Informatics 1 - Computation & Logic: Tutorial 3

Satisfiability and Resolution

Week 5: 19-23 October 2015

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.

1. Consider the encoding of a 9×9 Sudoku problem discussed in class. We use 729 propositional letters, or atoms, $P_{i,j,k}$, where i, j, k can each take integer values from 1 to 9. A Boolean valuation of these corresponds to a 9×9 square, filled in such that

the number k is written in the j^{th} row of the i^{th} column
iff
the valuation makes $P_{i,j,k}$ true

A Sudoku puzzle is specified by a partial valuation, that makes some atoms true, to place particular numbers in some squares. A typical puzzle will leave 50–60 squares blank. We consider each total valuation extending the one given by the puzzle to be a putative solution.

This question concerns the number of putative solutions and the Boolean constraints that we must impose to specify a correct solution.

(a) How many possible valuations are there? If we could check 2^{30} putative solutions per second, roughly how long would it take to check every putative solution for a typical puzzle?
Since there are 729 atoms, there are 2^{729} valuations. There are about 2^{25} seconds in a year, and the age of the universe is approximately 2^{34} years, or 2^{59} seconds. At 2^{30} valuations per second, we can therefore compute 2^{89} valuations in the age of the universe; meaning it would take approximately 2^{640} (4×10^{192}) times the age of the universe to check every valuation. However, since the last of the supermassive black holes are expected to evaporate via Hawking radiation in only 10^{100} years, leaving a cold, dark universe dominated by diffuse, low-energy photons, electrons, neutrinos, and their antiparticles, this may be impractical.

(b) How would you express the constraint that a solution can place at most one number in each square? How many clauses are required to express this in CNF? Assuming the same rate of checking as in (a), how long would it take to check only those putative solutions that satisfy this constraint?
For each square i, j, if $P_{i,j,x}$ is true, and y is a number different from x, then $P_{i,j,y}$ must be false. For example,

$$P_{i,j,1} \to \neg(P_{i,j,2})$$

So, for each i, j, x, y, with $x \neq y$ we add the clause

$$P_{i,j,x} \to \neg(P_{i,j,y})$$

In fact, this adds each pair of unequal numbers twice. It suffices, for each i, j, x, y, with $x < y$ to add the clause

$$P_{i,j,x} \to \neg(P_{i,j,y}) \qquad (9 \times 9 \times 9 \times 4 \text{ clauses})$$

There are 9^{81} valuations that meet this condition, or approximately 2^{257}; requiring a mere 2^{227} seconds, 2^{202} years, or 2^{168} times the current age of the universe. At this rate, the computation will be completed before the heat death of the universe, but may still be considered too long to be convenient.

(c) How would you express the constraint that a solution must place every number somewhere in each row? How many clauses are required to express this in CNF?
For each number, x and each row, j we must express the fact that x occurs in some column i. This can be done with a single clause for each of the 81 possible combinations of x and j.

$$(P_{1,j,x} \vee P_{2,j,x} \vee \ldots \vee P_{8,j,x} \vee P_{9,j,x}) \qquad (9 \times 9 \text{ clauses})$$

(d) How would you express the constraint that a solution must place every number somewhere in each column? How many clauses are required to express this in CNF?
This can be done with a single clause for each of the 81 possible combinations of x and i.

$$\left(P_{i,1,x} \vee P_{i,2,x} \vee \ldots P_{i,8,x} \vee P_{i,9,x}\right) \qquad (9 \times 9 \text{ clauses})$$

(e) How would you express the constraint that a solution can must place every number somewhere in each 3×3 subsquare? How many clauses are required to express this in CNF?
For each of the 9 sub squares, we need 9 clauses (one for each x) similar to the following (which gives the 9 clauses for the middle-right sub square),

$$\begin{pmatrix} P_{7,4,x} \vee P_{8,4,x} \vee P_{9,4,x} \\ P_{7,5,x} \vee P_{8,5,x} \vee P_{9,5,x} \\ P_{7,6,x} \vee P_{8,6,x} \vee P_{9,6,x} \end{pmatrix} \qquad (9 \text{ clauses for each sub square})$$

So, 9×9 clauses for this constraint.

For some problems, it may be hard to find a solution, but it is straightforward to check whether an answer is correct. Sudoku is an example: it is straightforward to check the correctness of a putative solution, but it can be hard to find a solution. All such (easy-to-check but hard-to find) search problems could be solved, in principle, by exhaustive search, but a 'combinatorial explosion' often makes this impractical.

We have encoded the Sudoku problem as an instance of the Boolean satisfiability problem (SAT). With a suitable technical definition of 'straightforward to check', it can be proved that every such problem can be reduced to SAT.

2. 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

   (b) $(A \vee B) \wedge C$ 96

   (c) $(A \to B) \to C$ 160

   (d)
   $$(A \to B) \wedge (B \to A) \wedge (C \to D) \wedge (D \to E) \wedge (E \to F) \wedge (F \to G) \wedge (G \to H)$$
   14

   (e)
   $$(A \to B) \wedge (B \to A) \wedge (C \to D) \wedge (D \to C) \wedge (E \to F) \wedge (F \to G) \wedge (G \to H)$$
   20

   (f)
   $$(H \to A) \wedge (A \to B \wedge C) \wedge (B \vee C \to D) \wedge (A \to E) \wedge (E \to F) \wedge (F \to G) \wedge (G \to H)$$
   6

3. This question concerns the resolution of the claim that

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

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

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

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

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

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

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

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

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

The resolution pool is;

$$\{\neg P, Q, R\}, \{\neg Q, \neg S\}, \{S, R\}, \{\neg R, Q\}, \{\neg Q, \neg R, S\}, \{P\}, \{\neg S\}$$

Note that resolving pairs of clauses does not eliminate them; it adds a new clause to the pool; clauses may therefore be reused, and it is not necessary to use all clauses. We can add the following clauses by resolution;

i. $\{Q, R\}$ by resolving $\{\neg P, Q, R\}$ and $\{P\}$ on P
ii. $\{\neg Q, \neg R\}$ by resolving $\{\neg Q, \neg R, S\}$ and $\{\neg S\}$ on S
iii. $\{S, \neg Q\}$ by resolving $\{\neg Q, \neg R\}$ and $\{S, R\}$ on R
iv. $\{\neg Q\}$ by resolving $\{\neg Q, \neg S\}$ and $\{S, \neg Q\}$ on S
v. $\{R\}$ by resolving $\{Q, R\}$ and $\{\neg Q\}$ on Q
vi. $\{\neg R\}$ by resolving $\{\neg R, Q\}$ and $\{\neg Q\}$ on Q
vii. $\{\}$ by resolving $\{\neg R\}$ and $\{R\}$ on R

If there is a valuation making all of the assumptions true then this same valuation will make each of the clauses introduced by resolution true. At each resolution step, the valuation makes both assumptions true, so it makes the conclusion true.

But no valuation can make both R and $\neg R$ true; the conclusions at 3(b)v and 3(b)vi contradict each other. We conclude that no valuation can make all of the assumptions true.

Normally we take one more step: we apply resolution to the contradictory clauses, to produce the empty clause. Since the empty clause is inconsistent, the premises conjoined with the negation of the conclusion must also be inconsistent.

(c) Is the original claim correct?

Yes.