Script generated by TTT

Title: Simon: Compilerbau (22.04.2013)

Date: Mon Apr 22 14:20:22 CEST 2013

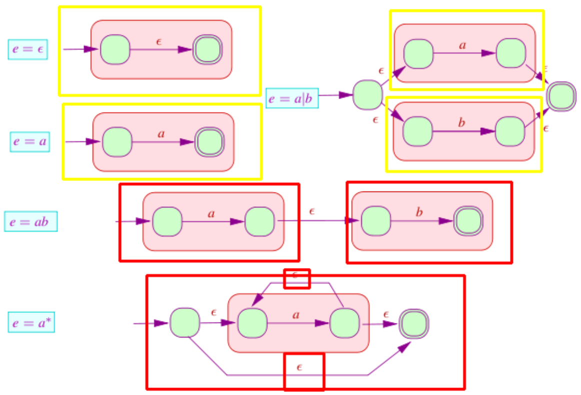
Duration: 91:47 min

Pages: 41

Kapitel 3:

Converting Regular Expressions to NFAs

In linear time from Regular Expressions to NFAs

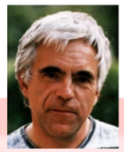



Ken Thompson

Thompson's Algorithm

Produces $\mathcal{O}(n)$ states for regular expressions of length n.

Berry-Sethi Approach


Gerard Berry


Ravi Sethi

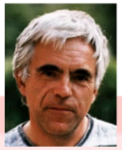
Berry-Sethi Algorithm

Produces exactly $n + 1$ states without ϵ-transitions and demonstrates → *Equality Systems* and → *Attribute Grammars*

Idea:

The automaton tracks (conceptionally via a marker "•"), in the syntax tree of a regular expression, which subexpression in e are reachable consuming the rest of input w.

Berry-Sethi Approach


Gerard Berry


Ravi Sethi

Berry-Sethi Algorithm

Produces exactly $n+1$ states without ϵ-transitions and demonstrates $\rightarrow$ *Equality Systems* and $\rightarrow$ *Attribute Grammars*

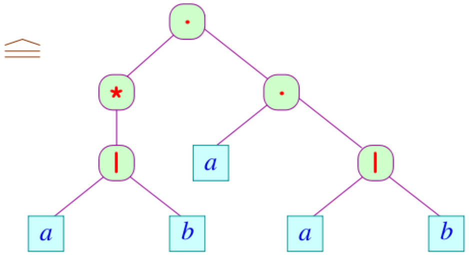
Idea:

The automaton tracks (conceptionally via a marker "$\bullet$"), in the syntax tree of a regular expression, which subexpression in e are reachable consuming the rest of input w.

Berry-Sethi Approach

... for example:

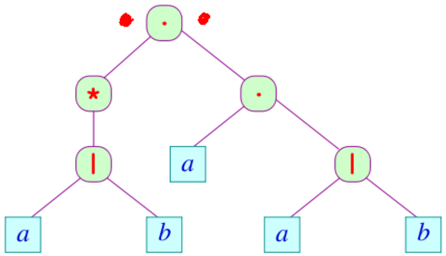
$\boxed{(a|b)^* a (a|b)}$ $\widehat{=}$



Berry-Sethi Approach

... for example:

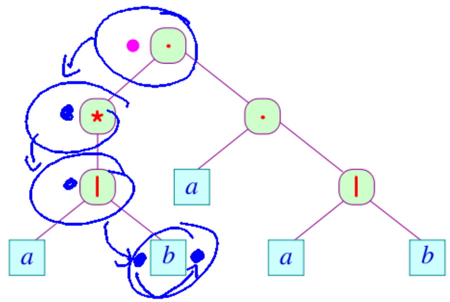
$w \quad = \quad bbaa \quad :$



Berry-Sethi Approach

... for example:

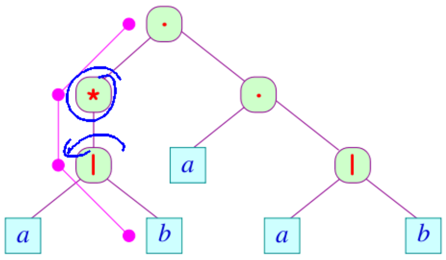
$w \quad = \quad bbaa \quad :$



Berry-Sethi Approach

... for example:

