Introduction to Algorithmic Differentiation

Derivative Code Automatically (Part III: Syntax-Directed Translation)

Uwe Naumann

Informatik 12:
Software and Tools for Computational Engineering (STCE)

RWTH Aachen

Contents

Outline

	STACK	STATE	PARSED	INPUT	ACTION
1		0	V	$= F(VNC);$	S
2	0	1	$V =$	$F(VNC);$	S
3	0,1	4	$V = F$	$(VNC);$	S
4	0,1,4	9	$V = F($	$VNC);$	S
5	0,1,4,9	11	$V = F(V$	$NC);$	S
6	0,1,4,9,11	7		$NC);$	R($P7$)
7	0,1,4,9	11	$V = F(e$	$NC);$	S
8	0,1,4,9,11	15	$V = F(eN$	$C);$	S
9	0,1,4,9,11,15	13	$V = F(eNC$	$);$	S
10	0,1,4,9,11,15,13	8		$);$	R($P8$)
11	0,1,4,9,11,15	13	$V = F(eNe$	$);$	S
12	0,1,4,9,11,15,13	17		$);$	R($P5$)
13	0,1,4,9	11	$V = F(e$	$);$	S

	STACK	STATE	PARSED	INPUT	ACTION
14	0,1,4,9,11	15	$V = F(e)$	;	S
15	0,1,4,9,11,15	18		;	R($P6$)
16	0,1	4	$V = e$	;	S
17	0,1,4	10	$V = e;$		S
18	0,1,4,10	14			R($P3$)
19		0	a		S
20	0	3			R($P1$)
21		0	s		S
22	0	2	$\$end$		S
23	0,2	5			R($P0$)
24		0	$\$accept$		ACCEPT

$P0$ is an auxiliary production rule for accepting the empty word.

Outline

Attribute grammars are a very powerful tool for rigorous definition of source code modification algorithms.

Transformation rules are associated with the syntactical structure of the language making their implementation often a straight-forward extension of the parser. A single pass over the input program may suffice to perform the desired transformation.

Moreover, attribute grammars describe exactly the corresponding traversal and modification pattern of the parse tree making them applicable to multi-pass compilation as well.

A synthesized attribute $v.s$ for the left-hand side v of a production $v : \phi(u_1, \ldots, u_k)$ is defined only in terms of its own attribute values or of attribute values of its children $u_1, \ldots, u_k$.

A grammar is called S-attributed if it contains only synthesized attributes.

The synthesized attribute s stores the number of subexpressions in the subtree rooted by the current nonterminal symbol.

$$
\begin{aligned}
a &: V = e; \\
e^l &: F(e^r) && \{ e^l.s := e^r.s + 1 \} \\
&: e^{r_1} L e^{r_2} && \{ e^l.s := e^{r_1}.s + e^{r_2}.s + 1 \} \\
&: e^{r_1} N e^{r_2} && \{ e^l.s := e^{r_1}.s + e^{r_2}.s + 1 \} \\
&: V && \{ e^l.s := 1 \} \\
&: C && \{ e^l.s := 1 \}
\end{aligned}
$$

S-Attributed Grammar

Example: Counting Subexpressions with `bison`

```
1  ...
2
3  %%
4
5  a : V '=' e ';' { printf("%d\n",$3); } ;
6  e : e L e { $$=$1+$3+1; }
7    | e N e { $$=$1+$3+1; }
8    | F '(' e ')' { $$=$3+1; }
9    | V { $$=1; }
10   | C { $$=1; }
11   ;
12
13 %%
14
15 ...
```

An inherited attribute $v.i$ for a symbol v on the right-hand side of a production $w : \phi(u_1, \ldots, v, \ldots, u_k)$ is defined in terms of its own attribute values, of those of its children, of those of its parent w, or of attribute values of its siblings $u_1, \ldots, u_k$.

Any S-attributed grammar is also L-attributed. Moreover, in an L-attributed grammar the values of all inherited attributes of an instance v of a non-terminal symbol are either functions of synthesized or inherited attributes of non-terminals to the left in the given production (including the parent on the left-hand side) or they are functions of synthesized or inherited attributes of v itself. Cyclic dependences among the attributes of v must not occur.

