Universität des Saarlandes
Fachrichtung 6.2 – Informatik
J. Siekmann, S. Autexier, K. Fischer, E. Melis, C. Zinn
M. Schiller, E. Schulz, R.Vollmann

4th Theoretical Assignment in Artificial Intelligence (WS 2006/2007)
Solutions

Note: You need not hand these exercises in, and they are not graded. But bring along your solutions to the tutorial. Impress your tutors by presenting your favourite exercise to the class.

Exercise 4.1 *Cryptarithmetic Puzzle*

Solve the following problem by hand

$$\begin{array}{cccc} & \mathrm{T} & \mathrm{W} & \mathrm{O} \\ + & \mathrm{T} & \mathrm{W} & \mathrm{O} \\ \hline \mathrm{F} & \mathrm{O} & \mathrm{U} & \mathrm{R} \end{array}$$

using:

- backtracking search with forward checking,
- minimum-remaining-value heuristic and
- least-constraining-value heuristic

Solution:

Initial Situation:

$$Alldiff(F, T, U, W, R, O)$$

$$\begin{array}{lrl} (1) & 2O &= R + 10X_1 \\ (2) & X_1 + 2W &= U + 10X_2 \\ (3) & X_2 + 2T &= O + 10X_3 \\ (4) & X_3 &= F \end{array}$$

$$\begin{aligned} T, U, W, R, O &\in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \\ F &\in \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \\ X_1, X_2, X_3 &\in \{0, 1\} \end{aligned}$$

1. *MRV picks one of X_1, X_2, X_3, say X_3; LCV chooses 1 (leaves 16 possible values for neighboring variables, instead of 14); that is, $X_3 = 1$*

   Forward Checking gives us:

$$\begin{array}{lrl} (3a) & X_2 + 2T &= O + 10 \\ (4a) & 1 &= F \end{array}$$

   and by using (3a):

$$\begin{aligned} F &\in \{1\} \\ T &\in \{5, 6, 7, 8, 9\} \\ O &\in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \end{aligned}$$

From before we have:

$$U, W, R \in \{0,1,2,3,4,5,6,7,8,9\}$$
$$X_1, X_2 \in \{0,1\}$$

2. *MRV picks F; LCV chooses 1 (only possible value); that is, $F = 1$*

   Forward Checking gives us:

$$U, W, R, O \in \{0,2,3,4,5,6,7,8,9\}$$

   From before we have:

$$T \in \{5,6,7,8,9\}$$
$$X_1, X_2 \in \{0,1\}$$

3. *MRV picks one of X_1, X_2, say X_2; LCV chooses one of $0, 1$ (both leave 19 possible values for neighboring variables), say 1; that is, $X_2 = 1$*

   Forward Checking gives us:

$$\begin{aligned} (2b) \quad X_1 + 2W &= U + 10 \\ (3b) \quad 1 + 2T &= O + 10 \end{aligned}$$

   and by using (2b) and (3b) and Alldiff

$$\begin{aligned} W &\in \{5,6,7,8,9\} \\ U &\in \{0,2,3,4,5,6,7,8\} \\ T &\in \{6,7,8\} \\ O &\in \{3,5,7\} \end{aligned}$$

   From before we have:

$$R \in \{0,2,3,4,5,6,7,8,9\}$$
$$X_1 \in \{0,1\}$$

4. *MRV picks X_1; LCV chooses 1 (leaves 10 possible values for neighboring variables instead of 4); that is, $X_1 = 1$*

   Forward Checking gives us:

$$\begin{aligned} (1c) \quad 2O &= R + 10 \\ (2c) \quad 1 + 2W &= U + 10 \end{aligned}$$

   and using (1c) and (2c) and Alldiff

$$\begin{aligned} O &\in \{5,7\} \\ R &\in \{0,4\} \\ W &\in \{6,7,8\} \\ U &\in \{3,5,7\} \end{aligned}$$

   From before we have:

$$T \in \{6,7,8\}$$

5. *MRV picks one of O, R, say O (this is what degree heuristic would have picked); LCV picks one of $5, 7$ (both leave two possible values for neighboring variables), say 7; that is $O = 7$*

   Forward Checking gives us:

$$\begin{aligned} (1d) \quad & 14 = R + 10 \\ (3d) \quad & 1 + 2T = 17 \end{aligned}$$

   and using (1d) and (3d)

$$R \in \{4\}$$
$$T \in \{8\}$$

   From before we have:

$$W \in \{6, 7, 8\}$$
$$U \in \{3, 5, 7\}$$

6. *two more iterations pick $R = 4$ and $T = 8$*

7. *MRV picks one of W, U, say U; LCV picks one of $3, 5, 7$ (all three leave one possible value for neighboring variables), say 3; that is $U = 3$*