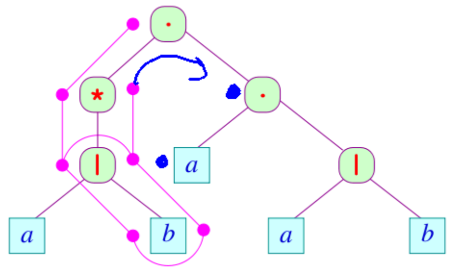
$w = bbaa$:



Berry-Sethi Approach

... for example:

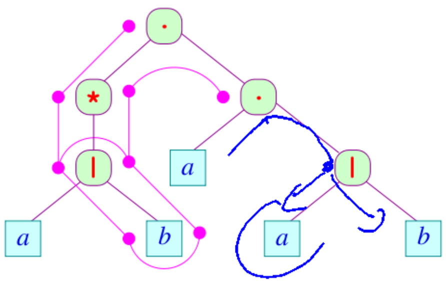
$w = \cancel{bb}aa$:



Berry-Sethi Approach

... for example:

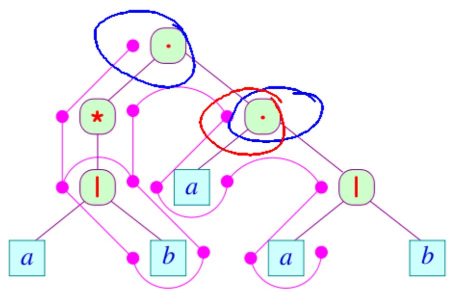
$w = \cancel{bba}a$:



Berry-Sethi Approach

... for example:

$w = bbaa$:



Berry-Sethi Approach

Attention:

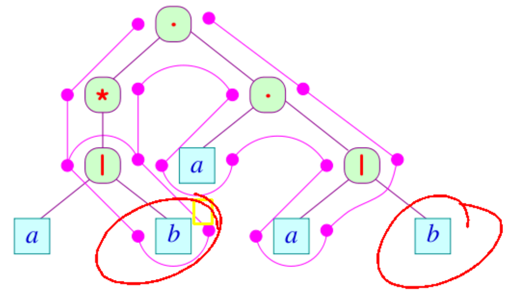
- Input is only consumed by the leaves.
- Navigation in the tree is done without consuming input, i.e. via ϵ-transition.
- For a formal construction we need to come up with identifiers for states.
- Therefore we use the subexpression, corresponding to the subtree, dominated by the particular node.
- There are possibly same subexpressions in one regular expression.

$\Longrightarrow$ we enumerate the leaves ...

Berry-Sethi Approach

... for example:

$w = bbaa$:



Berry-Sethi Approach

Attention:

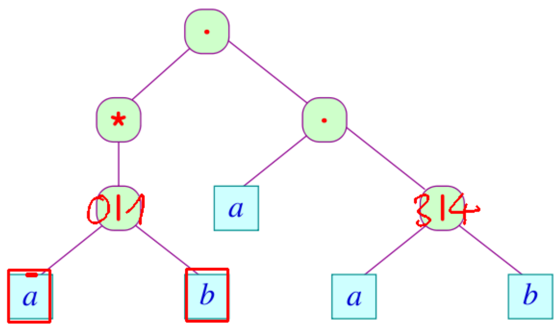
- Input is only consumed by the leaves.
- Navigation in the tree is done without consuming input, i.e. via ϵ-transition.
- For a formal construction we need to come up with identifiers for states.
- Therefore we use the subexpression, corresponding to the subtree, dominated by the particular node.
- There are possibly same subexpressions in one regular expression.

$\Longrightarrow$ we enumerate the leaves ...

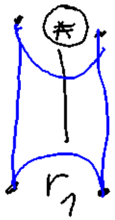
31 / 150

Berry-Sethi Approach

... for example:



Berry-Sethi Approach



Construction:

States: $\bullet r$, $r\bullet$ with r nodes of e;
Start state: $\bullet e$;
Final state: $e\bullet$;
Transitions: for leaves $r \equiv$ $\boxed{i \mid x}$ we require: $(\bullet r, x, r\bullet)$.

