Compiler
Assignment 6
Attribute Grammars and Top-Down Translator

403410033 資工三 曾俊宏

May 24, 2017

1 Question 1

a) S-attributed attribute grammar

production	semantic rules
$S \rightarrow L\ R$	S.val = L.val + R.val / R.base;

$R \rightarrow .\ L$	R.val = L.val
	R.base = L.base;

$R \rightarrow \epsilon$	R.val = 0;
	R.base = 0;

$L \rightarrow B\ L_s$	L.base = Ls.base;
	L.val = B.val * L.base + Ls.val;
	
	L.base *= 2;

$L_s \rightarrow B\ L_{s1}$	Ls.base = Ls1.base;
	Ls.val = B.val * Ls.base + Ls1.val;
	
	Ls.base *= 2;

$L_s \rightarrow \epsilon$	Ls.base = 1;
	Ls.val = 0;

$B \rightarrow 0$	B.val = 0;
$B \rightarrow 1$	B.val = 1;

b) S-attributed attribute grammar → top-down translator

. and $ are surrounded by single quotation mark due to latex rendering issue.

```
void S()
{
    switch (token) {
    case 0:
    case 1:
        L();
        R();
        break;
    case '$':
        break;
    default:
        error();
    }
}

void R()
{
    switch (token) {
    case '.':
        match('.');
        R();
        break;
    case '$':
        break;
    default:
        error();
    }
}

void L()
{
    switch (token) {
    case 0:
```

```
    case 1:
        B();
                Ls();
        break;
        case '$':
                break;
    default:
        error();
    }
}

void Ls()
{
    switch (token) {
    case 0:
    case 1:
        B();
        Ls();
        break;
    case '$':
        break;
    default:
        error();
    }
}

void B()
{
    switch (token) {
    default:
    case 0:
        match(0);
        break;
    case 1:
        match(1);
        break;
    default:
        error();
        }
    }
```

c) L-attributed attribute grammar

Assume `side = 1` means left-hand side, and `side = 0` means right-hand side

Assume `2^x` in the code means 2 to the power of x

production	semantic rules

$S \rightarrow L\ R$	L.side = 1;
	R.side = 0;
	S.val = L.val + R.val;

$R \rightarrow .\ L$	R.val = L.val
	L.side = R.side;

$R \rightarrow \epsilon$	R.val = 0;

$L \rightarrow B\ L_s$	L.len = 1 + Ls.len;
	Ls.side = L.side;
	
	L.val = (L.side == 1) ?
	B.val * (2^(L.len - 1)) + Ls.val :
	B.val / 2 + Ls.val / 2;}

$L_s \rightarrow B\ L_{s1}$	Ls.len = 1 + Ls1.len;
	Ls1.side = Ls.side;
	
	Ls.val = (Ls.side == 1) ?
	B.val * (2^(Ls.len - 1)) + Ls1.val :
	B.val / 2 + Ls1.val / 2;}

$L_s \rightarrow \epsilon$	Ls.len = 0;
	Ls.val = 0;

$B \rightarrow 0$	B.val = 0;
$B \rightarrow 1$	B.val = 1;

d) L-attributed attribute grammar → top-down translator

. and $ are surrounded by single quotation mark due to latex rendering issue.

```
void S()
{
    switch (token) {
    case 0:
    case 1:
        L();
        R();
        break;
    case '$':
        break;
    default:
        error();
    }
}

void R()
{
    switch (token) {
    case '.':
        match('.');
        R();
        break;
    case '$':
        break;
    default:
        error();
    }
}

void L()
{
    switch (token) {
    case 0:
```

```
    case 1:
        B();
                Ls();
        break;
        case '$':
                break;
    default:
        error();
    }
}

void Ls()
{
    switch (token) {
    case 0:
    case 1:
        B();
        Ls();
        break;
    case '$':
        break;
    default:
        error();
    }
}

void B()
{
    switch (token) {
    default:
    case 0:
        match(0);
        break;
    case 1:
        match(1);
        break;
    default:
        error();
        }
    }
```