Script generated by TTT

Title: Petter: Compiler Construction (18.06.2020)
- 36: Multiple Attributes

Date: Thu Jun 18 10:08:23 CEST 2020

Duration: 09:46 min

Pages: 5

Simultaneous Computation of Multiple Attributes

Computing empty, first, next from regular expressions:

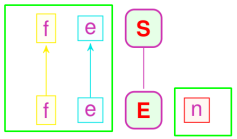
$S \rightarrow E$:

empty[0]	:=	empty[1]
first[0]	:=	first[1]
next[1]	:=	$\emptyset$

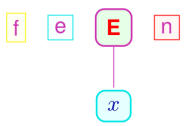
$E \rightarrow x$:

empty[0]	:=	$(x \equiv \epsilon)$
first[0]	:=	$\{x \mid x \neq \epsilon\}$

$D(S \rightarrow E)$:



$D(E \rightarrow x)$:



$D(S \rightarrow E) = \{ (empty[1], empty[0]), (first[1], first[0]) \}$

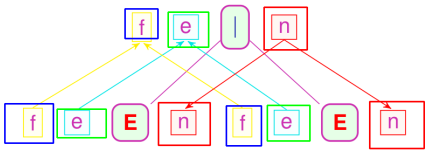
$D(E \rightarrow x) = \{ \}$

Regular Expressions: Rules for Alternative

$E \rightarrow E|E$:

empty[0]	:=	empty[1] $\vee$ empty[2]
first[0]	:=	first[1] $\cup$ first[2]
next[1]	:=	next[0]
next[2]	:=	next[0]

$D(E \rightarrow E|E)$:



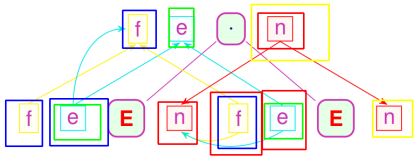
$D(E \rightarrow E|E) = \{ (empty[1], empty[0]), (empty[2], empty[0]), (first[1], first[0]), (first[2], first[0]), (next[0], next[2]), (next[0], next[1]) \}$

Regular Expressions: Rules for Concatenation

$E \rightarrow E \cdot E$:

empty[0]	:=	empty[1] $\wedge$ empty[2]
first[0]	:=	first[1] $\cup$ (empty[1] ? first[2] : $\emptyset$)
next[1]	:=	first[2] $\cup$ (empty[2] ? next[0] : $\emptyset$)
next[2]	:=	next[0]

$D(E \rightarrow E \cdot E)$:



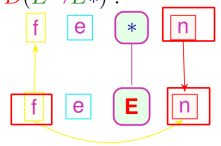
$D(E \rightarrow E \cdot E) = \{ (empty[1], empty[0]), (empty[2], empty[0]), (empty[2], next[1]), (empty[1], first[0]), (first[1], first[0]), (first[2], first[0]), (first[2], next[1]), (next[0], next[2]), (next[0], next[1]) \}$

Regular Expressions: Rules for Kleene-Star and Option

$E \to E*$:

$$\begin{array}{lcl} \mathsf{empty}[0] & := & t \\ \mathsf{first}[0] & := & \mathsf{first}[1] \\ \mathsf{next}[1] & := & \mathsf{first}[1] \cup \mathsf{next}[0] \end{array}$$

$D(E \to E*)$:

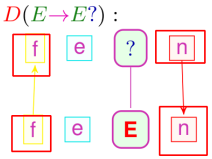


$$D(E \to E*) = \{ \quad (first[1], first[0]), \\ (first[1], next[2]), \\ (next[0], next[1])\}$$

$E \to E?$:

$$\begin{array}{lcl} \mathsf{empty}[0] & := & t \\ \mathsf{first}[0] & := & \mathsf{first}[1] \\ \mathsf{next}[1] & := & \mathsf{next}[0] \end{array}$$

$D(E \to E?)$:



$$D(E \to E?) = \{ \quad (first[1], first[0]), \\ (next[0], next[1])\}$$

Challenges for General Attribute Systems

Static evaluation

Is there a static evaluation strategy, which is generally applicable?

- an evaluation strategy can only exist, if for *any* derivation tree the dependencies between attributes are acyclic
- it is *DEXPTIME*-complete to check for cyclic dependencies [Jazayeri, Odgen, Rounds, 1975]