Lecture 3 - January 17

Math Review

Propositional Logic & Predicate Logic

Announcement

- Lab1 released
  + tutorial videos ✓ 2.5 hours → Book → Book.zip.
  + problems to solve
  + Study along with the Math Review lecture notes.

Look at the tips of Rockin installation

Logical Operator vs. Programming Operator

p	q	$p \wedge q$	$p \vee q$
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

short-circuit evaluation: L to R

e1 && e2

↳ if LHS evaluates to F, skip the evaluation of RHS

Q. Are the $\wedge$ and $\vee$ operators equivalent to, respectively, && and || in Java?

$\wedge$ → logical operator

&& → programming operator

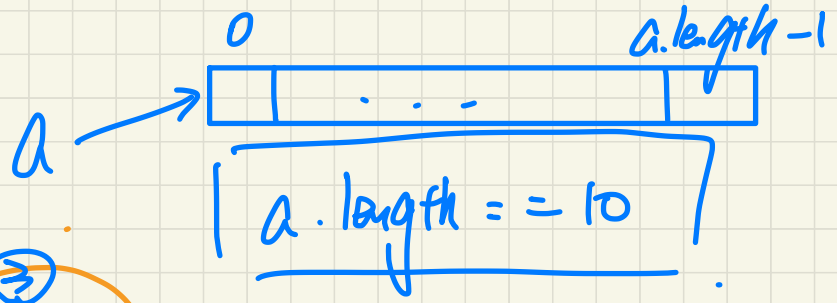
short-circuit

e1 || e2

① ↳ if LHS evaluates to T, skip the evaluation of RHS

↳ runtime evaluation

Accessing Array



① ② ③

`i >= 0 && i < a.length && a[i]`

guarding conditions

expression to protect

① `i == -2`

① evaluates to (F) ②, ③ skipped

overall: (F)

② `i == 12`

① evaluates (T) ② evaluates to (F)

③ skipped

overall: (F)

int[] a = ---

Exercise Assume `a.length == 10`

`i < a.length && a[i] > 10 && i >= 0`

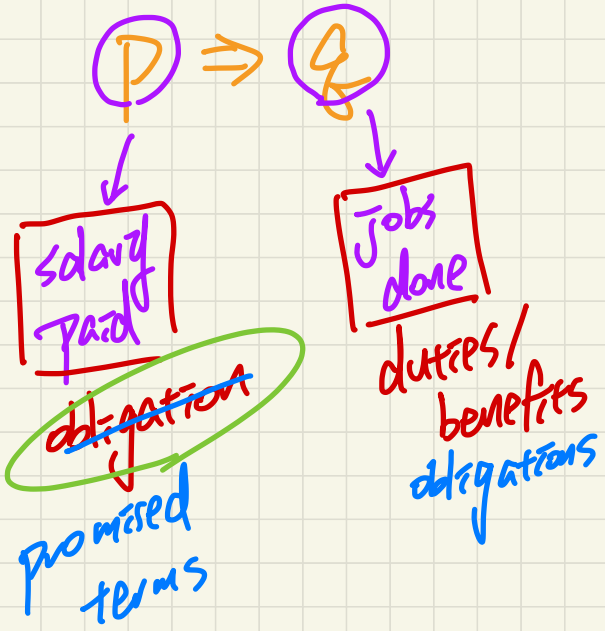
too late to evaluate for guarding.

↳ does this property guard `a[i]`?

↳ No! witness: `i == -2`

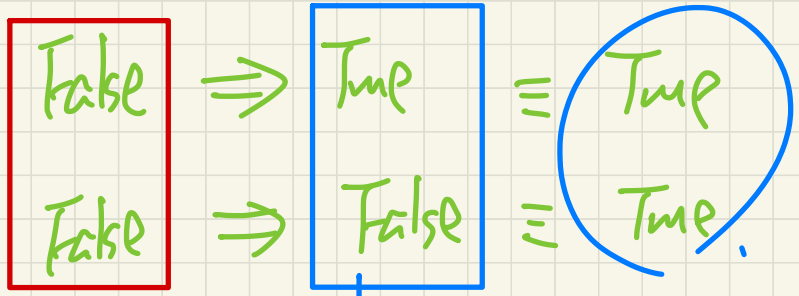
Exercises: Try other orderings of guarding conditions.

