# Coursework 3

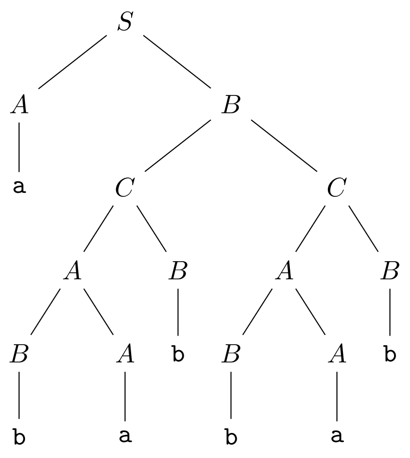
COMP2721 Algorithms and Data Structures II

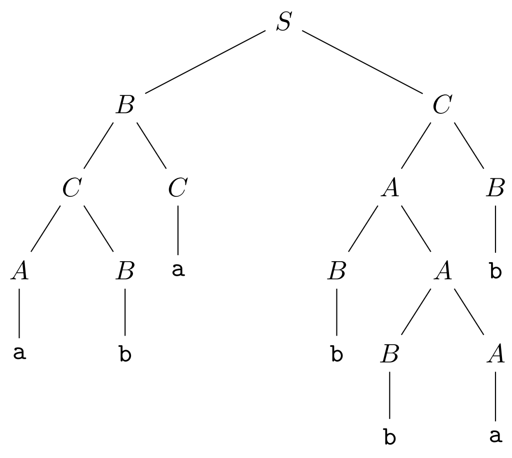
sample solutions

1. In the table we give the sets $V(i,k)$ for $1 \le i \le k \le 7$ for the input string `ababbab`. For $k > i$ the index at variable $X \in V(i,k)$ is a string made from all the values of $j$ such that $X \to YZ$ is a production of the grammar, $Y \in V(i,j)$ and $Z \in V(j+1,k)$.

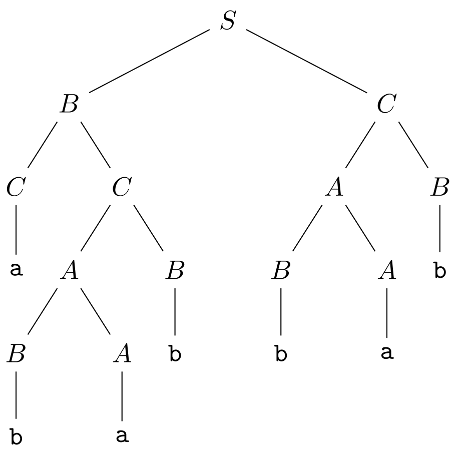
| | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ | $k=7$ |
|---|---|---|---|---|---|---|---|
| $i=1$ | $\{A,C\}$ | $\{S_1,C_1\}$ | $\{B_2\}$ | $\{B_{12}\}$ | $\varnothing$ | $\{A_{34}\}$ | $\{S_{1346},C_{16}\}$ |
| $i=2$ | | $\{B\}$ | $\{S_2,A_2\}$ | $\{S_{23},C_3\}$ | $\varnothing$ | $\varnothing$ | $\{B_4\}$ |
| $i=3$ | | | $\{A,C\}$ | $\{S_3,C_3\}$ | $\varnothing$ | $\varnothing$ | $\{B_{34}\}$ |
| $i=4$ | | | | $\{B\}$ | $\varnothing$ | $\{A_4\}$ | $\{S_{46},C_6\}$ |
| $i=5$ | | | | | $\{B\}$ | $\{S_5,A_5\}$ | $\{S_{56},C_6\}$ |
| $i=6$ | | | | | | $\{A,C\}$ | $\{S_6,C_6\}$ |
| $i=7$ | | | | | | | $\{B\}$ |

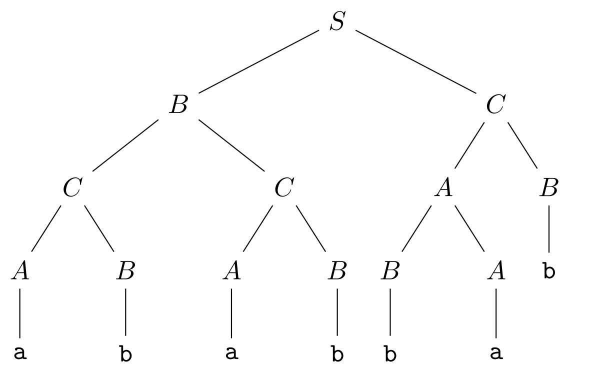
The $j$-values allow us to construct the following parse trees. Only one of them was required. For $j=1$ and $j=3$ on the top level:



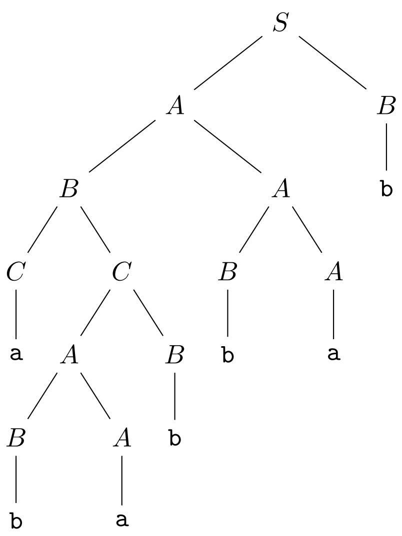



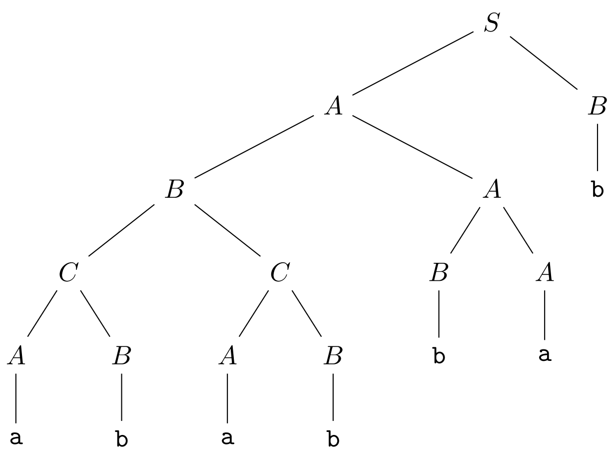
For $j = 4$ on the top level:

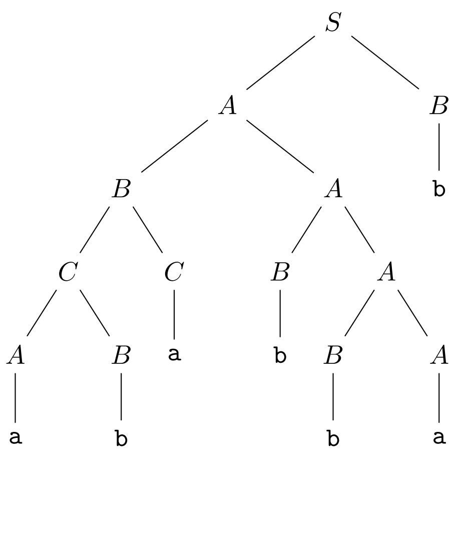




For $j = 6$ on the top level:







2. We choose an arbitrary vertex $r \in V$ as root of $T = (V, E)$. For each verex $x \in V$ let $V(x)$ denote the set of all vertices $y \in V$ such that $x$ is on the path from $y$ to $r$ in $T$. Especially we have $x \in V(x)$. Furthermore, let $T(x)$ be the subtree of $T$ induced by $V(x)$.

   For all $x \in V$ let

   - $m(x)$ denote the maximum weight of a matching of $T(x)$,
   - $s(x)$ denote the maximum weight of a matching of $T(x)$ that saturates $x$, and
   - $u(x)$ denote the maximum weight of a matching of $T(x)$ that leaves $x$ unsaturated.

   The *children* of $x$ are the vertices in the set $C(x) = N(x) \cap V(x)$. For the root $r$ all neighbours are children. For all other vertices $x$ there is a parent vertex in $N(x) \setminus V(x)$ which is not a child of $x$.

   We have the following recurrences

$$
\begin{aligned}
s(x) &= \begin{cases} \max\limits_{y \in C(x)} \big(w(xy) + u(y) + \sum\limits_{z \in C(x) \setminus \{y\}} m(z)\big) & \text{if } C(x) \neq \varnothing \\ -\infty & \text{if } C(x) = \varnothing \end{cases} \\
u(x) &= \begin{cases} \sum\limits_{y \in C(x)} m(y) & \text{if } C(x) \neq \varnothing \\ 0 & \text{if } C(x) = \varnothing \end{cases} \\
m(x) &= \max\{s(x), u(x)\}
\end{aligned}
$$

   To see these we first observe that $C(x) = \varnothing$ if and only if $x$ is a leaf of $T$. In this case $T(x)$ has only one matching, namely $\varnothing$, which leaves $x$ unsaturated. So $u(x) = 0$ and, striktly speaking, $s(x)$ is undefined; we set $s(x) = -\infty$ instead. If $C(x) \neq \varnothing$ then every matching of $T(x)$ that leaves $x$ unsaturated partitions into matchings in the subtrees $T(y)$ rooted at the children $y$ of $x$. If $x$ is saturated, then $x$ is matched to a child $y$. So $y$ cannot be matched to a vertex in $V(y)$. For all other children $z \neq y$ of $x$ it does not matter whether $z$ is saturated. If so, it is matched to one of its children. Finally, the line for $m(x)$ is obvious.

   The maximum weight of a matching in $T$ is $m(r)$.

   The recurrence can be converted into a dynamic programming algorithm. Therefore we first run BFS starting from the root vertex $r$, and then compute $s(x)$, $u(x)$ and $m(x)$ in reverse order. That is, if $|V| = n$ then we start with the vertex $x$ with $\sigma(x) = n$ down to the root with $\sigma(r) = 1$. Note that the cases for $C(x) = \varnothing$ are not special if we set, for any parameter $t : V \to \mathbb{N}$, the values $\max\limits_{y \in \varnothing} t(y) = -\infty$ and $\sum\limits_{y \in \varnothing} t(y) = 0$.