

Satisfiability Checking - WS 2023/2024

Series 5

teaching@ths.rwth-aachen.de
https://ths.rwth-aachen.de/teaching/

Exercise 1

- Give a formula describing the Unequal game instance of Series 3 Exercise 2 in *equality logic with uninterpreted functions*. Remember, that the formula shall be satisfiable iff the game instance has a solution. You must not use propositional variables in your solution!
- Compare the resulting formula to the propositional encoding. More precisely, compare the number of literals and clauses using the big \mathcal{O} notation. Draw a conclusion.

Solution:

- In the following, $g_{i,j}$ ($1 \leq i, j \leq n$) is an uninterpreted variable and we introduce the set of constants $N := \{1, \dots, n\}$. Then $g_{i,j}$ represents the number in the grid at the coordinates (i, j) . We define the uninterpreted function $greater(r, s)$ on $N \times N$ mapping to the constants T and F .

$$\varphi_{grid} := \bigwedge_{i=1}^n \bigwedge_{j=1}^n \bigvee_{k=1}^n g_{i,j} = k$$

$$\varphi_{row} := \bigwedge_{i=1}^n \bigwedge_{j=1}^n \bigwedge_{k=j+1}^n g_{i,j} \neq g_{i,k}$$

$$\varphi_{column} := \bigwedge_{j=1}^n \bigwedge_{i=1}^n \bigwedge_{k=i+1}^n g_{i,j} \neq g_{k,j}$$

$$\varphi_{>} := \bigwedge_{\substack{r=1 \\ s=1 \\ r > s}}^n greater(r, s) = T \quad \wedge \quad \bigwedge_{\substack{r=1 \\ s=1 \\ r \leq s}}^n greater(r, s) = F$$

Hence, we get the formula representing the game instance of Series 3 Exercise 2:

$$\begin{aligned} \varphi^{UF} := & g_{1,1} = 3 \quad \wedge \quad g_{4,1} = 1 \quad \wedge \quad greater(g_{1,2}, g_{1,3}) = T \\ & greater(g_{2,3}, g_{2,2}) = T \quad \wedge \quad greater(g_{1,4}, g_{2,4}) = T \\ & greater(g_{3,4}, g_{2,4}) = T \quad \wedge \quad greater(g_{4,3}, g_{4,4}) = T \\ & \varphi_{grid} \quad \wedge \quad \varphi_{row} \quad \wedge \quad \varphi_{column} \quad \wedge \quad \varphi_{>} \end{aligned}$$

- By comparing the literals/constraints and clauses of the result of part a) with the result of Series 3 Exercise 2 you see:

- As the number of variables in the formula using propositional logic of Series 3 Exercise 2 has been n^3 , the number of literals is:

$$\mathcal{O}(n^3)$$

- The number of clauses in the propositional formula is:

$$\mathcal{O}(n^2) + 2 \cdot \mathcal{O}(n^4) + 5 \cdot \mathcal{O}(n^2) + 2 = \mathcal{O}(n^4)$$

- The number of literals (different equations) in φ^{UF} is:

$$3 \cdot \mathcal{O}(n^3) + \mathcal{O}(n^2) + 5 + 2 = \mathcal{O}(n^3)$$

- The number of clauses in φ^{UF} is:

$$\mathcal{O}(n^2) + 2 \cdot \mathcal{O}(n^3) + \mathcal{O}(n^2) + 5 + 2 = \mathcal{O}(n^3)$$

Conclusion: The formula using propositional logic needs less literals but $\mathcal{O}(n)$ times more clauses than the formula using equality logic with uninterpreted functions.

Exercise 2

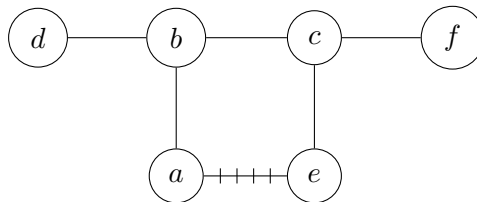
Consider the following formula in equality logic:

$$\varphi := a = b \wedge (b = c \vee c = e) \wedge (b = d \vee c = f) \wedge a \neq e$$

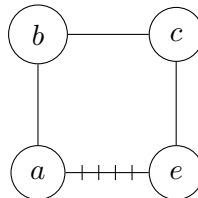
- Simplify the formula φ using the method presented in the lecture, based on polar equality graphs (slides 37-38).
- For the simplified formula, construct the equality graph without polarity and make it chordal. What are the chord-free simple cycles?
- Construct the satisfiability-equivalent propositional logic formula for φ using the previous results from b).

Solution:

- The equality graph with polarity for φ is given by:



The edges (b, d) and (c, f) are not part of any contradictory cycle, thus we can remove them and replace in the formula both $b = d$ and $c = f$ by *true*. We get the following simplified equality graph:

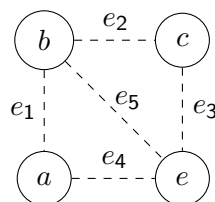


原prop需要先simplify为non-polarity cycle, 再进行后续步骤

The simplified formula is

$$a = b \wedge (b = c \vee c = e) \wedge a \neq e$$

- A chordal completion of the simplified equality graph without polarity is:



The chord-free simple cycles are (a, b, e, a) and (b, c, e, b) .

c) The satisfiability-equivalent propositional logic formula for φ using the previous results is:

$$\varphi^{prop} := \varphi_{sk} \wedge \varphi_{trans}$$

$$\varphi_{sk} := e_1 \wedge (e_2 \vee e_3) \wedge \neg e_4$$

$$\begin{aligned} \varphi_{trans} := & ((e_1 \wedge e_4) \rightarrow e_5) \\ & \wedge ((e_1 \wedge e_5) \rightarrow e_4) \\ & \wedge ((e_4 \wedge e_5) \rightarrow e_1) \\ & \wedge ((e_2 \wedge e_3) \rightarrow e_5) \\ & \wedge ((e_2 \wedge e_5) \rightarrow e_3) \\ & \wedge ((e_3 \wedge e_5) \rightarrow e_2) \end{aligned}$$