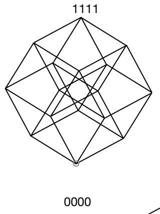
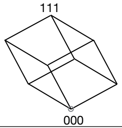
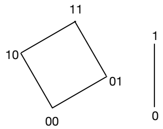
order !



inf1a-cl
Michael Fourman





Ordering

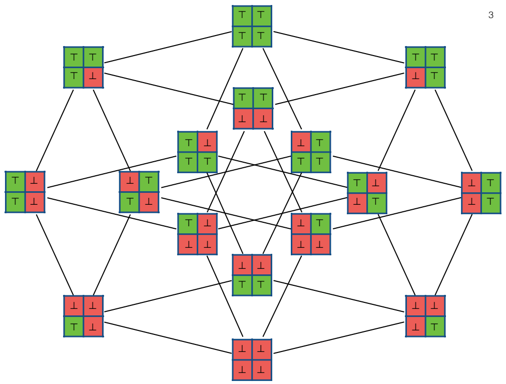
A→B	⊥	⊤
⊥	⊤	⊤
⊤	⊥	⊤

for 0-1 truth values,
A→B = ⊤ iff
A ≤ B

if A→B = ⊤ then
{ x | A} ⊆ { x | B}

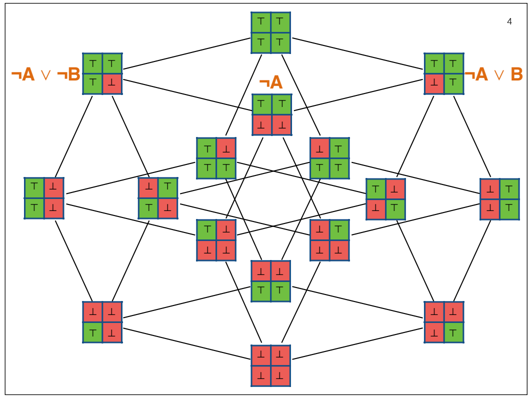
In any Boolean algebra, we define

A ≤ B iff A→B = ⊤ iff A∧B = A iff A∨B = B



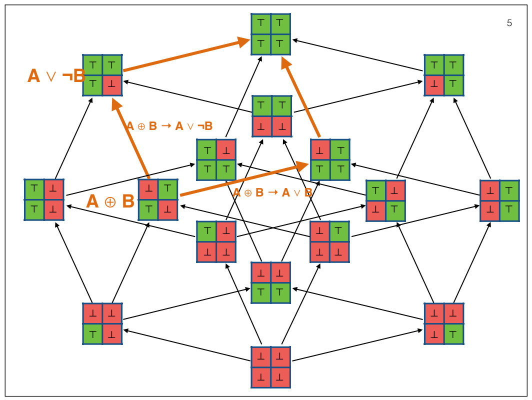
This diagram shows the truth tables for the 16 possible boolean functions of two variables.
We can also view it as a diagram of the subsets of a situation with four individuals, each representative of one of the four possible combinations of two boolean properties A and B. Each boolean function corresponds to a property P, and the diagram shows { x | P(x)

}.



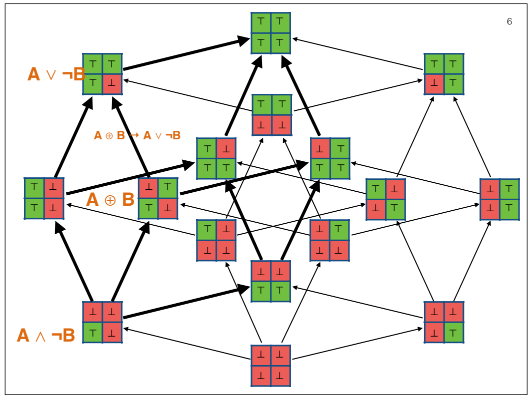
This diagram shows the truth tables for the 16 possible boolean functions of two variables.

