Name: Gabe Eapen
EID: eapengp
Class: EE382V (Software Testing Class)
HW#4 – Due Nov 28, 2014 @ 11:59pm

1. Section 5.2 Question 1 (Page 189) [answer this question only for mutants 2 and 5 in Figure 5.1 (and not the other mutants mentioned in the text)].
2. Section 5.2 Question 2 (Page 189)
3. Section 5.5 Question 5 (Page 209) [answer only parts (a) and (b)]
4. Section 5.5 Question 6 (Pages 209–210) [answer only part (a)]

Section 5.2 Question 1 (Page 189) [Answer]

For Mutant Δ2

Reachability: *true*
Infection: $A \neq B$
Propagation: $A \neq B$
Full Specification: $true \wedge (A \neq B) \wedge (A \neq B)$
Test Case: $A = 1, B = 2$

For Mutant Δ5

Reachability: $B < A$
Infection: $A \neq B$
Propagation: $A \neq B$
Full Specification: $(B < A) \wedge (A \neq B) \wedge (A \neq B)$
Test Case: $A = 2, B = 1$

Section 5.2 Question 2 (Page 189) [Answer]

For findVal()

a) The for loop is always executed. So a test input that skips it cannot be found
b) The for loop is always executed. So when it is executed, an infection occurs
c)
```
int[] numbers new int[2]; numbers [0] = 1; numbers [1] = 2; val = 2;
```
d)
```
int[] numbers new int[2]; numbers [0] = 2; numbers [1] = 1; val = 2;
```

For sum()

a) An empty integer array x (no elements)
```
int[] x new int[];
```
b)
```
int[] x new int[2]; x[0] = 0; x[1] = 0;
```
c)
```
int[] x new int[2]; x[0] = 1; x[1] = -1;
```
d)
```
int[] x new int[2]; x[0] = 1; x[1] = 2;
```

Section 5.5 Question 5 (Page 209) [Answer]

a)

For the provided grammar, the following strings can be generated

42

Derivation

$$val\ ::=\ \boldsymbol{number}$$
$$val\ ::=\ digit+$$

4 2 +

Derivation

$$val\ ::=\ \boldsymbol{val}\, pair$$
$$val\ ::=\ number\ \boldsymbol{pair}$$
$$val\ ::=\ \boldsymbol{number}\ \ \boldsymbol{number}\ \ op$$
$$val\ ::=\ digit+\ \ digit+\ \ op$$

4 2 7 - *

Derivation

$$val\ ::=\ \boldsymbol{val}\, pair$$
$$val\ ::=\ number\ \boldsymbol{pair}$$
$$val\ ::=\ number\ number\ \boldsymbol{pair}\ op$$
$$val\ ::=\ \boldsymbol{number}\ \boldsymbol{number}\ \boldsymbol{number}\ op\ \ op$$
$$val\ ::=\ digit+\ \ digit+\ \ digit+\ \ op\ \ op$$

4 2 - 7 *

Derivation

$$val\ ::=\ \boldsymbol{val}\, pair$$
$$val\ ::=\ number\ \boldsymbol{pair}\ \boldsymbol{pair}$$
$$val\ ::=\ \boldsymbol{number}\ \boldsymbol{number}\ \ op\ \ \boldsymbol{number}\ \ op$$
$$val\ ::=\ digit+\ \ digit+\ \ op\ \ digit+\ \ op$$

b)

The following strings can be generated only by the mutated grammar and not the original grammar

4 + 2

Derivation

$$val\ ::=\ \boldsymbol{val}\, pair$$
$$val\ ::=\ number\ \boldsymbol{pair}$$
$$val\ ::=\ \boldsymbol{number}\ op\ \boldsymbol{number}$$
$$val\ ::=\ digit+\ \ op\ \ digit+$$

Section 5.5 Question 6 (Page 209–210) [Answer]

a)

123-4567 (phone Number)

012-3456 (**non-phone** Number; exchangePart needs to start with **1 or 2**)

109-1212 (phone Number)

346-9900 (**non-phone** Number; exchangePart needs to start with **1 or 2**)

113-1111 (phone Number)

For exchangePart = $D_1D_2D_3$,

D_1 = 1 or 2

D_2 = 0 or 1 or 2

D_3 = 3 or 4 or 5 or 6 or 7 or 8 or 9

b)

Original:

$exchangePart\ ::=\ special\ zeroOrSpecial\ other$

Mutation:

$exchangePart\ ::=\ special\ ordinary\ other$

<u>In Mutated Grammar Only:</u>

None of the provided strings fall in this category. But an external example is: 1**3**3-4567

<u>In Original Grammar Only:</u>

$ordinary\ ::=\ zero\ |\ special\ |\ other$

$ordinary\ ::=\ zeroOrSpecial\ |\ other$

So we cannot find a string that is only in original grammar as exchangePart in original grammar is always a subset of exchangePart in mutation grammar.

<u>In both Original & Mutated Grammar:</u>

As exchangePart in original grammar is always a subset of exchangePart in mutation grammar, the following strings that satisfy the original grammar also satisfies the mutation grammar.

123-4567, 109-1212, 113-1111