Informatics 1 - Computation & Logic: Tutorial 4

Satisfiability and Resolution

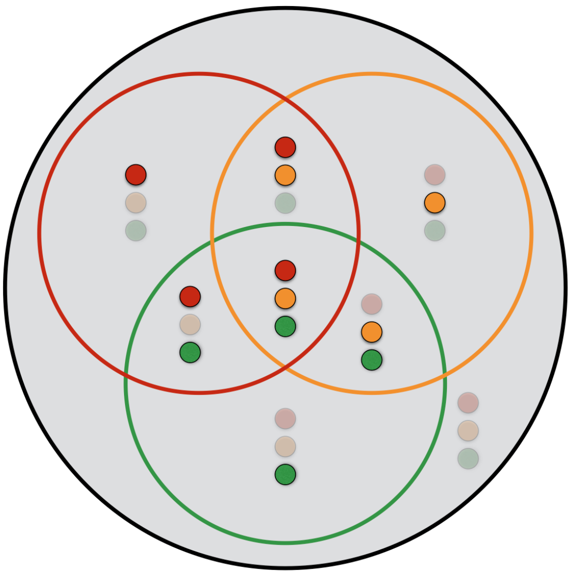
Week 6: 24-28 October 2016

Please attempt the entire worksheet in advance of the tutorial, and bring with you all work, including (if a computer is involved) printouts of code and test results. Tutorials cannot function properly unless you do the work in advance.

You may work with others, but you must understand the work; you can't phone a friend during the exam.

Assessment is formative, meaning that marks from coursework do not contribute to the final mark. But coursework is not optional. If you do not do the coursework you are unlikely to pass the exams.

Attendance at tutorials is **obligatory**; please let your tutor know if you cannot attend.



Consider the following clausal form:

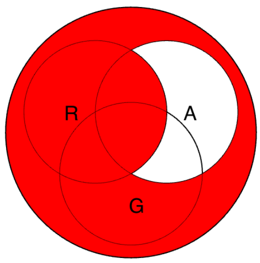
$$\{\, \{R, \neg A\}\, , \{G, A\}\, \}$$

Resolving on A, we get:

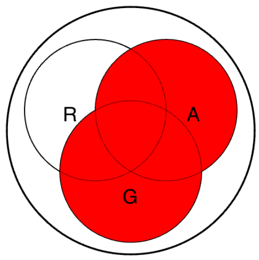
$$\frac{\{R, \neg A\} \qquad \{G, A\}}{\{R, G\}}\ (A)$$

It's worth remembering the reasoning behind resolution; If $(R \vee \neg A) \wedge (G \vee A)$ is true, if A is true, $\neg A$ must be false, and so for $R \vee \neg A$ to be true, R must be true; otherwise, A is false, and so for $G \vee A$ to be true, G must be true; therefore, $G \vee R$ must be true.

Recall the discussion in last week's tutorial sheet regarding clauses as *constraints*. We can visualise this resolution using Venn diagrams. Remember, here we are interested in the regions *excluded* by the clauses—the regions in white. Thus, the original clauses exclude:

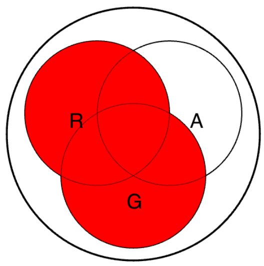


$\{R, \neg A\}$



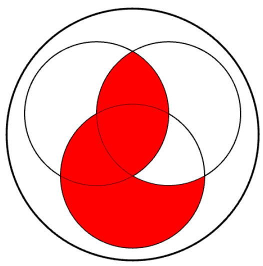
$\{G, A\}$

Our resolvent excludes (white regions):



$\{R, G\}$

Compare this to the valuations excluded by the whole clausal form (the union of the set of valuations excluded by the individual clauses):



$\{\, \{R, \neg A\}\, , \{G, A\}\, \}$

We see that the valuations excluded by the resolvent are a subset of the valuations excluded by the conjunction of the original clauses (resolvends); that is to say, the resolvent only excludes valuations that were already excluded by the resolvents; or again—*the resolvends entail the resolvent.*