The leftover transitions are:

r	Transitions
$r_1 \mid r_2$	$(\bullet r, \epsilon, \bullet r_1)$ $(\bullet r, \epsilon, \bullet r_2)$ $(r_1\bullet, \epsilon, r\bullet)$ $(r_2\bullet, \epsilon, r\bullet)$
$r_1 \cdot r_2$	$(\bullet r, \epsilon, \bullet r_1)$ $(r_1\bullet, \epsilon, \bullet r_2)$ $(r_2\bullet, \epsilon, r\bullet)$

r	Transitions
r_1^*	$(\bullet r, \epsilon, r\bullet)$ $(\bullet r, \epsilon, \bullet r_1)$ $(r_1\bullet, \epsilon, \bullet r_1)$ $(r_1\bullet, \epsilon, r\bullet)$
$r_1?$	$(\bullet r, \epsilon, r\bullet)$ $(\bullet r, \epsilon, \bullet r_1)$ $(r_1\bullet, \epsilon, r\bullet)$

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

$\Rightarrow$ Strategy: Avoid generating ϵ-transitions

Necessary node-attributes:

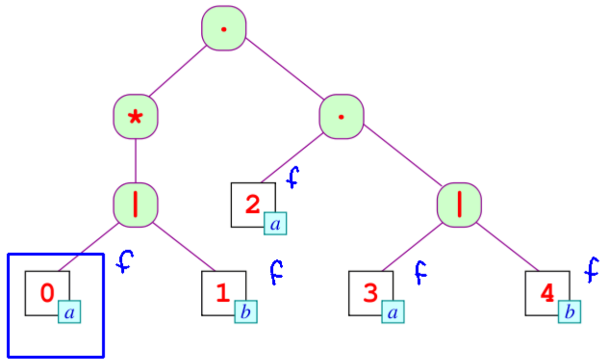
empty	can the subexpression r consume ϵ ?
first	the set of read states below r, which may be reached first, when descending into r.
next	the set of read states on the right of r, which may may be reached first in the traversal after r.
last	the set of read states below r, which may be reached last when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

Idea: Compute these attributes for the nodes via DFS!

Berry-Sethi Approach: 1st step

empty$[r] = t$ if and only if $\epsilon \in [\![r]\!]$

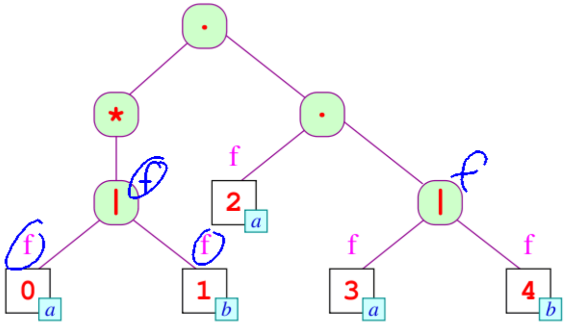
... for example:



Berry-Sethi Approach: 1st step

empty$[r] = t$ if and only if $\epsilon \in [\![r]\!]$

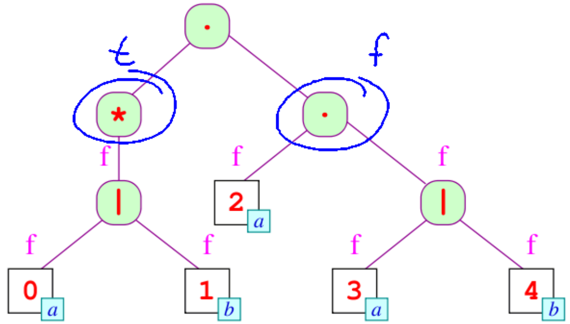
... for example:



