Towards a Theory of Bisimulation for the Higher-Order Process Calculi

Yong-Jian Li and Xin-Xin Liu
Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100080
P.R. China
E-mail: lyj238@ios.ac.cn
Received December 25, 2002; revised September 12, 2003.

Abstract In this paper, a labelled transition semantics for higher-order process calculi is studied. The labelled transition semantics is relatively clean and simple, and corresponding bisimulation equivalence can be easily formulated based on it. And the congruence properties of the bisimulation equivalence can be proved easily.

Keywords higher-order process, labelled transition semantics, barbed bisimulation, context-bisimulation

1 Introduction

Labelled semantics achieved great success in early studies of process calculi. Based on labelled semantics, the idea of bisimulation was introduced and formulated by Milner and Park and has become one of the most important notions in process calculi. In mobility-free calculi such as CCS which has later become a blueprint for new process calculi, a labeled transition system describes the operational behaviors of processes, and bisimulation is defined on top of the labeled transition system (LTS) by imposing the following requirement: two processes are bisimilar if any action by one of them can be matched by an equal action from the other in such a way that the resulting derivatives are again bisimilar. Note that two matching actions must be syntactically identical. It is reasonable in first-order process calculus, because the effect of the first-order object on the environment can be entirely represented by its syntax form.

When moving to more powerful process calculi with mobility and higher order features, the traditional labelled semantics become hard to work with, and corresponding bisimulation formulations are often much more elaborated, resulting in less satisfactory theories. It seems that two problems have made traditional labelled semantics difficult for richer process calculi. The first is passing of local names out of the scope where they are defined. The scope extrusion rule was introduced to cope with this, but some delicate structure was also introduced into the labels, which complicated the definition of labelled transition relation. The second problem is the meaning of the label of higher order output action in traditional LTS. It only takes into account the object that the process emits, but not the context in which the emitted object is supposed to be used. For example, let us consider an output action in HOn's LTS.

\[ P \xrightarrow{\nu\alpha(K)} P' \]

where \( \nu\alpha(K) \) represents the output object of \( P \), and \( P' \) represents the derivative of \( P \). And the whole transition rule does not contain any information about the context in which emitted object is supposed to be used.

This traditional higher order output transition will cause the non-naturalness problems in the definition of bisimulation. In more detail, when defining some kind of bisimulation \( R \) of two example processes \( P, Q \), we cannot use the traditional clause as below to define the matching of the higher-order output actions of the two processes.

- Whenever \( P \xrightarrow{\mu} P' \), then \( Q' \) exists s.t. \( Q \xrightarrow{\mu} Q' \) and \( P' R Q' \), where \( \mu \) is a higher-order output action.

Unlike first-order output action, the comparison of labels of high order output action, for example, processes, cannot be based on syntactic identity in formulation of bisimulation, that would be too strong for any reasonable semantic equivalence. Thomsen used bisimilarity instead of identity in comparing labels, following earlier ideas of Astesiano and Boudol, but the resulting equivalence is still too strong.
Sangiorgi had some very illustrative examples of the problems in his dissertations[5]. And he pointed out that the separation between the object part (i.e., the process emitted) and the context of a higher-order output prevents a satisfactory treatment of the channels private to the two, and then causes the problems of higher order bisimulation mentioned above. To avoid this separation between object part and context of an output action, he proposed a context bisimulation as below which explicitly takes into account the context in which the emitted object is supposed to be used.

- Whenever \( \frac{P \overset{(\nu)b a(K)}{\longrightarrow} P'}{Q \overset{(\nu)b a(K)}{\longrightarrow} Q'} \) exists s.t. \( Q \overset{P}{\longrightarrow} Q' \) and for all \( PG \) with \( fn(PG) \cap (b \cup c) = \emptyset, \) \( (\nu b)(P|PG{K_1/U})|Q \overset{(\nu c)}{\longrightarrow} Q'|PG{K_2/U}. \) \( (2) \)

But this style of context bisimulation still seems nonsensical and complex. First, the clauses of the definition of bisimulation are not defined in a uniform style. The clause, which defines matching between higher-order output actions of two processes, is different from other clauses. Especially, the principle that underlies the definition of matching of higher-order output actions adopts the contextual view, which is defined as the clause (2); on the other hand, the definition of matching of first-order output actions adopts the non-contextual view which is defined as the clause (1). Second, this context bisimulation is very complex. When he showed the correspondence between his context bisimulation and a version of barbed equivalence, the proof is too complex[5].

In [9], Sewell introduced the contextual point of view of labelled semantics and opened up a new alternative towards the solution to the above problems. He pinpointed the connection between labelled semantics and reduction semantics by making explicit the intuition that labelled transitions capture the possible interactions between a term and a surrounding context. Roughly his idea is that labels of transitions from a process \( P \) will be contexts that, when put together with \( P \), create an occurrence of a reduction rule. As a test of the idea, he it is shown that the new definition of labelled transition induces the same bisimulation for a fragment of CCS. Notice that the key difference between Sewell's contextual transition label and traditional transition label is in that the former comes from the process that performs the action, but the latter comes from the process environment (or context).

In this paper we want to show that the same idea can be successfully applied to higher-order process calculi. In our labelled transition semantics, there are three kinds of labels: \( \tau, a(U).Q, \bar{a}(K).0 \). As usual, \( P \overset{\tau}{\longrightarrow} P' \) represents internal communication as traditional LTS. However \( P \overset{a(U).Q}{\longrightarrow} P' \) and \( P \overset{\bar{a}(K).0}{\longrightarrow} P' \) represent that \( P \) can respond to the tests \( a(U).Q \) and \( \bar{a}(K).0 \) from context respectively, and then becomes \( P' \). We will show that our labelled transition semantics are relatively clean and the corresponding bisimulation equivalence can be easily obtained.

To show the correctness and rationality of our semantics, we want to show the correspondence between our labelled semantics and those well-established. First, we want to relate our LTS with reduction semantics. And we will characterize our new bisimulation in terms of a version of barbed equivalence. Second, we relate our LTS semantics to Sangiorgi's LTS. And our new bisimulation will be characterized in terms of a version of context bisimulation defined in his LTS.

The rest of the paper is organized as follows. Section 2 reviews the syntax of the calculus that will be considered in this paper. Section 3 specifies the contextual labelled transition semantics for the calculus. Sections 4 and 5 will show the correspondence between our labelled transition semantics and reduction semantics and the traditional labelled transition semantics. Section 6 is the conclusion and related work.

2 Syntax

The language we explored is similar to the second-order fragment of reduced HO\( \pi \) proposed in [5] by Sangiorgi, which satisfies the following restrictions: 1) all arities are unary, 2) each process is finitely describable, 3) only guarded choices are permitted, 4) only processes are allowed to transmit. In this calculus replication may be avoided because recursion and replication can be simulated using restriction and parallel composition in our calculus, as shown by Thomsen[6]. We still think it is convenient for us to have replication explicitly, and its presence does not complicate our theory. And we use \( K, L, P, Q, R \) to range over processes; \( a, b, c, \ldots \), to range over names; \( U, V, X, Y \) to range over variables. The syntax is as follows:

\[
P ::= \sum_{i \in I} [a_i.P]|P_1|P_2|\nu a.P|X|P
\]

\[
a ::= \bar{a}(P)|a(X)
\]

As usual, we write \( P\{Q/X\} \) for the capture-