# CS 173, Spring 19 Examlet 9, Part A

**Name:**________________________________________

**NetID:**____________________ **Lecture:** **A** **B**

**Discussion:** **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

(18 points) A Mouse tree is a binary tree containing 2D points such that:

- Each leaf node contains $(3,1)$, $(-2,-5)$, or $(2,2)$.
- An internal node with one child labelled $(a,b)$ has label $(a+1,b-1)$.
- An internal node with two childen labelled $(x,y)$ and $(a,b)$ has label $(x+a,y+b)$.

Use (strong) induction to prove that the point in the root node of any Mouse tree is on or below the line $x=y$.

**Solution:** The induction variable is named__**h**__and it is the __**height**__ of/in the tree.

**Base Case(s):** The shortest Mouse trees consist of a single node containing $(3,1)$, $(-2,-5)$, or $(2,2)$. All three of these points lie on or below the line $x=y$.

**Inductive Hypothesis** [Be specific, don't just refer to "the claim"]: Suppose that the point in the root node of any Mouse tree is on or below the line $x=y$, for trees of height $h=0,1,\ldots,k-1$. ($k \geq 1$).

**Inductive Step:** Let $T$ be a Mouse tree of height $k$. There are two cases.

Case 1: The root of $T$ has one child subtree, whose root contains $(a,b)$. By the inductive hypothesis, $(a,b)$ is on or below $x=y$, i.e. $b \leq a$. By the definition of a Mouse tree, the root of $T$ contains $(a+1,b-1)$. Since $b \leq a$, $b-1 \leq a+1$, so this point is on or below $x=y$.

Case 2: The root of $T$ has two child subtrees, whose roots contain $(x,y)$ and $(a,b)$. Then the root of $T$ contains $(x+a,y+b)$. By the inductive hypothesis, $y \leq x$ and $b \leq a$, so $y+b \leq x+a$. So $(x+a,y+b)$ is on or below $x=y$.

In both cases the root note contains a point on or below $x=y$, which is what we needed to show.

**Name:**__________________________________________

**NetID:**______________________ **Lecture: A B**

**Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6**

(18 points) A palindrome is a string that is the same if you reverse it. For example, `abbabba` and `abaaba` are palindromes. The empty (zero-length) string $\epsilon$ counts as a palindrome.

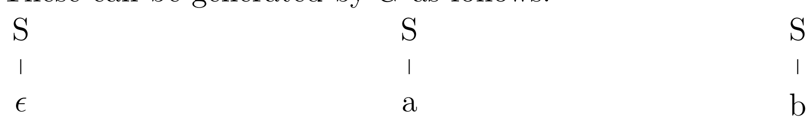
Here is a grammar $G$, with start symbol $S$ and terminal symbols $a$ and $b$.

$$S \rightarrow a\ S\ a \mid b\ S\ b \mid a \mid b \mid \epsilon$$

Use (strong) induction to prove that any palindrome made out of characters a and b can be generated by grammar $G$. That is, show how to build parse trees for these strings. Hint: remove the first and last character from the string.

**Solution:** The induction variable is named __**h**__ and it is the __**length**__ of/in the string.

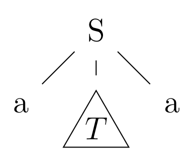
**Base Case(s)**: At $h = 0$, the only palindrome is the empty string $\epsilon$. At $h = 1$, the only palindromes are `a` and `b`. These can be generated by $G$ as follows:



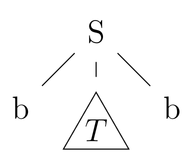

**Inductive Hypothesis** [Be specific, don't just refer to "the claim"]: Suppose that any palindrome made out of a's and b's can be generated by grammar $G$, for strings of length $h = 0, 1, \ldots, k-1$ $(k \geq 2)$.

