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/

Announcements

- new tutorial rooms (check website)
- A1 due in 10 days

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

$\mathbb{Z}$ ✗

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

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.

$j = a \times b$ $\quad a \in S \quad b \in S \quad a < j \quad b < j \quad \Rightarrow \Leftarrow$

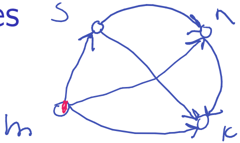
Every $n > 1$ has a prime factorization

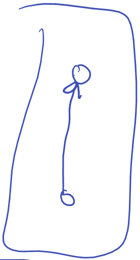
For sake of contradiction, assume this is false. i.e.

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

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 $j = a \times b \wedge 1 < a < j \wedge 1 < b < j$ (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.

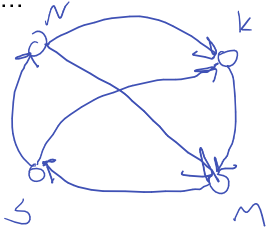
Round-robin tournament cycles



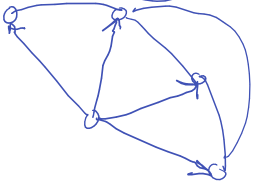


Round-robin tournament $\equiv$ every player faces every other player once.

Consider "cycles" of matchups such as...

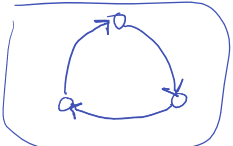


- Naomi beats Kim
- Kim beats Monica
- Monica beats Serena
- Serena beats Naomi

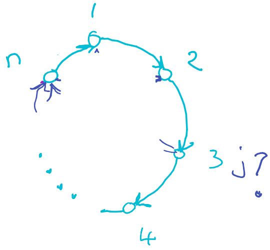


Claim: Any round-robin tournament having at least one cycle has a 3-cycle.

$\forall$



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



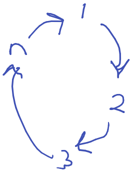
Assume there is a cycle of the form...

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

Let $S = \{ j \in \mathbb{N} \mid p_i > p_n \}$ # beat p_n

then S is non-empty, since $p_{n-1} > p_n$

So by PWO, S has a smallest ele, p_j. $j \neq 1$



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

by def'n of S

by defn of sequence $p_1, p_2 \dots p_n$

$j-1 < j$,

$j-1 \notin S$

∎

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

$P(t)$ $P(s)$ $P(q)$

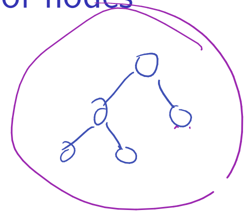
For some recursively defined set S...

1. Define predicate with domain S ~~$P(n)$~~
2. **Basis**: verify $P(x)$ for 'basic' element(s) $x \in S$ $P(\quad)$
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.

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

1. Predicate

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



(proof by CI?)

$P(n)$: any complete bin tree with n nodes ...

Prove: all ~~complete~~ full 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

Let t_1, t_2 be complete binary trees. Assume $P(t_1) \wedge P(t_2)$

Let t be the result of joining t_1 and t_2 under a new root node.

$$Nodes(t) = Nodes(t_1) + Nodes(t_2) + 1$$

By IH, let $k_1, k_2 \in \mathbb{N}$, such that

$$Nodes(t) = (2k_1 + 1) + (2k_2 + 1) + 1$$

$$= 2(k_1 + k_2 + 1) + 1$$

thus $P(t)$

Compare with simple induction

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

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

successor function

$-1 \in \mathbb{N}$

$\{-1, 0, 1, 2 \ldots\}$

[1] Why is this necessary?

Strings with matching parentheses

"language"

// aka set of strings

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

1. $\epsilon \in \mathcal{B}$ `# where` ϵ `denotes the empty string` " length(ϵ) = 0
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?

$(\epsilon) \equiv ()$

ϵ

$()$

$(())$

$((()))$

$()()$

$(())()$

$\epsilon\epsilon\epsilon(\epsilon\epsilon)\epsilon$

A claim about $\mathcal{B}$

$S = \epsilon ab = ab$

7

00007

Define...

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

$\mathcal{B}$ smallest set such that

- $\epsilon \in \mathcal{B}$ (empty string)
- $s \in \mathcal{B} \Rightarrow (s) \in \mathcal{B}$
- $s_1, s_2 \in \mathcal{B} \Rightarrow \underline{s_1 s_2} \in \mathcal{B}$

concatenation

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

s' is a prefix of s

if $\exists$ string z, $s = s'z$

ab has 3 prefixes

1. ϵ
2. a
3. ab

~~4.?~~ $\epsilon\epsilon$

equal

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

$P(s)$: for all prefixes s' of s, $L(s') \geq R(s')$

$P_2(s)$: $L(s) \geq R(s)$

WTS: $\forall s \in B$, $P(s)$

Basis:

Let $s = \epsilon$

s has only 1 prefix, ϵ

$L(\epsilon) = 0 \geq 0 = R(\epsilon)$, so $P(s)$

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

Inductive step

Let $s_1, s_2 \in B$, 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

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

Case 2: $len(s') > len(s_1)$

thus in all cases, $L(s') \geq R(s')$

$P(s)$

$P(n) = ?$

Case 1: $n = 0$

$P(n)$

Case 1: n is even

$P(n)$

Case: n is odd

$P(n)$