

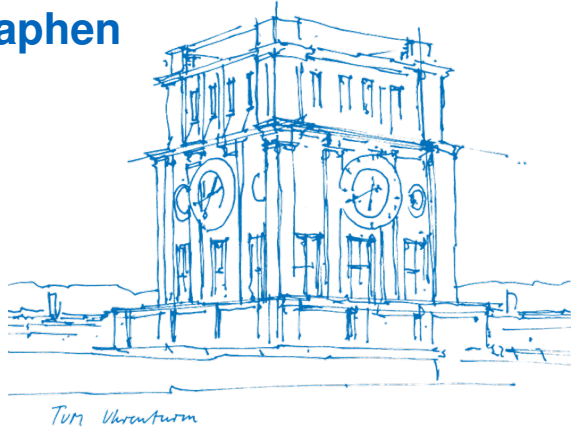
Übung 12: And-Inverter-Graphen und SAT-Solving

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

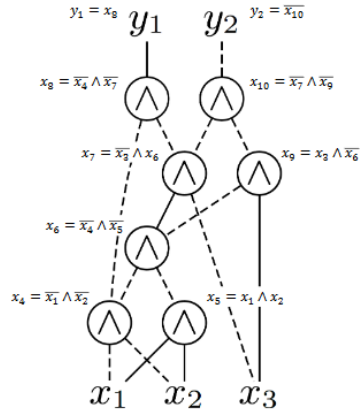
21. Januar 2024



Durchzählen!

Keine Garantie für die Richtigkeit der Tutorfolien: Bei Unklarheiten/Unstimmigkeiten haben VL/ZÜ-Folien Recht!

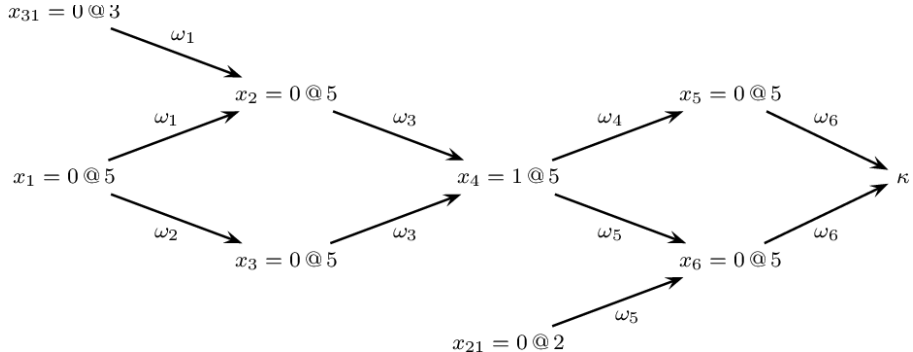
- $\{\wedge, \neg\}$ ist funktional vollständig \rightarrow NAND-Logik
- geringere Kosten für Darstellung einer booleschen Funktion
- AIG: Knoten stehen für Verundung beider Inputs, Output-Linie gestrichelt/durchgezogen für Negation/keine Negation
- x_0 ist reserviert für Konstante 0
- im Bild sind Inputs unten, Outputs oben
- algebraische Redundanzen siehe VL



Quelle: Vorlesungsmaterialien ERA

- Satisfiability \rightarrow Erfüllbarkeit einer booleschen Funktion feststellen
- Viele Probleme können durch Abbildung auf KNF in ein SAT-Problem überführt werden
- DPLL-Algorithmus (bekannt aus DS) zur Lösung von KNF-SAT:
 1. Wähle Variable x_i und belege sie in allen Klauseln ω_i
 2. Kann die Belegung anderer Variablen dadurch abgeleitet werden? (Propagation)
 3. Bei Konflikten aus Implikationsgraph „gelernte Klausel“ ableiten, dann Backtracking
 4. So lange fortführen bis erfüllende Belegung gefunden

Implikationsgraphen (Konfliktanalyse)



Quelle: Conflict-Driven Clause Learning SAT Solvers, Joao Marques-Silva and Inês Lynce and Sharad Malik

gelernte Klausel: $\omega_\kappa := (x_{31} + x_1 + x_{21})$

Fragen?

- H12 - Mastermind bis 28.01.2024 23:59 Uhr
- Finden von KNFs für Mastermind-Regeln

- Zulip: „ERA Tutorium - Mi-1600-MI4“ bzw. „ERA Tutorium - Fr-1100-MW2“
- Wikipedia zu SAT

Übung 12: And-Inverter-Graphen und SAT-Solving

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

21. Januar 2024