**Inductive Step:** Let $w$ be a palindrome of length $k$. Since $k \geq 2$, there are two cases

Case 1: $w$ starts with an a. Since $w$ is a palindrome, it must look like $w = ava$ where $v$ is a string of length $k-2$. By the inductive hypothesis, we can build a parse tree $T$ for $v$. We can then build a parse tree for $w$ using the rule $S \rightarrow a\ S\ a$ like this




Case 2: $w$ starts with a b. Since $w$ is a palindrome, it must look like $w = bvb$ where $v$ is a string of length $k-2$. By the inductive hypothesis, we can build a parse tree $T$ for $v$. We can then build a parse tree for $w$ using the rule $S \rightarrow b\ S\ b$ like this




In both cases, we can build a parse tree for the string $w$, which is what we needed to show.

**Name:**________________________________________

**NetID:**____________________ **Lecture: A B**

**Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6**

(18 points) Recall that a node in a full binary tree must have either 0 or 2 children. A Shark tree is a full binary tree in which each node is colored orange or blue, such that:

- If $v$ is a leaf node, then $v$ is colored orange.
- If $v$ has two children of the same color, then $v$ is colored blue.
- If $v$ has two children of different colors, then $v$ is colored orange.

Use (strong) induction to show that the root of a Shark tree is blue if and only if the tree has an even number of leaves. You may assume basic divisibility facts e.g. the sum of two odd numbers is even.

**Solution:** The induction variable is named **h** and it is the **height** of/in the tree.

**Base Case(s):** $h = 0$. The tree consists of a single node, which must be colored orange. The claim holds because the tree has an odd number of leaves (i.e. just one).

**Inductive Hypothesis** [Be specific, don't just refer to "the claim"]: Suppose that the root of a Shark tree is blue if and only if the tree has an even number of leaves, for trees of height $h = 0, 1, \ldots, k-1$ ($k$ an integer $\geq 1$).

**Inductive Step:** Let $T$ be a Shark tree of height $k$. Since $T$ is a full binary tree with height $\geq 1$, $T$ consists of a root plus two child subtrees. There are three cases:

Case 1: The root of $T$ is blue and the roots of the child subtrees are both orange. By the inductive hypothesis, both subtrees have an odd number of leaves. Therefore $T$ has an even number of leaves.

Case 2: The root of $T$ is blue and the roots of the child subtrees are both blue. By the inductive hypothesis, both subtrees have an even number of leaves. Therefore $T$ has an even number of leaves.

Case 3: The root of $T$ is orange, the root of one child subtree (call it $T_1$) is orange, and the root of the old child subtree (call it $T_2$) is blue. By the inductive hypothesis, $T_1$ has an odd number of leaves and $T_2$ has an even number of leaves. Therefore $T$ has an odd number of leaves.

In all three cases, the root of $T$ is blue if and only if $T$ has an even number of leaves, which is what we needed to prove.

**Name:**________________________________________

**NetID:**________________________ **Lecture:** **A** **B**

**Discussion:** **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

(18 points) A Horse tree is a binary tree whose nodes contain integers such that

- Every leaf node contains 5, 17, or 23.
- A node with one child contains the same number as its child.
- A node with two children contains the value $x(y+1)$, where $x$ and $y$ are the values in its children.

Use strong induction to prove that the value in the root of a Horse tree is always positive.

**Solution:** The induction variable is named **h** and it is the **height** of/in the tree.

**Base Case(s):** The smallest Horse trees consist of a single root node, which is also a leaf. By the definition of Horse tree, this must contain 5, 17, or 23, all of which are positive.

**Inductive Hypothesis** [Be specific, don't just refer to "the claim"]: Suppose that the root of a Horse tree of height $h$ is always positive, for $h = 0, \ldots, k-1$.

**Inductive Step:** Let $T$ be a Horse tree of height $k$. There are two cases for what the top of $T$ looks like.

Case 1: $T$ consists of a root $r$ with a single subtree $S$ under it. $r$ contains the same number as the root of $S$. Since $S$ must be shorter than $k$, its root contains a positive number by the inductive hypothesis. Since $r$ has the same label, $r$ contains a positive number.