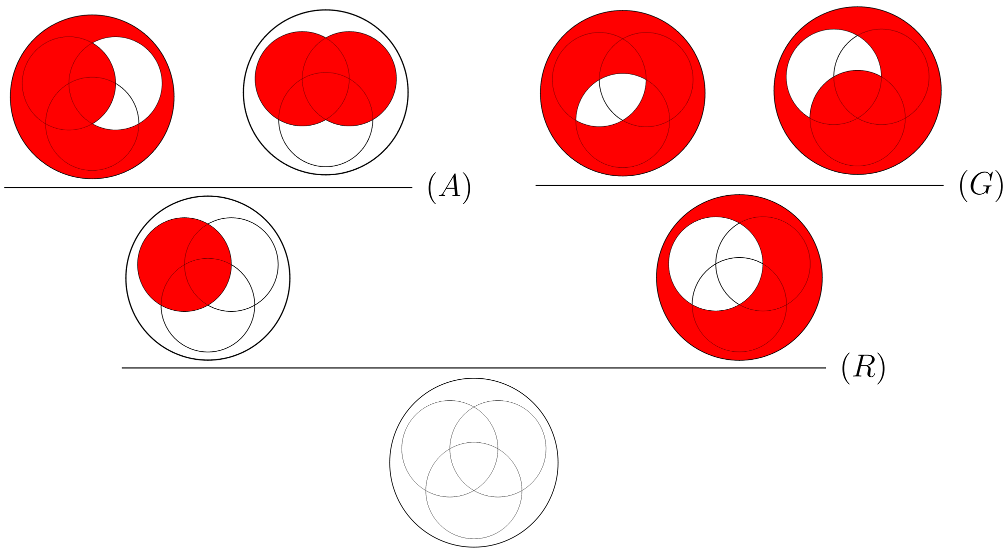
$$R \vee \neg A, G \vee A \vdash R \vee G$$

Now, let us consider a case that can be resolved to the empty set:

$$\{\, \{R, \neg A\}\, , \{R, A\}\, , \{\neg R, \neg G\}\, , \{\neg R, G\}\, \}$$

$$\dfrac{\dfrac{\{R, \neg A\} \qquad \{R, A\}}{\{R\}}\,(A) \qquad \dfrac{\{\neg R, \neg G\} \qquad \{\neg R, G\}}{\{\neg R\}}\,(G)}{\{\}}\,(R)$$

Using Venn diagrams:

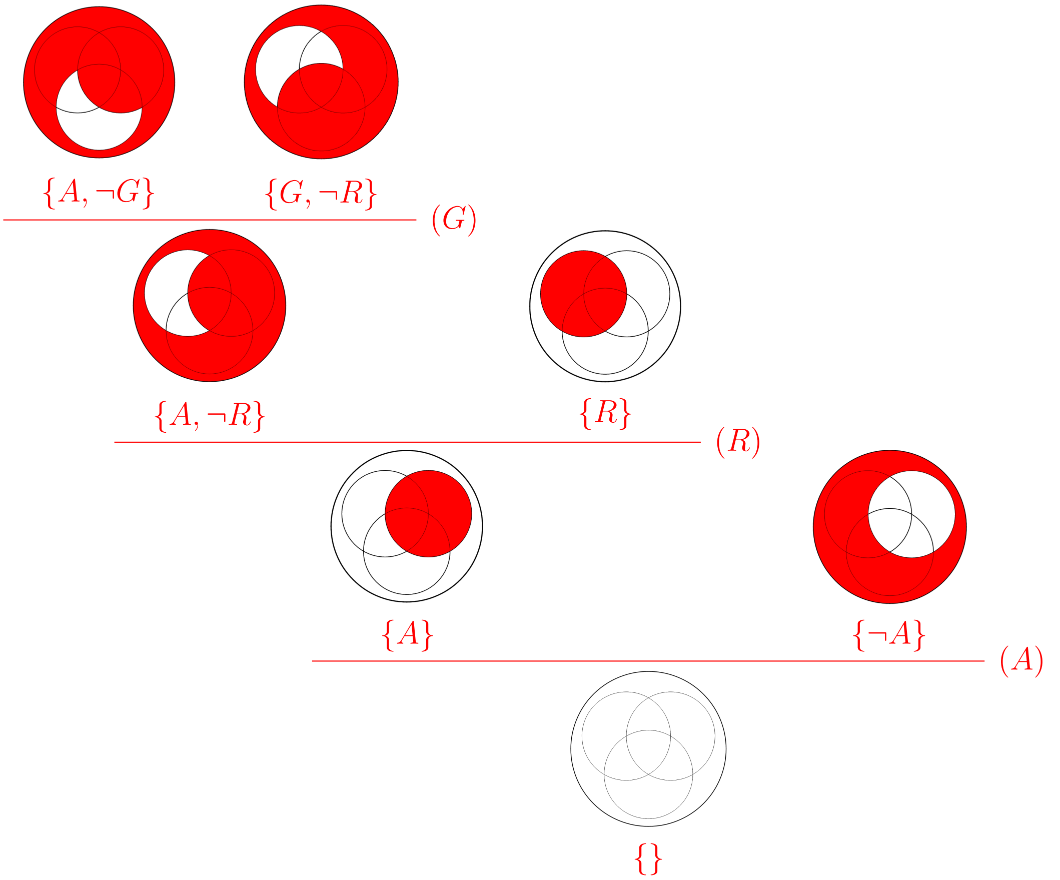


Here, the resolution process shows us that *all* valuations are excluded by the conjunction of the clauses; thus we can see that the clausal form has *no* satisfying valuations; it is contradictory.[1]

[1] You may have noticed that here, the sets of valuations excluded by the resolvents are *identical to* the unions of the sets excluded by their resolvends; this happens when the resolvends differ *only* in the literal to be resolved upon.

