Comment on the Paper
“Error Detection in Formal Languages”

GYÖRGY RÉVÉSZ

Institute for Coordination of Computer Techniques, Budapest, V. Akadémia u. 17, Hungary
Received July 28, 1972; revised January 12, 1973

In a paper of W. B. Smith [1], some normal form theorems have been presented about context-sensitive grammars. Professor A. Salomaa has noticed that the proof of Lemma 2 in this paper was incorrect and he has given an example in which the proof does not work.1 It was guessed that there was no simple way to repair Smith’s proof, while M. Penttonen has found a rather complicated proof of a somewhat stronger normal form [2].

We shall give here a construction that is quite simple and still works. Before doing so, we must mention that the proof of Lemma 1 in Smith’s paper is also incorrect and a similar bug can be found also in the well-known paper of Kuroda [3]: namely, on page 211 of the latter it is stated that each rule $AB \rightarrow CD$ of a context-sensitive grammar can be replaced by three rules: $AB \rightarrow A'B$, $A'B \rightarrow A'D$ and $A'D \rightarrow CD$ where $A'$ is a new non-terminal symbol. Consider, however, the following rules: $S \rightarrow AB$, $B \rightarrow DE$, $AB \rightarrow CD$. The above replacement yields a set of rules permitting a derivation $S \Rightarrow CD$ which does not exist in the original grammar.

In order to avoid this parasitical derivation we have to apply four rules: $AB \rightarrow A'B$, $A'B \rightarrow A'B'$, $A'B' \rightarrow CB'$ and $CB' \rightarrow CD$, where $A'$ and $B'$ are new non-terminal symbols.

The situation is more complex in the case of Smith’s proofs.

Lemma 1 asserts that any language generated by a left-context-sensitive grammar $G$ can be generated by some grammar $G'$ all of whose rules are of the form $A \rightarrow a$, $A \rightarrow BC$ or $BA \rightarrow BC$, where $A$, $B$, $C$ are non-terminals and $a$ is terminal.

The proof of this lemma claims that a rule of the form $A \rightarrow B$ is merely the replacing of one non-terminal symbol by another and, thus, it can be eliminated by replacing $A$ by $B$ in all rules of the grammar. But, again, this may lead to parasitical derivations as can be seen in this example: $S \rightarrow AC$, $A \rightarrow B$, $BC \rightarrow Bc$, $AC \rightarrow Ad$, $A \rightarrow a$, $B \rightarrow b$.

Here the elimination of $A \rightarrow B$ gives $S \rightarrow Bc$, $BC \rightarrow Bc$, $Bc \rightarrow Bd$, $B \rightarrow a$, $B \rightarrow b$.

1 The author thanks Professor A. Salomaa, R. V. Book, and M. Penttonen for their valuable communications on the problem concerned, during a conference in Paris, July 3–7, 1972.
that makes the terminal string $ac$ derivable which is not the case in the original grammar.

**Lemma 2** asserts that any context-sensitive grammar $G$ can be generated by a grammar $G'$ consisting of left-context-sensitive rules plus rules of the form $AB \rightarrow BA$.

The example of Salomaa is $S \rightarrow ABAB$, $AB \rightarrow CB$, $A \rightarrow a$, $B \rightarrow b$, $C \rightarrow c$. According to the construction of Smith's proof the rule $AB \rightarrow CB$ will be replaced by the following rules:

- $A \rightarrow D$,
- $DB \rightarrow DB'$,
- $DB' \rightarrow B'D$,
- $B'D \rightarrow B'E$,
- $B'E \rightarrow EB'$,
- $EB' \rightarrow EB$,
- $E \rightarrow C$,

where $B'$, $D$, $E$ are new non-terminals. In this new grammar we have the following derivation: $S \Rightarrow DB'DB' \Rightarrow B'DB'D \Rightarrow B'E'B'E \Rightarrow B'E'B'E \Rightarrow EB'E \Rightarrow EB' \Rightarrow cbbc$. However, the string $cbbc$ is not derivable in the original grammar.