The inherited attribute i stores the index of the subexpression associated with the current nonterminal symbol.

$$
\begin{aligned}
a : V = e; \quad & \{\ e.i := 0\ \} \\
e^l : F(e^r) \quad & \{\ e^l.s := e^r.s + 1;\ e^r.i := e^l.i + 1\ \} \\
: e^{r_1} L e^{r_2} \quad & \{\ e^l.s := e^{r_1}.s + e^{r_2}.s + 1 \\
& \qquad e^{r_1}.i := e^l.i + 1;\ e^{r_2}.i := e^{r_1}.i + e^{r_1}.s\ \} \\
: e^{r_1} N e^{r_2} \quad & \{\ e^l.s := e^{r_1}.s + e^{r_2}.s + 1 \\
& \qquad e^{r_1}.i := e^l.i + 1;\ e^{r_2}.i := e^{r_1}.i + e^{r_1}.s\ \} \\
: V \quad & \{\ e^l.s := 1\ \} \\
: C \quad & \{\ e^l.s := 1\ \}
\end{aligned}
$$

L-Attributed Grammar

Example: Enumerating Subexpressions with `bison`

A global counter (sacvc) can be used instead of an inherited attribute, which is would require storage of (parts of) the parser tree.

```
1  %{
2  ...
3  unsigned int sacvc; // sac variable counter
4  %}
5  ...
6  %%
7
8  a : V '=' { sacvc=0; } e ';'
9  e : e L e { $$=sacvc++; printf("(%d,%d)\n(%d,%d)\n",$1,$$,$3,$$); }
10   | e N e { $$=sacvc++; printf("(%d,%d)\n(%d,%d)\n",$1,$$,$3,$$); }
11   | F '(' e ')' { $$=sacvc++; printf("(%d,%d)\n",$3,$$); }
12   | V { $$=sacvc++; }
13   | C { $$=sacvc++; }
14   ;
15
16 %%
17 ...
```

Outline

The attributes n and j replace the previously used s and i. The SAC is synthesized into the string attribute c.

$$
\begin{aligned}
(P3) \quad a : V = e; \quad & e.j = 0 \\
& a.c = e.c \\
& \quad + V.c + \text{" =v0;"} \\
(P4) \quad e^l : e^{r_1} L e^{r_2} \quad & e^l.n = e^{r_1}.n + e^{r_2}.n + 1 \\
& e^{r_1}.j = e^l.j + 1 \\
& e^{r_2}.j = e^{r_1}.j + e^{r_1}.n \\
& e^l.c = e^{r_1}.c + e^{r_2}.c \\
& \quad + \text{" v"} + e^l.j + \text{" =v"} + e^{r_1}.j + L.c \\
& \quad + \text{" v"} + e^{r_2}.j + \text{" ;"}
\end{aligned}
$$

$$
\begin{aligned}
(P5)\quad e^l : e^{r_1} N e^{r_2} \quad & e^l.n = e^{r_1}.n + e^{r_2}.n + 1 \\
& e^{r_1}.j = e^l.j + 1 \\
& e^{r_2}.j = e^{r_1}.j + e^{r_1}.n \\
& e^l.c = e^{r_1}.c + e^{r_2}.c \\
& \qquad + \text{" v"} + e^l.j + \text{" =v"} + e^{r_1}.j + N.c \\
& \qquad + \text{" v"} + e^{r_2}.j + \text{" ;"} \\
(P6)\quad e^l : F(e^r) \quad & e^l.n = e^r.n + 1 \\
& e^r.j = e^l.j + 1 \\
& e^l.c = e^r.c \\
& \qquad + \text{" v"} + e^l.j + \text{" ="} + F.c + \text{" (v"} + e^r.j + \text{");"}
\end{aligned}
$$

$$
\begin{array}{lll}
(P7) & e : V & e.n = 1 \\
 & & e.c = \text{" v"} + e.j + \text{" ="} + V.c + \text{" ;"} \\
(P8) & e : C & e.n = 1 \\
 & & e.c = \text{" v"} + e.j + \text{" ="} + C.c + \text{" ;"}
\end{array}
$$

Syntax-Directed Assignment-Level SAC

Example: $y = \sin(x * 2);$

The following refers to the characteristic DFA of the SL^2 parser.

