

Homework 5

Abhi Agarwal

1 Name Analysis

I discussed Question 1.2 with Tyler Palsulich as I did not quite understand what it was trying to ask. He gave me some clarification, but we did not discuss the answer in detail.

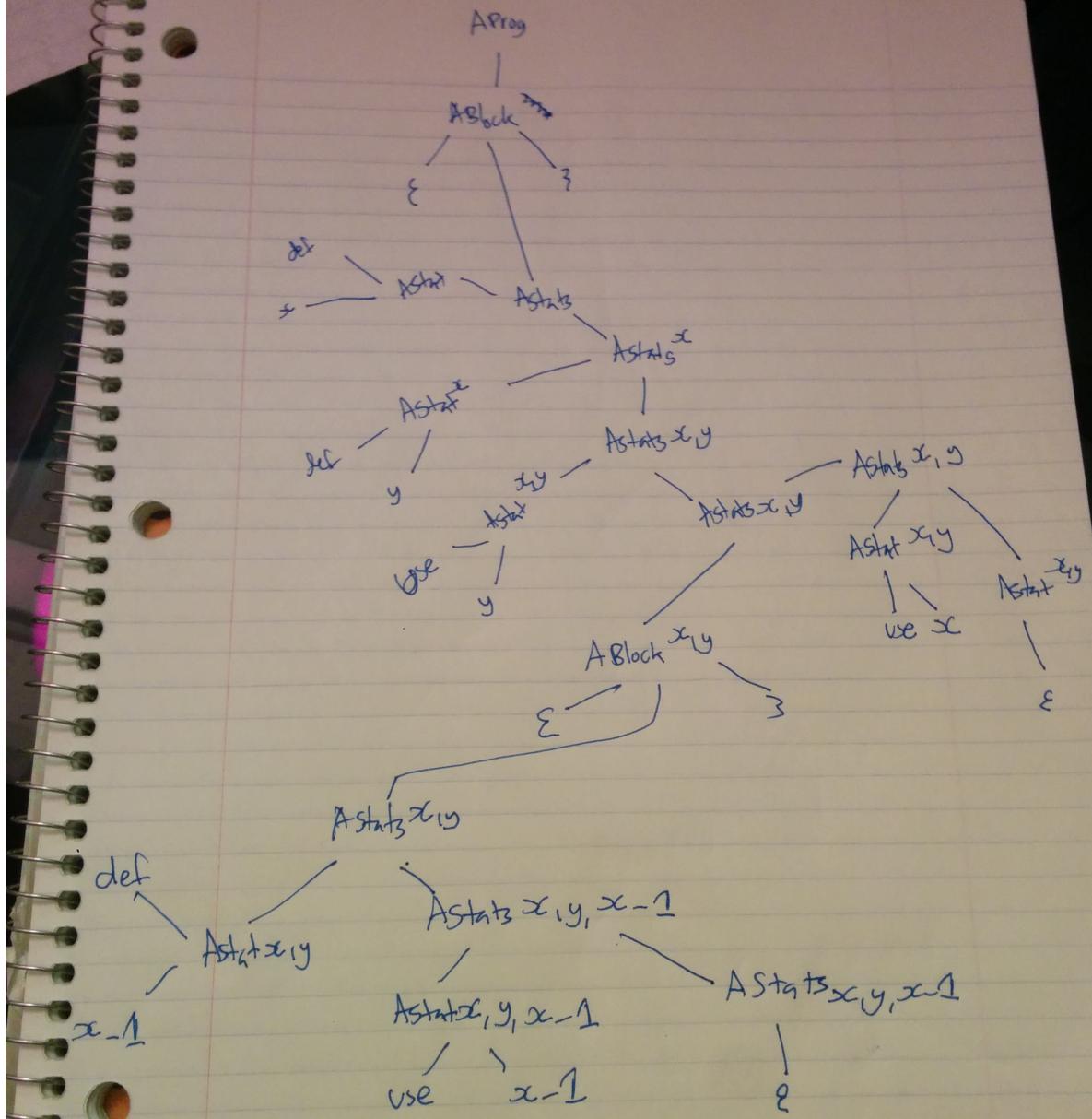
1.1 Question 1.1

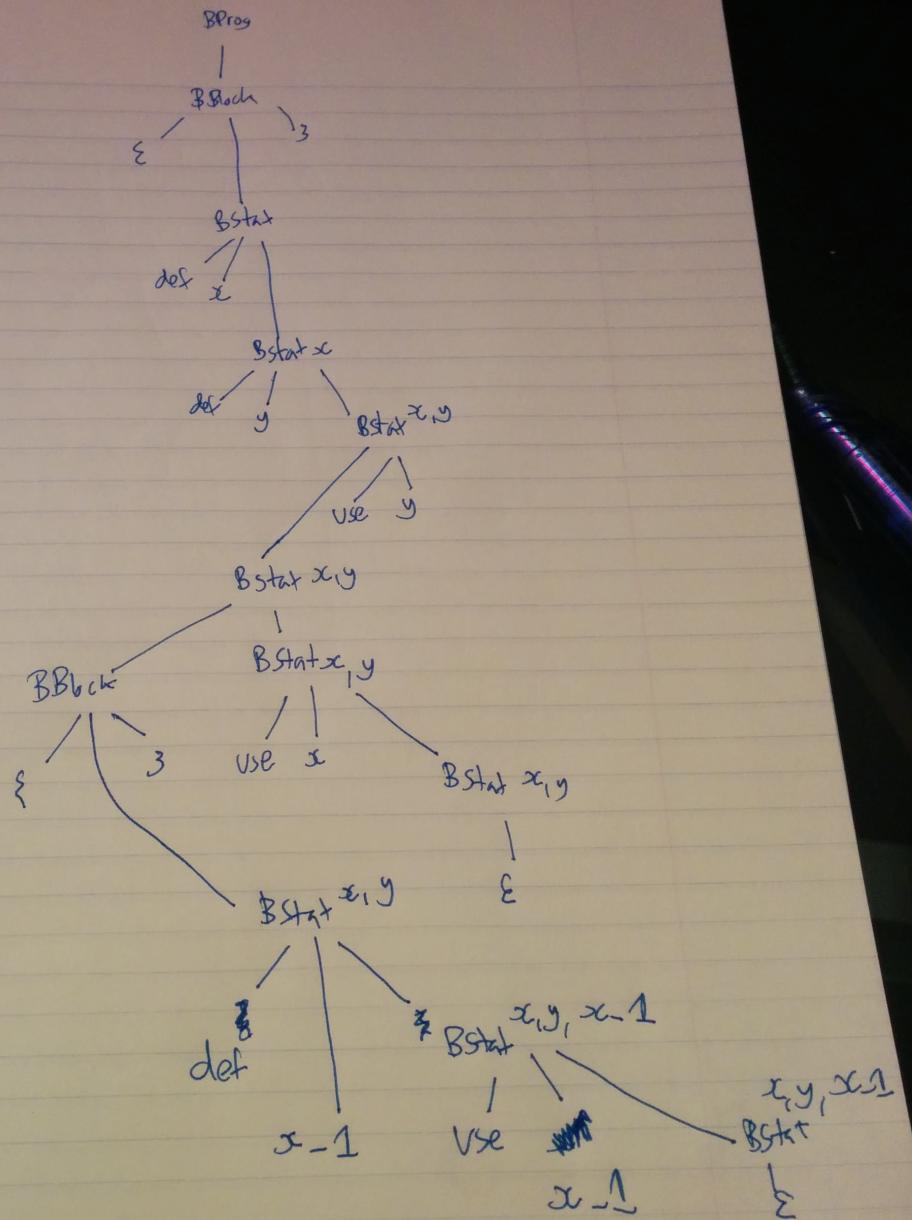
For the first grammar the section where $AStats \rightarrow AStats \rightarrow AStats \rightarrow \epsilon$, and $AStats \rightarrow AStat \rightarrow use \rightarrow x_1$ has to be in an order where $AStat \rightarrow use \rightarrow x_1$ is before the other. Both of the grammars generate the same language, but we can see that they generate different parse trees.

