Useless variables in context-free grammars

A variable X in a context-free grammar is called *useless* if it doesn't occur in any derivation of a word from that grammar.
Here is an easy algorithm to eliminate useless variable from CFGs.

Call a variable *generating* if it derives a string of terminals. Note that the language accepted by a context-free grammar is non-empty if and only if the start symbol is generating. Here is an algorithm to find the generating variables in a CFG:

1. Mark a variable X as "generating" if it has a production X -> w where w is a string of only terminals and/or variables previously marked "generating".
2. Repeat the step above until no further variables get marked "generating".

All variables not marked "generating" are non-generating (by a simple induction on the length of derivations).

Call a variable *reachable* if the start symbol derives a string containing that variable. Here is an algorithm to find the reachable variables in a CFG:

1. Mark the start variable as "reachable".
2. Mark a variable Y as "reachable" if there is a production X -> w where X is a variable previously marked as "reachable" and w is a string containing Y.
3. Repeat the step above until no further variables get marked "reachable".

All variables not marked "reachable" are non-reachable (by a simple induction on the length of derivations).

Observe that a CFG has no useless variables if and only if all its variables are reachable and generating.
Therefore it is possible to eliminate useless variables from a grammar as follows:

1. Find the non-generating variables and delete them, along with all productions involving non-generating variables.
2. Find the non-reachable variables in the resulting grammar and delete them, along with all productions involving non-reachable variables.

Note that step 1 does not make other variables non-generating, and step 2 does not make other variables non-reachable or non-generating. Therefore the end result is a grammar in which all variables are reachable and generating, and hence useful.

Reversing step 1 and 2 in the above algorithm would not work, as eliminating non-generating variables and their productions may make other variables unreachable. Example:

S -> AB | a

A -> aA

B -> b

Here A is non-generating, and after deleting A (along with the production S -> AB) the variable B becomes unreachable. So it must be a useless variable. However, if we would first test for reachability, all variables would be reachable, and subsequently eliminating non-generating variables would leave us with B.

Rob van Glabbeek rvg@cs.stanford.edu