COMP2620/6262 (Logic) Tutorial

Week 2

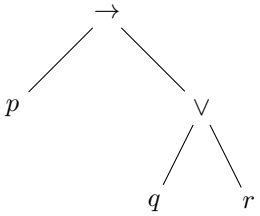
Semester 1, 2025

1. Recall the Backus-Naur form definition of the syntax of propositions:

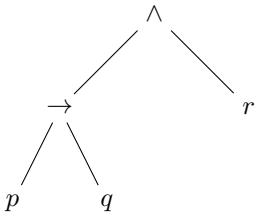
$$\varphi \quad := \quad p \mid \bot \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi$$

Draw trees showing how each of the following propositions, here expressed as strings, can be built with this definition.

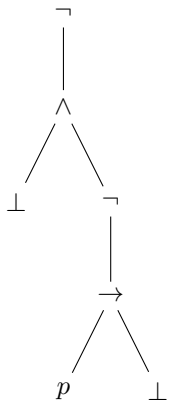
- $p \to (q \vee r)$
Solution.



- $(p \to q) \wedge r$
Solution.

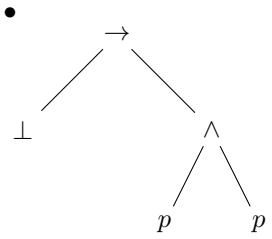


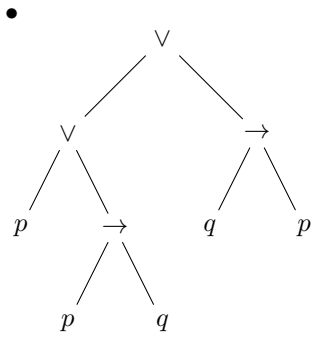
- $\neg(\bot \wedge \neg(p \to \bot))$
Solution.

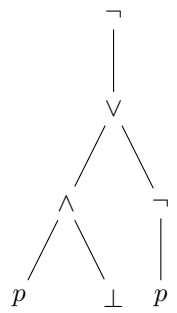


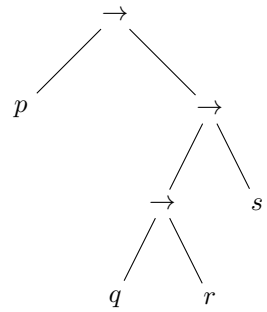
2. Convert the following trees into string syntax. Avoid excessive parentheses using the rules

- ¬ binds tighter than ∧ and ∨, which in turn bind tighter than →;
- accept ambiguity when writing e.g. $p \wedge q \wedge r$ and $p \vee q \vee r$;
- → is right associative.

- 

- 

- 

- 

Solution.

- Top left: $\bot \to p \wedge p$ - parentheses not needed because $\wedge$ binds tighter than $\to$.
- Top right: $p \vee (p \to q) \vee (q \to p)$ - we ignore which order the $\vee$ are applied.
- Bottom left: $\neg((p \wedge \bot) \vee \neg p)$
- Bottom right: $p \to (q \to r) \to s$ - we do not need parentheses around $(q \to r) \to s$ because $\to$ is right associative, but do need parentheses around $q \to r$ because we do not want $q \to (r \to s)$.

3. **The test at the start of the next tutorial will resemble this question. Truth tables for connectives will be provided, as below. Rules for eliminating parentheses will not be provided. The propositions asked about will have 3 variables.**

   Recall the truth tables of propositional logic:

p	q	$\bot$	$\neg p$	$p \wedge q$	$p \vee q$	$p \to q$
1	1	0	0	1	1	1
1	0	"	"	0	1	0
0	1	"	1	0	1	1
0	0	"	"	0	0	1

   For each proposition of question 1, and for each proposition you got as answers for question 2, construct their truth table. Are these propositions valid and/or satisfiable?

   (You have not seen a truth table in lectures where there are 4 variables in play, so for the bottom right example you will need to think about how to set up the rows!)

   Solution.

   - Satisfiable:

p	q	r	p	$\to$	$(q$	$\vee$	$r)$
1	1	1	1	**1**	1	1	1
1	1	0	1	**1**	1	1	0
1	0	1	1	**1**	0	1	1
1	0	0	1	**0**	0	0	0
0	1	1	0	**1**	1	1	1
0	1	0	0	**1**	1	1	0
0	0	1	0	**1**	0	1	1
0	0	0	0	**1**	0	0	0

- Satisfiable:

p	q	r	$(p$	$\to$	$q)$	$\wedge$	r
1	1	1	1	1	1	**1**	1
1	1	0	1	1	1	**0**	0
1	0	1	1	0	0	**0**	1
1	0	0	1	0	0	**0**	0
0	1	1	0	1	1	**1**	1
0	1	0	0	1	1	**0**	0
0	0	1	0	1	0	**1**	1
0	0	0	0	1	0	**0**	0

- Valid, and hence satisfiable:

p	$\neg$	$(\bot$	$\wedge$	$\neg$	$(p$	$\to$	$\bot))$
1	**1**	0	0	1	1	0	0
0	**1**	0	0	0	0	1	0