   Forward Checking gives us:

$$(2e) \quad 1 + 2W = 13$$

   and using (2e)

$$W \in \{6\}$$

8. *one more iteration picks $W = 6$ and we're done with the following assignment:*
   $T = 8, W = 6, O = 7, F = 1, U = 3, R = 4, X_1 = 1, X_2 = 1, X_3 = 1$

$$\begin{array}{cccc} & 8 & 6 & 7 \\ + & 8 & 6 & 7 \\ \hline 1 & 7 & 3 & 4 \end{array}$$

Exercise 4.2 *Arc Consistency*

Imagine you are responsible for organizing a seminar with four different speakers A, B, C and D, to each of whom you would like to assign one out of four timeslots.
The topics require that:

- Speaker A gives his talk one or two timeslots before speaker B gives his talk.
- Speaker C speaks before speaker A.
- Speaker C also speaks before speaker B.
- Speaker C gives his talk either one or two timeslots before speaker D.
- Speaker D speaks before speaker B.

Furthermore, speaker A is away during the third timeslot. Note that no two speakers may give their talks during the same timeslot.

1. Describe the constraint satisfaction problem (indicate variables, their domain, constraints) and draw the constraint graph. Representing the timeslots by numbers allows you to formulate the constraints as arithmetic expressions. Constraints that are subsumed by other constraints (e.g. $A < 5$ is subsumed by $A < 2$) can be ignored.

2. Is the network arc-consistent? If not, compute the arc-consistent network (show the whole process of enforcing arc-consistency and not just the final arc-consistent network).

3. Is the network consistent (is there a valid assignment for the variables that solves the problem)? If yes, give a solution.

Solution:

1. *We have four variables, A, B, C and D, wich stand for the speakers A, B, C and D. We have four timeslots, which are represented by the numbers $1, 2, 3$ and 4 in the corresponding order (timeslot 1 is before timeslot 2, and so forth), such that $A, B, C, D \in \{1, 2, 3, 4\}$. We have the following constraints:*

   (a) $0 < B - A \leq 2$

   (b) $C < A$

   (c) $C < B$

   (d) $0 < D - C \leq 2$

   (e) $D < B$

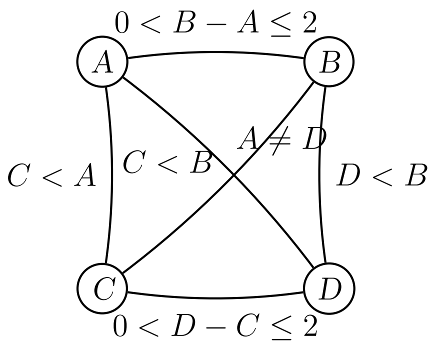
   (f) $A \neq 3$

   Furthermore, we know that no two speakers have their talk at the same time, which means that $X \neq Y$ for all pairs of different speakers X and Y. But we only need to add one constraint

   (g) $A \neq D$

   because this constraint is not already subsumed by the constraints above.

   The constraint graph is given by:



2. *The network is not arc consistent. An arc consistent network is a network where all the arcs are consistent. An arc (V_i, V_j) is consistent if for every value x in the domain of V_i there is a value y in the domain of V_j such that the assignment ($V_i = x$ and $V_j = y$) is permitted by the constraint between V_i and V_j. If we consider arc $C - A$ ($C < A$), then ($C = 4$) would imply that there is no suitable value for A.*

   Clearly, an arc (V_i, V_j) can be made consistent by simply deleting those values from the domain of V_i for which there does not exist corresponding value in the domain of V_j such that the constraint between V_i and V_j is satisfied.

