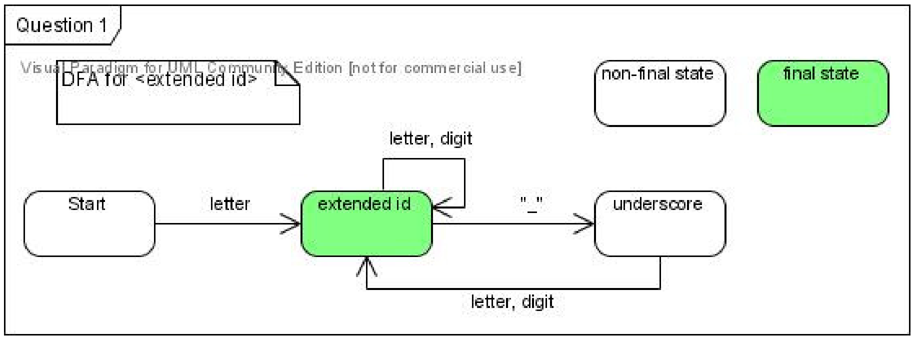
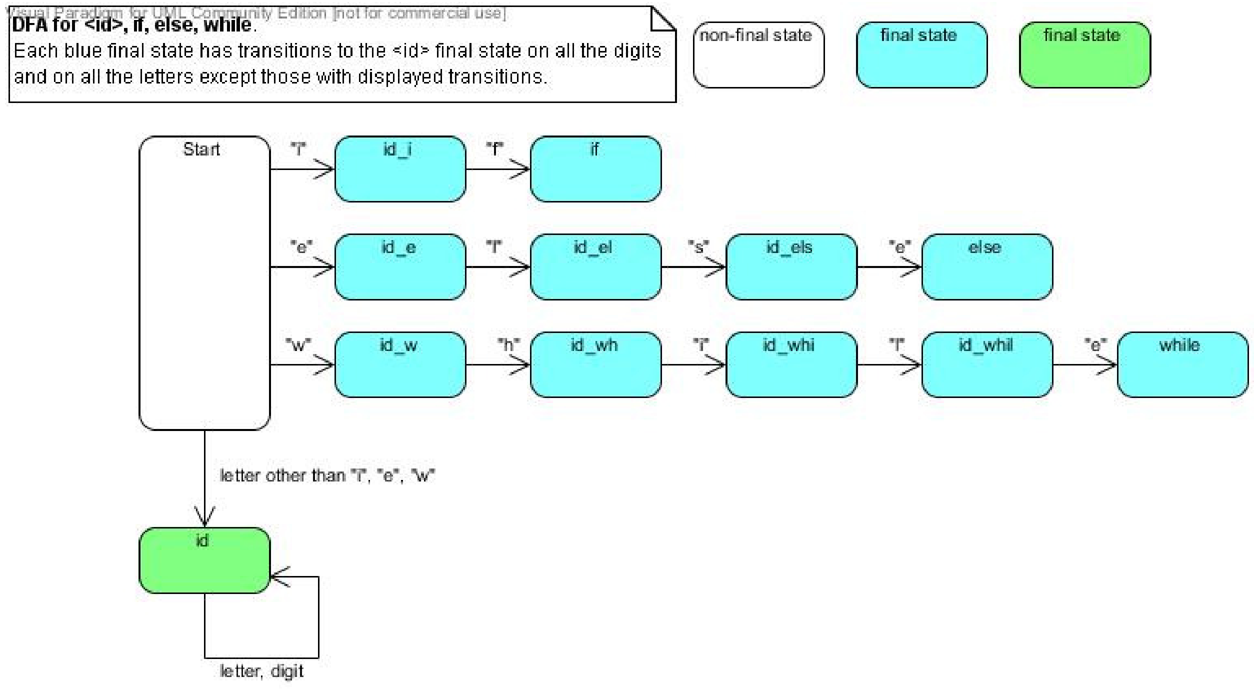
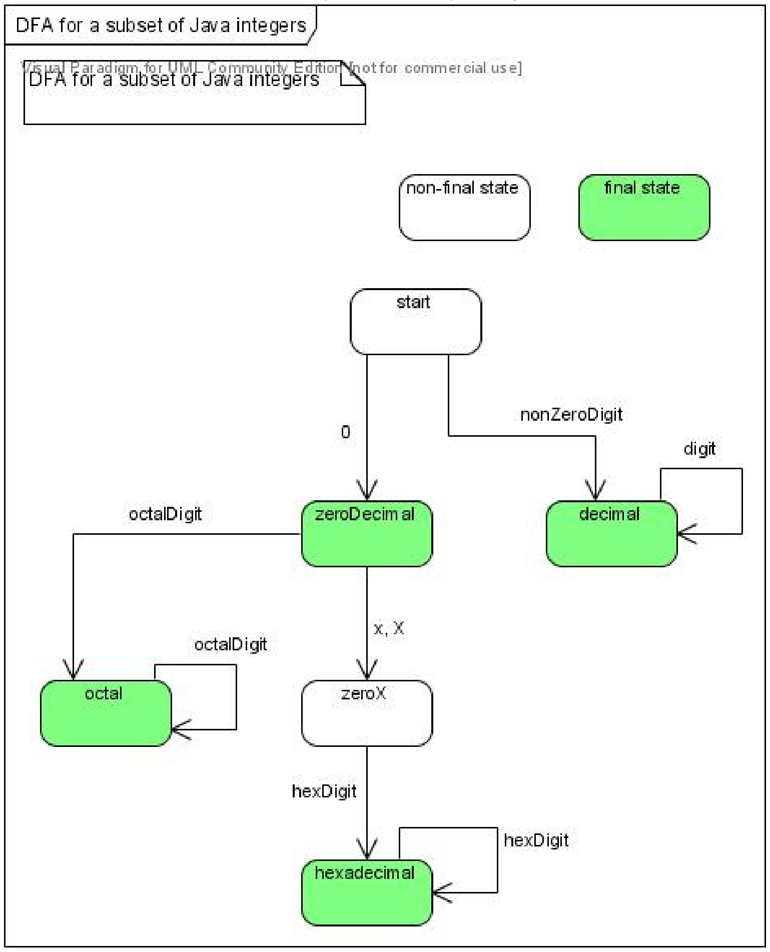
1.



