Lab 1 — Propositional Logic

Ondřej Lengál

Faculty of Information Technology
Brno University of Technology

IAM'19

Transformation to NNF

- NNF contains only $\wedge$, $\vee$, and literals

Transformation to NNF:

1. substitute $\leftrightarrow$ with equivalent formulae:

$$\varphi \leftrightarrow \psi \quad \Leftrightarrow \quad (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$$

2. substitute $\rightarrow$ with equivalent formulae:

$$\varphi \rightarrow \psi \quad \Leftrightarrow \quad \neg\varphi \vee \psi$$

3. push $\neg$ to atoms using De Morgan's laws and DNE:

$$\begin{aligned} \neg(\varphi \wedge \psi) \quad &\Leftrightarrow \quad \neg\varphi \vee \neg\psi \\ \neg(\varphi \vee \psi) \quad &\Leftrightarrow \quad \neg\varphi \wedge \neg\psi \\ \neg\neg\varphi \quad &\Leftrightarrow \quad \varphi \end{aligned}$$

Transformation to DNF

- DNF is NNF of the form $\bigvee_i \bigwedge_j \ell_{i,j}$

Transformation to DNF:

1. transform to NNF
2. use distributive law:

$$F \wedge (G \vee H) \quad \Leftrightarrow \quad (F \wedge G) \vee (F \wedge H)$$

Transformation to CNF

- CNF is NNF of the form $\bigwedge_i \bigvee_j \ell_{i,j}$

Transformation to CNF:

1. transform to NNF
2. use distributive law:

$$F \lor (G \land H) \quad \Leftrightarrow \quad (F \lor G) \land (F \lor H)$$

Equisatisfiability

- transformation to an equivalent CNF formula $\rightsquigarrow$ exponential size
- when testing satisfiability, an equisatisfiable formula would suffice
  - φ and ψ are equisatisfiable when φ is satisfiable iff ψ is satisfiable

- **Tseytin transformation**: linear size
  1. transform φ to ψ in NNF using the previous algorithm [linear size]
  2. transform ψ to a combinatorial circuit
     - **variables** of ψ are inputs
     - every **subformula** corresponds to a logical gate
     - an **auxiliary variable** for every gate (denotes its output)
     - **gates**: small CNF formulae relating inputs (A, B) with the output (C)
     - the result is the conjuction of formulae for all gates [still linear size]
     - one more clause is added denoting that the output value is $true$

Equisatisfiability

Type	**Operation**	**CNF expression**
NOT	$C \equiv \neg A$	$(A \vee C) \wedge (\neg A \vee \neg C)$
AND	$C \equiv A \wedge B$	$(A \vee \neg C) \wedge (B \vee \neg C) \wedge (\neg A \vee \neg B \vee C)$
OR	$C \equiv A \vee B$	$(\neg A \vee C) \wedge (\neg B \vee C) \wedge (A \vee B \vee \neg C)$

Table: Correct behaviour of gates (A, B are inputs, C is output, Z denotes when the gate is operating correctly)

NOT		
A	C	Z
0	0	0
0	1	1
1	0	1
1	1	0

AND			
A	B	C	Z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

OR			
A	B	C	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1