Knowledge Representation and Reasoning

Exercise Session 1

Exercise 1. Truth Tables

(*)

Build the **truth tables** of the following formulas, remembering the precedence of the operators.

- 1. $\neg(x \land y) \lor z$
- 2. $(x \land y \lor \neg x \land \neg w) \land z$
- 3. $(x \lor y) \land x$
- 4. $\neg (p \land (\neg p \lor q \lor s)) \lor q \lor s$

Exercise 2. Boolean Functions

(*)

For each of the following truth tables, build a formula expressing the same Boolean function.

x	y	z	φ_1	_	\boldsymbol{x}	z	φ_2		x	y	z	w	φ_3
0	0	0	0		0	0	1		0	0	0	0	0
0	0	1	1		0	1	0		0	0	0	1	1
0	1	0	0		1	0	0		0	0	1	0	0
0	1	1	1		1	1	1		0	0	1	1	0
1	0	0	1				•		0	1	0	0	1
1	0	1	0						0	1	0	1	0
1	1	0	0						0	1	1	0	1
1	1	1	1						0	1	1	1	1
									1	0	0	0	0
									1	0	0	1	1
									1	0	1	0	0
									1	0	1	1	0
									1	1	0	0	1
									1	1	0	1	0
									1	1	1	0	0
									1	1	1	1	0

If the tables represent Boolean functions, what is their arity?

Exercise 3. Types of Formulas

(*)

Use truth tables (and, if needed the equivalences seen during the lecture) to verify whether the following formulas are tautologies, contradictions, or non-tautological satisfiable formulas.

- 1. $x \rightarrow \neg x$
- 2. $(x \to y) \land (\neg y \to \neg x)$
- 3. $(x \to y) \to (\neg y \to \neg x)$
- 4. $x \wedge y \wedge (\neg x \vee \neg y) \wedge z$

Exercise 4. NNF (**)

A formula is in **negation normal form** (NNF). For example, the second and third formulas in Exercise 1 are in NNF, while the other two formulas are not.

- 1. Using the DeMorgan equivalences, transform the formulas of Exercise 1 to NNF
- 2. Do you notice something from the constructions in Exercise 2?

Exercise 5. Sheffer Functions

(* * *)

We have seen that \neg, \land, \lor form a **functionally complete** set of logical connectives in the sense that every Boolean function can be described using those connectives. In fact, we \neg, \land and \neg, \lor are functionally complete by themselves.

A logical connective is called a **Sheffer function** or **sole sufficient operator** if it is functionally complete by itself.

- 1. show that the NAND connective (denoted as \uparrow) is a Sheffer function
- 2. are there other Sheffer functions?
- 3. could a unary connective be a Sheffer function?

Exercise 6. Knowledge Bases

(**)

For the following truth table, construct a knowledge base (expressed as a set of propositional rules) which represents the same knowledge.

\boldsymbol{x}	y	z	K
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Exercise 7. Expressivity

(**)

Can any propositional formula be expressed as a knowledge base?

If the answer is **yes**, argue why; if it is **no**, give at least two examples of formulas which cannot be expressed.

Exercise 8. Reasoning

(*)

Consider the following knowledge base K:

$$\begin{array}{l} x \leftarrow \\ y \leftarrow x, \ z, \ w \\ x \leftarrow v \\ w \leftarrow y, \ z \\ z \leftarrow v, \ x \\ z \leftarrow y, \ w \\ z \leftarrow u, \ x \\ u \leftarrow \\ p \leftarrow \\ t \leftarrow w, \ u \\ r \leftarrow s, \ t \end{array}$$

- 1. Compute the redux \hat{K}
- 2. Find all the facts that are entailed by K
- 3. Decide whether the following clauses are consequences of K
 - a) $v \leftarrow u$
 - b) $t \leftarrow y$
 - c) $q \leftarrow q$
 - d) $r \leftarrow w$

Exercise 9. Revision

(**)

In the knowledge base from Exercise 8, substitute the fact $x \leftarrow$ with $u \leftarrow$. Call this new knowledge base K'.

- 1. Do your answers from Exercise 8 change?
- 2. Which fact(s) should you remove to ensure that z is **not** a consequence of K'?
- 3. If facts cannot be removed, which rules would you remove to ensure that z is not entailed?

Exercise 10. Tautologies

(* * *)

The soundness of the fact extraction algorithm seen during the lecture depends on the following property:

if $\varphi \to \psi$ and $\psi \to \xi$ are both tautologies, then $\varphi \to \xi$ is also a tautology.

Show that this property holds always in propositional logic.

