CSE 311 - HW 6
Eric Boris
Tim Coulter, Alyssa Cote, Josie Thompson, Felix Ayres
November 2019

1 Up the Ladder to the Proof

1. Define x as a binary string, $x \in \{0,1\}*$ with an integer length n, i.e. $|x| = n$ or, when expanded, $x = x_1x_2x_3...x_{n-1}x_n$. Also define the function $f_x(n)$ to be the number of 1s minus the number of 0s in x such that $f_x(n)$ is positive when there are more 1s than 0s, negative when there are more 0s than 1s, and zero when they are equal. Further, define a grammar S such that $S \rightarrow SS|0S1|1S0|\epsilon$ and let $S \implies x$ mean that there are a sequence of substitutions that can be applied to S to generate x. Let P(n) be the claim that if there are an equal number of 1s and 0s in a string x of length n then x could have been generated by S, i.e. $(f_x(n) = 0) \rightarrow (S \rightarrow SS|0S1|1S0|\epsilon \implies x)$. Prove P(n) holds for all n by strong induction.

   · Note that if x is an odd length string, i.e. n is odd, then $f_x(n)$ will never equal zero. (There must be an equal number of 1s and 0s for $f_x(n)$ to equal zero.) Because the premise of the implication is false, the implication is vacuously true. Similarly, if $f_x(n) \neq 0$ then the premise of the implication is false and the implication is vacuously true.

   · Additionally, define a substring y of x such that $y = [x_2, x_n - 1]$, i.e. y is the second element of x to the second to last element of x. And define m to be the integer length of y, $|y| = m$ where m is less than n.

2. Base Case: P(0) holds because $f_x(0) = 0$ (There are an equal number of 1s and 0s. None, to be precise.) and x is the empty string ϵ which is generated by S.

3. Inductive Hypothesis: Assume P(j) holds for arbitrary integers j, k where $0 \leq j \leq k$.

4. Inductive Step: Goal, show that P(k+1) holds for an arbitrary integer k+1, i.e. $(f_x(k+1) = 0) \rightarrow (S \rightarrow SS|0S1|1S0|\epsilon \implies x)$. We will show P(k+1) with 3 cases.

   Case 1: $\forall j \in [k](f_x(j) > 0)$

   1.1 The first element of x must be 1. Case 1 is defined as $\forall j \in [k](f_x(j) > 0)$. If the first element of x were 0 then $\forall j \in [k](f_x(j) > 0)$ would be false. Since this would be a contradiction, x_1 must equal 1.

   1.2 The last element of x must be 0. We know that $f_x(k+1) = 0$. And we know that $\forall j \in [k](f_x(j) > 0)$. Since $f_x(j)$ is greater than 0 for all elements j up to k but is 0 at k+1, we know the value of $f_x(j)$ had to be reduced from being greater than 0 to equal to 0 with the addition of a 0 on the end of x.

   1.3 x is of the form $x = 1x_2x_3...x_{k-1}x_k0$. By 1.1 and 1.2.

   1.4 Substitute y into x, $x = 1y0$ by applying the definition of y.

1.5 y can be generated by S, i.e. $S \implies y$. By the inductive hypothesis, because y is a substring of x it's length is less than k+1. Additionally, x begins with 1 and ends with 0, i.e. an equal number of 1 and 0 so we know that the substring y will also have an equal number of 1s and 0s, i.e. $f_y(m) = 0$.

1.6 $x = 1S0$. Because y can be generated by S we can rewrite x with S in place of y.

1.7 $S \to 1S0 \implies x$. By the definition of S, S can generate 1S0, and since x = 1S0, S can generate x.

Case 2: $\forall j \in [k](f_x(j) < 0)$

2.1 The first element of x must be 0. Case 1 is defined as $\forall j \in [k](f_x(j) < 0)$. If the first element of x were 1 then $\forall j \in [k](f_x(j) < 0)$ would be false. Since this would be a contradiction, x_1 must equal 0.

2.2 The last element of x must be 1. We know that $f_x(k+1) = 0$. And we know that $\forall j \in [k](f_x(j) > 0)$. $f_x(j)$ is less than 0 for all elements j up to k but is 0 at k+1, therefore we know the value of $f_x(j)$ had to be increased from being less than 0 to equal to 0 with the addition of the last element on x. Adding 0 to x would have only reduced the value of $f_x(j)$ further and the only element left, that will also decrease the value of $f_x(j)$, is 1.

2.3 x is of the form $x = 0x_2x_3...x_{k-1}x_k1$. By 2.1 and 2.2.

2.4 Substitute y into x, $x = 0y1$ by applying the definition of y.

2.5 y can be generated by S, i.e. $S \implies y$. By the inductive hypothesis, because y is a substring of x it's length is less than k+1. Additionally, x begins with 0 and ends with 1, i.e. an equal number of 1 and 0 so we know that the substring y will also have an equal number of 1s and 0s, i.e. $f_y(m) = 0$.

2.6 $x = 0S1$. Because y can be generated by S we can rewrite x with S in place of y.

2.7 $S \to 0S1 \implies x$. By the definition of S, S can generate 0S1, and since x = 0S1, S can generate x.

