Lecture 5 (Section 1.2 continued)

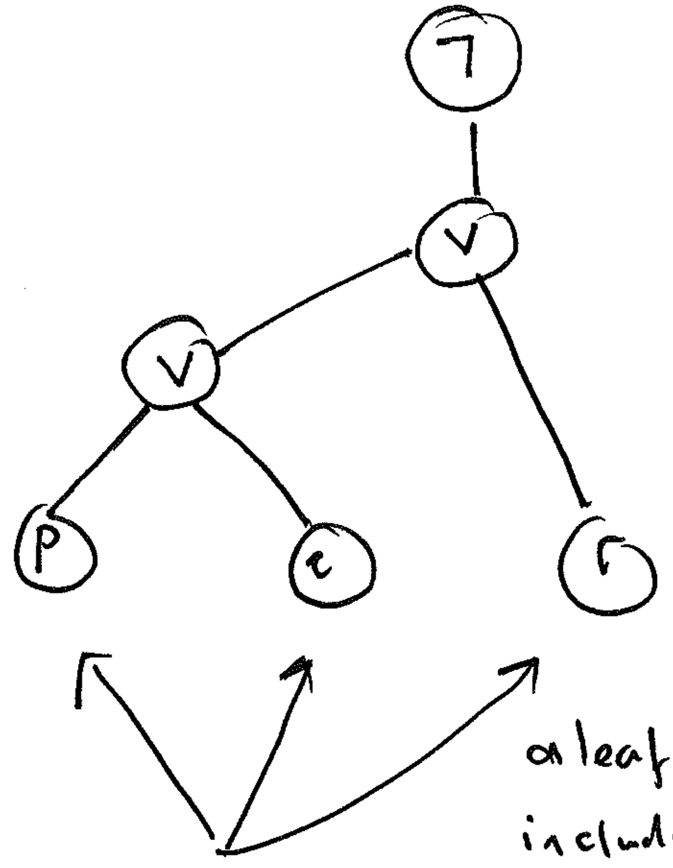
Parse trees

These enable us to write wff without brackets but we pay the price of having to write them in 2D rather than in 1D.

Example We construct the parse tree of

$\neg((p \vee q) \vee r)$

$$p,\ q,\ r$$
$$\Downarrow$$
$$p \vee q$$
$$\Downarrow$$
$$(p \vee q) \vee r$$
$$\Downarrow$$
$$\neg((p \vee q) \vee r)$$



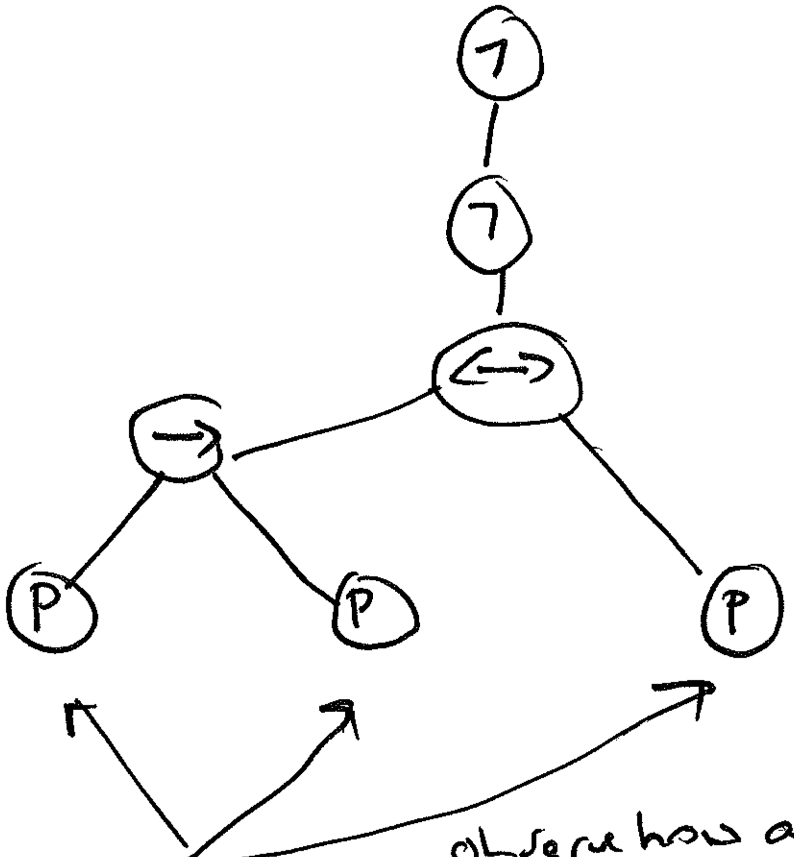
a leaf for every atom including repeats.