7, N, V

1. $\neg(x \land y) \lor z$

x y z	γΛy	1(212)	1(x xy) VZ
1 1 1	1	0	1111 X
1.1.0	11171	0	· · · · · · · · · · · · · · · · · · ·
101	0		111111
0.1.4			1::::1
100	0		111111
010	· · · · · · · · · · · · · · · · · · ·	1	1::::1
001	· · · · 0 · · · ·		111111
000	0	1	
HIVIW		/REI MORGAN)	Tove paremesus

IS TRUE IF AT LEAST OVE IS TRUE

0111

1101

1110

1 0 11

Pas	78	7689	7PV 9V3	PA(1P V4VS)	7(PA (7PVqV3)	7(PA(1P Vq V5))Vq	1(PA(79 V9VS)) V4VS
111	0	11:	11:	1.4		111111	4
110	•	11	11:	11:		: : 1	1 1 1 1
101	0	0	11		· · · · •	0	1 1
011	1	: 1 ::::	1	0	11111	11111	<u> </u>
100	.0 .	O	O	0	1 1 1 1 1 1	1. 1. 1. 1. 1. 1.	
010	1	111	1 1		1 1 1 1 1 1		11:
001	4	11:	11:	.0		· · • · · · · ·	. 1
000	1	1.1.	[1]		1111	: : 1 : : : :	1111

yz φ_1 x0 0 0 0 (17 1 74 12) V (1x 12) 1 (27 4) 0 1 1 (1x 1 y 12) V ARITY 3 (x / -y / 12) / { X / (1 / / -Z) / (Y / 2) 1 1 0 1 1 0 0 (XAYAZ) (1xx72) V or both face φ_2

0

0

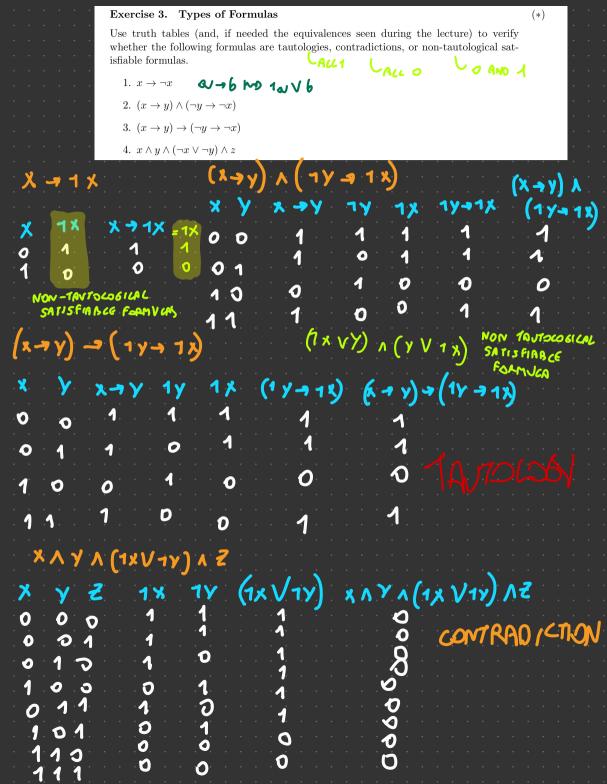
0 1

(WA 35 AVFA XF) (1×1×1×1×1×1) V (7XXY N & NW) V ARITY-4 S TAKES N ARGUMEND (XX7YX72XV)V

(W TASTA YAK)

ARITY 2

ARITY: MUMBER OF PARAMETERS THAT THE FUNLTAN HAVE TO TAKE IN CONSIDERATION





Exercise 4. NNF (**)

A formula is in **negation normal form** (NNF). For example, the second and third formulas in Exercise 1 are in NNF, while the other two formulas are not.

- 1. Using the DeMorgan equivalences, transform the formulas of Exercise 1 to NNF
- 2. Do you notice something from the constructions in Exercise 2?



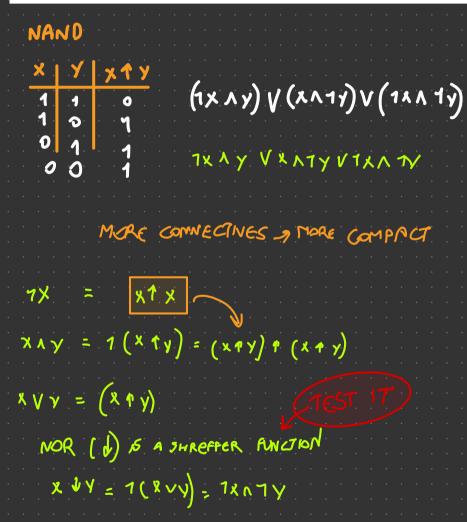
- 1. $\neg(x \land y) \lor z$
- 2. $(x \wedge y \vee \neg x \wedge \neg w) \wedge z$
- 3. $(x \lor y) \land x$
- 4. $\neg (p \land (\neg p \lor q \lor s)) \lor q \lor s$

