Permutations Generated by a Stack Machine
Jayadev Misra
10/11/98

Let S be an infinite stack. An output string is computed as follows from an input string. In each step either the next input symbol is added to the stack, or the top of the stack is moved to the output. Let $x \longrightarrow y$ denote that y is a possible output string given x as the input string. We explore some of the properties of $\longrightarrow$.

$$\epsilon \longrightarrow \epsilon$$
$$x \longrightarrow x', y \longrightarrow y' \Rightarrow axy \longrightarrow x'ay'.$$

The second rule states that given an input string axy the item a appears in the output at some point. Prior to output of a, item a is at the bottom of the stack and hence, some portion of the input, x, is converted to x' and appended to the output. Following the output of a, the stack is empty and the remaining input sequence y is converted to y'.

We assume henceforth that the output is a permutation of the input, which can be proved from the rules. Symbols a, b are in the alphabet; all other symbols are strings over the alphabet, possibly empty.

Lemma 1: $x \longrightarrow x', y \longrightarrow y' \Rightarrow xy \longrightarrow x'y'$.

Proof: by induction on x. For empty x the result is immediate. For the inductive case, let x be of the for auv and x' be $u'av'$ where $u \longrightarrow u'$ and $v \longrightarrow v'$.

$$\begin{array}{ll} & xy \\ = & \{x = auv\} \\ & au(vy) \\ \longrightarrow & \{\text{from } v \longrightarrow v',\ y \longrightarrow y' \text{ by induction } vy \longrightarrow v'y'; \\ & \quad \text{rule 2: } u \longrightarrow u',\ vy \longrightarrow v'y'\} \\ & u'av'y' \\ = & \{x' = u'av'\} \\ & x'y' \end{array}$$

Lemma 2: $y \longrightarrow y', z \longrightarrow z' \Rightarrow xyz \longrightarrow y'\overline{x}z'$, where $\overline{x}$ is the reverse of x.

Proof: By induction on x. For empty x the result follows from lemma 1. For x of the form au:

$$\begin{array}{ll} & xyz \\ = & \{x = au\} \\ & auyz \\ \longrightarrow & \{uy \longrightarrow y'\overline{u}, \text{ by induction; } z \longrightarrow z'; \text{ apply rule 2}\} \end{array}$$

$$
\begin{array}{ll}
 & y'\overline{u}az' \\
= & \{\overline{u}a = \overline{x}\} \\
 & y'\overline{x}z'
\end{array}
$$

Corollary 1: $x \longrightarrow \overline{x}$, setting y, z to ϵ in Lemma 2.

Corollary 2: $x \longrightarrow x$.
Proof: First show $a \longrightarrow a$ from definition. Next apply lemma 1 to show $ax \longrightarrow ax$, by induction.

Corollary 3: $y \longrightarrow y' \Rightarrow xyz \longrightarrow xy'z$.
Proof: Apply corollary 2 and lemma 1.

Note that $\longrightarrow$ is not symmetric: $cab \longrightarrow abc$, though $abc \not\longrightarrow cab$, which can be seen by appealing to the stack machine.

Given an input string x and an alleged output string y, it can be determined if y is a possible output given x as input. The algorithm employs a stack. In each step: if the stack is empty or the top element of the stack differs from the next output symbol then do an input step (read the next input symbol and push it onto the stack), else (the stack is non-empty and top element of the stack matches the next output symbol) then remove the top symbol of the stack and move to the next output symbol.