We shall now present a detailed proof of Lemma 2.

**Proof.** Kuroda has shown in [3] (with the above correction) that any context-sensitive language can be generated by some grammar, whose rules are all of the form $A \rightarrow a$, $A \rightarrow B$, $A \rightarrow BC$, $AB \rightarrow AC$ or $AB \rightarrow CB$. Thus, we have to eliminate only the rules of the form $AB \rightarrow CB$. A rule of this form can be replaced by the following rules:

- $AB \rightarrow AB'$,  $A \rightarrow A'$,
- $A'B' \rightarrow A'D$,  $A' \rightarrow E$,
- $ED \rightarrow DE$,  $D \rightarrow C'$,
- $CE \rightarrow CE'$,  $C' \rightarrow C$,
- $CE' \rightarrow CB$,

where $A'$, $B'$, $C'$, $D$, $E$, and $E'$ are new non-terminal symbols. (It can be observed that only one inverting rule $ED \rightarrow DE$ is applied here, while Smith's construction applies two of them.)

Clearly the rule $AB \rightarrow CB$ can be simulated with the aid of these new rules. Reversely, we have to show that every terminal string derivable in the new grammar is also derivable in the original one. For this purpose let us consider all possible derivations from the string $AB$ using only the new rules as shown in Fig. 1. As can be seen in
the figure, every node of this derivation-tree, except the starting node and the leftmost ending one, corresponds to a pair of symbols containing at least one of the new non-terminal symbols. We shall see, further, that we cannot get rid of these new non-terminals if we deviate from the leftmost path or do not follow it up to the end: namely, an inconvenient (interfering) replacement of these symbols may occur in two ways:

![Figure 1](image_url)

1. Two pairs of symbols shown in Figure 1 appear next to each other in a derivation, and the last symbol of the first pair followed by the first symbol of the second pair occurs on the left side of a rule.

2. A nested pair appears in a derivation with matching symbols within the nest. This may happen to a pair containing only one of the new non-terminals if the other nonterminal symbol $A$, $B$, or $C$ can be replaced by $AB$ in the original grammar. Such pairs are $AB'$, $A'C$, $CE$, $CE'$, $A'B$, $EB$, $EC$. (See Figure 1.)

All these possibilities are denoted by "yes" in Table 1, where each row corresponds to some symbol that can be the last symbol of a pair of Figure 1 or the first symbol of a nest, while each column corresponds to some symbol that can be the first
symbol of a pair of Figure 1 or the last symbol of a nest. Here the minus sign means that no interference between the two symbols may occur.

<table>
<thead>
<tr>
<th>A</th>
<th>C</th>
<th>A'</th>
<th>C'</th>
<th>E</th>
<th>D</th>
<th>B'</th>
<th>E</th>
<th>E'</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>yes</td>
</tr>
<tr>
<td>B'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C'</td>
<td></td>
<td></td>
<td></td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>yes</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

According to Table 1 the following subcases arise:

(a) $A'C'EB$, $A'C'EB'$, $A'C'EC$, $A'C'ED$, $EC'EB$, $EC'EB'$, $EC'EC'$, $EC'EC$, $EC'ED$,
(b) $C'EDE$, $CEDE$, $DEDE$,
(c) $A'CE'$, $ECE'$,
(d) $A'C'E$, $EC'E$,
(e) $EDE$.

After the application of the rules in question we have:

(b') $C'DEE$, $CDEE$, $DDEE$,
(c') $A'CB$, $ECB$,
(d') $A'C'E'$, $EC'E$,
(e') $DEE$.

For (a') it is easy to see that $A'$ can be replaced only by $E$, and the leading $E$ cannot be replaced by another symbol in these strings. A similar situation holds for (c') and (d').

For (b') and (e') the rightmost $E$ cannot be replaced by another symbol.
The remaining symbols $A'$ and $E$ can be interferred by other symbols only in the
same way as already discussed, and this completes the proof.

References