1. Use resolution to show whether the following clausal forms are satisfiable or not. Use Venn diagrams to check your answers, including intermediate resolvents.

   (a) $\{\{A, \neg G\}, \{G, \neg R\}, \{R\}, \{\neg A\}\}$

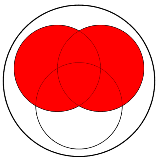


The empty set was found: therefore, all valuations are excluded — every valuation fails to satisfy at least one of the clauses; the clausal form is not satisfiable.

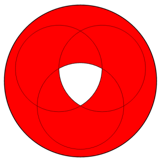
Once we have a derivation of the empty clause, given a valuation **V**, we can find a constraint that it violates.

Every valuation, in particular, **V**, violates the impossible constraint represented by the empty clause. So we can work our way up the derivation tree, starting from the root, then successively looking at a clause the violates **V** and choosing a parent that also violates **V**, until we reach one of the original clauses that violates **V**.

(b) $\{\, \{R, A\}\, , \{\neg R, \neg A, \neg G\}\, \}, \{G\}\, \}$

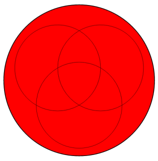


$\{R, A\}$

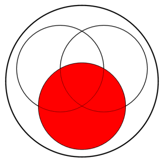


$\{\neg R, \neg A, \neg G\}$

(R)



$\{A, \neg A, \neg G\}$



$\{G\}$

Here, resolution leads to a trivial constraint, which is dropped. There are no further opportunities for resolution and we have not produced the empty clause. This is enough to show that we cannot produce the empty clause.

We can go further and produce a satisfying valuation. The resolution step replaces the clauses mentioning R by the single clause $A, \neg A, \neg G$. So we know that any valuation, $\mathbf{V}$ of A and G, that satisfies both this clause, $A, \neg A, \neg G$, and the remaining clause, G, can be extended with a suitable value for R to satisfy the original clauses.

To satisfy G, $\mathbf{V}(G)$ must be $\top$, but $\mathbf{V}(A)$ can take either Boolean value, to satisfy $A, \neg A, \neg G$. Whichever value we choose for $\mathbf{V}(A)$, the assumptions of the rule we have used can be satisfied by taking $\mathbf{V}(R)$ to be the complementary value, $\neg\mathbf{V}(A)$.

A Karnaugh Map (or *K-map*), like a Venn diagram, is a visual representation of a boolean expression. For domains consisting of four boolean letters, the map may be presented as a four-by four grid—but note that the two-dimensional plane wraps around on itself, from its bottom edge to its top, and from left to right - like the screen in Pac-Man or Asteroids. This geometry may be represented on the surface of a torus.

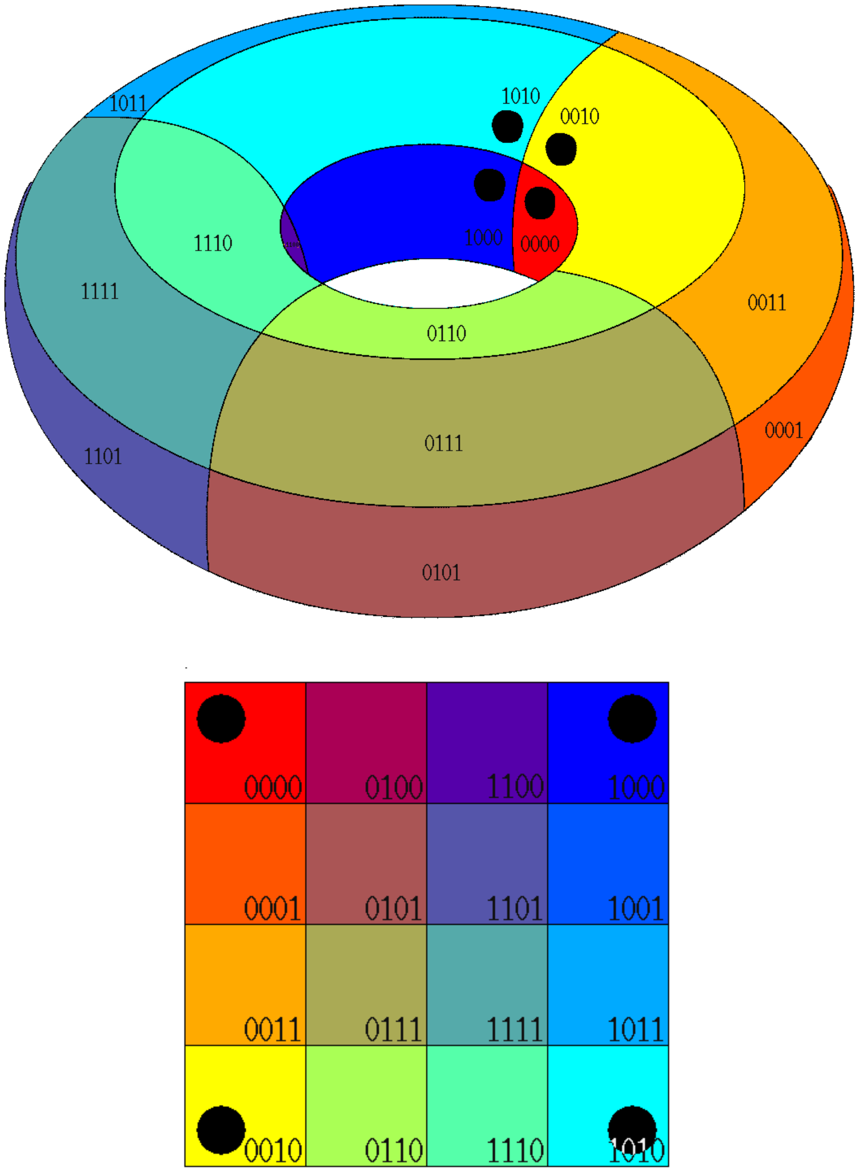
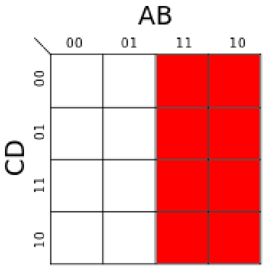


