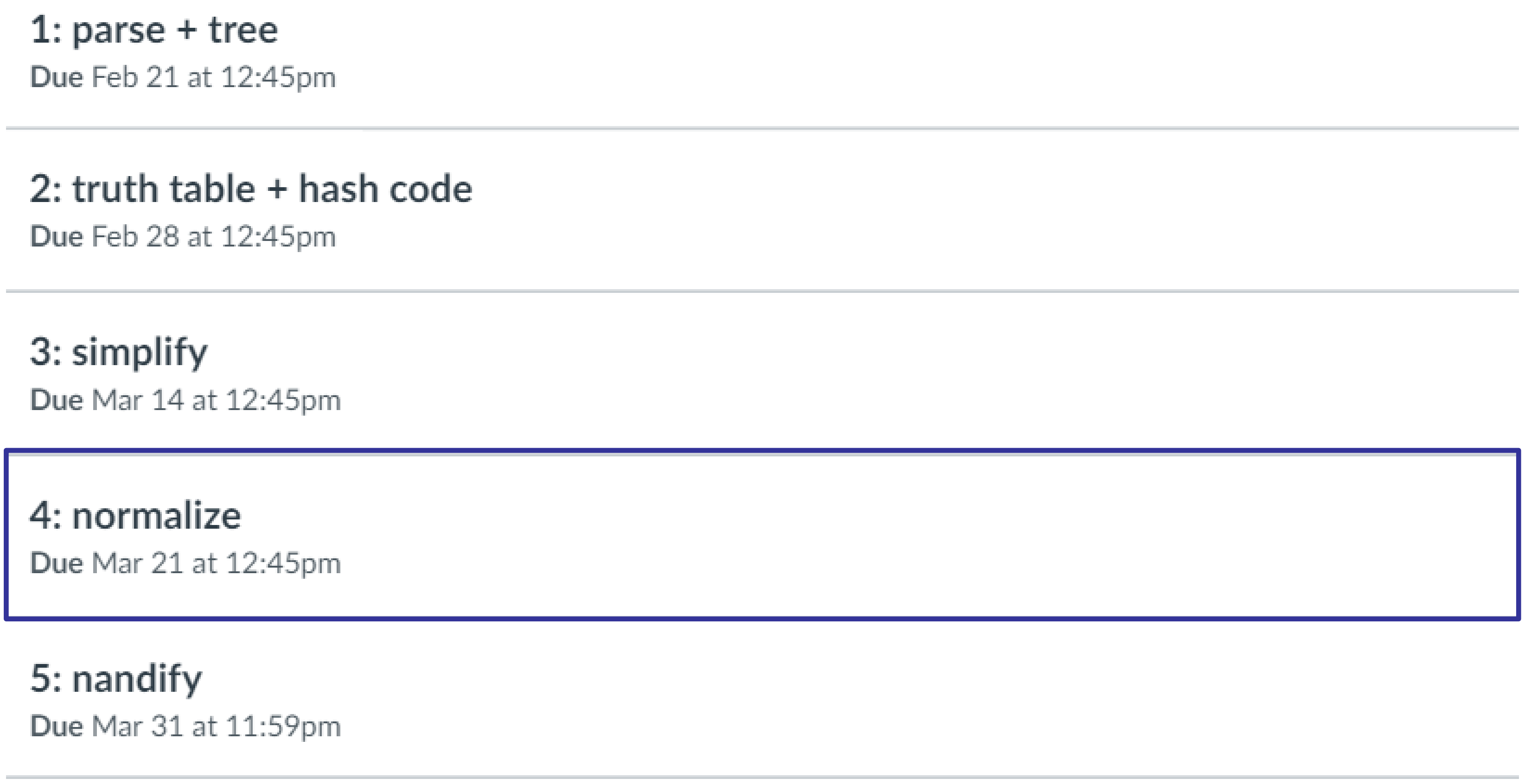
Automata and Logic Engineering 1 (ALE1)
Feb 2019