i	PARSED	ACTION	$\$\$.j$	$\$\$.c$	Comment
0	V	S			
1	$V =$	S			
4	$V = F$	S			
9	$V = F($	S			
11	$V = F(V$	S			
7		R($P7$)	2	$\boxed{v_2 = x;}$	
11	$V = F(e$	S			
15	$V = F(eN$	S			
13	$V = F(eNC$	S			
8		R($P8$)	3	$\boxed{v_3 = 2;}$	
13	$V = F(eNe$	S			
17		R($P5$)		$v_2 = x;$	$< \ldots = e^{r_1}.c$
				$v_3 = 2;$	$< \ldots = e^{r_2}.c$
			1	$\boxed{v_1 = v_2 * v_3;}$	$N.c =$ " $*$ "
					$e^{r_1}.j = 2,\ e^{r_2}.j = 3$

Syntax-Directed Assignment-Level SAC

Example: $y = \sin(x * 2);$

i	PARSED	ACTION	$\$\$.j$	$\$\$.c$	Comment
11	$V = F(e$	S			
15	$V = F(e)$	S			
18		R($P6$)		$v_2 = x;$	$<$
				$v_3 = 2;$	$<$
				$v_1 = v_2 * v_3;$	$< \ldots = e^r.c$
			0	$\boxed{v_0 = \sin(v_1);}$	$F.c =$ " sin ", $e^r.j = 1$
4	$V = e$	S			
10	$V = e;$	S			
14		R($P3$)		$v_2 = x;$	$<$
				$v_3 = 2;$	$<$
				$v_1 = v_2 * v_3;$	$<$
				$v_0 = \sin(v_1);$	$< \ldots = e.c$
				$\boxed{y = v_0;}$	$V.c =$ "y", $e.j = 0$

Syntax-Directed Assignment-Level SAC

Example: $y = \sin(x * 2);$

i	PARSED	ACTION	$\$\$.j$	$\$\$.c$	Comment
0	a	S			
3		R($P1$)			
0	s	S			
2	$\$end$	S			
5		R($P0$)			
0	\$accept	ACCEPT			

Outline

$$
\begin{array}{lll}
(P7) & e : V & e.n = 1 \\
 & & e.c = \text{" v"} + e.j + \text{" \_="} + V.c + \text{" \_;"} \\
 & & \quad + \text{" v"} + e.j + \text{" ="} + V.c + \text{" ;"} \\
(P8) & e : C & e.n = 1 \\
 & & e.c = \text{" v"} + e.j + \text{" \_=0;"} \\
 & & \quad + \text{" v"} + e.j + \text{" ="} + C.c + \text{" ;"}
\end{array}
$$

$$
\begin{aligned}
(P6) \quad e^l : F(e^r) \quad & e^l.n = e^r.n + 1 \\
& e^r.j = e^l.j + 1 \\
& e^l.c = e^r.c \\
& \quad + \text{" v"} + e^l.j + \text{" \_="} \\
& \quad + \partial_{e^r.j} F + \text{" * v"} + e^r.j + \text{" \_;"} \\
& \quad + \text{" v"} + e^l.j + \text{" ="} + F.c + \text{" (v"} + e^r.j + \text{");"}
\end{aligned}
$$

where $\partial_{e^r.j} F = \text{"cos(v"} + e^r.j + \text{")"}$ if $F.c = \text{"sin"}$ etc.

$$
\begin{aligned}
&(P4) \quad e^l : e^{r_1} O e^{r_2} \quad && e^l.n = e^{r_1}.n + e^{r_2}.n + 1 \\
&\text{and} && e^{r_1}.j = e^l.j + 1 \\
&(P5) && e^{r_2}.j = e^{r_1}.j + e^{r_1}.n \\
& && e^l.c = e^{r_1}.c + e^{r_2}.c \\
& && \quad + \text{" v"} + e^l.j + \text{" \_="} + \partial_{e^{r_1}.j} O \\
& && \quad + \text{" *v"} + e^{r_1}.j + \text{" \_+"} \\
& && \quad + \partial_{e^{r_2}.j} O + \text{" *v"} + e^{r_2}.j + \text{" \_;"} \\
& && \quad + \text{" v"} + e^l.j + \text{" =v"} + e^{r_1}.j + O.c \\
& && \quad + \text{" v"} + e^{r_2}.j + \text{" ;"}
\end{aligned}
$$

where $O \in \{L, N\}$, $\partial_{e^{r_1}.j} O = \partial_{e^{r_2}.j} O = \text{"1"}$ if $O.c = \text{"+"}$, $\partial_{e^{r_1}.j} O = \text{"v"} + e^{r_2}.j$ and $\partial_{e^{r_2}.j} O = \text{"v"} + e^{r_1}.j$ if $O.c = \text{"*"}$, etc.