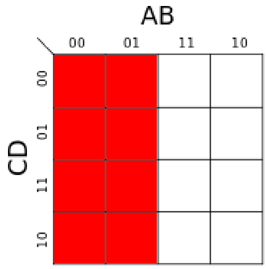
Figure 1: A Karnaugh map, wrapped around a torus and flattened out as a 4×4 grid. Note that the four dotted corners on the grid are adjacent to each other, as shown on the torus. The four-bit numbers show the state $ABCD$ represented by each square .

A K-map is often filled in with a 1 or 0 in each square, to indicate the truth or falsity, in each state, of a Boolean function of four variables. You can use ones and zeros, or colours or your choice, to indicate truth and falsity.

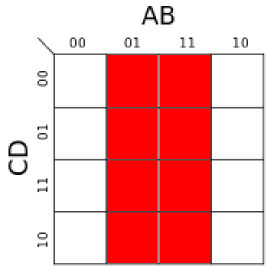
2. First fill in the Karnaugh maps below to show the area corresponding to (the truth of) each of the atomic propositions, and the negations of two of them.



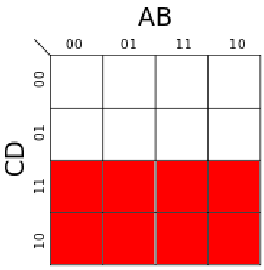
A



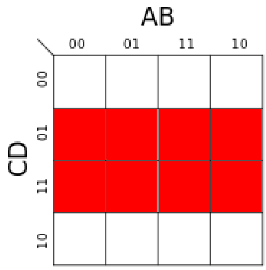
$\neg A$



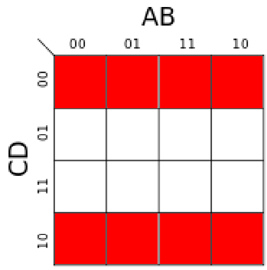
B



C



D

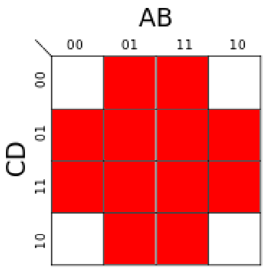


$\neg D$

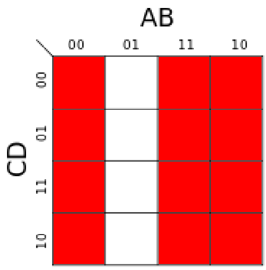
3. Consider the clausal form:

$$\{\, \{D, B\}\,, \{A, \neg B\}\,, \{\neg A, \neg B\}\,, \{\neg C, \neg D\}\,, \{C, \neg D\}\, \}$$

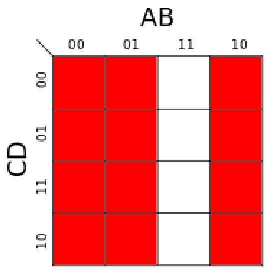
(a) Show the region corresponding each of the clauses in its own K-map:



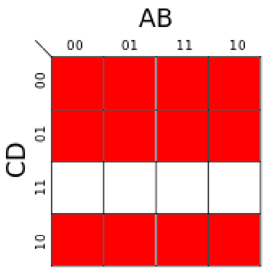
$\{D, B\}$



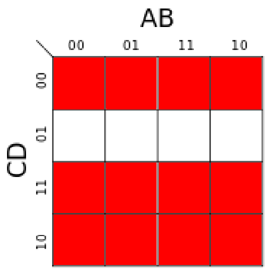
$\{A, \neg B\}$



$\{\neg A, \neg B\}$

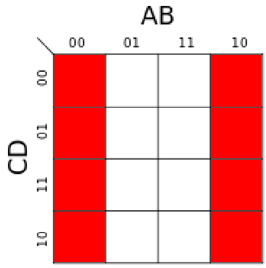


$\{\neg C, \neg D\}$



$\{C, \neg D\}$

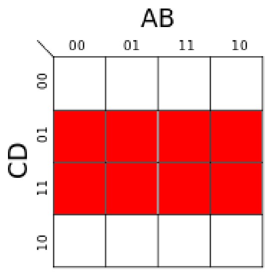
(b) Now, take all clauses containing A or $\neg A$, and resolve on A every possible pairing. Cross out the resolvends, and show the resolvent(s) in on the provided K-map. The remaining clauses from the original clausal form plus the new resolvents are your *resolution pool*.