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

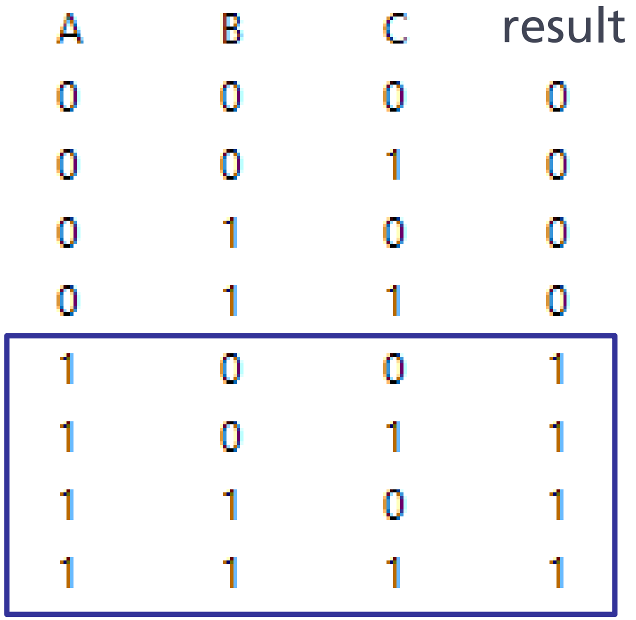
5: nandify

Due Mar 31 at 11:59pm

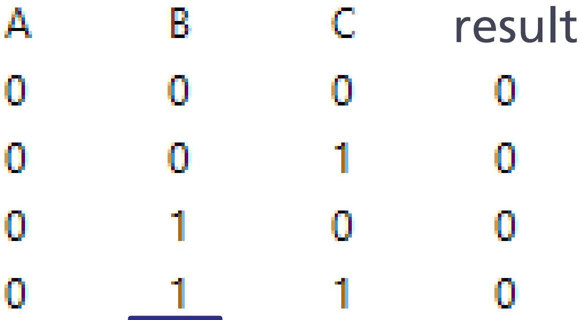
Last time

- Simplification

Simplification



A	B	C	result
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



A	B	C	result
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	*	1
1	1	*	
1	*	1	
1	*	0	

Normalization

WHY and **HOW**?

Normalization

- Or **Disjunctive Normal Form (DNF)** or **standardization**
- Is a **disjunction** of **conjunctive** clauses
  - *disjunction*: **A B**
  - *conjunction*:**A ∧ B**

Example

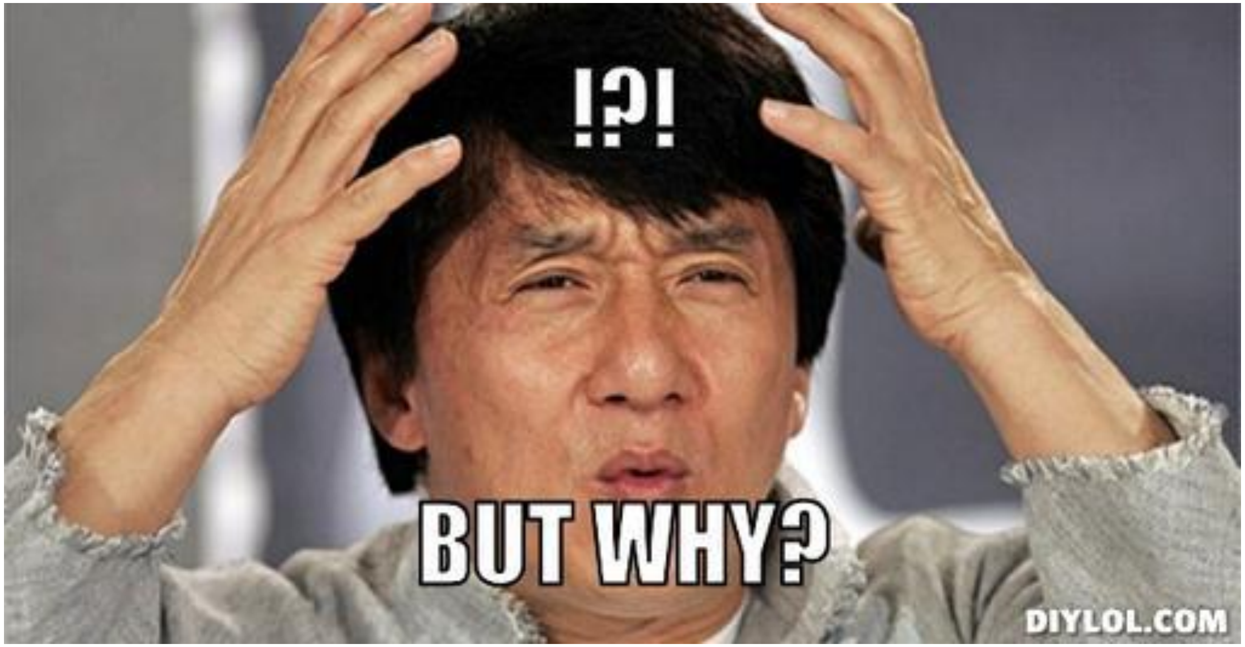
Logic expression:

$$(\neg(A \Rightarrow B)) \lor (A \land (C \Rightarrow B))$$

DNF:

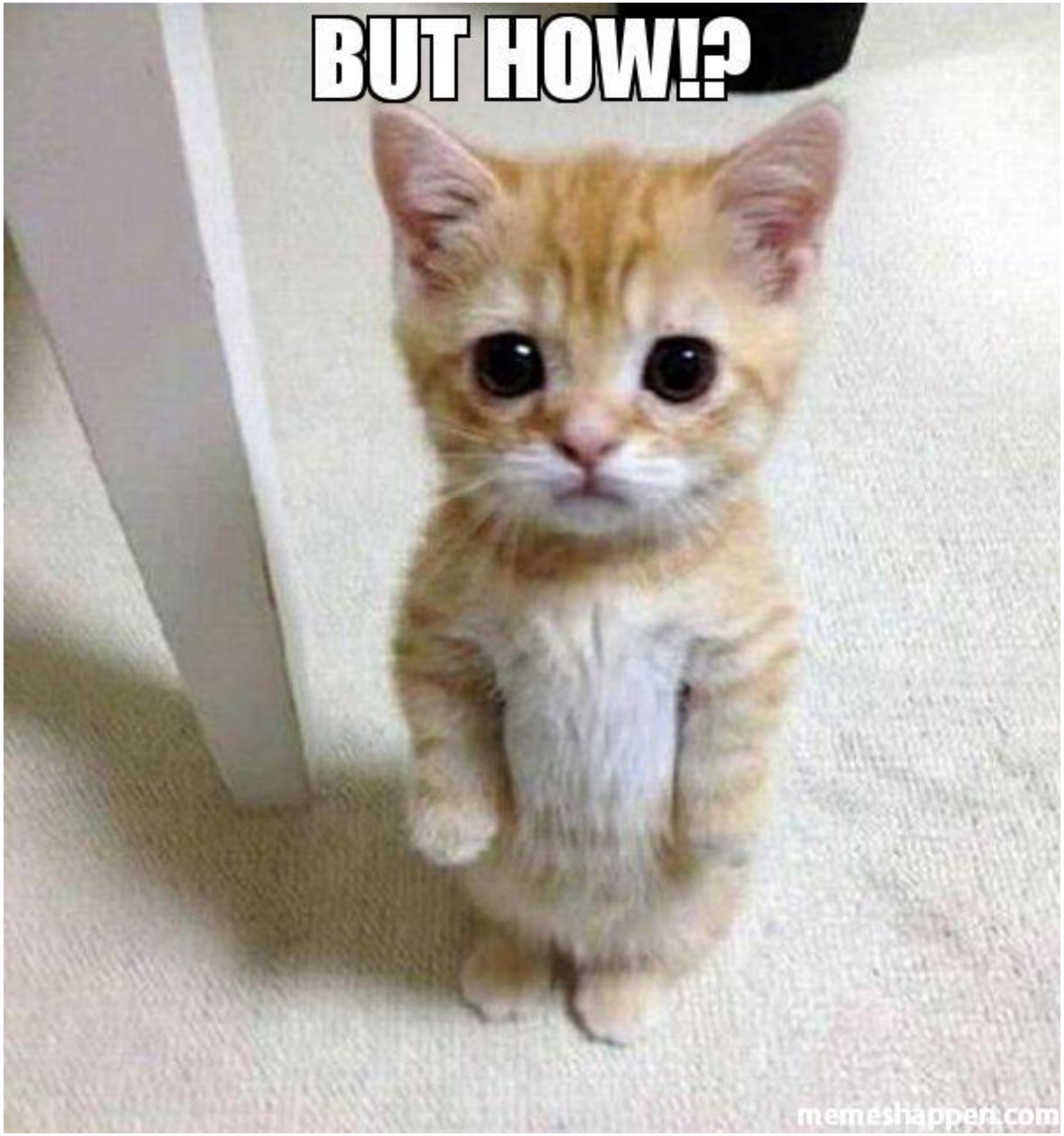
$$(A \land \neg B \land \neg C) \lor (A \land \neg B \land C) \lor (A \land B \land \neg C) \lor (A \land B \land C)$$