$\vee$ $+$ $\wedge$ $\rightarrow$ $\leftrightarrow$

all have two inputs from below and one output from the top (unless they are the last connective/ root connective).

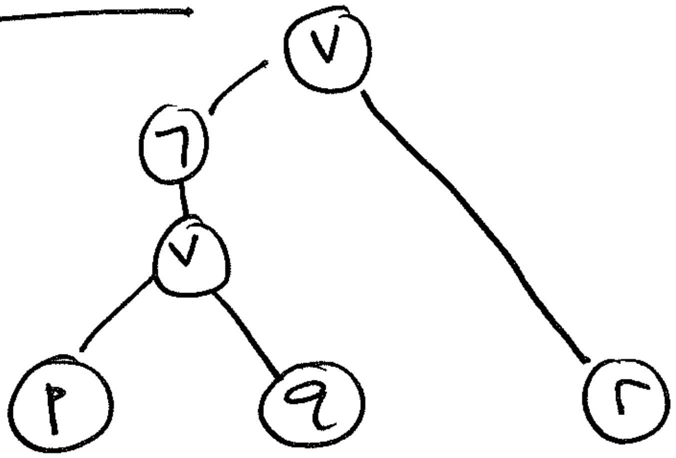
$\neg$ has one input from below and one output from the top unless it is the root connective.

<u>Example</u> The parse tree of $\neg\neg((p \rightarrow p) \leftrightarrow p)$



observe how all repeated atoms are written down.

<u>Example</u> Parse tree for $\neg(p \vee q) \vee r$



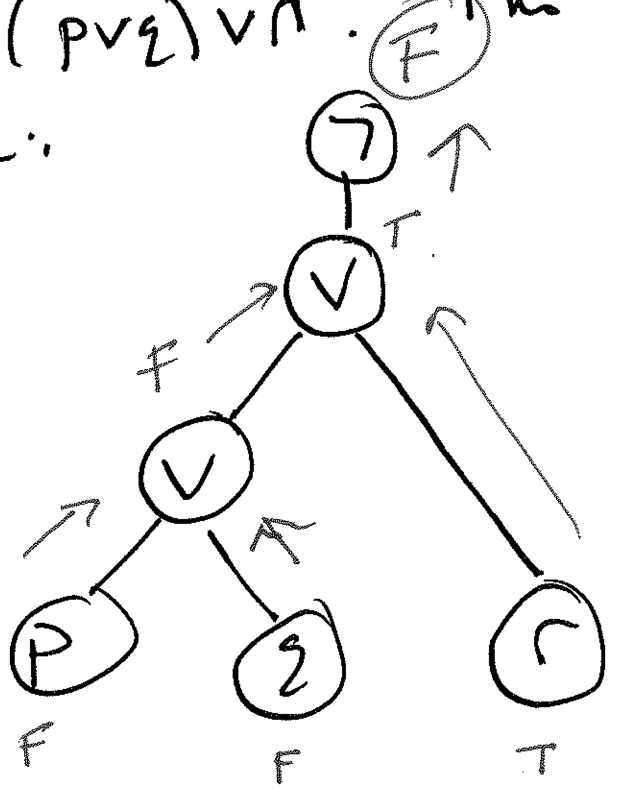
In addition to describing the syntax of a wff without using brackets, parse trees are, in fact, the first (and, actually, key step) in constructing a circuit. We shall return to this point later.

Section 1.3: Semantics

We have described the syntax of PL, we now describe its semantics. This will simply be the truth table of the wff.

<u>Example</u> (Idea) Consider the wff.

$\neg((p \vee q) \vee r)$. This has the following parse tree:



Using this tree, we can see very clearly how truth values assigned to the atoms (p, q, r)

lead to a truth value being assigned to the wff $\neg((p \vee q) \vee r)$.

p	q	r	$\neg((p \vee q) \vee r)$
F	F	T	F

However, we want to show all possibilities (not just a few).