We can also view it as a diagram of the subsets of a situation with four representative individuals for the four possible combinations of two boolean properties A and B. Each boolean function corresponds to a property P, and the diagram shows { x | P(x) }.



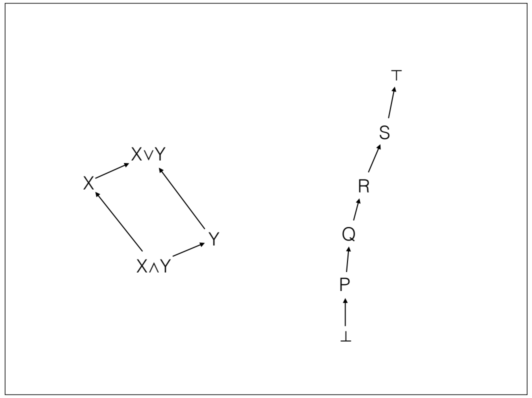
Each line in the diagram represents the addition of an additional element to the set.

Each arrow represents a valid implication



Each line in the diagram represents the addition of an additional element to the set.

Each arrow represents a valid implication



Ordering

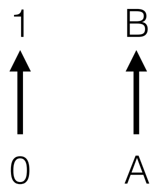
A→B	⊥	⊤
⊥	⊤	⊤
⊤	⊥	⊤

for 0-1 truth values,
A→B = ⊤ iff
A ≤ B

if A→B = ⊤ then
{ x | A} ⊆ { x | B}

In any Boolean algebra, we define

$A \le B$ iff $A \to B = \top$ iff $A \wedge B = A$ iff $A \vee B = B$



$0 \leq 1$

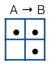
$\bot \leq \top$

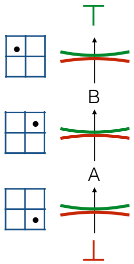
for booleans

$A \rightarrow B = \top$

iff

$A \leq B$





Suppose $A \rightarrow B$
there are three possible
truth valuations for A and B
(we exclude only ($A = \top$, $B = \bot$))

Propositions are ordered
by $x \leq y$ iff $x \rightarrow y = \top$
Any valid truth assignment
must draw a line
between $\bot$ and $\top$

Binary constraints

You may not take both Archeology and Chemistry
If you take Biology you must take Chemistry
You must take Biology or Archeology
If you take Chemistry you must take Divinity
You may not take both Divinity and Biology

$(\neg A \vee \neg C) \wedge (\neg B \vee C) \wedge (B \vee A) \wedge (\neg C \vee D) \wedge (\neg D \vee \neg B)$

$(A \rightarrow \neg C) \wedge (B \rightarrow C) \wedge (\neg B \rightarrow A) \wedge (C \rightarrow D) \wedge (D \rightarrow \neg B)$

each binary constraint is equivalent to an arrow

$\neg R \vee \neg A$	$\equiv$	$R \to \neg A$
$A \vee \neg G$	$\equiv$	$\neg A \to \neg G$
$R \vee A$	$\equiv$	$\neg R \to A$
$\neg R \vee B$	$\equiv$	$R \to B$

each binary constraint is equivalent to two arrows

$\neg R \vee \neg A$	$\equiv$	$R \to \neg A$	$\equiv$	$A \to \neg R$
$A \vee \neg G$	$\equiv$	$\neg A \to \neg G$	$\equiv$	$G \to A$
$R \vee A$	$\equiv$	$\neg R \to A$	$\equiv$	$\neg A \to R$
$\neg R \vee B$	$\equiv$	$R \to B$	$\equiv$	$\neg B \to \neg R$

each binary constraint is equivalent to two arrows

$\neg R \vee \neg A$	$\equiv$	$R \to \neg A$	$\equiv$	$A \to \neg R$	$\equiv$	$\neg A \vee \neg R$
$A \vee \neg G$	$\equiv$	$\neg A \to \neg G$	$\equiv$	$G \to A$	$\equiv$	$\neg G \vee A$
$R \vee A$	$\equiv$	$\neg R \to A$	$\equiv$	$\neg A \to R$	$\equiv$	$A \vee R$
$\neg R \vee B$	$\equiv$	$R \to B$	$\equiv$	$\neg B \to \neg R$	$\equiv$	$B \vee \neg R$