Normalization



Normalization - WHY?

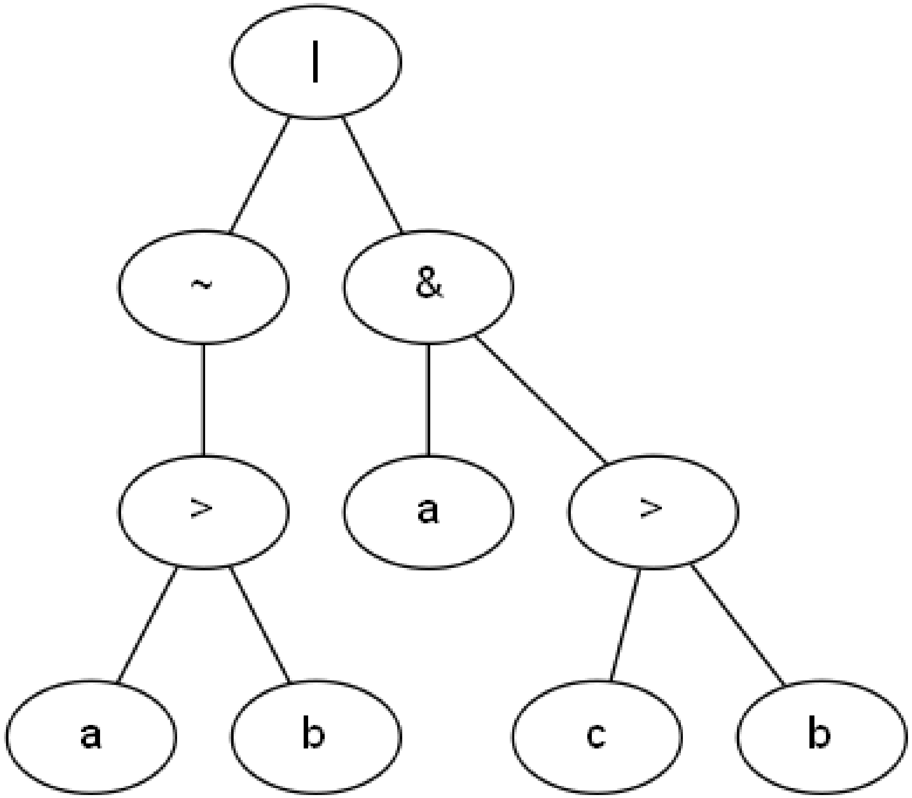
- Easier for **programmers** to understand and maintain
  - Simple syntax
  - Fewer parentheses
    - operator precedence: ¬ ,∧, ∨
    - omitting the parentheses is not an error in this form