- Valid, and hence satisfiable:

p	$\bot$	$\to$	p	$\wedge$	p
1	0	**1**	1	1	1
0	0	**1**	0	0	0

- Valid, and hence satisfiable:

p	q	p	$\vee$	$(p$	$\to$	$q)$	$\vee$	$(q$	$\to$	$p)$
1	1	1		1	1	1	**1**	1	1	1
1	0	1		1	0	0	**1**	0	1	1
0	1	0		0	1	1	**1**	1	0	0
0	0	0		0	1	0	**1**	0	1	0

  Note that we exploited the associativity of $\vee$ to write in 1 when any of the three disjoined propositions are 1, instead of calculating the two $\vee$ separarately.

- Satisfiable:

p	$\neg$	$((p$	$\wedge$	$\bot)$	$\vee$	$\neg$	$p)$
1	**1**	1	0	0	0	0	1
0	**0**	0	0	0	1	1	0

- We need $2^4 = 16$ rows! Truth tables are starting to feel a little unwieldy. But it is satisfiable:

p	q	r	s	p	$\to$	$(q$	$\to$	$r)$	$\to$	s
1	1	1	1	1	**1**	1	1	1	1	1
1	1	1	0	1	**0**	1	1	1	0	0
1	1	0	1	1	**1**	1	0	0	1	1
1	1	0	0	1	**1**	1	0	0	1	0
1	0	1	1	1	**1**	0	1	1	1	1
1	0	1	0	1	**0**	0	1	1	0	0
1	0	0	1	1	**1**	0	1	0	1	1
1	0	0	0	1	**0**	0	1	0	0	0
0	1	1	1	0	**1**	1	1	1	1	1
0	1	1	0	0	**1**	1	1	1	0	0
0	1	0	1	0	**1**	1	0	0	1	1
0	1	0	0	0	**1**	1	0	0	1	0
0	0	1	1	0	**1**	0	1	1	1	1
0	0	1	0	0	**1**	0	1	1	0	0
0	0	0	1	0	**1**	0	1	0	1	1
0	0	0	0	0	**1**	0	1	0	0	0

4. Determine the validity of the following sequents using truth tables. Recall that you only need to consider the rows in which every premise has truth value 1.

- $p \to \bot \vdash \neg p$

  Solution. We will not give quite as much detail for the truth tables in this section.

p	$p \to \bot$	$\neg p$
1	0	
0	1	1

  The sequent is valid. We left off the first entry for $\neg p$ because we did not need it for determining validity, but if we had included it then we would see that $p \to \bot$ and $\neg p$ in fact have the same truth table, so we would also have a valid sequent if we swapped premise and conclusion.

- $p, \neg p \vdash q$

  Solution. There is no line that will set both p and $\neg p$ to 1, so the sequent is valid vacuously, regardless of the conclusion. We could even replace q with the always false $\bot$, and the sequent remains valid!

- $p \to r, q \to r \vdash r \to (p \vee q)$

  Solution. Not valid:

p	q	r	$p \to r$	$q \to r$	$r \to (p \vee q)$
1	1	1	1	1	1
1	1	0	0	0	
1	0	1	1	1	1
1	0	0	0	1	
0	1	1	1	1	1
0	1	0	1	0	
0	0	1	**1**	**1**	**0**
0	0	0	1	1	1

- $(p \to q) \to p, \neg q \to \neg p \vdash q$

  Solution. Valid:

p	q	$(p \to q) \to p$	$\neg q \to \neg p$	q
1	1	1	1	1
1	0	1	0	
0	1	0	1	
0	0	0	1	

5. Some of our connectives can be defined in terms of some of the others; for example we could define $\neg p$ as $p \to \bot$, or $p \vee q$ as $\neg(\neg p \wedge \neg q)$.

   The Sheffer Stroke, also called nand, or 'not both', is a connective with truth table

p	q	$p \uparrow q$
1	1	0
1	0	1
0	1	1
0	0	1

   - How can we define $p \uparrow q$ using our usual connectives?

     Solution. $\neg(p \wedge q)$ is one correct answer; $\neg p \vee \neg q$ is another. There are others that work also. Just check that the truth table is the same.

   - (Tricky) show that all our usual connectives can be defined using $\uparrow$ only. Hint: start by thinking about how you would define $\neg$.

     Solution. Define $\neg p$ as $p \uparrow p$, because 'not both p and p' is the same as 'not p'.

     Now $\neg(p \uparrow q)$, or writing it out in full, $(p \uparrow q) \uparrow (p \uparrow q)$, gives exactly the truth table for $p \wedge q$ by flipping the truth table for the Sheffer stroke.

     This is enough to get us home because we can define all our other connectives in terms of $\neg$ and $\wedge$: $\bot$ is $p \wedge \neg p$, $p \vee q$ is $\neg(\neg p \wedge \neg q)$, and $p \to q$ is $\neg p \vee q$. You might find some more elegant encodings by working directly with the Sheffer stroke.