$\to$	0	1
0	1	1
1	0	1

$$A \to B = \top$$

iff

$$A \leq B$$

$$A \to B = \top$$

iff

$$A \leq B$$

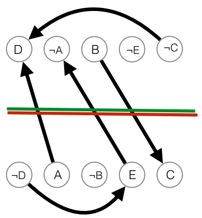
$$\begin{array}{cccc}
\top & \top & \top & \top \\
\uparrow & \uparrow & \uparrow & \uparrow \\
B & \top & \top & \bot \\
\uparrow & \uparrow & \uparrow & \uparrow \\
A & \top & \bot & \bot \\
\uparrow & \uparrow & \uparrow & \uparrow \\
\bot & \bot & \bot & \bot
\end{array}$$

A **valuation,** or **state,** makes some atoms true and the rest false. Once we have a valuation, for each atom, we can compute the truth value of every expression. If an atom is true its negation is false, and vice versa.

We draw a line to visualise a valuation, placing the true literals above the line, and the false literals below it.

Every binary constraint

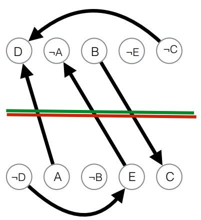
We draw a line to visualise a valuation, placing the true literals above the line, and the false literals below it.



An implication between literals is represented by an arrow.

Every binary constraint

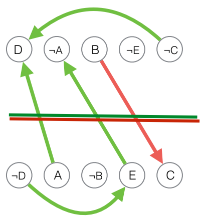
We draw a line to visualise a valuation, placing the true literals above the line, and the false literals below it.



An implication between literals is represented by an arrow.

Every binary constraint

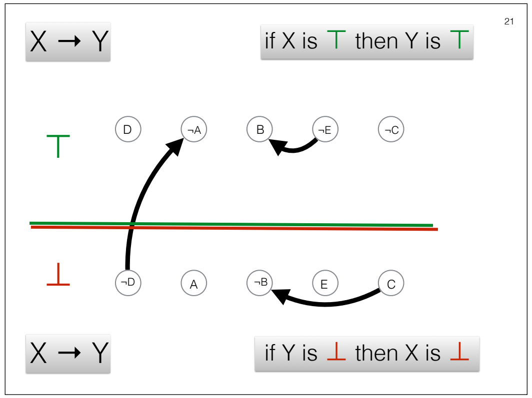
We draw a line to visualise a valuation, placing the true literals above the line, and the false literals below it.



An implication between literals is represented by an arrow.

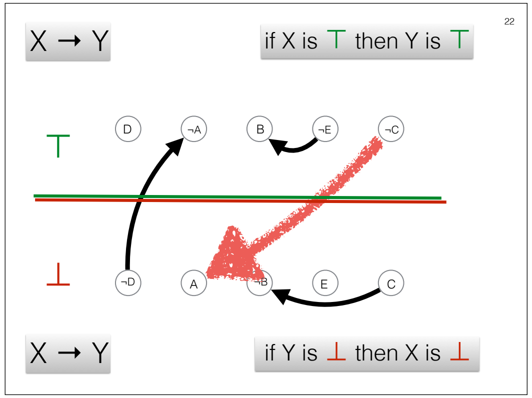
The valuation makes the implication true, unless the arrow goes from true to false.

