_{1,1,1} & \mapsto & 1 \\
& \vdots & \\
x_{n,1,1} & \mapsto & n \\
x_{1,2,1} & \mapsto & n+1 \\
& \vdots & \\
x_{n,2,1} & \mapsto & 2 \times n \\
& \vdots & \\
x_{1,n,1} & \mapsto & (n-1) \times n + 1 \\
& \vdots & \\
x_{n,n,1} & \mapsto & n \times n
\end{array}$$

$$\begin{array}{lcc}
x_{1,1,2} & \mapsto & n \times n + 1 \\
& \vdots & \\
x_{1,n,2} & \mapsto & n \times n + (n-1) \times n + 1 \\
& \vdots & \\
x_{n,n,2} & \mapsto & 2 \times n \times n \\
& \vdots & \\
x_{n,n,m} & \mapsto & m \times n \times n
\end{array}$$

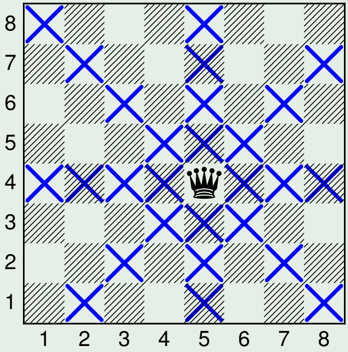
$$x_{i,j,k} \mapsto (k-1) \times n \times n + (j-1) \times n + i$$

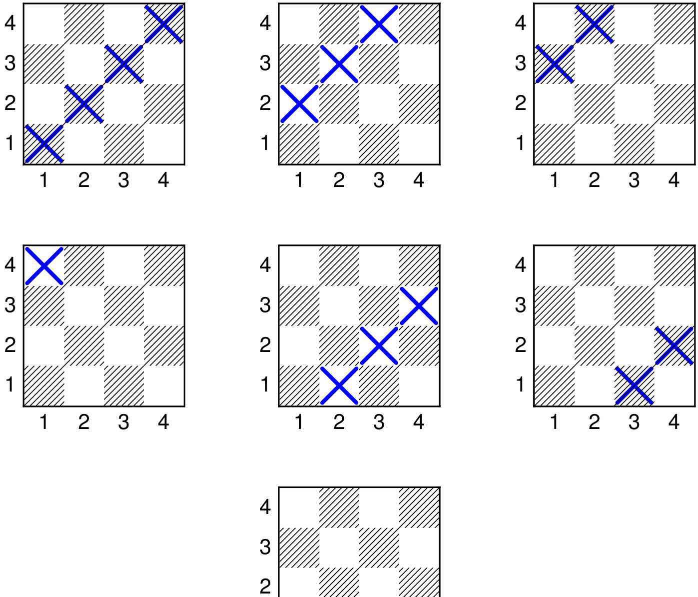
Queen Independence Problem

Queen Independence Problem — QIP(m,n)

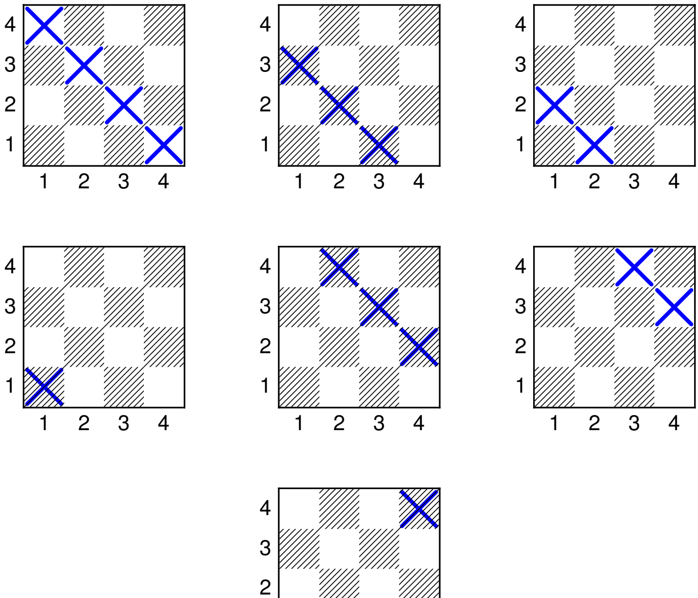
Place m queens on an $n \times n$ chessboard so that they do not threaten each other