$$\frac{\{A, \neg B\} \quad \{\neg A, \neg B\}}{\{\neg B\}}\ (A)$$

Here the result, $\neg B$, of the resolution step excludes all those states excluded by either of the resolved clauses.

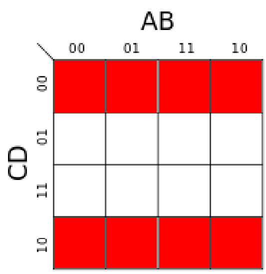
(c) If the previous step did not result in the empty clause being found, repeat the procedure, this time resolving on B, again using every pair of clauses in your resolution pool with complementary B-literals. Again, cross out the resolvends. The remaining clauses, plus the new resolvents are your updated resolution pool.



$$\frac{\{\neg B\} \quad \{D, B\}}{\{D\}}\ (B)$$

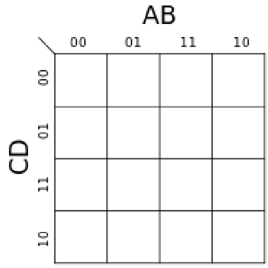
Here the result, D, of the resolution step is harder to describe. It consists of those states $abcd$ such that each of the states $0bcd$ and $1bcd$ is excluded by at least one of the resolved clauses. [Note that this description would also cover the previous step.]

(d) If the previous step did not result in the empty clause being found, repeat resolving on C.



$$\frac{\{\neg C, \neg D\} \quad \{C, \neg D\}}{\{\neg D\}}\ (C)$$

(e) If the previous step did not result in the empty clause being found, repeat resolving on D.



$$\frac{\{D\} \quad \{\neg D\}}{\{\}}\ (D)$$

(f) Was the empty clause found? Yes

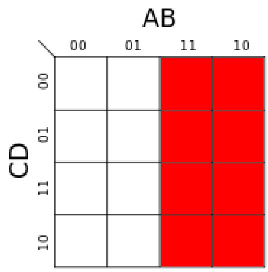
(g) Is the clausal form satisfiable? No

The previous example was somewhat trivial, because at each stage you in fact only had to resolve one pair of resolvends. This will not always be the case. Thus, if you have three clauses containing C and three containing $\neg C$, you will have nine resolutions to perform on C. In the worst case scenario, this can be computationally intensive for large clausal forms; however, it has the advantage of guaranteeing that once each atom has been resolved upon, if the clauses are not satisfiable, the empty clause *will* have been found.

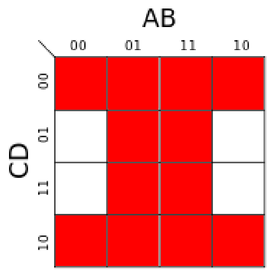
4. Consider the following clausal form:

$$\{\,\{A\}\,, \{B, \neg D\}\,, \{\neg A, \neg B, C, \neg D\}\,, \{\neg A, D\}\,\}$$

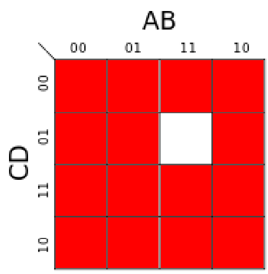
(a) Again, show the region corresponding to each clause in its own K-map:



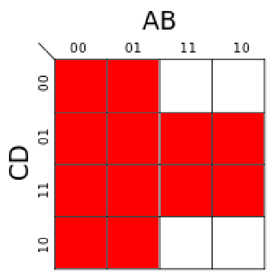
$\{A\}$



$\{B, \neg D\}$

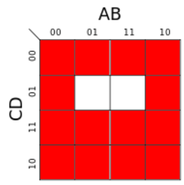
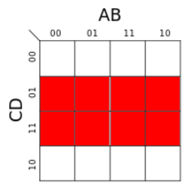
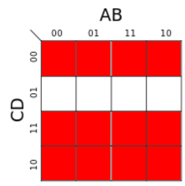
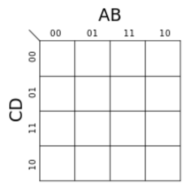
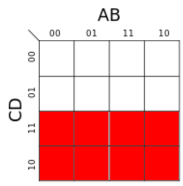


$\{\neg A, \neg B, C, \neg D\}$



$\{\neg A, D\}$

(b) Following the same procedure as before, resolve on each atom in turn:
Superscript letters are prepended to each clause used in the resolution, to indicate the resolution variable used to eliminate that clause.

