Maximal Confluent Processes

Xu Wang

International Institute of Software Technology, United Nations University
P.O. Box 3058, Macau
wx@iist.unu.edu

Abstract. In process semantics of Petri Net, a non-sequential process is a concurrent run of the system represented in a partial order-like structure. For transition systems it is possible to define a similar notion of concurrent run by utilising the idea of confluence. Basically a confluent process is an acyclic confluent transition system that is a partial unfolding of the original system. Given a non-confluent transition system $G$, how to find maximal confluent processes of $G$ is a theoretical problem having many practical applications.

In this paper we propose an unfolding procedure for extracting maximal confluent processes from transition systems. The key technique we utilise in the procedure is the construction of granular configuration structures (i.e. a form of event structures) based on diamond-structure information inside transition systems.

1 Introduction

Confluence is an important notion of transition systems. Previously there has been extensive work devoted to its study, e.g. \cite{10,7,6,9}. In \cite{7} confluence is studied from the perspective of non-interleaving models, where it was concluded that in order to characterise the class of confluent transition systems the underlying event-based models needs to support the notion of or-causality \cite{16,19}.

In this paper we are going to study the idea of maximal confluent sub-systems of a non-confluent transition system, also from a non-interleaving perspective. It can be regarded as an extension of the notion of non-sequential processes in Petri Net \cite{5,2,3} onto transition systems. We call it maximal confluent process (MCP). Intuitively a maximal confluent process is a concurrent run of the system that is maximal both in length and in degree of concurrency. A non-confluent system has multiple such runs. Non-maximal concurrent runs can be deduced from maximal ones, e.g. by restricting concurrency (i.e. strengthening causality relation).

Like non-sequential processes, which can be bundled together to form branching processes of Petri Net, the set of maximal confluent processes (extracted from a given transition system) can coalesce into a MCP branching processes of the original system. Such branching processes record, in addition to causality information, also the ‘choice points’ of the system at which different runs split from each other. In a non-interleaving setting the ‘choice points’ are formalised as (immediate) conflicts on events. The arity of the conflicts can be non-binary,
thus giving rise to the so called finite conflicts. For instance, in state $s_0$ of Figure 1, actions $a$, $b$ and $c$ form a ternary conflict, which induces the three maximal concurrent runs of the system (i.e. the three subgraphs on the right).

In this paper we propose an unfolding procedure to construct granular configuration structures from transition systems. The procedure preserves the maximality of confluence in such a way that each generated configuration corresponds to a prefix of some maximal concurrent run. Configuration structures are an event structure represented in a global-state based fashion \cite{15,14}. They support or-causality as well as finite conflicts.

2 Motivating Examples

We first look at two examples in order to build up some intuitions for maximal confluent processes.

![Full Graph](image1)

**Fig. 1.** A running example

The first example is the left-most graph in Figure 1, which is an LTS in the shape of a broken cube (i.e. replacing transition $s_4 \xrightarrow{a} s_8$ by $s_4 \xrightarrow{a} s_7$ will give rise to a true cube-shaped LTS). The three subgraphs on its right are confluent subgraphs of the broken cube. Moreover, they are maximal such subgraphs; adding any state or transition to them will invalidate their confluence. They are exactly the maximal confluent processes we are looking for.

![LTS MCP 1 MCP 2 MCP 3](image2)

**Fig. 2.** The second example

For the general cases, however, maximal confluent processes do not coincide with maximal confluent subgraphs. Let us look at the left-most LTS in Figure 2. The maximal confluent subgraphs of such system are the four maximal simple paths in the graph, i.e. $s_1 \xrightarrow{a} s_2 \xrightarrow{a} s_3$, $s_1 \xrightarrow{a} s_2 \xrightarrow{b} s_3$, $s_1 \xrightarrow{b} s_2 \xrightarrow{a} s_3$, and...