7 (P & (7PVQVS)) V 9 VS
7 (P & 7P) V (P & 9) V (P & 5)) V9 VS
7 (P & 7P) & 1 (P & 9) & 7 (P & 5) V9 VS
G P V 7P) & (1PV 79) & (1PV 75) V9 VS
9 V 5 V 7PV PA 79 & 15

We have seen that \neg, \wedge, \vee form a **functionally complete** set of logical connectives in the sense that every Boolean function can be described using those connectives. In fact, we \neg, \wedge and \neg, \vee are functionally complete by themselves.

A logical connective is called a **Sheffer function** or **sole sufficient operator** if it is functionally complete by itself.

- 1. show that the NAND connective (denoted as \uparrow) is a Sheffer function
- 2. are there other Sheffer functions?
- 3. could a unary connective be a Sheffer function? NO, NECO TO BE ATCEST & MAAY



Exercise 6. Knowledge Bases

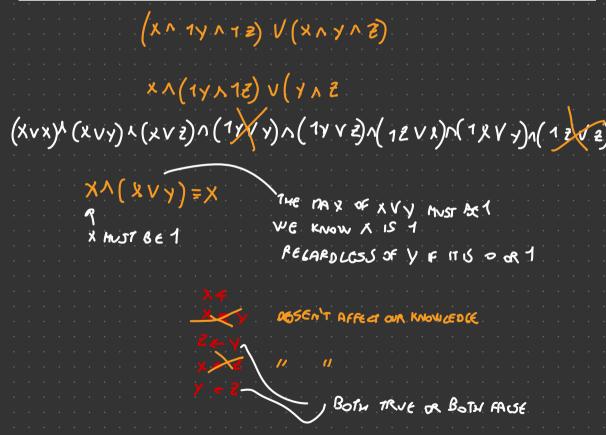
(**)

For the following truth table, construct a knowledge base (expressed as a set of propositional rules) which represents the same knowledge.

֡֡֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜֜

REPRESENT 1RV1H

1ADLE AS A FRANKA



Exercise 7. Expressivity

(**)

Can any propositional formula be expressed as a knowledge base?

If the answer is **yes**, argue why; if it is **no**, give at least two examples of formulas which cannot be expressed.

MO XVY CAN'T BE EXPRES) AS HORM CLAUSES

IN B: MAJE SPECIFIC SHAPE, NEED TO

SE HARN CEAUSES, NEED &

POSITINE CITERIOC

THERE ARE THINGS THAN CAN'T BE EXPAIN

Exercise 8. Reasoning

Consider the following knowledge base K:

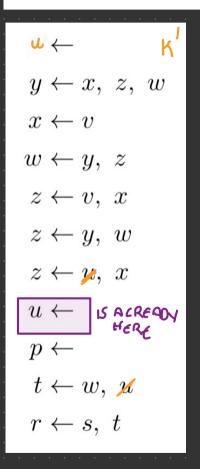
- 1. Compute the redux \hat{K}
- 2. Find all the facts that are entailed by K
- 3. Decide whether the following clauses are consequences of K
 - a) $v \leftarrow u$ to check we assume u (the copy) is true use b) $t \leftarrow u$ to make the consider $v \in u$
 - c) $q \leftarrow q$ ACWAYS TRUE AON $q \in AAD$ AAD q is a fact
 - d) $r \leftarrow w \cdot N$

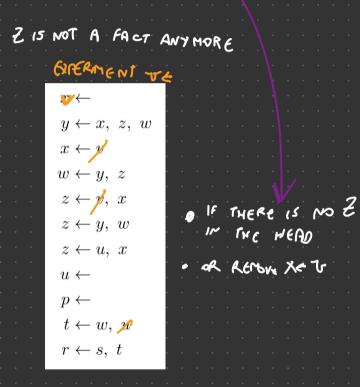
Exercise 9. Revision

(**)

In the knowledge base from Exercise 8, substitute the fact $x \leftarrow$ with $u \leftarrow$. Call this new knowledge base K'.

- 1. Do your answers from Exercise 8 change?
- 2. Which fact(s) should you remove to ensure that z is **not** a consequence of K'?
- 3. If facts cannot be removed, which rules would you remove to ensure that z is not entailed?





The soundness of the fact extraction algorithm seen during the lecture depends on the following property:

if $(\varphi \to \psi)$ and $\psi \to \xi$ are both tautologies, then $\varphi \to \xi$ is also a tautology.

Show that this property holds always in propositional logic.