4. For the sake of clarity, arrows for these transitions are omitted in the diagram: **Each blue final state has transitions to the *id* final state on all the digits and on all the letters except those with displayed transitions.**


Visual Paradigm for UML Community Edition [not for commercial use]
DFA for <id>, if, else, while.
Each blue final state has transitions to the <id> final state on all the digits and on all the letters except those with displayed transitions.
non-final state
final state
final state
Start
"i"
id_i
"f"
if
"e"
id_e
"l"
id_el
"s"
id_els
"e"
else
"w"
id_w
"h"
id_wh
"i"
id_whi
"l"
id_whil
"e"
while
letter other than "i", "e", "w"
id
letter, digit

5.


DFA for a subset of Java integers
Visual Paradigm for UML Community Edition [not for commercial use]
DFA for a subset of Java integers
non-final state
final state
start
0
nonZeroDigit
digit
octalDigit
zeroDecimal
decimal
x, X
octalDigit
octal
zeroX
hexDigit
hexDigit
hexadecimal

7.

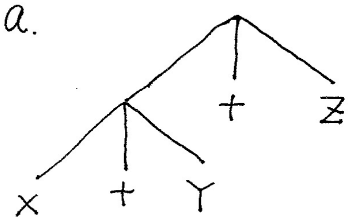
```
void sequence()
{
    if ( t is "(" )
    {
        getToken();
        elements();
        if ( t is ")" )
            getToken();
        else
            print( "Error: ) expected" );
    }
    else
        print( "Error: ( expected" );
}

void elements()
{
    element();
    while ( t is "," )
    {
        getToken();
        element();
    }
}

void element()
{
    if ( t is <id> )
        getToken();
    else
        sequence();
}
```

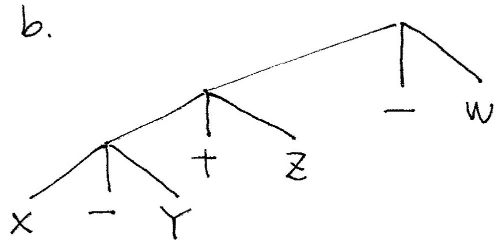
8.2 Since the given production rules are in iterative form, left associativity is used.

a.



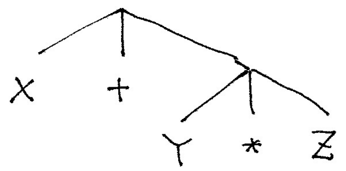
```
push X
push Y
add
push Z
add
```

b.



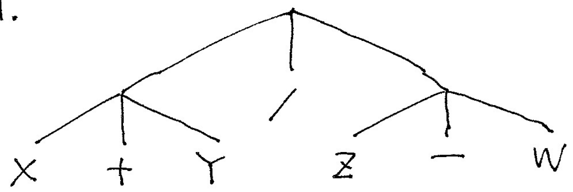
```
push X
push Y
sub
push Z
add
push W
sub
```

c.



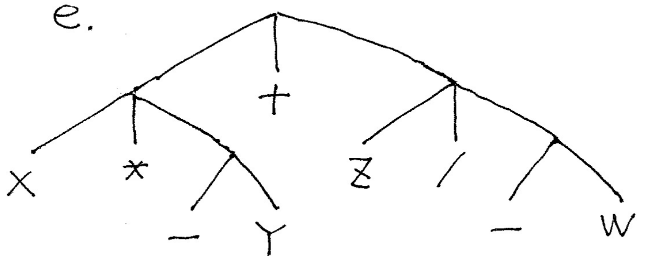
```
push X
push Y
push Z
mul
add
```

d.



```
push X
push Y
add
push Z
push W
sub
div
```

e.



```
push X
push Y
neg
mul
push Z
push W
neg
div
add
```

9.

```
void A()
{
    if ( t is "+" || t is "-" )
    {
        getToken();
        B();
        B();
    }
    else
        print( "Error: + or - expected" );
}

void B()
{
    if ( t is <id> )
        getToken();
    else
        A();
}
```