	Resolve A	Resolve B	Resolve C	Resolve D
^{A}A $^{B}B, \neg D$ $^{A}\neg A, \neg B, C, \neg D$ $^{A}\neg A, D$	$^{B}\{\neg B, C, \neg D\}$ $^{D}\{D\}$	$^{D}\{C, \neg D\}$	none!	$\{C\}$
				
				

(c) Was the empty clause found? No

(d) Is the clausal form satisfiable? Yes

Resolution terminates with the single clause C. We can satisfy this by taking $\mathbf{V}(C) = \top$.

Then we can extend the valuation to a satisfying valuation for all of the original clauses, by working backwards through each step.

Looking first at the clauses labelled ^{D} we see that $\mathbf{V}(D)$ must be $\top$, to make both $C, \neg D$ and D true.

The next step is to consider clauses labelled ^{B}. It is trivial to check that taking $\mathbf{V}(B) = \top$ makes both $\neg B, C, \neg D$ and $B, \neg D$ true.

Finally, consider the clauses labelled ^{A}. It is trivial to check that taking $\mathbf{V}(A) = \top$ makes these remaining clauses true.

A satisfying valuation is given by A, B, C, D.

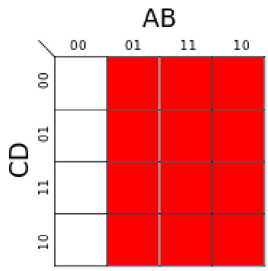
The properties of the resolution step tell us that whenever this resolution procedure fails to derive the empty clause, there will be a satisfying valuation for the remaining clauses, and that any such valuation can be extended stepwise to give a valuation that satifies all of the original clauses.

The resolution procedure is *complete*: if the constraints are inconsistent we will derive the empty clause, and refute every valuation. Otherwise, we can use the failure to produce the empty clause to produce a valuation that satisfies all the constraints.

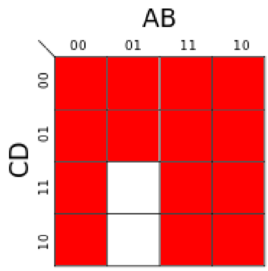
5. Our next example includes 6 clauses:

$$\{\, \{A, B\}\, , \{A, \neg B, \neg C\}\, , \{\neg A, D\}\, , \{\neg B, C, D\}\, , \{\neg B, \neg D\}\, , \{\neg A, B, \neg D\}\, \}$$

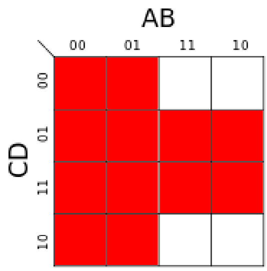
(a) Show each clause in its own K-map:



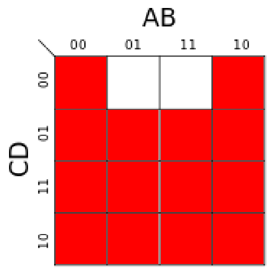
$\{A, B\}$



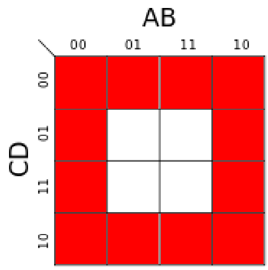
$\{A, \neg B, \neg C\}$



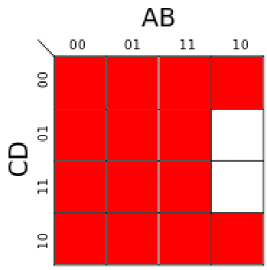
$\{\neg A, D\}$



$\{\neg B, C, D\}$

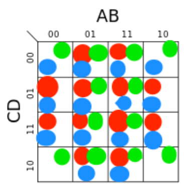
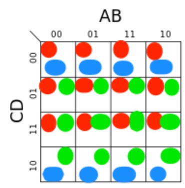
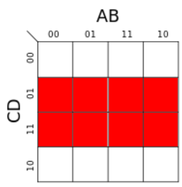
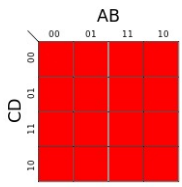


$\{\neg B, \neg D\}$



$\{\neg A, B, \neg D\}$

(b) Following the same procedure as before, resolve on each variable in turn

	Resolve A	Resolve B	Resolve C	Resolve D
$^{A}A, B$	$^{B}\{B, D\}$	$^{C}\{\neg C, D\}$	$^{D}\{D\}$	$\{\}$
$^{A}A, \neg B, \neg C$	$^{B}\{B, \neg D\}$	$^{C}\{C, D\}$		
$^{A}\neg A, D$	$^{B}\{\neg B, \neg C, D\}$	$\{D, \neg D\}$		
$^{B}\neg B, C, D$	$\{B, \neg B, \neg C, \neg D\}$	$\{\neg C, D, \neg D\}$		
$^{B}\neg B, \neg D$	*Note: this clause*	$\{C, D, \neg D\}$		
$^{A}\neg A, B, \neg D$	*is tautologous,*	$^{D}\{\neg D\}$		
	it cannot			
	contribute to			
	further resolution.			
				