Every binary constraint



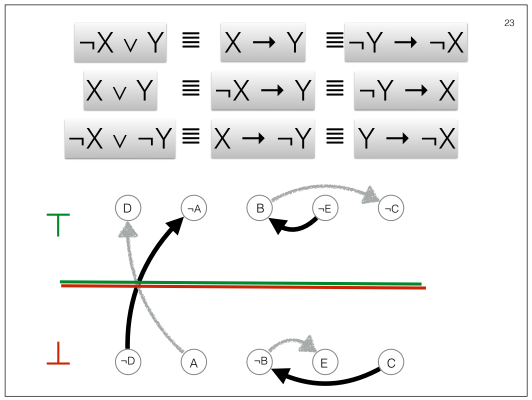
This valuation makes B and D true, and A, C,and E false.

It makes ¬D → ¬A, C→ ¬B, and ¬E → B true.



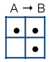
This valuation makes B and D true, and A, C, and E false.

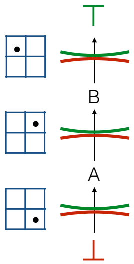
It makes ¬D → ¬A, C→ ¬B, and ¬E → B true, and ¬C → A is false



The arrows actually come in pairs, since each arrow is just one way of expressing a binary constraint:

A





Suppose A → B
there are three possible
truth valuations for A and B
(we exclude only (A = ⊤, B = ⊥))

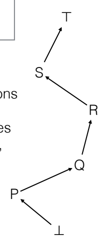
Propositions are ordered
by $x \leq y$ iff $x \to y = \top$
Any valid truth assignment
must draw a line
between ⊥ and ⊤

⊤
↑
Z
↑
Y
↑
X
↑
⋮
↑
C
↑
B
↑
A
↑
⊥

⊤
↑
Z
↑
Y
↑
X
↑
⋮
↑
C
↑
B
↑
A
↑
⊥

$P \to Q$
$\wedge$
$Q \to R$
$\wedge$
$R \to S$

$\neg P \vee Q$
$\wedge$
$\neg Q \vee R$
$\wedge$
$\neg R \vee S$



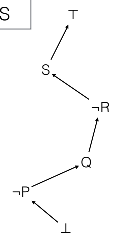
If we have a chain of n-1 implications
between n variables
we can draw the line in n+1 places
making any number, from 0 to n,
of these variables true.

We can draw the line
before each letter, or after them all.

27

$$\begin{array}{c} \neg P \rightarrow Q \\ \wedge \\ Q \rightarrow \neg R \\ \wedge \\ \neg R \rightarrow S \end{array} \qquad \begin{array}{c} P \vee Q \\ \wedge \\ \neg Q \vee \neg R \\ \wedge \\ R \vee S \end{array}$$

If some of the variables are negated we can do the same (but making the negated variables false when they fall above the line and true when they fall below)



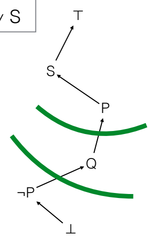
28

$$\begin{array}{c} \neg P \rightarrow Q \\ \wedge \\ Q \rightarrow P \\ \wedge \\ P \rightarrow S \end{array} \qquad \begin{array}{c} P \vee Q \\ \wedge \\ \neg Q \vee P \\ \wedge \\ \neg P \vee S \end{array}$$

If a variable appears together with its negation, we have to draw the line between them.

Here, P must be true.

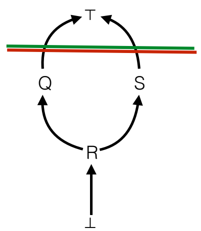
$(\neg P \rightarrow P) \rightarrow P$
is a tautology



29

$$\begin{array}{c} R \rightarrow Q \\ \wedge \\ R \rightarrow S \end{array} \qquad \begin{array}{c} \neg R \vee Q \\ \wedge \\ \neg R \vee S \end{array}$$

