

## Reasoning About Programs Can Be Hard

- Textbook example of a program that is hard to reason about (Chapter 1).
- (Use a C/Java-like notation for a bit.)
- Program searches a function `int f(int)` for a zero.
- Two pieces of sequential code run in parallel:
- $S_1$  is the program

```
found = false; x = 0;
while !found { found = (f(++x) == 0); }
```
- $S_2$  is the program

```
found = false; y = 1;
while !found { found = (f(--y) == 0); }
```
- This is buggy!
  - If  $S_1$  finds an `x` and sets `found` to `true` before  $S_2$  starts running, then  $S_2$  sets `found` to `false` before trying `y` equals `0`, `-1`, .... (and maybe none of those have `f(y)` equal to zero).
  - Should set `found` to `false` **before** starting  $S_1$  and  $S_2$ . Also, drop the `found = false;` in  $S_1$  and  $S_2$ :
- Program: `found = false;` run  $S_1$  and  $S_2$  in par-

allel where

- $S_1$  is `x = 0; while !found { found = (f(++x) == 0); }`
- $S_2$  is `y = 1; while !found { found = (f(--y) == 0); }`
- No, that's buggy too!
  - $S_2$  finds `found` is false but before it can test `f(--y)`
  - $S_1$  tests `f(++x)`, finds zero, sets `found = true`;
  - Then  $S_2$  sets `found = false`.
- Problem is that we can reset `found` to `false` once it becomes `true`.
  - Use the following:
  - $S_1$  is `x = 0;`  
`while !found {`  
    `if (f(++x) == 0) found = true }`
  - $S_2$  is `y = 1;`  
`while !found {`  
    `if (f(--y) == 0) found = true }`

## Review Propositional Logic

### Propositional Logic

- Example of proposition:  $p \wedge q \leftrightarrow \neg(\neg p \vee \neg q)$ .
- Propositions (= Propositional formulas)
  - Use boolean variables as simplest kind of proposition (“proposition letters”).
- Combine using basic connectives.
  - $p \wedge q$  – logical and
  - $p \vee q$  – logical or (at least 1 of  $p, q$  are true)
  - $p \rightarrow q$  – implication ( $p$  implies  $q$ )
  - $T \rightarrow F$  is false;  $F \rightarrow F, F \rightarrow T$ , and  $T \rightarrow T$  are all true.
  - $p \leftrightarrow q$  – equivalence ( $p$  iff  $q$ ,  $p$  is equivalent to  $q$ )
  - $\neg p$  – logical negation ( $\neg T \leftrightarrow F, \neg F \leftrightarrow T$ )
  - Precedences:  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$  (strong to weak)
    - E.g.  $((((\neg p) \wedge q) \vee r) \rightarrow s) \leftrightarrow t$  has all parens optional.
    - $\neg p \wedge q \vee r \rightarrow s \leftrightarrow t$
  - Associativity:
    - $\rightarrow$  is right associative
    - $\wedge, \vee, \leftrightarrow$  are associative, so it sort of doesn't

matter; use right associativity

- $\rightarrow$  is not associative:  $(p \rightarrow (q \rightarrow r))$  is not logically equivalent to  $((p \rightarrow q) \rightarrow r)$
- Note  $p \rightarrow q \rightarrow r$  is short for  $(p \rightarrow (q \rightarrow r))$ .
- Propositional logic is good for showing how the logical connectives are related.
  - $(p \leftrightarrow q) \leftrightarrow (\neg p \leftrightarrow \neg q)$ , for example.
- Typical semantics: Truth tables.
  - Two propositions are *logically equivalent* when they have the same truth table.
- A *tautology* has a truth table with all true results.
  - $p$  and  $q$  are logically equivalent exactly when  $p \leftrightarrow q$  is a tautology.
- A *contradiction* has a truth table with all false results.

**Some basic logical equivalences**

- Commutativity
  - $p \vee q \Leftrightarrow q \vee p$
  - $p \wedge q \Leftrightarrow q \wedge p$
  - $(p \Leftrightarrow q) \Leftrightarrow (q \Leftrightarrow p)$
- Associativity
  - $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
  - $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$
- Distributivity
  - $(p \vee q) \wedge r \Leftrightarrow (p \wedge r) \vee (q \wedge r)$
  - $(p \wedge q) \vee r \Leftrightarrow (p \vee r) \wedge (q \vee r)$
- Transitivity
  - [Note:  $\rightarrow$ , not  $\Leftrightarrow$ ]
  - $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$
- Redefinition of  $\rightarrow$ ,  $\wedge$ , and  $\Leftrightarrow$  from  $\neg$  and  $\vee$ .  
(Other possible combinations:  $\neg$  and  $\wedge$ ; or  $\neg$  and
- $(p \Leftrightarrow q) \wedge (q \Leftrightarrow r) \rightarrow (p \Leftrightarrow r)$
- Identity
  - $p \wedge T \Leftrightarrow p$
  - $p \vee F \Leftrightarrow p$
- Domination
  - $p \vee T \Leftrightarrow T$
  - $p \wedge F \Leftrightarrow F$
- Contradiction
  - $p \wedge \neg p \Leftrightarrow F$
- Excluded middle
  - $p \vee \neg p \Leftrightarrow T$
- Idempotentcy
  - $p \vee p \Leftrightarrow p$
  - $p \wedge p \Leftrightarrow p$
- Double negation (Pierce's Law)
  - $\neg\neg p \Leftrightarrow p$

$\rightarrow$ ):

- $(p \leftrightarrow q) \leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$
- $p = q \leftrightarrow p \leftrightarrow q$
- $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$
- $\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$  [DeMorgan's law 1]
- $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$  [DeMorgan's law 2]
- Substitutability
  - If  $p \leftrightarrow q$  and  $p$  appears in a proposition  $r$ , then any/some/all occurrences of  $p$  inside  $r$  can be replaced by  $q$ .
  - Example: Say  $r \leftrightarrow \neg\neg r$  and use  $p \wedge q \rightarrow p$  for  $r$ . Then by substitution,  $(p \wedge q \rightarrow p) \leftrightarrow \neg\neg(p \wedge q \rightarrow p)$ .

### Example of Manipulation of Propositions

$$\begin{aligned}
 &(a \rightarrow b) \wedge a \\
 \Leftrightarrow &(\neg a \vee b) \wedge a && \text{Defn of } \rightarrow \\
 \Leftrightarrow &(\neg a \wedge a) \vee (b \wedge a) && \text{Distribution of } \wedge \text{ over } \vee \\
 \Leftrightarrow &(a \wedge \neg a) \vee (b \wedge a) && \text{Commut. of } \wedge \text{ [omit?]} \\
 \Leftrightarrow &F \vee (b \wedge a) && \text{Contradiction} \\
 \Leftrightarrow &b \wedge a && \text{Identity}
 \end{aligned}$$

In general we often give names to the properties we prove so that we can reuse them more easily

- $(a \rightarrow b) \wedge a \rightarrow b$       Modus ponens

Sample Truth Table:

<b>p</b>	<b>q</b>	<b><math>p \rightarrow q</math></b>	<b><math>\neg p</math></b>	<b><math>\neg p \vee q</math></b>	<b><math>\neg p \wedge q</math></b>
T	T	T	F	T	F
T	F	F	F	F	F
F	T	T	T	T	T
F	F	T	T	T	F

So  $(p \rightarrow q)$  is logically equivalent to  $(\neg p \vee q)$

I.e.,  $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$