Case 3: Neither Case 1 nor Case 2 holds.

3.1 The value of $f_x(j)$ changes by at most 1, from the property that $|f_x(j) - f_x(j+1)| \leq 1$ for all j.

3.2 There is an integer index b of x, $0 \leq b \leq k+1$ where $f_x(b) = 0$. We know that the function on x has values less than 0 and values greater than 0 so we know that x has some index a where $f_x(a) < 0$ and an index c where $f_x(c) > 0$. This fact and the fact that $f_x(k)$ is continguous on that range informs us that there must be an integer index b such that $f_x(a) \leq f_x(b) \leq f_x(c)$ where $f_x(b) = 0$.

3.3 x can be split into substrings on index b such that $x_{left} = [x_1, x_b]$ and $x_{right} = [x_{b+1}, x_{k+1}]$.

3.4 $S \implies x_{left}$ and $S \implies x_{right}$. We apply the Inductive Hypothesis because $|x_{left}| \leq k$ and $|x_{right}| \leq k$ and since $f_x(k+1) = 0$ and $f_{xleft}(b) = 0$ (there are an equal number of 1s and 0s in both x and x_{left}) there must also be an equal number of 1s and 0s in x_{right}.

3.5 $S \to SS \implies x_{left}x_{right} \implies x$. Because x_{left} and x_{right} can both be generated by S we can say that x can be generated by S.

5. Therefore because an arbitrary length binary string with an equal number of 1s and 0s can be generated by the grammar S with some application of 3 cases, P(n) holds. $\square$

2 Zero Hour: Extra Credit

3 Grammar School

3(a)

$S \to S1|S01|S001|000$

3(b)

$S \to 0S0|1S1|0S|1S|2$

3(c)

$S \to ST|TS|0S1|1S0|2$
$T \to 1T0|0T1|TT|T$
T generates binary strings with an equal number of 1s and 0s.

4 As If: Extra Credit

5 All Your Base

5(a)

$$\begin{aligned}
\text{value}_{10}(\text{Node}(1, \text{Node}(9, \text{Node}(3, \text{null})))) &= 1 + 10 * \text{value}_{10}(\text{Node}(9, \text{Node}(3, \text{null}))) \\
&= 1 + 10 * (9 + 10 * \text{value}_{10}(\text{Node}(3, \text{null}))) \\
&= 1 + 10 * (9 + 10 * (3 + 10 * \text{value}_{10}(\text{null}))) \\
&= 1 + 10 * (9 + 10 * (3 + 10 * (0))) \\
&= 1 + 10 * (9 + 10 * (3)) \\
&= 1 + 10 * (9 + 30) \\
&= 1 + 10 * (39) \\
&= 1 + 390 \\
&= 391
\end{aligned}$$

5(b)

Because value_b will the value of the list L in some base b onto which can be added or multiplied any integers y or r, respectively.

5(c)

Because the the conversion of a number's base to another base, preserves the value of that number, it must be the case that the value of L when converted from base b to base c has the same value despite having a different representation.

5(d)

1. Let P(N) be the claim that $\text{value}_c(\text{convert}_{b\to c}(N)) = \text{value}_b(N)$ where $b, c \in \mathbb{Z}$ and $b, c \geq 2$ for all $L \in$ Lists. Prove P(n) by structural induction.

2. Base Case: P(null) holds because, $\text{value}_c(\text{convert}_{b\to c}(\text{null})) = \text{value}_c(\text{null}) = 0 = \text{value}_b(\text{null})$.

3. Inductive Hypothesis: Assume P(M) holds for an arbitrary $M \in$ Lists, i.e. $\text{value}_c(\text{convert}_{b\to c}(M)) = \text{value}_b(M)$.

4. Inductive Step: Goal, show that P(Node(x, M)) holds for any $x \in \mathbb{Z}$, i.e. $\text{value}_c(\text{convert}_{b\to c}(\text{Node}(x, M))) = \text{value}_b(\text{Node}(x, M))$.

$$\begin{aligned}
\text{value}_c(\text{convert}_{b\to c}(\text{Node}(x, M))) &= \text{value}_c(\text{add}_c(\text{mult}_c(\text{convert}_{b\to c}(M), b), x)) && \text{D. convert}_{b\to c} \\
&= \text{value}_c(\text{mult}_c(\text{convert}_{b\to c}(M), b)) + x && \text{Property of add}_c \\
&= \text{value}_c(\text{convert}_{b\to c}(M)) * b + x && \text{Property of mult}_c \\
&= \text{value}_b(M) * b + x && \text{Inductive Hypothesis} \\
&= \text{value}_b(\text{Node}(x, M)) && \text{Definition of value}_b
\end{aligned}$$

5. Therefore, by structural induction we show that P(n) holds.

6 Acting Up

6(a)

reflexive, symmetric, not antisymmetric, not transitive

6(b)

not reflexive, symmetric, not antisymmetric, not transitive

6(c)

reflexive, symmetric, not antisymmetric, transitive

6(d)

not reflexive, not symmetric, antisymmetric, transitive

7 Machine Shop

Submitted Online