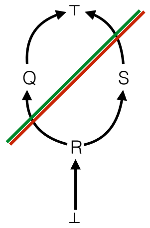
$\neg R \quad \neg Q \quad \neg S$



30

$R \to Q$
$\wedge$
$R \to S$

$\neg R \vee Q$
$\wedge$
$\neg R \vee S$

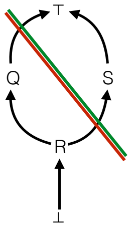


$\neg R$ Q $\neg S$

31

$R \to Q$
$\wedge$
$R \to S$

$\neg R \vee Q$
$\wedge$
$\neg R \vee S$

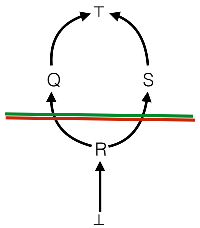


$\neg R$ $\neg Q$ S

32

$R \to Q$
$\wedge$
$R \to S$

$\neg R \vee Q$
$\wedge$
$\neg R \vee S$

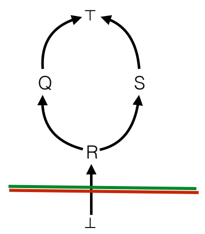


$\neg R$ Q S

$$R \rightarrow Q \;\wedge\; R \rightarrow S$$

$$\neg R \vee Q \;\wedge\; \neg R \vee S$$

R Q S

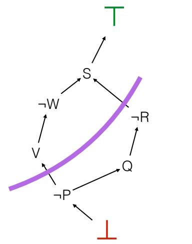


$$\neg P \rightarrow V \wedge V \rightarrow \neg W \wedge \neg W \rightarrow S \quad \wedge \quad \neg P \rightarrow Q \wedge Q \rightarrow \neg R \wedge \neg R \rightarrow S$$

$$P \vee V \wedge \neg V \vee \neg W \wedge W \vee S \quad \wedge \quad P \vee Q \wedge \neg Q \vee \neg R \wedge R \vee S$$

The same trick works if our implications form a partial order.
But we have more options since we can draw a wavy line.

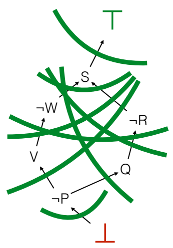
The **arrow rule** says that, whenever our line cuts an arrow, then the head must be on the side of true and the tail on the side of false.



The same trick works if our implications form a partial order.
But we have more options since we can draw a wavy line.

Not all of the valid truth assignments are represented in this diagram.

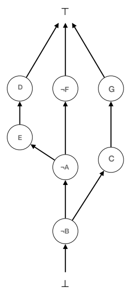
How many are missing?



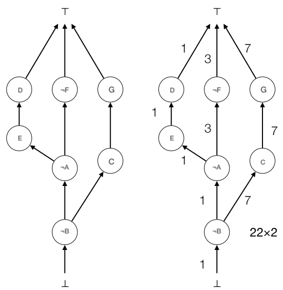
$ABCDEFGH$

eight letters, 256 valuations; only 7 letters used so multiply result by 2

$(\neg B \to \neg A) \wedge (\neg A \to E) \wedge (\neg A \to \neg F) \wedge (E \to D) \wedge (\neg B \to C) \wedge (C \to G)$

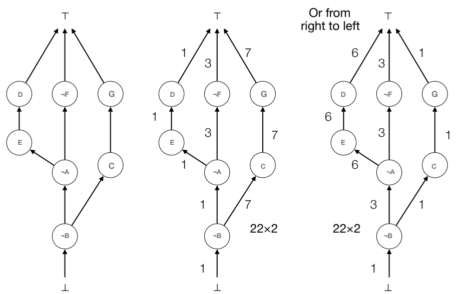


$(\neg B \to \neg A) \wedge (\neg A \to E) \wedge (\neg A \to \neg F) \wedge (E \to D) \wedge (\neg B \to C) \wedge (C \to G)$