Berry-Sethi Approach: 1st step

empty$[r] = t$ if and only if $\epsilon \in [\![r]\!]$

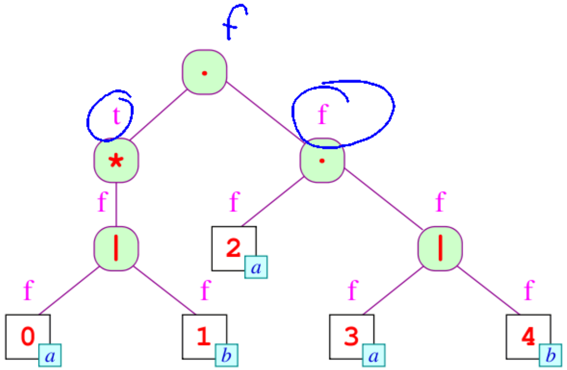
... for example:



Berry-Sethi Approach: 1st step

empty$[r] = t$ if and only if $\epsilon \in [\![r]\!]$

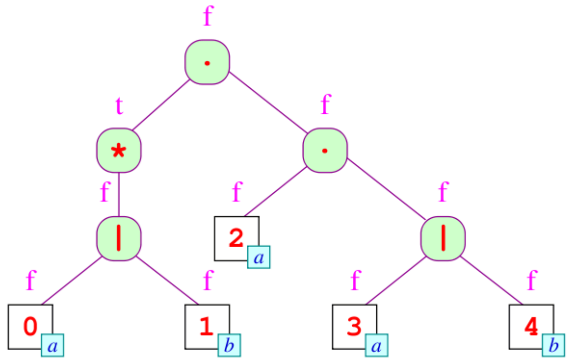
... for example:



Berry-Sethi Approach: 1st step

empty$[r] = t$ if and only if $\epsilon \in [\![r]\!]$

... for example:



Berry-Sethi Approach: 2nd step

Implementation:
DFS post-order traversal

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

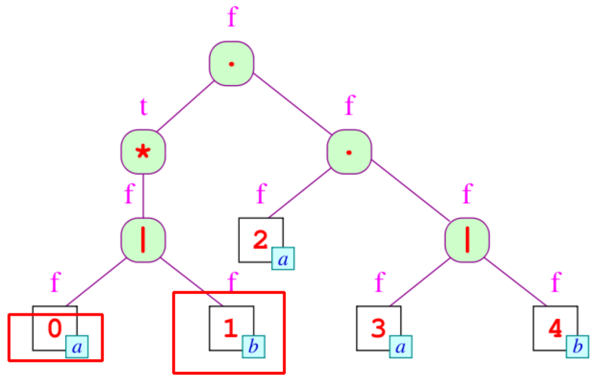
Otherwise:

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

Berry-Sethi Approach: 2nd step

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

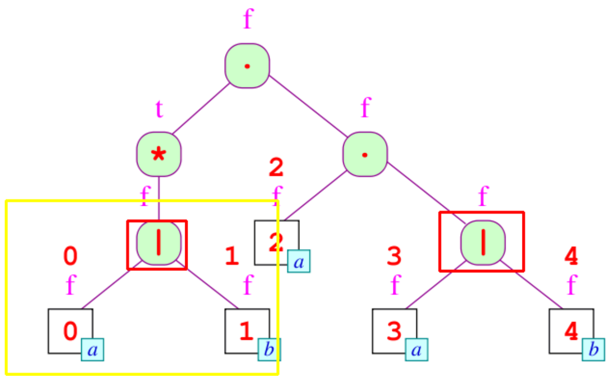
... for example:



Berry-Sethi Approach: 2nd step

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

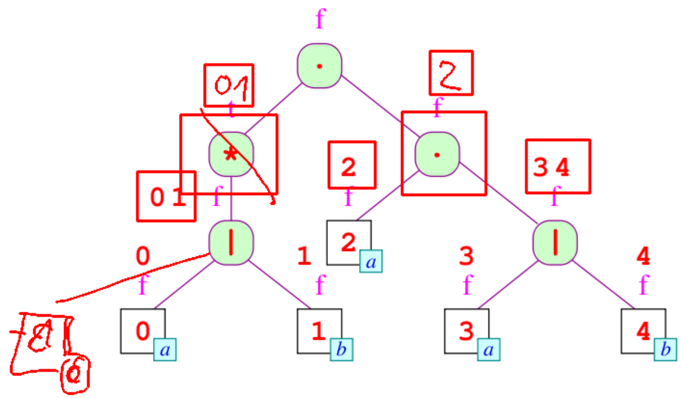
... for example:



Berry-Sethi Approach: 2nd step

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

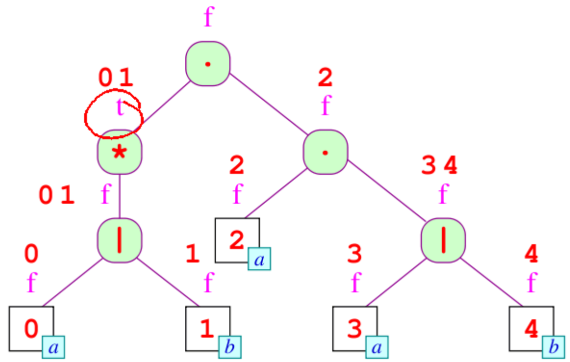
... for example:



Berry-Sethi Approach: 2nd step

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

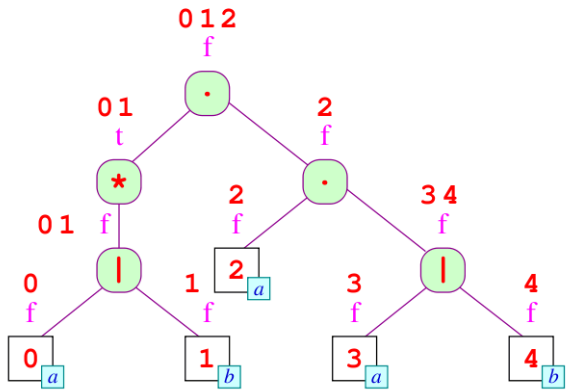
... for example:



Berry-Sethi Approach: 2nd step

The may-set of first reached read state: 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:

$$\text{first}[r_1 \mid r_2] = \text{first}[r_1] \cup \text{first}[r_2]$$

$$\text{first}[r_1 \cdot r_2] = \begin{cases} \text{first}[r_1] \cup \text{first}[r_2] & \text{if } \text{empty}[r_1] = t \\ \text{first}[r_1] & \text{if } \text{empty}[r_1] = f \end{cases}$$

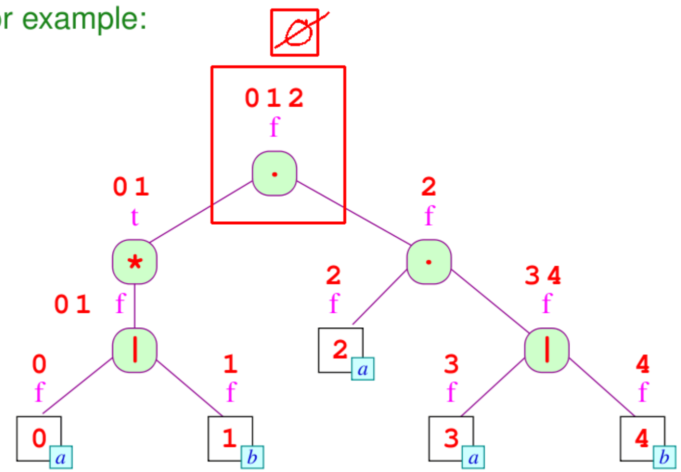
$$\text{first}[r_1^*] = \text{first}[r_1]$$

$$\text{first}[r_1?] = \text{first}[r_1]$$

Berry-Sethi Approach: 3rd step

The may-set of next read states: The set of read states within the subtrees right of $r\bullet$, that may be reached next via sequences of ϵ-transitions. $\text{next}[r] = \{i \mid (r\bullet, \epsilon, \bullet\boxed{i \mid x}) \in \delta^*, x \neq \epsilon\}$