(c) Was the empty clause found? Yes

(d) Is the clausal form satisfiable? No

6. Of course, logical claims can consist of arbitrarily many atoms, and our ability to usefully represent them visually is eventually exhausted. However, resolution can be applied to sets of clauses with any number of atoms. This question concerns the resolution of the claim that:

$$P \rightarrow (Q \vee R), Q \rightarrow \neg S, S \vee R, R \rightarrow Q, (Q \wedge R) \rightarrow T \vdash P \rightarrow T$$

(a) Express each of the assumptions, and the negation of the conclusion, in clausal form.

i. $P \rightarrow (Q \vee R)$
$\neg P \vee (Q \vee R)$ by arrow elimination
$\neg P \vee Q \vee R$ by associativity
$\{\neg P, Q, R\}$

ii. $Q \rightarrow \neg S$
$\neg Q \vee \neg S$ by arrow elim
$\{\neg Q, \neg S\}$

iii. $S \vee R$ $\{S, R\}$

iv. $R \rightarrow Q$
$\neg R \vee Q$ by arrow elim
$\{\neg R, Q\}$

v. $(Q \wedge R) \rightarrow T$
$\neg(Q \wedge R) \vee T$ by arrow elim
$(\neg Q \vee \neg R) \vee T$ by De Morgan
$\neg Q \vee \neg R \vee T$ by associativity
$\{\neg Q, \neg R, T\}$

vi. $\neg(P \rightarrow T)$
$\neg(\neg P \vee T)$ by arrow elim
$\neg\neg P \wedge \neg T$ by De Morgan
$P \wedge \neg T$ by double negation elimination
$\{P\}, \{\neg T\}$

(b) Use resolution to determine whether the negation of the conclusion is consistent with the conjunction of the assumptions.

	Resolve P	Resolve Q	Resolve R	Resolve S	Resolve T
${}^{P}\neg P, Q, R$	${}^{Q}Q, R$	${}^{R}R, \neg S$	${}^{S}\neg S$		
${}^{Q}\neg Q, \neg S$		${}^{R}\neg R, \neg S$	${}^{S}\neg S, T$	${}^{T}T$	
${}^{R}S, R$			${}^{S}S, T$		$\{\ \}$
${}^{Q}\neg R, Q$		$R, \neg R, T$	$S, \neg S$		
${}^{Q}\neg Q, \neg R, T$					
${}^{P}P$		${}^{R}\neg R, T$			
${}^{T}\neg T$					

(c) Is the original claim correct? YES!

7. This question concerns the 256 possible truth valuations of the following eight propositional letters A, B, C, D, E, F, G, H. For each of the following expressions, say how many of the 256 valuations satisfy the expression, and briefly explain your reasoning. For example, the expression D is satisfied by half of the valuations, that is 128 of the 256, since for each valuation that makes D true there is a matching valuation that make D false.

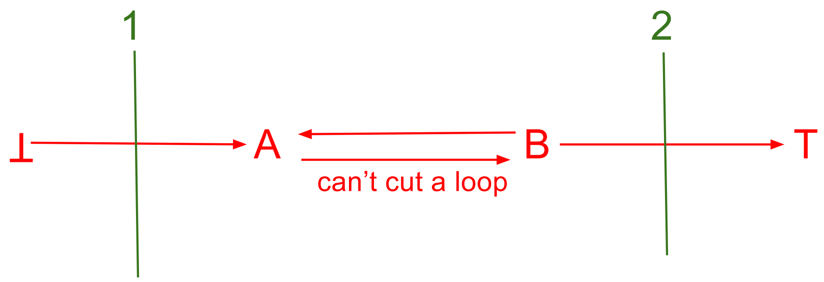
   (a) $A \wedge B$ 64 — one quarter of 256.

   (b) $(A \vee B) \wedge C$ 96 — 3/4 of the 256 satisfy $(A \vee B)$ and one half of these have C true, so the answer is 3/8 * 256

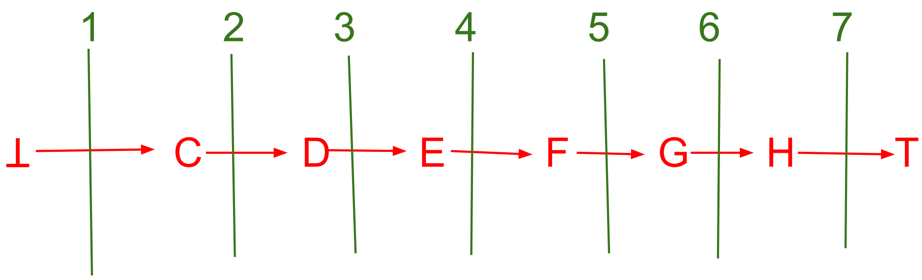
   (c) $(A \to B) \to C$ 160 === this is equivalent to $\neg(A \to B) \vee C$; the first disjunct is true for one quarter of the 256, and the second for half of them. The overlap is one half of the first disjunct, so the answer is 5/8 * 256

