CS 511, Fall 2018, Handout 02

Syntax of Propositional Logic

Assaf Kfoury

September 04, 2018
(corrected on Sept 08, 2018)

Syntax of the wff's of Propositional Logic

- Reading: [LCS, Section 1.3]

Syntax of the wff's of Propositional Logic

- Reading: [LCS, Section 1.3]
- The wff's of propositional logic are obtained by applying the construction rules below, and only these, **finitely many times**.

  One **basis step**:

  0. every propositional atom (*i.e.*, propositional variable) p is a WFF

  Four **induction steps**:

  1. if φ is a wff, then so is $(\neg\varphi)$
  2. if φ and ψ are wff's, then so is $(\varphi \wedge \psi)$
  3. if φ and ψ are wff's, then so is $(\varphi \vee \psi)$
  4. if φ and ψ are wff's, then so is $(\varphi \to \psi)$

Remember this inductive definition: We use it later in syntax-directed proofs.

Syntax of the wff's of Propositional Logic

- More succintly, in **BNF (Backus Naur Form)**:

$$\varphi \ ::= \ p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi)$$

  This is the same as in [LCS, page 33].

Syntax of the wff's of Propositional Logic

- More succintly, in **BNF (Backus Naur Form)**:

  $$\varphi \ ::= \ p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi)$$

  This is the same as in [LCS, page 33].

- Or, in **Extended BNF** – some occurrences of 'φ' are replaced by 'ψ':

  $$\varphi \ ::= \ p \mid (\neg\varphi) \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\varphi \to \psi)$$

Syntax of the wff's of Propositional Logic

- More succintly, in **BNF (Backus Naur Form)**:

  $$\varphi \ ::= \ p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi)$$

  This is the same as in [LCS, page 33].

- Or, in **Extended BNF** – some occurrences of 'φ' are replaced by 'ψ':

  $$\varphi \ ::= \ p \mid (\neg\varphi) \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\varphi \rightarrow \psi)$$

- Or, more abstractly by omitting parentheses, in **Extended BNF**:

  $$\varphi \ ::= \ p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi$$

  Parentheses are used only to set an order of precedence among logical connectives $\{\neg, \wedge, \vee, \rightarrow\}$.

Parse Trees of wff's

- A fully-parenthesized wff:

$$\Big(\big(\neg((\neg P) \lor (Q \land (\neg P)))\big) \\ \to \big(\neg((\neg P) \to (Q \lor (\neg R)))\big)\Big)$$

Parse Trees of wff's

- A fully-parenthesized wff:

$$\Big(\big(\neg((\neg P) \vee (Q \wedge (\neg P)))\big)$$
$$\rightarrow \big(\neg((\neg P) \rightarrow (Q \vee (\neg R)))\big)\Big)$$

- Same wff with all parentheses omitted:

$$\neg\ \neg P\ \ \vee\ \ Q \wedge\ \ \neg P$$
$$\rightarrow\ \ \neg\ \neg P\ \ \rightarrow\ \ Q \vee\ \ \neg R$$

(an incomprehensible mess!)

Parse Trees of wff's

- A fully-parenthesized wff:

$$\Big(\big(\neg((\neg P) \vee (Q \wedge (\neg P)))\big) \\ \quad \to \big(\neg((\neg P) \to (Q \vee (\neg R)))\big) \Big)$$

- Same wff with all parentheses omitted:

$$\neg \; \neg P \;\; \vee \;\; Q \wedge \;\; \neg P \\ \quad \to \;\; \neg \; \neg P \;\; \to \;\; Q \vee \;\; \neg R$$

(an incomprehensible mess!)

- Same wff minimally parenthesized:

$$\neg \; (\neg \, P \;\; \vee \;\; (Q \wedge \neg \, P)) \\ \quad \to \;\; \neg \; (\neg \, P \;\; \to \;\; (Q \vee \;\; \neg \, R))$$

Parse Trees of wff's

- A fully-parenthesized wff:

$$\Big(\big(\neg((\neg P)\vee(Q\wedge(\neg P)))\big)$$
$$\qquad\to\big(\neg((\neg P)\to(Q\vee(\neg R)))\big)\Big)$$

- Same wff with all parentheses omitted:

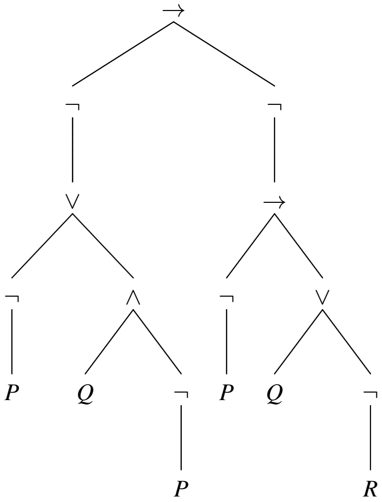
$$\neg\ \neg P\ \vee\ Q \wedge\ \neg P$$
$$\qquad\to\ \neg\ \neg P\ \to\ Q \vee\ \neg R$$

(an incomprehensible mess!)

- Same wff minimally parenthesized:

$$\neg\,(\neg\,P\ \vee\ (Q\wedge\neg\,P))$$
$$\qquad\to\ \neg\,(\neg\,P\ \to\ (Q\vee\ \neg\,R))$$

No parentheses in the parse tree



Parse trees are very nice,

but more difficult to store.

(THIS PAGE INTENTIONALLY LEFT BLANK)