[With post-lecture annotations (in green)]

CSC236 winter 2020, week 3: structural induction, well-ordering

See section 1.2-1.3 of course notes

Colin Morris

colin@cs.toronto.edu

http://www.cs.toronto.edu/~colin/236/W20/

January 20, 2020

Outline

- Well-ordering
  - Principle of well-ordering
  - Example: prime factorizations, revisited
  - Example: round-robin tournament cycles
- Structural induction
  - Introduction
  - Example: complete binary trees
  - Comparison with simple induction
  - Example: strings of matching parentheses

Principle of well-ordering

Not true of...

$\mathbb{Z}$

$\mathbb{Q}^+$ $\left\{ \frac{1}{1}, \frac{1}{2}, \frac{1}{3} \ldots \right\}$ ← counterexample

Every non-empty subset of $\mathbb{N}$ has a smallest element.

Surprisingly, turns out to be equivalent to principle of mathematical induction / complete induction. (Theorem 1.1 in Vassos course notes)

Every $n > 1$ has a prime factorization

For sake of contradiction, assume this is false. i.e.

$$S = \{n \in \mathbb{N} \mid n > 1 \land n \text{ is not the product of primes}\}$$

is non-empty. By PWO, S has a smallest element, call it j.

because S non-empty, $S \subseteq \mathbb{N}$

$j = a \times b$ $\quad$ $a < j$, $a \notin S$

$\quad\quad\quad\quad\quad$ $b < j$ $\quad$ $b \notin S$

Every $n > 1$ has a prime factorization

[Looks very similar to our complete induction proof from last week - not a coincidence.]

For sake of contradiction, assume this is false. i.e.

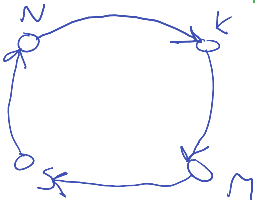
$$S = \{n \in \mathbb{N} \mid n > 1 \land n \text{ is not the product of primes}\}$$

is non-empty. By PWO, S has a smallest element, call it j.
Case 1: j is prime. **Contradiction!**
Case 2: j is composite. Let $a, b \in \mathbb{N}$ such that $j = a \times b \land 1 < a < j \land 1 < b < j$ (by definition of composite).
$a, b \notin S$, since j was chosen to be the smallest element. So a and b each have a prime factorization. We can concatenate them to form a prime factorization of j.
Contradiction!
In each case, we derived a contradiction, so our premise is false. S must be empty.
$\forall n \in \mathbb{N} - \{0, 1\}$, n has a prime factorization.

Round-robin tournament cycles

Round-robin tournament $\equiv$ every player faces every other player once.
Consider "cycles" of matchups such as...

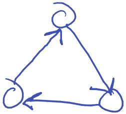
- Naomi beats Kim
- Kim beats Monica
- Monica beats Serena
- Serena beats Naomi



Note: a complete specification of this graph would have edges btwn N + M, and S + K, but the direction of those edges is irrelevant in this context.

Claim: Any round-robin tournament having at least one cycle has a 3-cycle.

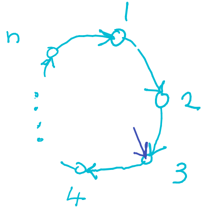
$\forall$ $\Rightarrow$



Proof: if a RR tournament has a cycle, it has a 3-cycle

For an arbitrary RR tournament, assume there is some cycle

$$p_1 > p_2 > \dots p_n > p_1$$



$$S = \{ i \in N \mid p_i > p_1 \}$$

$n \in S$, let j be smallest ele of S

since $j-1 < j$, $\therefore$ $j-1 \notin S$

$$p_j > p_1 > p_{j-1} > p_j$$

since $j \in S$ by def'n of sequence $p_1, p_2 \dots p_n$

then there is a 3-cycle, $a > b > c > a$

Recursively defined sets

Sets defined in terms of one or more ‘simple’ examples, plus rules for generating elements from other elements.

Example:

- A single node is a complete binary tree
- If t_1 and t_2 are complete binary trees, then a new node joined to t_1 and t_2 as its children form a complete binary tree

We can use structural induction to prove properties of such sets.

Structural induction proof outline

For some recursively defined set S...

1. Define predicate with domain S $\cancel{P(n)}$ $P(t)$ $P(s)$ $P(q)$
2. **Basis**: verify $P(x)$ for 'basic' element(s) $x \in S$
3. **Inductive step**: show that each rule that generates other elements of S preserves P-ness. i.e. for each rule...
   3.1 Choose arbitrary elements of S
   3.2 Assume predicate holds for those elements
   3.3 Use assumption to show that $P(z)$ holds, where z is an element generated from our previously chosen elements.