$$
\begin{aligned}
(P3) \quad a : V = e; \quad & e.j = 0 \\
& a.c = e.c \\
& \quad + V.c + \text{" \_=v0\_;"} \\
& \quad + V.c + \text{" =v0;"}
\end{aligned}
$$

For $y = v_0$ we get $y^{(1)} = \frac{\partial y}{\partial v_0} \cdot v_0^{(1)} = v_0$.

Syntax-Directed Straight-Line Tangent Code

Example: $y = \sin(x * 2);$

i	PARSED	ACTION	$\$\$.j$	$\$\$.c$
0	V	S		
1	$V =$	S		
4	$V = F$	S		
9	$V = F($	S		
11	$V = F(V$	S		
7		R($P7$)	1	$v_2^{(1)} = x^{(1)};\ v_2 = x;$
11	$V = F(e$	S		
15	$V = F(eN$	S		
13	$V = F(eNC$	S		
8		R($P8$)	2	$v_3^{(1)} = 0;\ v_3 = 2;$
13	$V = F(eNe$	S		
17		R($P5$)		$v_2^{(1)} = x^{(1)};\ v_2 = x;$ $v_3^{(1)} = 0;\ v_3 = 2;$
			3	$v_1^{(1)} = v_2^{(1)} * v_3 + v_2 * v_3^{(1)};\ v_1 = v_2 * v_3;$

Syntax-Directed Straight-Line Tangent Code

Example: $y = \sin(x * 2);$

i	PARSED	ACTION	$\$\$.j$	$\$\$.c$
11	V=$F(e$	S		
15	$V = F(e)$	S		
18		R($P6$)		$v_2^{(1)} = x^{(1)};\ v_2 = x;$ $v_3^{(1)} = 0;\ v_3 = 2;$ $v_1^{(1)} = v_2^{(1)} * v_3 + v_2 * v_3^{(1)};\ v_1 = v_2 * v_3;$
			4	$\boxed{v_0^{(1)} = \cos(v_1) * v_1^{(1)};\ v_0 = \sin(v_1);}$
4	$V = e$	S		
10	$V = e;$	S		
14		R($P3$)		$v_2^{(1)} = x^{(1)};\ v_2 = x;$ $v_3^{(1)} = 0;\ v_3 = 2;$ $v_1^{(1)} = v_2^{(1)} * v_3 + v_2 * v_3^{(1)};\ v_1 = v_2 * v_3;$ $v_0^{(1)} = \cos(v_1) * v_1^{(1)};\ v_0 = \sin(v_1);$ $\boxed{y^{(1)} = v_0^{(1)};\ y = v_0;}$

Syntax-Directed Straight-Line Tangent Code

Example: $y = \sin(x * 2)$;

i	PARSED	ACTION	$\$\$.j$	$\$\$.c$
0	a	S		
3		R($P1$)		
0	s	S		
2	$\$end$	S		
5		R($P0$)		
0	\$accept	ACCEPT		

Outline

The inherited attribute k enumerates all assignments. The augmented primal is synthesized into the string attribute c^f; the adjoint is synthesized into c^b.

$$
\begin{aligned}
(P0) \quad \$accept : \quad & \\
& s.k = 0 \\
& s\$end \\
& \$accept.c^f = s.c^f \\
& \quad + \text{"int } i\text{;"} \\
& \quad + \text{"while}(pop\_c(i))\{\text{"} \\
& \quad + \quad \text{"if}(i == 1)\{\text{"} \\
& \quad + \quad\quad s.c_1^b
\end{aligned}
$$

$$
\begin{aligned}
&+ \quad \text{"}\}\text{else if}(i == 2)\{\text{"} \\
&+ \quad\quad s.c_2^b \\
&\quad\vdots \\
&+ \quad \text{"}\}\text{else if}(i == \text{"} + s.k + \text{"})\{\text{"} \\
&+ \quad\quad s.c_{s.k}^b \\
&+ \quad \text{"}\}\text{"}
\end{aligned}
$$