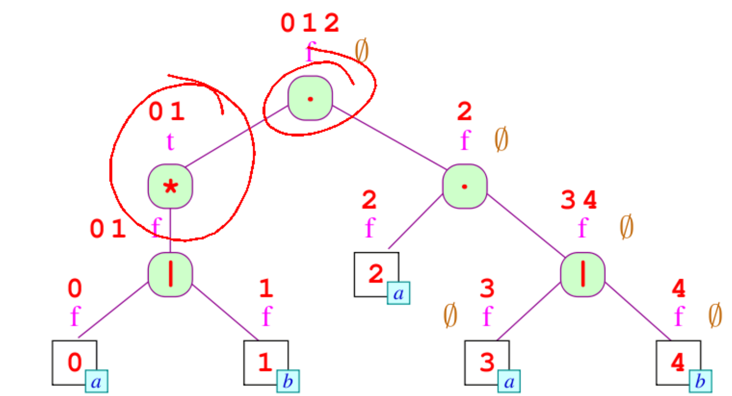
... for example:



Berry-Sethi Approach: 3rd step

The may-set of next read states: The set of read states within the subtrees right of $r\bullet$, that may be reached next via sequences of ϵ-transitions. $\text{next}[r] = \{i \mid (r\bullet, \epsilon, \bullet\boxed{i \mid x}) \in \delta^*, x \neq \epsilon\}$

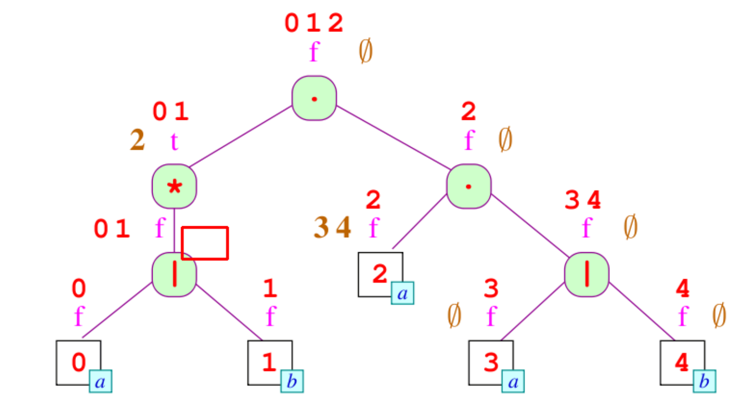
... for example:



Berry-Sethi Approach: 3rd step

The may-set of next read states: The set of read states within the subtrees right of $r\bullet$, that may be reached next via sequences of ϵ-transitions. $\text{next}[r] = \{i \mid (r\bullet, \epsilon, \bullet\boxed{i \mid x}) \in \delta^*, x \neq \epsilon\}$

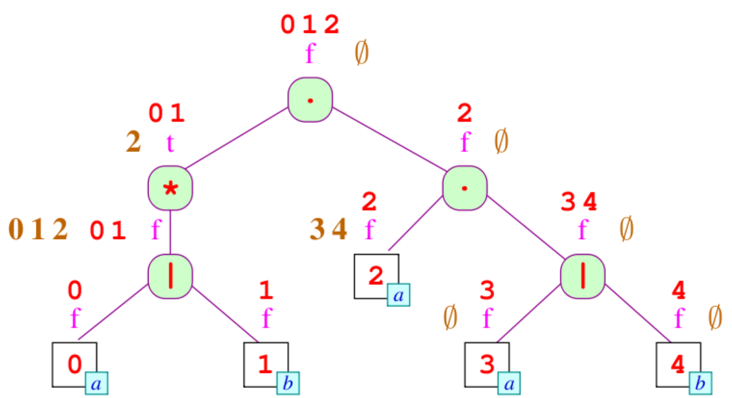
... for example:



Berry-Sethi Approach: 3rd step

The may-set of next read states: The set of read states within the subtrees right of $r\bullet$, that may be reached next via sequences of ϵ-transitions. $\text{next}[r] = \{i \mid (r\bullet, \epsilon, \bullet\boxed{i \mid x}) \in \delta^*, x \neq \epsilon\}$