For the second grammar I notice that my syntax is not too different for { and }, and ϵ . But under BBlock it is { and }, and under BStat it is ϵ

My x_1 is a different x, I did not find any other way to define it. It is just an x defined in a different scope.

The two grammars are the same language.





1.2 Question 1.2

I didn't quite understand this, but my logic was to try and keep track of the variables and when that particular id was used I would assign yes. I could not just put everything into i, but I had to put it into ok so I used it as a store, as a yes, and as a false.

$\begin{array}{l} AProg \rightarrow ABlock \\ ABlock \rightarrow \{AStats\} \\ (Converting) AStats \rightarrow AStat \quad AStats \mid \epsilon \\ AStat \rightarrow def\ id; \mid use\ id; \mid ABlock \end{array}$		$\begin{array}{l} AProg \rightarrow ABlock_1 \\ ABlock \rightarrow \{AStats_1\} \\ AStats \rightarrow AStats_1 \quad AStats_2 \mid \epsilon \\ AStat \rightarrow def\ id; \mid use\ id; \\ \mid ABlock_1 \end{array}$
<u>Production</u>		<u>Rules</u>
$AProg \rightarrow ABlock_1$		$ABlock_1.i = \{\}$
$ABlock \rightarrow \{AStats_1\}$		$AStats_1.i = ABlock.i$
$AStats \rightarrow AStat_1 \quad AStats_2 \mid \epsilon$		$AStat_1 = AStats.i; \quad AStat_2 = AStats.i \quad AStat_1.OK$
$AStat \rightarrow def\ id;$		$AStats.OK = \{id.name\}$
$\mid use\ id;$		$AStats.OK = true$
$\mid ABlock_1$		$ABlock_1.i = AStat.i; \quad AStat.OK = false$
<ul style="list-style-type: none"> One inherited attribute, i Synthesized attribute, ok 		
<ul style="list-style-type: none"> - Need to have all the inside variables within a variable - {} means empty right now - PLUS meaning combine the two sets. - I'm not too familiar with how to store the "Yes". 		

1.3 Question 1.3

Converting $BProg \Rightarrow BBlock$ to $BProg \Rightarrow BBlock - 1$ $BBlock \Rightarrow \{BStat\}$ to $BBlock \Rightarrow \{BStat - 1\}$ $BStat \Rightarrow def id; BStat$ to $BStat \Rightarrow def id; BStat - 1$ $ use id; BStat$ to $ use id; BStat - 1$ $ BBlock BStat$ to $ BBlock - 1 BStat - 2$ $ \epsilon$ to $ \epsilon$	
<u>Production</u>	<u>rule</u>
$BProg \rightarrow BBlock - 1$	$BBlock - 1 := \{\}$ empty
$BBlock \rightarrow \{BStat - 1\}$	$BStat - 1 := BBlock - 1.i = BBlock.i;$ $BStat - 1 := BBlock - 1.i = BBlock.i;$
$BStat \rightarrow def id; BStat - 1$	$BStat - 1.i = BStat.i PLUS \{id.name\}$
$ use id; BStat - 1$	$BStat - 1.i = BStat.i PLUS \{id.name\}$
$ BBlock - 1 BStat - 2$	$BBlock.i = BStat.i; BStat - 2.i = BStat.i;$ $BBlock.i = BStat.i; BStat - 2.i = BStat.i;$ $BStat.ok = false$
$ \epsilon$	

- following same framework with
 $i \& ok$,
 - PLUS is adding of the sets