Implication ≈ Whether a Contract is Honoured



True ⇒ True ≡ True

True ⇒ False ≡ False



Expressing Implications

p: snow storm
q: cancel class

q if p, **p is sufficient for q**

p	q	p ⇒ q
true	true	true
true	false	false
false	true	true
false	false	true

q is true if p is true

p is not true, no guarantee what q is.

p only if q, **q is necessary for p**

p	q	p ⇒ q
true	true	true
true	false	false
false	true	true
false	false	true

p is not T, don't care.

one condition for ⇒ to be T.

q unless ¬p

p	q	p ⇒ q
true	true	true
true	false	false
false	true	true
false	false	true

if p already T, for ⇒ to be T, necessary for q to be

Prove $p \Leftrightarrow q$

(1) $p \Rightarrow q$ (p only if q) T

(2) $q \Rightarrow p$ (p if q)

$p \Rightarrow q$

$p \Rightarrow q \equiv \neg q \Rightarrow \neg p$

(1) Inverse: $\neg p \Rightarrow \neg q$

(2) Converse: $q \Rightarrow p$

(3) Contrapositive: $\neg q \Rightarrow \neg p$
(Inverse of Converse)

Given

$x > 0 \wedge x \leq 10 \Rightarrow$

$y \geq 3 \vee y < 5$

(1)

(2)

(3) (apply de morgan when applicable)

Identity

$$true \Rightarrow P \equiv P$$

$$true \wedge P \equiv P$$

$$false \vee P \equiv P$$

$$0 + \bar{\imath} = \bar{\imath}$$

$$1 * \bar{\imath} = \bar{\imath}$$

Zero

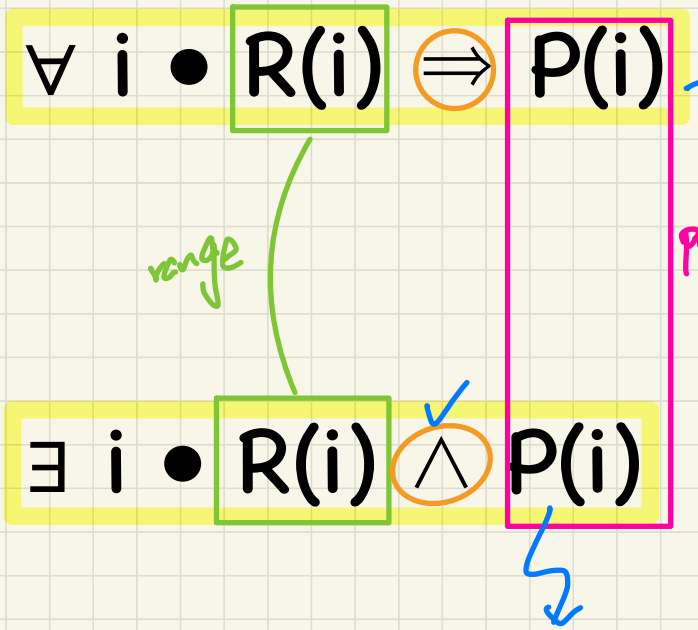
$$false \Rightarrow P \equiv True$$

$$false \wedge P \equiv false$$

$$true \vee P \equiv true$$

Predicate Logic: Quantifiers

- syntax
- base cases in programming



for each i,
if i satisfies R,
then P is satisfied.
(implicitly, if no such
i satisfies R,
then ∀ is T)

there's at least one i,
s.t. i is in the range and i satisfies P.
(implicitly, if no such i satisfies R, then ∃ is F)