Homework 4

Answer the questions below, and submit electronically via elearning. Make sure you submit a couple hours early at the latest and double check your submission to ensure everything is in order *before* the submission time. Your answers should be submitted as a ".doc", ".docx", or ".pdf" file. Your answers should be typed, but scans or pictures of hand drawn figures and diagrams may be included in the file when needed.

Due: Friday, March 29th 11:59pm

Syntax and Semantics (Semantics only)

1. **(10pt)** Recall the following grammar from homework 3.

1	<prod_file>	→	<prod_list>
2	<prod_list>	→	<prod>
3		|	<prod> **ENDLINE** <prod_list>
4	<prod>	→	**ID** "->" <body>
5	<body>	→	<body_item>
6		|	<body_item> "," <body>
7	<body_item>	→	**ID**
8		|	"&"
9		|	"%"
10		|	"*"

Consider the following attribute computation functions and predicate functions associated with the rules above.

1. Predicate: "main" ∈ <prod_list>.ids
2. <prod_list>.ids = {<prod>.id}
3. <prod_list>[0].ids = {<prod>.id} ∪ <prod_list>[1].ids
4. <prod>.id = **ID**.id
   <body>.id = **ID**.id
5. <body_item>.id = <body>.id
6. <body_item>.id = <body>[0].id
   <body>[1].id = <body>[0].id
7. Predicate: <body_item>.id ≠ ID.id

In class, we constructed the attributed grammar to specify the details of the statement variables must be defined before they are used.'' In this question, you will be doing the opposite. I have provided the attributed grammar and you must identify the informal statements it is

detailing. There are two such statements. Specify these statements informally with an English sentence for each. Do not directly reference any of the attributes, predicate functions, or attribute computation functions. For convenience here are some definitions you can use.

- Anything generated from <prod> is a *production*.
- Anything generated from <body> is a *body* of the *production*.
- Anything generated from <body_item> is an *item*.
- The ID to the left of the arrow (->) in a production is the *name* of the production.

Give a brief description on how they are enforced (The explanation can and should reference the attributed grammar). For each statement, list the attributes involved (do not forget they are attached to symbols), and specify whether or not each attribute is synthesized, inherited, or intrinsic.

For the following programs determine if they satisfy the static semantics.

(a)
```
main -> step,show
step -> &,&
show -> %,&,*
```

(b)
```
compute -> *,&,%,compute2
compute2 -> *,&,%,compute3
compute3 -> *,&,%,&
main -> &,&,&,%,&,&,compute
```

2. **(20pt)** Recall the attributed grammar from question 1. Consider the operational semantics given as a translation in the attached "operational_semantics.py".

   NOTE: The format is similar to our example from class. The semantics in this questions is *different* from the one presented in the denotational semantics questions.

   For the following programs, what will be printed when it program is ran? Briefly explain why.

(a)
```
main -> step,show
step -> &,&
show -> %,&,*
```

(b)
```
compute -> *,&,%,compute2
compute2 -> *,&,%,compute3
compute3 -> *,&,%,&
main -> &,&,&,%,&,&,compute
```

3. **(20pt)** Recall the attributed grammar from question question 1. Consider the following denotational semantics.

$$
\begin{aligned}
&\text{denote}(< \text{prod} > \text{ENDLINE} < \text{prod\_list} >) &&= \text{denote}(< \text{prod} >) \cup \text{denote}(< \text{prod\_list} >)\\
&\text{denote}(\text{ID } \text{“- >”} < \text{body} >) &&= \{(\text{ID}, \text{bodysem}(< \text{body} >))\}\\
&\text{bodysem}(< \text{body\_item} > \text{“,”} < \text{body} >) &&= \text{bodysem}(< \text{body\_item} >) + \text{bodysem}(< \text{body} >)\\
&\text{bodysem}(\text{ID}) &&= \text{if ID} = \text{main then } 10 \text{ else } 1\\
&\text{bodysem}(\text{“\&”}) &&= 2\\
&\text{bodysem}(\text{“\%”}) &&= 100\\
&\text{bodysem}(\text{“*”}) &&= -10
\end{aligned}
$$

NOTE: The format is similar to our example from class. The semantics in this questions is *different* from the operational semantics question.

For the following programs, what is the mathematical meaning? Briefly explain why. HINT: What mathematical object is being used to represent the meaning of programs?

(a)
```
main -> step,show
step -> &,&
show -> %,&,*
```

(b)
```
compute -> *,&,%,compute2
compute2 -> *,&,%,compute3
compute3 -> *,&,%,&
main -> &,&,&,%,&,&,compute
```

Basic Fuzzing

4. **(50pt)** For this question you will be writing three fuzzers: A mutation-based, a generation-based, and a protocol-based fuzzer. Choose between one an three programs. The programs could be code from a personal project of yours, a project from another class, or a program you found online. Make sure the program(s) chosen will properly demonstrate the uniqueness of each fuzzer.

   - A mutation-based fuzzer needs a sophisticated enough input format to make mutating it useful. There should be multiple ways to mutate the input.
   - A generation-based fuzzer needs sophisticated enough behavior to make knowledge of it useful.
   - A protocol-based fuzzer needs a sophisticated enough input format to make a grammar useful. Find a program that uses an input format with a specifications (do not use JSON)

   Write three fuzzers (like I did in class) to test these programs: A mutation fuzzer, generation fuzzer, and a protocol-based fuzzer. Run a few tests from each fuzzer on the program. What did you learn about the programs? Did you discover any bugs you did not know were there? You may use Java,C,C++, or Python to write your fuzzers. Be sure to turn in the fuzzer code with your assignment.