Script generated by TTT

Title: Petter: Compiler Construction (30.04.2020) - 05: Advanced Berry Sethi

Date: Wed Apr 22 10:10:59 CEST 2020

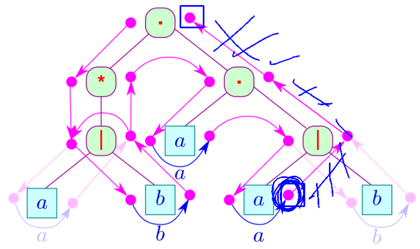
Duration: 59:31 min

Pages: 13

Berry-Sethi Approach

... for example:

$w \quad = \quad :$



Berry-Sethi Approach

Discussion:

- Most transitions navigate through the expression
- The resulting automaton is in general nondeterministic

Berry-Sethi Approach

Discussion:

- Most transitions navigate through the expression
- The resulting automaton is in general nondeterministic

⇒ Strategy for the sophisticated version:
Avoid generating ϵ-transitions

Idea:
Pre-compute helper attributes during D(epth)F(irst)S(earch)!

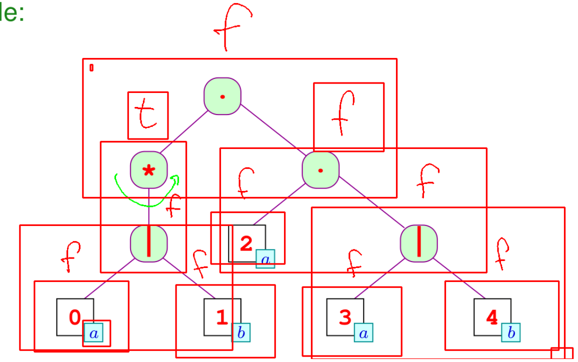
Necessary node-attributes:

- first: the set of read states below r, which may be reached first, when descending into r.
- next: the set of read states, which may be reached first in the traversal after r.
- last: the set of read states below r, which may be reached last when descending into r.
- empty: can the subexpression r consume ϵ?

Berry-Sethi Approach: 1st step

$\boxed{\text{empty}[r] = t \quad \text{if and only if} \quad \epsilon \in [\![r]\!]}$

... for example:



Berry-Sethi Approach: 1st step

Implementation:
DFS post-order traversal

for leaves $r \equiv \boxed{i \mid x}$ we find $\text{empty}[r] = (x \equiv \epsilon)$.

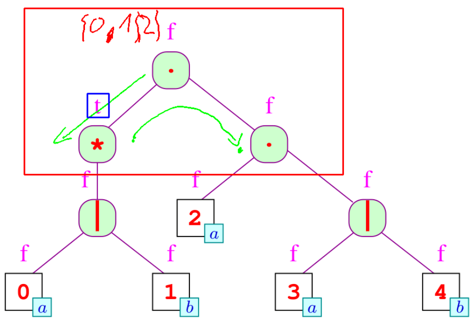
Otherwise:

$$\begin{array}{lll} \text{empty}[r_1 \mid r_2] & = & \text{empty}[r_1] \vee \text{empty}[r_2] \\ \text{empty}[r_1 \cdot r_2] & = & \text{empty}[r_1] \wedge \text{empty}[r_2] \\ \text{empty}[r_1^*] & = & t \\ \text{empty}[r_1?] & = & t \end{array}$$

Berry-Sethi Approach: 2nd step

The may-set of first reached read states: The set of read states, that may be reached from $\bullet r$ (i.e. while descending into r) via sequences of ϵ-transitions:
$\text{first}[r] = \{i \text{ in } r \mid (\bullet r, \epsilon, \bullet\boxed{i \mid x}) \in \delta^*, x \neq \epsilon\}$

... for example:



Berry-Sethi Approach: 2nd step

Implementation:
DFS post-order traversal

for leaves $r \equiv \boxed{i \mid x}$ we find $\text{first}[r] = \{i \mid x \neq \epsilon\}$.

Otherwise:

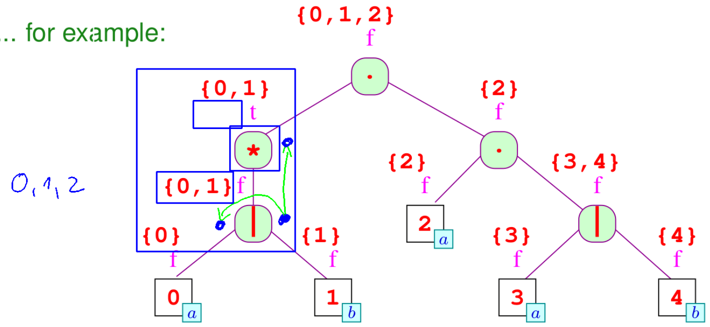
$$\begin{array}{lll} \text{first}[r_1 \mid r_2] & = & \text{first}[r_1] \cup \text{first}[r_2] \\ \text{first}[r_1 \cdot r_2] & = & \begin{cases} \text{first}[r_1] \cup \text{first}[r_2] & \text{if } \text{empty}[r_1] = t \\ \text{first}[r_1] & \text{if } \text{empty}[r_1] = f \end{cases} \\ \text{first}[r_1^*] & = & \text{first}[r_1] \\ \text{first}[r_1?] & = & \text{first}[r_1] \end{array}$$

Berry-Sethi Approach: 3rd step

The may-set of next read states: The set of read states reached after reading r, that may be reached next via sequences of ϵ-transitions.