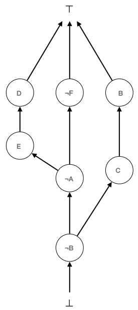


Count the ways of threading a path from left to right

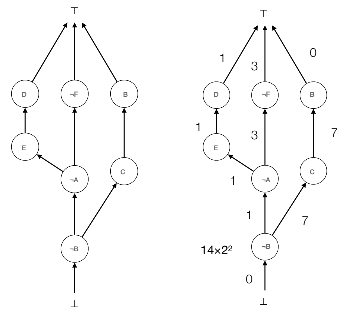
$(\neg B \to \neg A) \wedge (\neg A \to E) \wedge (\neg A \to \neg F) \wedge (E \to D) \wedge (\neg B \to C) \wedge (C \to G)$

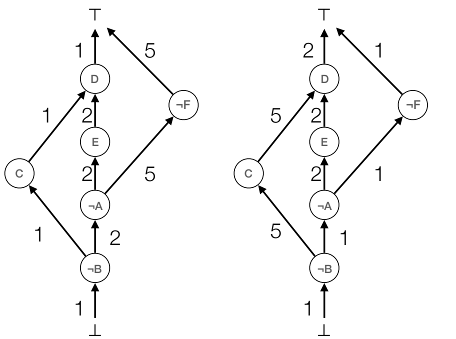


$(\neg B \to \neg A) \wedge (\neg A \to E) \wedge (\neg A \to \neg F) \wedge (E \to D) \wedge (\neg B \to C) \wedge (C \to B)$



$(\neg B \to \neg A) \wedge (\neg A \to E) \wedge (\neg A \to \neg F) \wedge (E \to D) \wedge (\neg B \to C) \wedge (C \to B)$





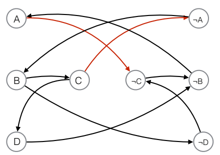
Binary constraints

You may not take both Archeology and Chemistry
If you take Biology you must take Chemistry
You must take Biology or Archeology
If you take Chemistry you must take Divinity
You may not take both Divinity and Biology

$(\neg A \lor \neg C) \land (\neg B \lor C) \land (B \lor A) \land (\neg C \lor D) \land (\neg D \lor \neg B)$

$(A \rightarrow \neg C) \land (B \rightarrow C) \land (\neg B \rightarrow A) \land (C \rightarrow D) \land (D \rightarrow \neg B)$

$(\neg A \lor \neg C) \land (\neg B \lor C) \land (B \lor A) \land (\neg C \lor D) \land (\neg D \lor \neg B)$

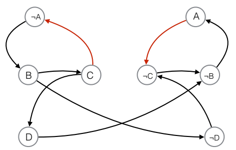


$(\neg A \lor \neg C) \land (\neg B \lor C) \land (B \lor A) \land (\neg C \lor D) \land (\neg D \lor \neg B)$

$\equiv$

$(A \rightarrow \neg C) \land (B \rightarrow C) \land (\neg B \rightarrow A) \land (C \rightarrow D) \land (D \rightarrow \neg B)$

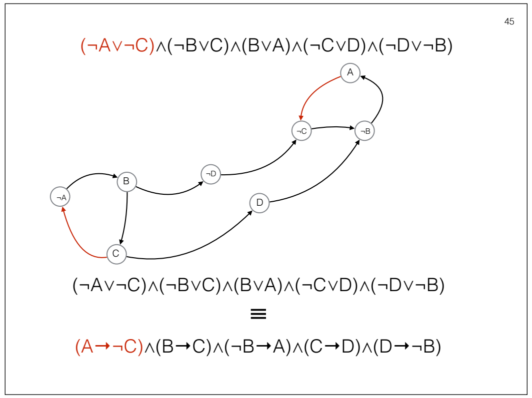
$(\neg A \lor \neg C) \land (\neg B \lor C) \land (B \lor A) \land (\neg C \lor D) \land (\neg D \lor \neg B)$