The complete process of enforcing arc consistency is illustrated below, using the arc consistency algortihm AC-3 (illustrated in the Russell/Norvig book on p. 146). The algorithm starts with a queue of all arcs in the problem, i.e. $(A, B), (A, C), (A, D), (B, A), ...$ *For unary constraints, a node consistency rather than an arc consistency is enforced. This is simply achieved by removing the values that do not satisfy the unary constraint from the domain of the node.*

- *Node A consistency:* $A \neq 3$
$$A \in \{1, 2, 4\}$$
$$B, C, D \in \{1, 2, 3, 4\}$$
Arcs B-A,C-A,D-A are added to the queue.
- *Arc A-B consistency:* $0 < B - A \leq 2$
$$A \in \{1, 2\}$$
$$B, C, D \in \{1, 2, 3, 4\}$$
Arcs B-A,C-A,D-A are added to the queue.
- *Arc A-C consistency:* $C < A$
$$A \in \{2\}$$
$$B, C, D \in \{1, 2, 3, 4\}$$
Arcs B-A,C-A,D-A are added to the queue.
- *Arc A-D consistency:* $A \neq D$
$$A \in \{2\}$$
$$B, C, D \in \{1, 2, 3, 4\}$$
- *Arc B-A consistency:* $0 < B - A \leq 2$
$$A \in \{2\}$$
$$B \in \{3, 4\}$$
$$C, D \in \{1, 2, 3, 4\}$$
Arcs A-B,C-B,D-B are added to the queue.
- *Arc B-C consistency:* $C < B$
$$A \in \{2\}$$
$$B \in \{3, 4\}$$
$$C, D \in \{1, 2, 3, 4\}$$
- *Arc B-D consistency:* $D < B$
$$A \in \{2\}$$
$$B \in \{3, 4\}$$
$$C, D \in \{1, 2, 3, 4\}$$
- *Arc C-A consistency:* $C < A$
$$A \in \{2\}$$
$$B \in \{3, 4\}$$
$$C \in \{1\}$$
$$D \in \{1, 2, 3, 4\}$$
Arcs A-C,B-C,D-C are added to the queue.

- *Arc C-B consistency:* $C < B$

$$A \in \{2\}$$
$$B \in \{3, 4\}$$
$$C \in \{1\}$$
$$D \in \{1, 2, 3, 4\}$$

- *Arc C-D consistency:* $0 < D - C \leq 2$

$$A \in \{2\}$$
$$B \in \{3, 4\}$$
$$C \in \{1\}$$
$$D \in \{1, 2, 3, 4\}$$

- *Arc D-A consistency:* $A \neq D$

$$A \in \{2\}$$
$$B \in \{3, 4\}$$
$$C \in \{1\}$$
$$D \in \{1, 3, 4\}$$

  Arcs A-D,B-D,C-D are added to the queue.

- *Arc D-B consistency:* $D < B$

$$A \in \{2\}$$
$$B \in \{3, 4\}$$
$$C \in \{1\}$$
$$D \in \{1, 3\}$$

  Arcs A-D,B-D,C-D are added to the queue.

- *Arc D-C consistency:* $0 < D - C \leq 2$

$$A \in \{2\}$$
$$B \in \{3, 4\}$$
$$C \in \{1\}$$
$$D \in \{3\}$$

  Arcs A-D,B-D,C-D are added to the queue.

- *Arc B-A consistency:* $0 < B - A \leq 2$ *... nothing changes*
- *Arc C-A consistency:* $C < A$ *... nothing changes*
- *Arc D-A consistency:* $A \neq D$ *... nothing changes*

  . . .

- *Arc B-D consistency:* $D < B$

$$A \in \{2\}$$
$$B \in \{4\}$$
$$C \in \{1\}$$
$$D \in \{3\}$$

  Thus, arc consistency can be achieved, and the solution to the problem is A=2, B=4, C=1 and D=3.

Exercise 4.3 *Entailment*

Assume that A, B and C are propositional constants.

1. Use truth tables to show that $\{A \vee B, \neg A \vee C\} \models B \vee C$
2. Does also $\{B \vee C\} \models (A \vee B) \wedge (\neg A \vee C)$ hold? Justify your answer.
3. Does $\{\neg(B \vee C)\} \models \neg((A \vee B) \wedge (\neg A \vee C))$ hold? Justify your answer.
4. Explain why the resolution rule preserves satisfiability.
5. Let F be a propositional formula and KB be a finite set of propositional formulas. Assume we can derive the empty clause from $KB \wedge \{\neg F\}$. Explain why we can conclude $KB \models F$

Solution:

The truth table for that problem is

A	B	C	$A \vee B$	$\neg A \vee C$	$KB (A \vee B) \wedge (\neg A \vee C)$	$B \vee C$	
true	*true*	*true*	*true*	*true*	*true*	*true*	$*$
true	*true*	*false*	*true*	*false*	*false*	*true*	$\circ$
true	*false*	*true*	*true*	*true*	*true*	*true*	$*$
true	*false*	*false*	*true*	*false*	*false*	*false*	$+$
false	*true*	*true*	*true*	*true*	*true*	*true*	$*$
false	*true*	*false*	*true*	*true*	*true*	*true*	$*$
false	*false*	*true*	*false*	*true*	*false*	*true*	$\circ$
false	*false*	*false*	*false*	*true*	*false*	*false*	$+$

Checking all lines, where the knowledge base is true, then $B \vee C$ is also true (see all the lines marked with $*$). Thus, $\{A \vee B, \neg A \vee C\} \models B \vee C$ holds.