Reminder: Queen moves





Queen Independence Problem: Diagonals (2)



Homework (1)

1. Write a CNF formula generator for the Bishop Independence Problem
2. Write a CNF formula generator for the King Independence Problem

Homework (2)

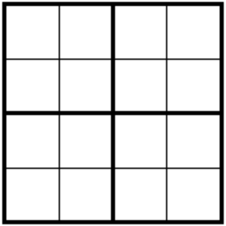
Latin Square

`http://mathworld.wolfram.com/LatinSquare.html`

1. Model the Latin Square problem
   - Start with a few examples
   - Then, work out a general formula
   - Finally, transform the formula in CNF
2. Write a CNF generator for Latin Square
3. Test your CNF with a SAT solver!

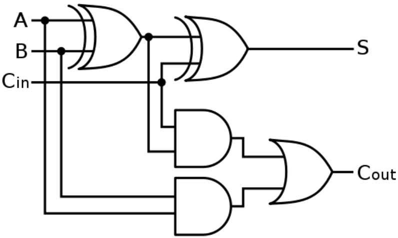
Homework (3)

1. Represent a 4×4 Sudoku with 2×2 squares using a propositional logic formula. This means that there is a 4×4 grid of fields, each of which should be filled with exactly one number between 1 and 4. The grid is divided into 4 non-overlapping regions of dimension 2×2.

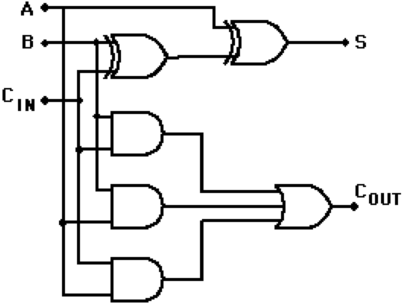


2. Generalize the formula for grids of $n^2 \times n^2$ fields, into each of which a number between 1 and n^2 should be written. The grid is divided into n^2 non-overlapping square regions of dimension $n \times n$. Here n is an arbitrary positive integer (so the previous example was a special case for $n = 2$).

Full Adder



Standard circuit



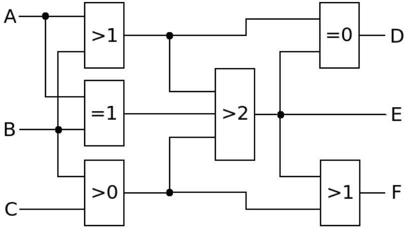
Alternative circuit

Full Adder Equivalence Problem

Do these two circuits implement the same functionality?

Homework

Represent the following Boolean circuit using a set of wffs:

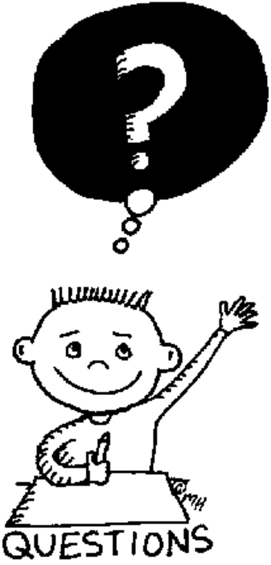


Each component of the circuit has inputs at its left-hand side and output at its right-hand side. The output of a component is 1 if and only if the sum of its inputs satisfies the condition written inside the component; so if the sum of the inputs does not satisfy the condition, the output is 0. The inputs and outputs of the circuit (A, B, C, D, E, F) may have values 1 or 0. Bifurcations are indicated using a dot; crossing wires without a dot.

The modelled formula must have models that correspond exactly to the admissible input and output values of the circuit.

Homework (2)

1. In the previous exercise, how can you find out whether the value of F can ever be 1 in an admissible state of the circuit?
2. Whether the value of E can ever be equal to the value of A?
3. Whether the value of F is 1 if and only if the value of D is 0?



END OF THE LECTURE