DM534 — Øvelser Uge 46

Introduktion til Datalogi, Efterår 2021

Jonas Vistrup

---

1 I

1.1

Which of the following formulas are satisfiable (give a satisfying assignment)? Which are not (give reasons)?

(a) A ∧ B. **SVAR:** Satisfiable, A=t B=t.

(b) A ∨ B. **SVAR:** Satisfiable, A=t B=t.

(c) A ⇒ B. **SVAR:** Satisfiable, A=t B=t.

(d) A ∧ -A. **SVAR:** Unsatisfiable, as any assignment has either A=t or A=f, and the expression is false in either case.

(e) A ∨ -A. **SVAR:** Satisfiable, A=t.

1.2

Two formulas are equivalent, if the same assignments satisfy both of them.

Which of the following formulas are equivalent?

(a) -A ∧ B.

(b) -A ∨ B.

(c) A ⇒ B.

(d) (A ⇒ B) ∧ (-B ⇒ A).

(e) (-A ⇒ B) ∧ (-B ⇒ -A).

SVAR: (b) is equivalent to (c), (d) is equivalent tod (e).

1.3

Convert the following formulas into CNF:

(a) -A ∧ B. **SVAR:** (-A) ∧ (B).

(b) -A ∨ B. **SVAR:** (-A ∨ B).

(c) A ⇒ B. **SVAR:** (-A ∨ B).

(d) (A ⇒ B) ∧ (-B ⇒ A). **SVAR:** (-A ∨ B) ∧ (B ∨ A).

(e) A ⇒ (-(B ∨ D)). **SVAR:** (-A ∨ -B) ∧ (-A ∨ -D)

1.4

Write two clauses that forbid solutions where there is a queen in the right half of the first row.

SVAR: $(\neg X_{1,3}) \wedge (\neg X_{1,4})$

1.5

Run SAT solver.

1.6

The formula from Slide 11 contains redundant information. For example, $X_{1,1} \Rightarrow -X_{1,2}$ and $X_{1,2} \Rightarrow -X1,1$ are equivalent. Understand and remove these redundancies:

(a) Why do these redundancies occur?

(b) Identify all such redundancies!

(c) Write down a simplified formula without redundancies!

(d) Convert the simplified formula into CNF!

(e) Write the formula in DIMACS format!

(f) Run the lingeling solver on it and interpret the result!

SVAR a: A tower in spot x might exclude a tower in spot y, and the same is true reverse, but both of these statements are equivalent to stating, that a tower can't be in both x and y.

SVAR b:

- $X_{1,1} \Rightarrow -X_{1,2}$ og $X_{1,2} \Rightarrow -X_{1,1}$.
- $X_{1,1} \Rightarrow -X_{2,1}$ og $X_{2,1} \Rightarrow -X_{1,1}$.
- $X_{1,2} \Rightarrow -X_{2,2}$ og $X_{2,2} \Rightarrow -X_{1,2}$.
- $X_{2,1} \Rightarrow -X_{2,2}$ og $X_{2,2} \Rightarrow -X_{2,1}$.

SVAR c:

$$(X_{1,1} \Rightarrow -X_{1,2}) \wedge (X_{1,1} \Rightarrow -X_{2,1}) \wedge (X_{1,2} \Rightarrow -X_{2,2}) \wedge (X_{2,1} \Rightarrow -X_{2,2})$$
$$\wedge (X_{1,1} \vee X_{1,2}) \wedge (X_{2,1} \vee X_{2,2})$$

SVAR d:

$$(-X_{1,1} \vee -X_{1,2}) \wedge (-X_{1,1} \vee -X_{2,1}) \wedge (-X_{1,2} \vee -X_{2,2}) \wedge (-X_{2,1} \vee -X_{2,2})$$
$$\wedge (X_{1,1} \vee X_{1,2}) \wedge (X_{2,1} \vee X_{2,2})$$

SVAR e: p cnf 4 6   -1 -2 0   -1 -3 0   -2 -4 0   -3 -4 0   1 2 0   3 4 0

SVAR f: Did in 1.5. Got 1, -2, -3, 4. This translates to a tower on $X_{1,1}$ and $X_{2,2}$.

2 II

2.1

Which of the following formulas are satisfiable (give a satisfying assignment)? Which are not (give reasons)?

(a) (A ⇒ B) ∧ (B ⇒ A). **SVAR:** Satisfiable, A=t B=t.

(b) (A ⇒ B) ∧ (B ⇒ A) ∧ A. **SVAR:** Satisfiable, A=t B=t.

(c) (A ⇒ B) ∧ (B ⇒ A) ∧ -A. **SVAR:** Satisfiable, A=f B=f.

(d) (A ⇒ B) ∧ (B ⇒ -A) ∧ (-A ⇒ -B) ∧ (-B ⇒ A). **SVAR:** Unsatisfiable. Argument: The first part (A ⇒ B) must be true, which excludes the assignment A=t B=f. Similarly, the next three parts exclude the assignments A=t B=t, A=f B=t, and A=f B=f, respectively. Hence, no assignment of A og B can make the formula true.

2.2

Two formulas are equivalent, if the same assignments satisfy both of them.

Which of the following formulas are equivalent?

(a) (A ⇒ B) ∧ (-B ⇒ A).

(b) (A ⇒ -B) ∧ (B ⇒ A).

(c) (-A ∨ -B) ∧ (A ∨ -B).

(d) (B ∨ A) ∧ (-A ∨ B).

SVAR: (a) is equivalent to (d) and (b) is equivalent to (c).

2.3

Convert the following formulas into CNF:

(a) (-A ⇒ B) ∧ (-B ⇒ -A). **SVAR:** (A ∨ B) ∧ (B ∨ -A).

(b) A ⇒ (-(B ∧ D)). **SVAR:** (-A ∨ -B ∨ -D)

(c) A ⇒ (-(B ⇒ (C ∧ D))). **SVAR:** (-A ∨ B) ∧ (-A ∨ -C ∨ -D).

2.4

Write a Java program that generates the input for a SAT solver to solve the 3-Towers problem:

(a) Write a method **pair2int(int r, int c)**. Should map $(1,1), (1,2), ..., (3,3)$ to 1 to 9 using the formula $3 \cdot (r-1) + c$.

(b) Write nested for-loops that go through all positions on the board from (1,1) to (3,3) and produces clauses that represent attacks.

(c) Write a for-loop that produces clauses that specify that all 3 rows contain a tower.

(d) Using (a)–(c), write a DIMACS file and test it using lingeling.

2.5

Generalize your Java program from Exercise II-4 to generate the input for a SAT solver to solve the N-Towers problem:

(a) The method **pair2int(int n,int r,int c)** should map pairs (r,c) to the integers 1 to n^2 using the formula $n \cdot (r-1)+c$.

(b) Write nested for-loops that go through all positions on the board from (1,1) to (n,n) and produces clauses that represent attacks.

(c) Write a for-loop that produces clauses that specify that all rows contain a tower.

(d) Using (a)–(c), write a DIMACS file and test it using lingeling.

2.6

Extend your Java program from Exercise II-5 to generate the input for a SAT solver to solve the N-Queens problem:

(a) Reuse your function pair2int(n,r,c) from Exercise II-5.

(b) Adapt your for-loops from Exercise II-5 to produce also clauses for the diagonals.

(c) Reuse the for-loop from Exercise II-5 that produces clauses that specify that all rows contain a tower.

(d) Using (a)–(c), write a DIMACS file and test it using lingeling.