

# **Automata and Logic Engineering 1**

## **(ALE1)**

**Feb 2019**

# Assignments

## 1: parse + tree

Due Feb 21 at 12:45pm

---

## 2: truth table + hash code

Due Feb 28 at 12:45pm

---

## 3: simplify

Due Mar 14 at 12:45pm

---

## 4: normalize

Due Mar 21 at 12:45pm

---

## 5: nandify

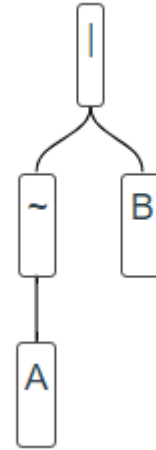
Due Mar 31 at 11:59pm

Final deadline

## Last time

- Normalization

Input:  $|(\sim(A), B)$



A	B	Result
0	0	1
0	1	1
1	0	0
1	1	1

DNF:  $|(|(\&(\sim(A), \sim(B)), \&(\sim(A), B)), \&(A, B))$

# NAND

- **NAND (negative-AND)**
- Operator (logic gate) which produces an output which is **false** only if all its inputs are **true**

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

# NAND

- Created only with either the connectives  
~ NOT    & AND

formula	becomes
$A \Rightarrow B$	$\neg A \vee B$
$A \Leftrightarrow B$	$(\neg A \wedge \neg B) \vee (A \wedge B)$
$A \vee B$	$\neg(\neg A \wedge \neg B)$

# Nandify

formula	becomes
$A \Rightarrow B$	$\neg A \vee B$
$A \Leftrightarrow B$	$(\neg A \wedge \neg B) \vee (A \wedge B)$
$A \vee B$	$\neg(\neg A \wedge \neg B)$
$\sim A$	?
$A \& B$	?

# Nandify

$$\neg( \dots \wedge \dots ) = \%( \dots , \dots )$$

formula	becomes	NAND
$A \Rightarrow B$	$\neg A \vee B$	$\%(A, \%(B, B))$
$A \Leftrightarrow B$	$(\neg A \wedge \neg B) \vee (A \wedge B)$	$\%( \%( \%(A, A), \%(B, B)), \%(A, B) )$
$A \vee B$	$\neg(\neg A \wedge \neg B)$	$\%( \%(A, A), \%(B, B) )$
$\sim A$	$\neg(A \wedge A)$	$\%(A, A)$
$A \& B$	$\neg(\neg(A \wedge B))$	$\%( \%(A, B), \%(A, B) )$

# ALE1 Deadline

- 31<sup>st</sup> March (Sunday) 23:59
  - Source code
  - Executable
  - Report
    - Don't forget to brag about your code! (all extra work you did in your project and advanced implementation that I might oversee)
- Ask for extension at least 1 day before