Case 2: $T$ consists of a root $r$ with a two subtrees $S_1$ and $S_2$. Suppose that the roots of $S_1$ and $S_2$ contain the numbers $x$ and $y$. Then, by the definition of Horse tree, $r$ contains $x(y+1)$.

Since $S_1$ and $S_2$ are shorter than $k$, $x$ and $y$ must be positive by the inductive hypothesis. Since $y$ is positive, so is $y+1$. Since $x$ and $y+1$ are positive, so is $x(y+1)$. So the root of $T$ contains a positive number.

So, in either case, the root of $T$ contains a positive number.

**CS 173, Spring 19** **Examlet 9, Part A** 

**Name:**________________________________________

**NetID:**____________________ **Lecture:** **A** **B**

**Discussion:** **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

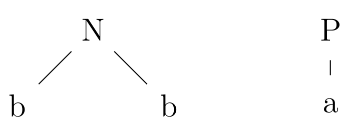
(18 points) Here is a grammar $G$, with start symbols $N$ and $P$, and terminal symbols $a$ and $b$.

$$
\begin{aligned}
N &\rightarrow P\,a \mid b\,b \\
P &\rightarrow P\,N \mid a
\end{aligned}
$$

Use (strong) induction to prove that any tree matching (aka generated by) grammar $G$ has an even number of leaves if and only if its root has label $N$.

**Solution:** The induction variable is named **h** and it is the **height** of/in the tree.

**Base Case(s)**: The shortest trees matching grammar $G$ have height $h = 1$. There are two such trees, which look like




The tree with root $N$ has an even number of leaves and the tree with root $P$ has an odd number of leaves, so the claim holds.

**Inductive Hypothesis** [Be specific, don't just refer to "the claim"]: Suppose that all trees $T$ matching grammar $G$ with heights $h = 1, 2, \ldots, k-1$ have an even number of leaves if and only if the root of $T$ has label $N$, for some integer $k \geq 2$.

**Inductive Step:** Let $T$ be a tree of height $k$ matching grammar $G$, where $k \geq 2$. There are two cases:

Case 1: $T$ consists of a root with label $P$, with a left child $T_1$ with root label $P$ and a right child $T_2$ with root label $N$. By the inductive hypothesis, $T_1$ has an odd number of leaves and $T_2$ has an even number of leaves. Since the leaves in $T$ are exactly the leaves in $T_1$ plus the leaves in $T_2$, $T$ has an odd number of leaves.

Case 2: $T$ consists of a root with label $N$, whose children are a subtree $T_1$ on the left and a leaf node on the right. $T_1$ has root label $P$ and the leaf has label $a$. By the inductive hypothesis, $T_1$ has an odd number of leaves, so $T$ must have an even number of leaves.

In both cases, $T$ has an even number of leaves if and only if the root of $T$ has label $N$, which is what we needed to show.

**Name:**________________________________________

**NetID:**____________________ **Lecture:** **A** **B**

**Discussion:** **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

(18 points) Recall that a node in a full binary tree is either a leaf or has exactly two children. A Possum tree is a full binary tree whose leaves are all orange and whose root is blue.

Use (strong) induction to prove that a Possum tree contains a blue node with (at least) one orange child.

**Solution:** The induction variable is named **h** and it is the **height** of/in the tree.

**Base Case(s):** A Possum tree must have height at least 1 because the root is a different color from the leaves. At $h = 1$, the tree consists of a blue root with two orange children, so the claim is true.

**Inductive Hypothesis** [Be specific, don't just refer to "the claim"]: Suppose that every Possum tree contains a blue node with (at least) one orange child, for heights $h = 0, \ldots, k - 1$.

**Inductive Step:** Consider a Possum tree $T$ with height $k$. $T$ consists of a root node $x$ (colored blue) and two child subtrees $T_L$ and $T_R$.

There are three cases:

Case 1: $T_L$ has an orange root. Then $x$ is the required blue node with at least one orange child.

Case 2: $T_R$ has an orange root. Then $x$ is the required blue node with at least one orange child.