> Let A be a wff with n atoms $p_1, \ldots, p_n$. Then the truth table for A will have 2^n rows.

The semantics/meaning of a wff *is* its truth table.

Example 1 Construct the truth table of

$$(p \vee q) \wedge \neg r$$

p	q	r	$p \vee q$	$\neg r$	$(p \vee q) \wedge \neg r$
T	T	T	T	F	F
T	T	F	T	T	T *
T	F	T	T	F	F
T	F	F	T	T	T *
F	T	T	T	F	F
F	T	F	T	T	T *
F	F	T	F	F	F
F	F	F	F	T	F

Example 2 Construct the truth table

d) $(p \vee q) \rightarrow (r \leftrightarrow \neg s)$

p	q	r	s	$p \vee q$	$\neg s$	$r \leftrightarrow \neg s$	$(p \vee q) \rightarrow (r \leftrightarrow \neg s)$
T	T	T	T	T	F	F	F
T	T	T	F	T	T	T	T
T	T	F	T	T	F	T	T
T	T	F	F	T	T	F	F
T	F	T	T	T	F	F	F
T	F	T	F	T	T	T	T
T	F	F	T	T	F	T	T
T	F	F	F	T	T	F	F
F	T	T	T	T	F	F	F
F	T	T	F	T	T	T	T
F	T	F	T	T	F	T	T
F	T	F	F	T	T	F	F
F	F	T	T	F	F	F	T
F	F	T	F	F	T	T	T
F	F	F	T	F	F	T	T
F	F	F	F	F	T	F	T

Truth table produced using the truth table generator

p	q	r	$(p \vee q) \wedge \neg r$
T	T	T	F
T	T	F	T
T	F	T	F
T	F	F	T
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	F

p	q	r	s	$(p \vee q) \rightarrow (r \leftrightarrow \neg s)$
T	T	T	T	F
T	T	T	F	T
T	T	F	T	T
T	T	F	F	F
T	F	T	T	F
T	F	T	F	T
T	F	F	T	T
T	F	F	F	F
F	T	T	T	F
F	T	T	F	T
F	T	F	T	T
F	T	F	F	F
F	F	T	T	T
F	F	T	F	T
F	F	F	T	T
F	F	F	F	T

Important definitions

- An atom or the negation of an atom is called a literal.

- Let A be a wff with atoms $p_1, \dots, p_n$. We say that A is satisfiable if there is at least one truth assignment to the atoms $p_1 \dots p_n$ that makes A true.

- If A is always false, it is called a contradiction. We write

$$\boxed{A \models}$$

- If A is always true, it is called a tautology. We write $\boxed{\models A}$

The symbol $\models$ is called the <u>semantic turnstile</u>.

- A truth assignment to the atoms of A which makes A true is said to <u>satisfy A</u>.

- The wff $A_1, \ldots, A_n$ are said to be (simultaneously) <u>satisfiable</u> if $((A_1 \wedge A_2) \wedge A_3) \ldots) \wedge A_n$ is satisfiable.

- A <u>decision problem</u> is a question that has only a yes or no answer. Eg. 'Is the natural number n prime?'

<u>The Satisfiability problem</u>

<u>(SAT)</u>

Given a wff A decide whether A is satisfiable or not.

A program that solves SAT is called a <u>SAT solver</u>

What's the problem with truth tables?

If there are only a few atoms then truth tables are fine. The problem arises when you scale up to wff you want to study.

S'pose a wff has 90 atoms.

Then its truth table will have 2^{90} rows.

Observe that $2 \approx 10^{0.3}$

$$\therefore 2^{90} \approx (10^{0.3})^{90} = 10^{0.3 \times 90} \approx 10^{27}$$

a nanosecond

S'pose that it takes 10^{-9} seconds to construct each row of the truth table. Then it will take 10^{18} $(= 10^{-9} \times 10^{27})$ seconds to construct the whole truth table.

For comparison purposes, the age of the universe is $\approx 4.35 \times 10^{17}$ seconds.

Thus, while truth tables are SAT solvers, they are very inefficient ones.

The question therefore is: can we do better than truth tables?