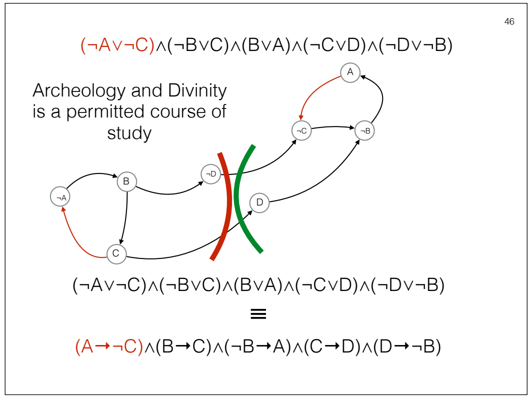
$(\neg A \lor \neg C) \land (\neg B \lor C) \land (B \lor A) \land (\neg C \lor D) \land (\neg D \lor \neg B)$

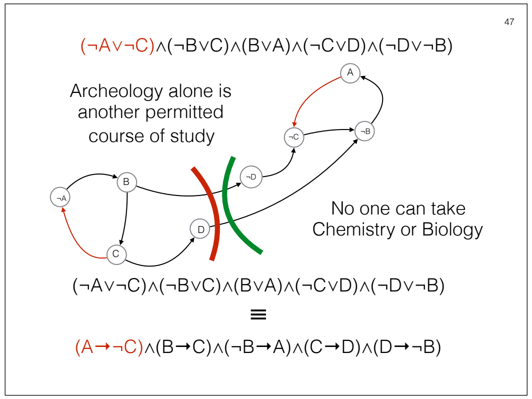
$\equiv$

$(A \rightarrow \neg C) \land (B \rightarrow C) \land (\neg B \rightarrow A) \land (C \rightarrow D) \land (D \rightarrow \neg B)$



If we have cycles of implications, then all nodes in the cycle must take the same truth value.





The algebraic transformation `Wff -> Form String`
which you implemented in Haskell
can produce a CNF whose size is **exponential** in the size of the Wff

The Tseytin procedure produces
a pattern of fixed size for each operation in the Wff,
so the size of the resulting CNF is
linear in the number of operations in the Wff.

Further readings on logic :

https://en.wikipedia.org/wiki/Propositional_formula
https://en.wikipedia.org/wiki/Propositional_calculus
https://en.wikipedia.org/wiki/Literal_(mathematical_logic)
https://en.wikipedia.org/wiki/Karnaugh_map
https://en.wikipedia.org/wiki/Conjunctive_normal_form
https://en.wikipedia.org/wiki/Valuation_(logic)
https://en.wikipedia.org/wiki/Satisfiability
https://en.wikipedia.org/wiki/DPLL_algorithm
https://en.wikipedia.org/wiki/Unit_propagation
https://en.wikipedia.org/wiki/Boolean_function

Logic

- Boolean functions and logical connectives
- representing constraints using logic
  e.g. no neighbouring cities have the same colours.
- derive CNF
  using km, using Boolean algebra, using Tseytin
- counting satisfying valuations
  various methods, e.g. arrow rule, simplifying
- simplifying a wff by setting a variable true or false
- understanding concepts underpinning DPLL
  CNF, valuation, reduction,
- simulate aspects of DPLL on small problems
  unit propagation

Logic

- Boolean functions and logical connectives
- representing constraints using logic
  e.g. no neighbouring cities have the same colours.
- derive CNF
  using km, using Boolean algebra, using Tseytin
- counting satisfying valuations
  various methods, e.g. arrow rule, simplifying
- simplifying a wff by setting a variable true or false
- understanding concepts underpinning DPLL
  CNF, valuation, satisfaction, refutation,
- simulate aspects of DPLL on small problems
  backtracking tree traversal
  literal selection
  unit propagation
  termination

How much of this can you do
without assistance?

University's Common Marking Scheme

50% is the pass mark

Grading system

This is quite different from, for example, the US system

a mark of 60% is very good

a mark of 90% is rare!

Numeric	Equivalent letter grade
80-100	A
70-79	A
60-69	B
55-59	C
50-54	D
46-49	E
40-45	F
35-39	F
0-34	G

Remember

if you can
do half of the questions
perfectly you will pass


After exams there will be holidays
but before exams we have to study machines