$(P1) \qquad s :$

$$a.k = s.k + 1$$

a

$$s.k = a.k; \quad s.c^f = a.c^f; \quad s.c^b = a.c^b$$

$$(P1a) \quad s : \quad b.k = s.k$$
$$b \quad s.k = b.k; \quad s.c^f = b.c^f; \quad s.c^b = b.c^b$$

$$(P1b) \quad s : \quad l.k = s.k$$
$$l \quad s.k = l.k; \quad s.c^f = l.c^f; \quad s.c^b = l.c^b$$

$$(P2) \quad s^l :$$

$$a.k = s^l.k + 1$$

$$a$$

$$s^r.k = a.k$$

$$s^r$$

$$s^l.k = s^r.k$$

$$s^l.c^f = a.c^f + s^r.c^f; \quad s^l.c^b = s^r.c^b + a.c^b$$

$$(P2a) \quad s^l : \quad b.k = s^l.k$$

$$b \quad s^r.k = b.k$$

$$s^r \quad s^l.k = s^r.k \quad s^l.c^f = b.c^f + s^r.c^f; \quad s^l.c^b = s^r.c^b + b.c^b$$

$$(P2b) \quad s^l : \quad l.k = s^l.k$$

$$l \quad s^r.k = l.k$$

$$s^r \quad s^l.k = s^r.k \quad s^l.c^f = l.c^f + s^r.c^f; \quad s^l.c^b = s^r.c^b + l.c^b$$

$(P3) \quad a :$

$$e.k = a.k; \quad e.i = 0$$

$$V = e;$$

$$a.c^f = e.c^f + \text{" push\_c(" } + a.k + \text{");"}$$
$$+ \text{" push\_d(" } + V.c^f + \text{");"}$$
$$+ V.c^f + \text{" =v0;"}$$
$$a.c^b_{a.k} = \text{" pop\_d(" } + V.c^f + \text{");"}$$
$$+ \text{" v0\_=" } + V.c^f + \text{" \_;"}$$
$$+ V.c^f + \text{" \_=0;"}$$
$$+ e.c^b_{a.k}$$

$(P4/5) \quad e^l :$

$$e^{r_1}.i = e^l.i + 1; \quad e^{r_i}.k = e^l.k \ \text{ for } i = 1, 2$$

e^{r_1}

$$e^{r_2}.i = e^{r_1}.i + e^{r_1}.s + 1$$

Oe^{r_2}

$$e^l.s = e^{r_1}.s + e^{r_2}.s + 1$$

$$e^l.c^f = e^{r_1}.c^f + e^{r_2}.c^f$$
$$+ \text{" push\_d(v"} + e^l.i + \text{");"}$$
$$+ \text{" v"} + e^l.i$$
$$+ \text{" =v"} + e^{r_1}.i + O.c^f + \text{" v"} + e^{r_2}.i + \text{" ;"}$$

$$
\begin{aligned}
e^l.c^b_{e.k} = &\text{" pop\_d(v"} + e^l.i + \text{");"} \\
&+ \text{" v"} + e^{r_2}.i + \text{" \_="} + O_{e^{r_2}.i} + \text{" *v"} + e^l.i + \text{" \_;"} \\
&+ \text{" v"} + e^{r_1}.i + \text{" \_="} + O_{e^{r_1}.i} + \text{" *v"} + e^l.i + \text{" \_;"} \\
&+ e^{r_2}.c^b_{e.k} + e^{r_1}.c^b_{e.k}
\end{aligned}
$$

$(P6) \quad e^l :$

$$e^r.i = e^l.i + 1; \quad e^r.k = e^l.k$$

$$F(e^r)$$

$$e^l.s = e^r.s + 1$$

$$e^l.c^f = e^r.c^f$$
$$+ \text{" push\_d(v"} + e^l.i + \text{");"}$$
$$+ \text{" v"} + e^l.i + \text{" ="} + F.c^f + \text{" (v"} + e^r.i + \text{");"}$$

$$e^l.c^b_{e.k} = \text{" pop\_d(v"} + e^l.i + \text{");"}$$
$$+ \text{" v"} + e^r.i + \text{" _="} + F_{e^r.i} + \text{" *v"} + e^l.i + \text{" _;"}$$
$$+ e^r.c^b_{e.k}$$