BUT HOW!?
memeshappen.com

So far…

$(\neg(A\Rightarrow B))\vee(A\wedge(C\Rightarrow B))$

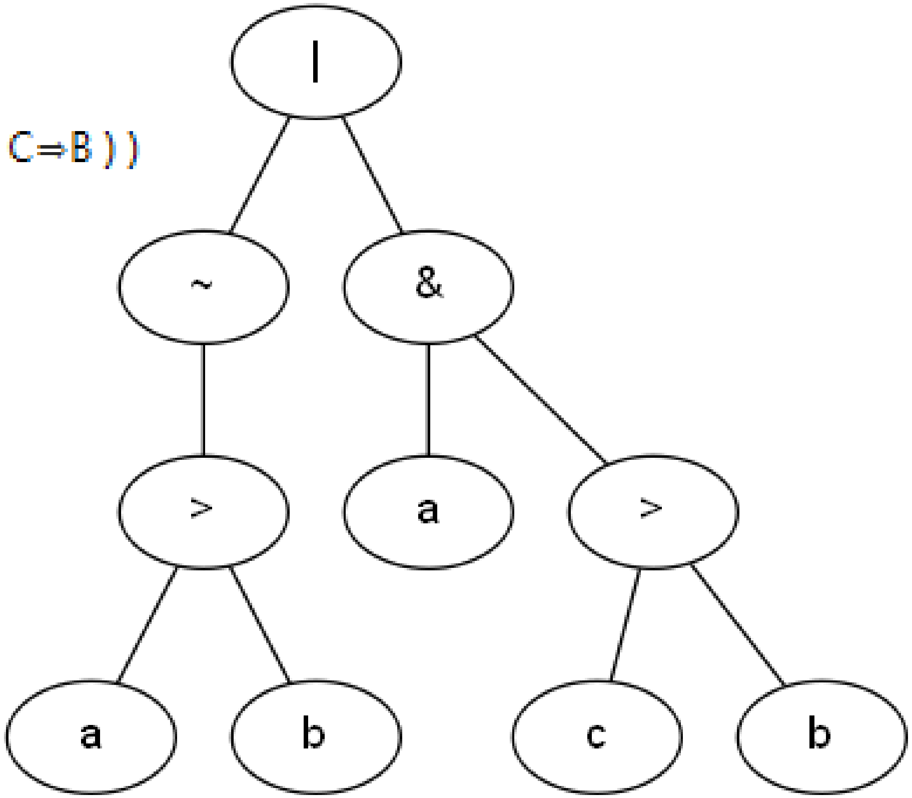
|(~(>(A,B)),&(A,>(C,B)))



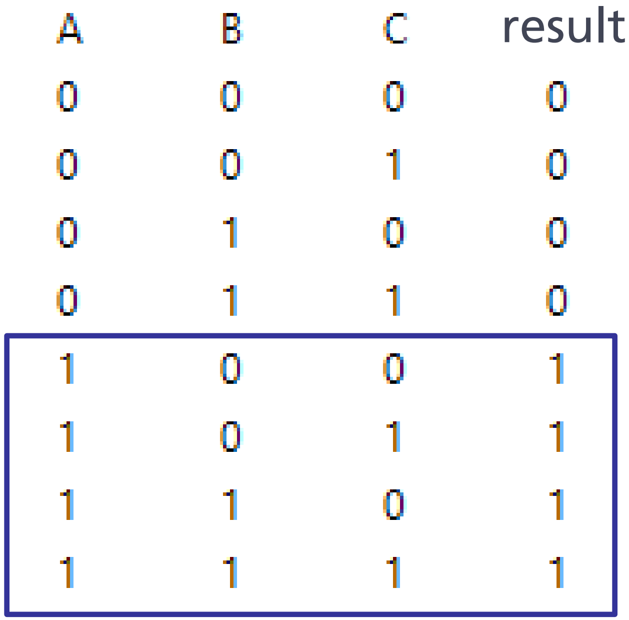
So far…

(¬(A⇒B))V(A⋀(C⇒B))

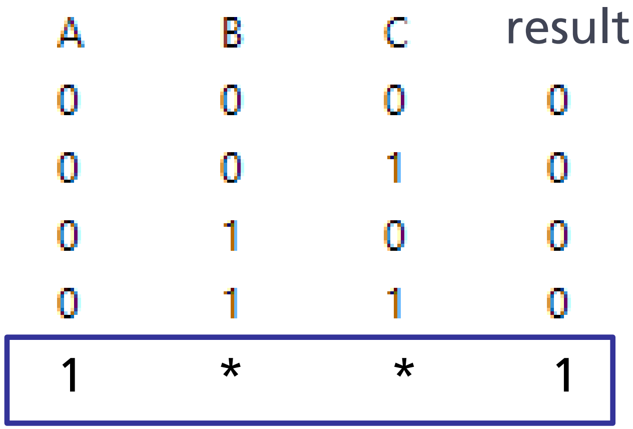
A	B	C	(¬(A⇒B))V(A⋀(C⇒B))
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



So far…



A	B	C	result
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



A	B	C	result
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	*	*	1

*' is not '0' and not '1'

Recap

- Normalization
- In your code?

Fontys