$\forall x \in S, \; P(x)$

Prove: all ~~complete~~ binary trees have an odd number of nodes

1. Predicate

$$P(t): \exists k \in \mathbb{N},\ Nodes(t) = 2k+1$$

Prove: all complete binary trees have an odd number of nodes

2. Basis

Let t be a single node

$\text{Nodes}(t) = 1 = 2 \times 0 + 1$, so $P(t)$

Prove: all complete binary trees have an odd number of nodes

3. Inductive step $T :=$ set of complete bin trees

Let $t_1, t_2 \in T$

Assume $P(t_1) \wedge P(t_2)$ # I.H.

Consider the tree formed by joining t_1 and t_2 under a new node. Call it t.

$\text{Nodes}(t) = \text{Nodes}(t_1) + \text{Nodes}(t_2) + 1$

Let $k_1, k_2 \in \mathbb{N}$, s.t.

$\text{Nodes}(t) = (2k_1 + 1) + (2k_2 + 1) + 1$ # by IH

$= 2(k_1 + k_2 + 1) + 1$, so $P(t)$. $\blacksquare$

Compare with simple induction

$-1 \in \mathbb{N}$? Possible if we omit "smallest"

Define $\mathbb{N}$ as the smallest[1] set such that:

1. $0 \in \mathbb{N}$
2. $n \in \mathbb{N} \implies \underline{n + 1} \in \mathbb{N}$ successor function

[1] Why is this necessary?

Strings with matching parentheses

set of strings, aka 'language'

Define $\mathcal{B}$ as the smallest set such that...

1. $\epsilon \in \mathcal{B}$ `# where` ϵ `denotes the empty string` ()
2. If $b \in \mathcal{B}$, then $(b) \in \mathcal{B}$
3. If $b_1, b_2 \in \mathcal{B}$, then $b_1 b_2 \in \mathcal{B}$ `# closed under concatenation`

Examples of elements?

ϵ

()   (())

(())()()

A claim about $\mathcal{B}$

s' is a prefix of s, if $\exists$ string z, s.t.

$s = s'z$

Define...

- $L(s) = \#$ of occurrences of (in s s.count('(')
- $R(s) = \#$ of occurrences of) in s

Claim: $\forall s \in \mathcal{B}$, if s' is a prefix of s, then $L(s') \geq R(s')$. (()))()()

String ab has 3 prefixes:

1. ϵ
2. a
3. ab

ab

Prove: prefixes of strings of balanced parens are left-heavy

$P(s)$: for all s', if s' is a prefix of s, $L(s') \geq R(s')$

Basis: ϵ has only one prefix, and that is ϵ

$L(\epsilon) = 0$, $R(\epsilon) = 0$

$L(\epsilon) \geq R(\epsilon)$

Thus $P(\epsilon)$

Prove: prefixes of strings of balanced parens are left-heavy

Inductive step [Part 1]

Let $s_1 \in B$, assume $P(s_1)$

let $s = (s_1)$

Let s' be an arbitrary prefix of s.

WTS: $L(s') \geq R(s')$ Not a valid claim at this point in the proof.

Case 1: $s' = \epsilon$

$L(\epsilon) = 0 \geq 0 = R(\epsilon)$, ~~thus $P(s)$~~ $L(s') \geq R(s')$

Case 2: $s' = ($

$L(() = 1 \geq 0 = R(()$

redundant w/ case 4.

Case 3: $s' = s$

$L(s') = L(s_1) + 1 \geq R(s_1) + 1$ # by I.H., and adding 1 to each side

$= R(s')$

Case 4: s' is of the form $(s''$, where s'' is a prefix of s_1

$L(s') = 1 + L(s'')$

$R(s') = R(s'')$

by I.H., $L(s'') \geq R(s'')$ # since s'' is a prefix of s_1

$L(s'') + 1 \geq R(s'')$

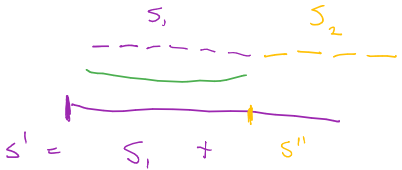
$L(s') \geq R(s')$

Thus for any prefix s', $L(s') \geq R(s')$, so $P(s)$

Prove: prefixes of strings of balanced parens are left-heavy

Inductive step [for concatenation rule]

Let $s_1, s_2 \in \mathcal{B}$, assume $P(s_1) \wedge P(s_2)$



Let $s = s_1 s_2$

Let s' be an arbitrary prefix of s

Case 1: $len(s') \leq len(s_1)$

then s' is a prefix of s_1, so $L(s') \geq R(s')$ by $P(s_1)$

Case: $len(s') > len(s_1)$

then $\exists\ s_2'$ such that $s' = s_1 s_2'$, where s_2' is a prefix of s_2

[algebra]

thus $L(s') \geq R(s')$

then $P(s)$