Observing Reductions in Nominal Calculi *Via* a Graphical Encoding of Processes*

Fabio Gadducci and Ugo Montanari

Dipartimento di Informatica, Università di Pisa

Abstract. The paper introduces a novel approach to the synthesis of labelled transition systems for calculi with name mobility. The proposal is based on a graphical encoding: Each process is mapped into a (ranked) graph, such that the denotation is fully abstract with respect to the usual structural congruence (i.e., two processes are equivalent exactly when the corresponding encodings yield the same graph).

Ranked graphs are naturally equipped with a few algebraic operations, and they are proved to form a suitable (bi)category of cospans. Then, as proved by Sassone and Sobocinski, the synthesis mechanism based on *relative pushout*, originally proposed by Milner and Leifer, can be applied. The resulting labelled transition system has ranked graphs as both states and labels, and it induces on (encodings of) processes an observational equivalence that is reminiscent of early bisimilarity.

Keywords: Nominal calculi, reduction semantics, synthesised labelled transition systems, relative pushouts, graph transformations.

1 Introduction

The dynamics of many computational devices is often defined in terms of *reduction relations*. Let us consider for example the paradigmatic functional language, the λ-calculus. Its operational semantics is aptly provided by the *β-reduction rule* $(\lambda x.M)N \Rightarrow M[N/x]$ that models the application of a functional process $\lambda x.M$ to the actual argument N. The reduction relation is then obtained by freely instantiating and contextualising the rule. This is quite typical in many calculi, since such a rule represents an *internal reduction* of a system component.

Moving towards calculi for interaction, let us consider now the reduction rule $a.P \mid \bar{a} \Rightarrow P$ for *asynchronous CCS-like communication*. The metavariable P actually denotes any possible process, let it be $P = \bar{b}$, and the rule can be contextualised in unary contexts such as $C[\_] = b.\mathsf{0} \mid [\_]$. Under those assumptions, the mechanism yields the rewriting step $b.\mathsf{0} \mid a.\bar{b} \mid \bar{a} \Rightarrow b.\mathsf{0} \mid \bar{b}$.

Reduction semantics have the advantage of conveying the semantics of calculi with relatively few compact rules. Its main drawback is poor compositionality, in

* Partly supported by the EU within the project HPRN-CT-2002-00275 SegraVis (*Syntactic and Semantic Integration of Visual Modelling Techniques*); and within the FETPI Global Computing, project IST-2004-16004 SEnSOria (*Software Engineering for Service-Oriented Overlay Computers*).

A. Middeldorp et al. (Eds.): Processes... (Klop Festschrift), LNCS 3838, pp. 106–126, 2005.

the sense that the dynamic behaviour of arbitrary stand alone terms (like $a.P$ in the example above) can be interpreted only by inserting them in the appropriate context (i.e., $[\_] \mid \bar{a}$), where a reduction may take place.

In different terms, reduction semantics is often less suitable whenever specific behaviours other than confluence (termination, reachability) are of interest. In fact, simply using the reduction relation for defining equivalences between components (e.g. in terms of *bisimulation*) fails to obtain a compositional framework, and in order to recover a suitable notion of equivalence it is often necessary to verify the behaviour of single components under any viable execution context. This is the way leading from the research on termination-under-context-closure equivalences for the λ-calculus to barbed and dynamic equivalences for the π-calculus. In these approaches, though, proofs of equivalence are often tedious as well as involuted, and they are left to the ingenuity of the researcher.

A standard way out of the empasse, reducing the complexity of such analyses, is to express the behaviour of a computational device by a *labelled transition system* (LTS). Should the label associated to a component evolution faithfully express how that component might interact with the whole of the system, it would be possible to analyse *in vitro* the behaviour of a single component, without considering all contexts. Thus, a "well-behaved" LTS represents a fundamental step towards a compositional semantics of the computational device.

Milner's proposal for an alternative semantics for the π-calculus [18] based on reactive rules modulo a suitable structural congruence, inspired by the CHAM paradigm [4], has been the source of an ongoing stream of research focussing on the investigation of the relationship between the LTS based semantics for nominal calculi and their more abstract reduction semantics.

Early attempts by Sewell [24] devised a strategy for obtaining an LTS from a reduction relation by adding contexts as labels on transitions. The technique was refined by Leifer and Milner [16] who introduced *relative pushouts* (RPOs) in order to capture the notion of *minimal context* activating a reduction. The generality of this proposal (and its bicategorical formulation due to Sassone and Sobocinski [22]) allows it to be applied to a large class of formalisms. More importantly, such attempts share the basic property of synthesising a congruent bisimulation equivalence, thus ensuring that the resulting LTS semantics is compositional. However, for the time being there are few case studies which either involve rich calculi, or succeed in making comparisons with standard behavioural equivalences. To tackle a full-fledged case study is the main aim of this paper.

Our starting point for the synthesis of an LTS are the graphical techniques proposed by the authors for modelling the reduction semantics of nominal calculi [11]. There is a long tradition in the use of graphical formalisms for describing the operational semantics of a computational device. They are often biased towards an implementation view, ranging from the functional paradigm (culminating on the works on *optimal implementation* [17]) to the imperative one (using *term graph rewriting* as an efficient technique for equational deduction [2]).

Only recent years have seen proposals concerning the use of graphical techniques for simulating reduction in process calculi, in particular for their mobile