$$(P7) \quad e : V \quad e.s = 1$$
$$a.c^f = \text{" push\_d(v"} + e.i + \text{");"}$$
$$+ \text{" v"} + e.i + \text{" ="} + V.c^f + \text{" ;"}$$
$$a.c^b_{e.k} = \text{" pop\_d(v"} + e.i + \text{");"}$$
$$+ V.c^f + \text{" \_+=v"} + e.i + \text{" \_;"}$$

$$(P8) \quad e : C \qquad e.s = 1$$

$$a.c^f = \text{" push\_d(v"} + e.i + \text{");"}$$

$$+ \text{" v"} + e.i + \text{" ="} + C.c^f + \text{" ;"}$$

$$a.c^b_{e.k} = \text{" pop\_d(v"} + e.i + \text{");"}$$

$$(P9) \quad b : IF(r)$$

$$s.k = b.k$$

$$\{s\}$$

$$b.k = s.k$$

$$b.c^f = \text{"if"} + \text{"("} + r.c^f + \text{")"} + \text{"\{"} s.c^f + \text{"\}"}$$

$$b.c^b = s.c^b$$

$(P10) \quad I : WHILE(r)$

$$s.k = I.k$$

$$\{s\}$$

$$I.k = s.k$$

$$I.c^f = \text{"while"} + \text{"("} + r.c^f + \text{")"} + \text{"\{"} s.c^f + \text{"\}"}$$

$$I.c^b = s.c^b$$

$(P11) \quad r : V^{r_1} R V^{r_2}$

$$r.c^f = V^{r_1}.c^f + R.c^f + V^{r_2}.c^f$$

Syntax-Directed Adjoint Code

Example

i	$\$\$.c^f$	$\$\$.c^b$
...		
11		
7	$push(v_2);\ v_2 = x;$	$pop(v_2);\ x_{(1)} \mathrel{+}= v_{2(1)};$
13		
8	$push(v_3);\ v_3 = 2;$	$pop(v_3);$
13		
14	$push(v_2);\ v_2 = x;$	$pop(v_1);\ v_{2(1)} = v_3 * v_{1(1)};\ v_{3(1)} = v_2 * v_{1(1)};$
	$push(v_3);\ v_3 = 2;$	$pop(v_3);$
	$push(v_1);\ v_1 = v_2 * v_3;$	$pop(v_2);\ x_{(1)} \mathrel{+}= v_{2(1)};$
11		
15		
...		

Syntax-Directed Adjoint Code

Example

i	$\$\$.c^f$	$\$\$.c^b$
18	$push(v_2);\ v_2 = x;$	$\boxed{pop(v_0);\ v_{1(1)} = \cos(v_1) * v_{0(1)};}$
	$push(v_3);\ v_3 = 2;$	$pop(v_1);\ v_{2(1)} = v_3 * v_{1(1)};\ v_{3(1)} = v_2 * v_{1(1)};$
	$push(v_1);\ v_1 = v_2 * v_3;$	$pop(v_3);$
	$\boxed{push(v_0);\ v_0 = \sin(v_1);}$	$pop(v_2);\ x_{(1)} \mathrel{+}= v_{2(1)};$
4		
10		
14	$push(v_2);\ v_2 = x;$	$\boxed{pop(y);\ v_{0(1)} = y_{(1)};\ y_{(1)} = 0;}$
	$push(v_3);\ v_3 = 2;$	$pop(v_0);\ v_{1(1)} = \cos(v_1) * v_{0(1)};$
	$push(v_1);\ v_1 = v_2 * v_3;$	$pop(v_1);\ v_{2(1)} = v_3 * v_{1(1)};\ v_{3(1)} = v_2 * v_{1(1)};$
	$push(v_0);\ v_0 = \sin(v_1);$	$pop(v_3);$
	$\boxed{push(y);\ y = v_0;}$	$pop(v_2);\ x_{(1)} \mathrel{+}= v_{2(1)};$
...		

Summary

Recall: Shift-Reduce Parser

Attribute Grammars

- Synthesized Attributes
- Inherited Attributes

Assignment-Level Single Assignment Code

Straight-Line Tangent Code

Adjoint Code