

Correction to

*An axiomatic approach to context-free rewriting
and its application to NLC graph grammars*

Theoretical Computer Science **55**(1987)141-181

by B. Courcelle

Graphs are finite, undirected, simple, loopless and vertex-labelled. A *concrete graph* is a triple consisting of a finite set of vertices, a symmetric irreflexive binary relation on this set representing the edges, and a mapping from vertices to labels. An *abstract graph* is the isomorphism class of a concrete graph.

The confluence of an NNLC graph grammar (see the paper for the definitions) can be defined in two ways, according to whether one deals with concrete or with abstract graphs. These two notions are implicitly assumed to be equivalent in the paper whereas they are not. Proposition 3.13 actually decides concrete confluence whereas it is stated for (abstract) confluence.

We now give more details. Let G be an NNLC grammar. Let (u, D) and (v, E) be two (possibly identical) production rules of G . Let H be a graph derived from the axiom of G , that has two distinct vertices x and y with respective labels u and v . Let K and L be two graphs respectively isomorphic to D and E such that H , K and L are pairwise disjoint. The concrete graphs $M = H[K/x][L/y]$ and $N = H[L/y][K/x]$ are both well-defined. (We denote by $H[K/x]$ the concrete graph obtained by the replacement in H of the vertex x by the graph K ; edges between the vertices of K and the neighbours of x in H are defined in terms of the connecting relation of G ; see the paper for details.) We say that G is *concretely confluent* if in every such case $M = N$; we say that G is *confluent* (as in the paper) if in every such case M and N are isomorphic (where isomorphisms must preserve a certain ordering of the nonterminal vertices; see the paper for details.)

The following example (due to J. Engelfriet who pointed out the error) shows that the two notions of confluence are not equivalent. We take the following rules:

$$(s, u \bullet \dots v), \quad (u, a \bullet \dots b), \quad (v, a \bullet \dots b).$$

The axiom is s , the connecting relation is $\{(u, a), (v, a), (a, b)\}$. The following two derivations give the same abstract graphs but different concrete graphs:

$$\begin{aligned} s &\rightarrow u \bullet \dots v &\rightarrow b \bullet \dots a \bullet \dots v &\rightarrow b \bullet \dots a \bullet \dots b \bullet \dots a \\ s &\rightarrow u \bullet \dots v &\rightarrow u \bullet \dots a \bullet \dots b &\rightarrow a \bullet \dots b \bullet \dots a \bullet \dots b. \end{aligned}$$

We use Roman letters a and b , as opposed to Italics, to represent the concrete copy of the rule with lefthand side u that is disjoint with the other rule. In this way one can see that the two generated graphs are isomorphic (same letters) but not equal (because of differences of polices.)

This grammar is confluent but not concretely confluent.

We now review the results of the paper that may be concerned with this fact. See the paper for statements and proofs.

Lemma 3.11 holds for concrete pairwise disjoint graphs H, H', H'' and with $=$ representing the equality (not the isomorphism) of concrete graphs.

Remark 3.12: BNLC and NUNLC grammars are always concretely confluent.

In **Proposition 3.13**, one should read "concretely confluent" instead of "confluent".

Whether the confluence of an NNLC grammar is decidable is an open question.

Lemma 3.14 holds for concrete graphs with $=$ representing equality as well as isomorphism: with the notation of the proof, the graph H_1 is always a subgraph of H_2 . Hence they are isomorphic iff they are equal.

It follows that we need not distinguish "concrete" associativity from associativity: the two notions coincide. We shall say that a grammar is *concretely context-free* if it is concretely confluent and associative. It is *context-free* (as in the paper) if it is confluent and associative.

Proposition 3.15 holds true.

In **Corollary 3.16**, one should read "concretely context-free" instead of "context-free". Whether the context-freeness of an NNLC grammar is decidable is an open question.

Proposition 3.18 holds true; in addition, it holds for concretely confluent and concretely context-free grammars. The given proof concerns actually the cases of concretely confluent and of associative grammars. Another proof can be written to give the case of confluent grammars. All other cases follow easily.

Proposition 3.19 holds true; in addition, it holds for concretely context-free grammars.

All results of **Section 4** hold without any change.

In **Section 5**, one should read everywhere "concretely confluent" and "concretely context-free" instead of "confluent" and "context-free".