Difference Logic

Satisfiability Checking Seminar

Alex Ryndin
Supervisor: Gereon Kremer

WS 2016/2017

Outline

Main Literature

- Scott Cotton, Eugene Asarin, Oded Maler and Peter Niebert. **"Some progress in satisfiability checking for difference logic"**. In Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, pages 263–276. Springer, 2004.
- Andrew V. Goldberg and Tomasz Radzik. **"A heuristic improvement of the Bellman-Ford algorithm"**. Applied Mathematics Letters, 6(3):36, 1993.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. **"Introduction to algorithms"**. MIT press, third edition, 2009.
  Note: the chapter 24 **"Single-Source Shortest Paths"** is relevant for the topic.

Difference Logic and Why It Is Important

- Difference logic (DL) is a special case of linear arithmetic (LA) logic.
- It is a Propositional Logic (PL) enhanced with constraints of the following form:

$$x - y \prec c \tag{1}$$

  where x, y are variables, c is a constant and $\prec \in \{<, \leq\}$ is a comparison operator.
- x, y, c can be defined either over Integers $\mathbb{Z}$ or over Reals $\mathbb{R}$.

Difference Logic and Why It Is Important

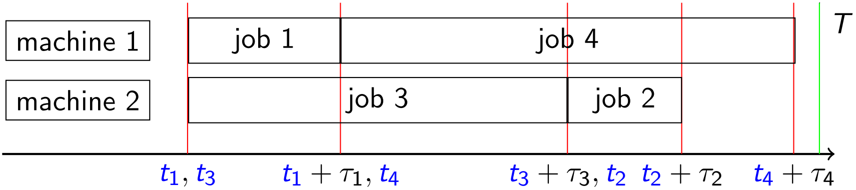
- A couple of examples:

$$\phi_1 = (p \vee q) \wedge (p \rightarrow (u - v < 3.3)) \wedge (q \rightarrow (v - w < -5.15)) \quad (2)$$

$$\begin{aligned} \phi_2 = {} & (u - v < 1) \wedge (v - w < 5) \\ & \wedge (w - x \leq -3) \wedge (x - y < 1) \\ & \wedge (y - z \leq -5) \wedge (y - v \leq 0) \end{aligned} \quad (3)$$

$$\begin{aligned} \phi_3 = {} & (u - v < 1) \wedge (v - w < 5) \\ & \wedge (w - x \leq -3) \wedge (x - y < -3) \\ & \wedge (y - z \leq -5) \wedge (y - w < 4) \end{aligned} \quad (4)$$

Scheduling Problem Example



- $p_{mj} = True$ if job j is scheduled on machine m:
  e.g. $p_{11} = p_{14} = p_{23} = p_{22} = True$ for the figure above
- job i starts at t_i and lasts τ_i
- a machine cannot process two or more jobs simultaneously:
  $(p_{mi} \wedge p_{mj}) \rightarrow ((t_i + \tau_i \leq t_j) \vee (t_j + \tau_j \leq t_i)) \quad \Leftrightarrow$
  $(p_{mi} \wedge p_{mj}) \rightarrow ((t_i - t_j \leq -\tau_i) \vee (t_j - t_i \leq -\tau_j))$
- the overall processing time should not exceed T:
  $t_i + \tau_i \leq T \quad \Leftrightarrow \quad t_i - 0 \leq T - \tau_i$

Scheduling Problem Example

$$\phi = \bigwedge_{j=1}^{4}(p_{1j} \vee p_{2j}) \quad \wedge$$

each task is executed on at least one machine

$$\bigwedge_{j=1}^{4}((p_{1j} \rightarrow \neg p_{2j}) \wedge (p_{2j} \rightarrow \neg p_{1j})) \quad \wedge$$

each task can be scheduled on one machine only

$$\bigwedge_{j=1}^{4}(t_j \geq 0) \ \wedge \ \bigwedge_{j=1}^{4}(t_j \leq T - \tau_j) \quad \wedge$$

general time constraints

$$\bigwedge_{m=1}^{2}\bigwedge_{i=1}^{3}\bigwedge_{j=i+1}^{4}((p_{mi} \wedge p_{mj}) \rightarrow ((t_i - t_j \leq -\tau_i) \vee (t_j - t_i \leq -\tau_j)))$$

no time overlap rule

SAT Checking of Propositional Logic

ttt

SAT Checking of Difference Logic

ttt

Constraint Graph

ttt

Negative Cycles in a Constraint Graph

ttt

How to Find a Negative Cycle in a Graph

ttt

Conclusion

- a
- b
- c

Thank you for your attention