1. *Do the check by checking when $B \vee C$ holds, then the KB is true. This fails for the lines marked by $\circ$.*
2. *Negate the values in the columns KB and $B \vee C$ above, and check whenever $\neg(B \vee C)$ is true, then so is $\neg(KB)$. This are the lines marked by $+$.*
3. *Consider the second question above. It is a simplified form of resolution and shows for any A, B, and C, that if the knowledge base $\{A \vee B, \neg A \vee C\}$ is satisfiable, then so is the resolvent $B \vee C$. Hence, if KB is satisfiable, then so is $KB \cup \{B \vee C\}$.*
4. *The resolution rule is*
$$\frac{l_1 \vee ... \vee l_k \quad m_1 \vee ... \vee m_n}{l_1 \vee ... l_{i-1} \vee l_{i+1} \vee ... \vee l_k \vee m_1 \vee ... m_{j-1} \vee m_{j+1} \vee ... \vee m_n},$$
where l_i and m_j are complementary literals.
For the proof, we assume that the premise is satisfiable, i.e.that there is a model M such that $M \vDash l_1 \vee ... \vee l_k$ and $M \vDash m_1 \vee ... \vee m_n$. We have two cases, either $M \vDash l_i$ or $M \nvDash l_i$. In the first case, since l_i and m_j are complementary, $M \nvDash m_j$. Therefore $M \vDash m_{j-1} \vee ... m_{j-1} \vee m_{j+1} \vee ... \vee m_n$ holds, and thus $M \vDash l_1 \vee ... l_{i-1} \vee l_{i+1} \vee ... \vee l_k \vee m_1 \vee ... m_{j-1} \vee m_{j+1} \vee ... \vee m_n$ In the second case, since $M \nvDash l_i$, it holds that $M \vDash l_1 \vee ... l_{i-1} \vee l_{i+1} \vee ... \vee l_k$, and again, $M \vDash l_1 \vee ... l_{i-1} \vee l_{i+1} \vee ... \vee l_k \vee m_1 \vee ... m_{j-1} \vee m_{j+1} \vee ... \vee m_n$. Therefore, the conclusion of the resolution rule is satisfiable.
5. *The empty clause is unsatisfiable, and so is any set of clauses that contains the empty clause. Thus by resolution we have derived from $KB \wedge \neg F$ an unsatisfiable knowledge base. Since resolution preserves satisfiability (see before), we know conversely that $KB \wedge \neg F$ must have been unsatisfiable, i.e. $\not\models KB \wedge \neg F$.*

$$\begin{aligned} \not\models KB \wedge \neg F \quad &\Leftrightarrow \quad \not\models \neg(KB \Rightarrow F) \\ &\Leftrightarrow \quad \models KB \Rightarrow F \\ &\overset{\text{Deduction Theorem}}{\Leftrightarrow} \quad KB \models F \end{aligned}$$

Exercise 4.4 *Resolution*

The unicorn is a mammal if it is horned. If the unicorn is either immortal or a mammal, then it is horned. If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal.

1. Encode these statements in propositional logic.
2. Use resolution to prove that the unicorn is a mammal.

Solution:

1. $Horned \Rightarrow Mammal$
2. $(Immortal \vee Mammal) \Rightarrow Horned$
3. $Mythical \Rightarrow Immortal$
4. $\neg Mythical \Rightarrow (\neg Immortal \wedge Mammal)$

The conversion to CNF leads to the following clauses:

1. $\neg Horned \vee Mammal$
2. (a) $\neg Immortal \vee Horned$
   (b) $\neg Mammal \vee Horned$
3. $\neg Mythical \vee Immortal$
4. (a) $Mythical \vee \neg Immortal$ (b) $Mythical \vee Mammal$
5. $\neg Mammal$ *(the negated conclusion)*

   Resolution now yields the following inferences:

6. $\neg Hormed (Res 1 \& 5)$
7. $Mythical (Res 4(b) \& 5)$
8. $Immortal (Res 7 \& 3)$
9. $\neg Immortal (Res 6 \& 2(a))$
10. $\square (Res 8 \& 9)$