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)
- AI 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

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 $j$.

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).

$a, b \not\in S$, 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.
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 $j$.

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

Claim: Any round-robin tournament having at least one cycle has a 3-cycle.
Proof: if a RR tournament has a cycle, it has a 3-cycle

Assume there is a cycle of the form ...

\[ P_i > P_2 > \ldots > P_n > P_i \]

Let \( S = \{ j \in \{N \mid P_i > P_n \} \} \) # 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 defn of \( S \)

by defn of sequence \( P_1, P_2, \ldots, P_n \)

\( j-1 < j \),

\( j-1 \in S \)
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

For some recursively defined set $S$...

1. Define predicate with domain $S$

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

1. Predicate

\[ P(\uparrow) : \forall k \in \mathbb{N} \quad \text{Nodes}(\uparrow) = 2k + 1 \]

(Proof by CS?)

\[ P(n) : \text{any complete bin tree with } n \text{ nodes} \ldots \]
Prove: all \textit{full complete} binary trees have an odd number of nodes

2. Basis

Let \( t \) be a single node.

\[
\text{Nodes}(t) = 1 = 2 \times 0 + 1, \text{ so } P(t)
\]
Prove: all complete binary trees have an odd number of nodes

3. Inductive step

Let $t_1$ and $t_2$ be complete binary trees. Assume $P(t_1) \land P(t_2)$

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

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

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

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

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

thus $P(t)$
Compare with simple induction

Define \( \mathbb{N} \) as the smallest\(^1 \) set such that:

1. \( 0 \in \mathbb{N} \)
2. \( n \in \mathbb{N} \rightarrow n + 1 \in \mathbb{N} \)

Why is this necessary?
Strings with matching parentheses

"language"

// a set of strings

Define $B$ as the smallest set such that...

1. $\varepsilon \in B$                       # where $\varepsilon$ denotes the empty string       \text{length}(\varepsilon) = 0
2. If $b \in B$, then $(b) \in B$
3. If $b_1, b_2 \in B$, then $b_1 b_2 \in B$  # closed under concatenation

Examples of elements?

- $\varepsilon$
- $(\ )$
- $( ( ))$
- $( ( ( )))$
- $( ( ( ( ))))$

- $(\varepsilon) \equiv ( )$
- $(())( )$
- $\varepsilon \varepsilon \in (\varepsilon \varepsilon) \varepsilon$
A claim about $B$

$s = eeab = a6$

Define...

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

Claim: $\forall s \in B$, if $s'$ is a prefix of $s$, then $L(s') \geq R(s')$. 
Prove: prefixes of strings of balanced parens are left-heavy

\[ P(s) : \text{for all prefixes } s' \text{ of } s, \ L(s') \geq R(s') \]

\[ P_2(s) : L(s) = R(s) \]

\text{WTS: } \forall s \in B, \ P(s) \]

\text{Basis:}

Let \( s = \varepsilon \)
\( \varepsilon \) has only 1 prefix, \( \varepsilon \)
\( L(\varepsilon) = 0 \geq 0 = R(\varepsilon), \) so \( P(s) \)
Prove: prefixes of strings of balanced parens are left-heavy

Inductive step

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

Let $s = s_1 s_2$

Let $s'$ be an arbitrary prefix of $s$

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

Then $s'$ is a prefix of $s_1$, 
$L(s') \geq R(s')$

Case 2: $\text{len}(s') > \text{len}(s_1)$

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

$P(s)$
\( p(n) = \begin{cases} 1 & \text{if } n \text{ is odd} \\ 0 & \text{if } n \text{ is even} \end{cases} \)