$\text{next}[r] = \{i \mid (r\bullet, \epsilon, \bullet\boxed{i \mid x}) \in \delta^*, x \neq \epsilon\}$

... for example:



Berry-Sethi Approach: 3rd step

Implementation:
DFS pre-order traversal

For the root, we find: $\boxed{\text{next}[e] = \emptyset}$
Apart from that we distinguish, based on the context:

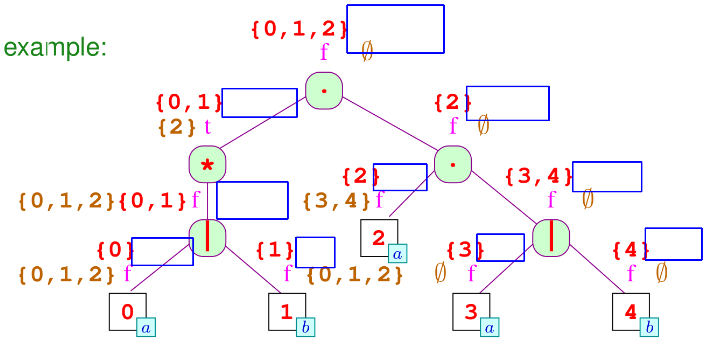
r	Equalities
$r_1 \mid r_2$	$\text{next}[r_1] = \text{next}[r]$ $\text{next}[r_2] = \text{next}[r]$
$r_1 \cdot r_2$	$\text{next}[r_1] = \begin{cases} \text{first}[r_2] \cup \text{next}[r] & \text{if } \text{empty}[r_2] = t \\ \text{first}[r_2] & \text{if } \text{empty}[r_2] = f \end{cases}$ $\text{next}[r_2] = \text{next}[r]$
r_1^*	$\text{next}[r_1] = \text{first}[r_1] \cup \text{next}[r]$
$r_1?$	$\text{next}[r_1] = \text{next}[r]$

Berry-Sethi Approach: 4th step

The may-set of last reached read states: The set of read states, which may be reached last during the traversal of r connected to the root via ϵ-transitions only:

$\text{last}[r] = \{i \text{ in } r \mid (\boxed{i \mid x}\bullet, \epsilon, r\bullet) \in \delta^*, x \neq \epsilon\}$

... for example:



Berry-Sethi Approach: 4th step

Implementation:
DFS post-order traversal

for leaves $r \equiv \boxed{i \mid x}$ we find $\text{last}[r] = \{i \mid x \neq \epsilon\}$.

Otherwise:

$$\begin{array}{lcll}
\text{last}[r_1 \mid r_2] & = & \text{last}[r_1] \cup \text{last}[r_2] & \\
\text{last}[r_1 \cdot r_2] & = & \begin{cases} \text{last}[r_1] \cup \text{last}[r_2] \\ \text{last}[r_2] \end{cases} & \begin{array}{l} \text{if } \text{empty}[r_2] = t \\ \text{if } \text{empty}[r_2] = f \end{array} \\
\text{last}[r_1^*] & = & \text{last}[r_1] & \\
\text{last}[r_1?] & = & \text{last}[r_1] &
\end{array}$$

Berry-Sethi Approach: (sophisticated version)

Construction (sophisticated version):

Create an automanton based on the syntax tree's new attributes:

States: $\{\bullet e\} \cup \{i\bullet \mid i$ a leaf not $\epsilon\}$ $n+1$ n # chars

Start state: $\bullet e$

Final states: $\text{last}[e]$ if $\text{empty}[e] = f$

$\{\bullet e\} \cup \text{last}[e]$ otherwise

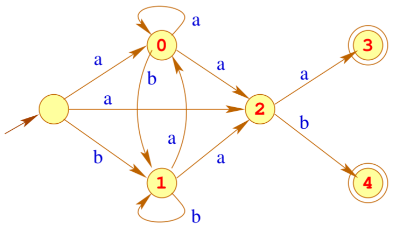
Transitions: $(\bullet e, a, i\bullet)$ if $i \in \text{first}[e]$ and i labled with a.

$(i\bullet, a, i'\bullet)$ if $i' \in \text{next}[i]$ and i' labled with a.

We call the resulting automaton A_e.

Berry-Sethi Approach

... for example:



Remarks:

- This construction is known as Berry-Sethi- or Glushkov-construction.
- It is used for XML to define Content Models
- The result may not be, what we had in mind...