Case 3: $T_L$ and $T_R$ both have blue roots. Then we can apply the inductive hypothesis to $T_L$. By the inductive hypothesis, $T_L$ must contain a blue node with at least one orange child. Since $T_L$ is a subtree of $T$, this node and its children also live in $T$.

In all three cases, $T$ contains a blue node with at least one orange child, which is what we needed to prove.

**Name:**________________________________________

**NetID:**____________________ **Lecture:** **A** **B**

**Discussion:** **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

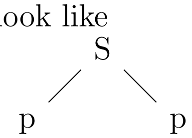
(18 points) Here is a grammar $G$, with start symbol $S$ and terminal symbols $a$ and $p$.

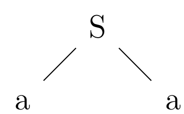
$$S \rightarrow S\ S \mid p\ S\ p \mid p\ p \mid a\ a$$

Use (strong) induction to prove that any tree matching (aka generated by) grammar $G$ has an even number of nodes with label p. Use $P(T)$ as shorthand for the number of p's in a tree $T$.

**Solution:** The induction variable is named **h** and it is the **height** of/in the tree.

**Base Case(s)**: The shortest trees matching grammar $G$ have height $h = 1$. There are two such trees, which look like




Both of these contain an even number of nodes with label p.

**Inductive Hypothesis** [Be specific, don't just refer to "the claim"]:

Suppose that all trees $T$ matching grammar $G$ with heights $h = 1, 2, \ldots, k-1$ have $P(T)$ even, for some integer $k \geq 2$.

**Inductive Step:** Let $T$ be a tree of height $k$ matching grammar $G$, where $k \geq 2$. There are two cases:

Case 1: $T$ consists of a root with label $S$ plus two child subtrees $T_1$ and $T_2$. By the inductive hypothesis $P(T_1)$ and $P(T_2)$ are both even. But $P(T) = P(T_1) + P(T_2)$. So $P(T)$ is also even.

Case 2: $T$ consists of a root with label $S$ plus three children. The left and right children are single nodes containing label $p$. The center child is a subtree $T_1$. By the inductive hypothesis, $P(T_1)$ is even. $P(T) = P(T_1) + 2$. So $P(T)$ is also even.

In both cases $P(T)$ is even, which is what we needed to show.

**Name:**________________________________________

**NetID:**________________________ **Lecture: A B**

**Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6**

(18 points) Recall that a node in a full binary tree is either a leaf or has exactly two children. A Snake tree is a full binary tree whose leaves are all blue and whose root is orange.

Use (strong) induction to prove that a Snake tree contains an orange node with two blue children.

**Solution:** The induction variable is named **h** and it is the **height** of/in the tree.

**Base Case(s):** A Snake tree must have height at least 1 because the root is a different color from the leaves. At $h = 1$, the tree consists of an orange root with two blue children, so the claim is true.

**Inductive Hypothesis** [Be specific, don't just refer to "the claim"]: Suppose that every Snake tree contains an orange node with two blue children, for heights $h = 0, \ldots, k-1$.

**Inductive Step:** Consider a Snake tree $T$ with height $k$. $T$ consists of a root node $x$ (colored orange) and two child subtrees $T_L$ and $T_R$.

There are three cases:

Case 1: Both $T_L$ and $T_R$ have blue roots. Then $x$ is the required orange node with two blue children.

Case 2: $T_L$ has an orange root. Then we can apply the inductive hypothesis to $T_L$. By the inductive hypothesis, $T_L$ must contain an orange node with two blue children. Since $T_L$ is a subtree of $T$, this node and its children also live in $T$.

Case 3: $T_R$ has an orange root. Then we can apply the inductive hypothesis to $T_R$. By the inductive hypothesis, $T_R$ must contain an orange node with two blue children. Since $T_R$ is a subtree of $T$, this node and its children also live in $T$.

In all three cases, $T$ contains an orange node with two blue children, which is what we needed to prove.