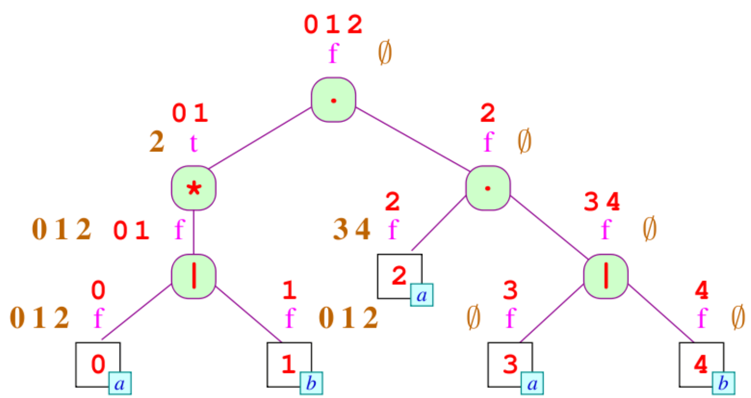
... for example:



Berry-Sethi Approach: 3rd step

The may-set of next read states: The set of read states within the subtrees right of $r\bullet$, 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:

$$\text{next}[e] = \emptyset$$

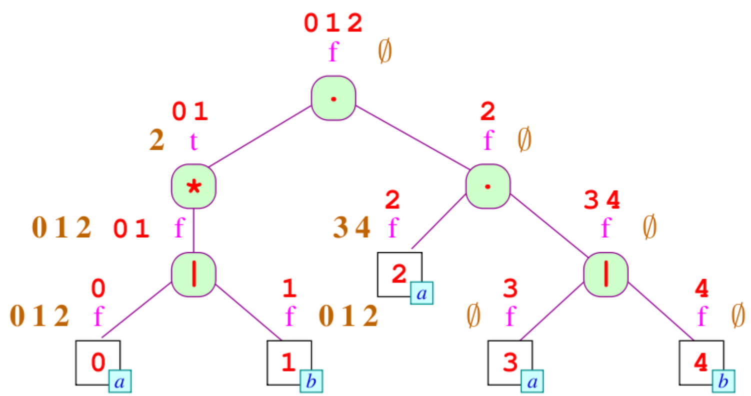
Apart from that we distinguish, based on the context:

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

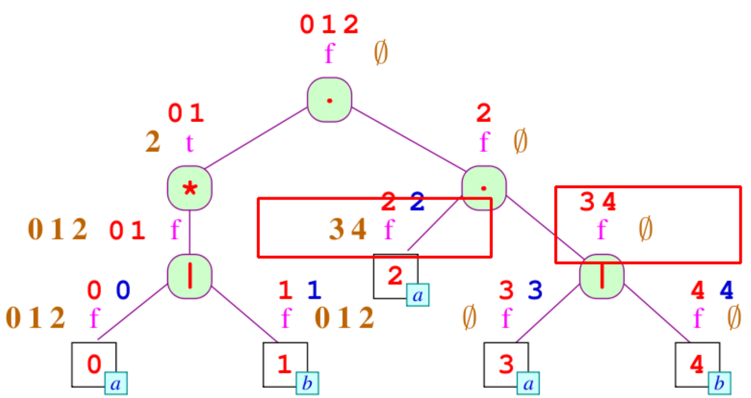
... for example:



Berry-Sethi Approach: 4th step

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: Integration

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

States: $\{\bullet e\} \cup \{i\bullet \mid i \text{ a leaf}\}$

Start state: $\bullet e$

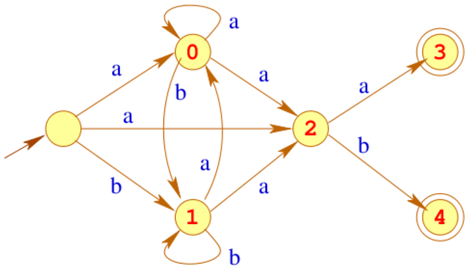
Final states: $\text{last}[e]$ if $\text{empty}[e] = f$.
$\{\bullet e\} \cup \text{last}[e]$ else.

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 beschriftet ist.

We call the resulting automaton $\boxed{A_e}$.

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...

Kapitel 4:

Turning NFAs deterministic