# $getNodeT(x, i)$ and $getNodeP(X\alpha \cdot \beta, w, z)$ for BNF grammars

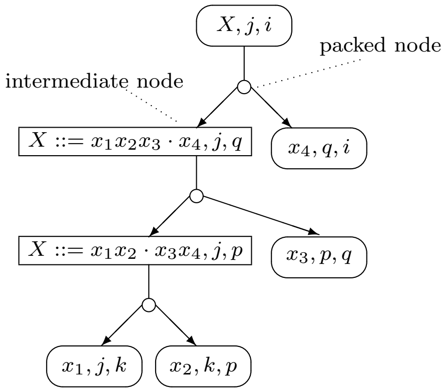
A GLL parser constructs binarised SPPFs. These are bipartite graphs which merge all of the derivation trees of a given input string. Derivation tree nodes are labelled with both a grammar symbol, $x$, and two integers which define the substring of the input derive from $x$. So in a derivation tree for $a_1 \dots a_m$, $(x, j, i)$ labels a node which is the root of a subtree whose leaves are $a_{j+1}, \dots, a_i$.

To ensure that the resulting SPPF is worst-case cubic in size, the derivation trees are binarised in a simple way by introducing intermediate nodes from the left.

The binarised derivation trees are packed together with nodes with the same label being merged. A derivation tree node may have many packed node children but each packed node will have at most two children as the original trees were binary.

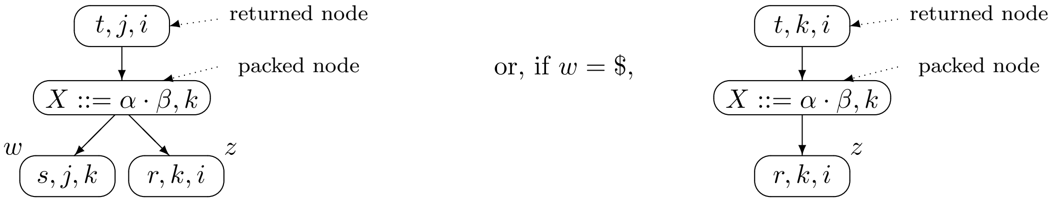
Then, a binarised SPPF has three types of SPPF nodes: symbol nodes, with labels of the form $(x, j, i)$ where $x$ is a terminal, nonterminal or $\epsilon$ and $0 \leq j \leq i \leq m$; intermediate nodes, with labels of the form $(t, j, i)$; and packed nodes, with labels for the form $(t, k)$, where $0 \leq k \leq m$ and $t$ is of the form $X ::= \alpha \cdot \beta$.

For example, for the rule $X ::= x_1x_2x_3x_4$ we have SPPF fragment



$getNodeT(x, i)$ creates and returns an SPPF node labelled $(x, i, i + 1)$ or $(\epsilon, i, i)$ if $x = \epsilon$.

$getNodeP(X ::= \alpha \cdot \beta, w, z)$ takes a grammar position (slot) $X ::= \alpha \cdot \beta$ and two SPFF nodes $w$, $z$, the first of which may be the dummy node \$. The nodes $w$ and $z$ are not packed nodes and will have labels of the form $(s, j, k)$ and $(r, k, i)$. The function creates an SPPF fragment of the form below, where $t$ is $X ::= \alpha \cdot \beta$ if $\beta \neq \epsilon$ and $t$ is $X$ otherwise.


$t, j, i$
returned node
packed node
$X ::= \alpha \cdot \beta, k$
$w$
$z$
$s, j, k$
$r, k, i$
or, if $w = \$$,
$t, k, i$
returned node
packed node
$X ::= \alpha \cdot \beta, k$
$z$
$r, k, i$