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

We can use the arrow rule to solve this:



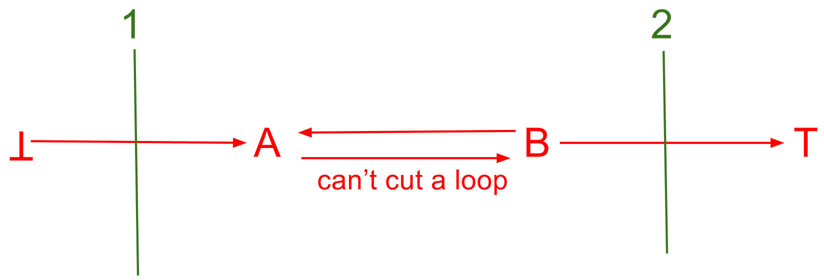
We find 2 valuations for A and B.



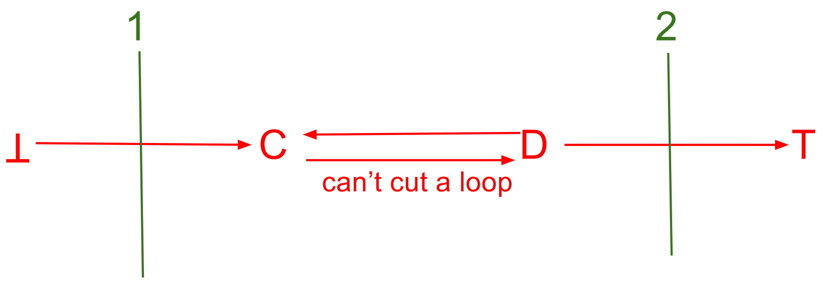
...and 7 for C, D, E, F, G, and H, giving $2 \times 7 = 14$ valuations in total

(e)

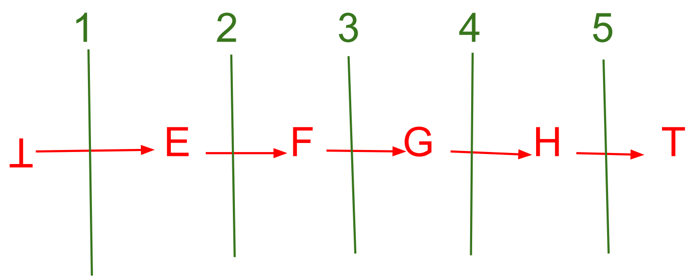
$$
\begin{aligned}
(A \to B) \wedge (B \to A) \wedge (C \to D) \wedge (D \to C) \\
\wedge (E \to F) \wedge (F \to G) \wedge (G \to H)
\end{aligned}
$$



We find 2 valuations for A and B.



...2 valuations for C and D.

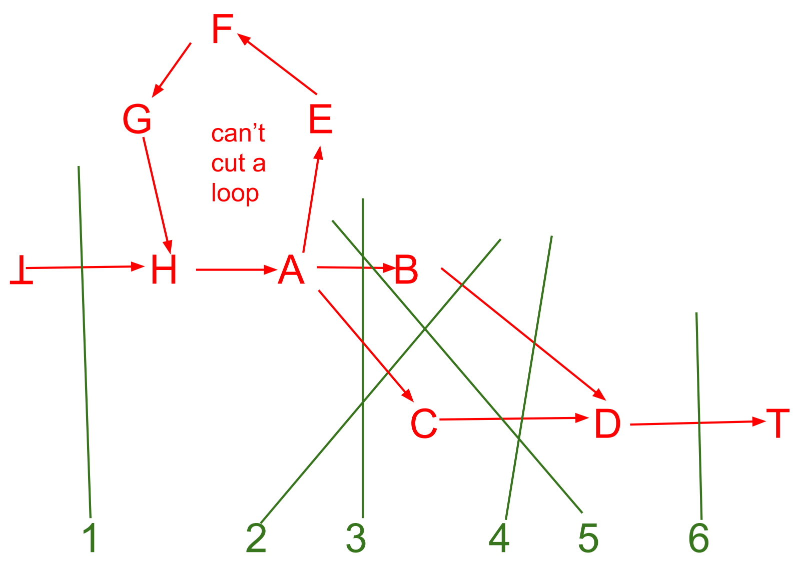


...and 5 for E, F, G, and H, giving $2 \times 2 \times 5 = 20$ valuations in total

(f)

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

Noting that $A \to B \wedge C$ is equivalent to $(A \to B) \wedge (A \to C)$ and $(B \vee C \to D)$ is equivalent to $(B \to D) \wedge (C \to D)$, we derive the following graph, giving 6 satisfying valuations:



This tutorial exercise sheet was written by Dave Cochran and Michael Fourman, with additional contributions from an earlier tutorials produced by Paolo Besana, Thomas French, and Areti Manataki. Send comments to `Michael.Fourman@ed.ac.uk`