NoCo: System Description for CoCo 2015

Takaki Suzuki, Kentaro Kikuchi and Takahito Aoto

RIEC, Tohoku University, Sendai, Japan
{ takaki, kentaro, aoto }@nue.riec.tohoku.ac.jp

Nominal rewriting [2, 3] is a framework that extends first-order term rewriting by a binding mechanism. Studies of nominal rewriting are preceded by extensive studies of a *nominal approach* to terms and unifications [4, 5, 7]. A distinctive feature of the nominal approach is that α-conversion and capture-avoiding substitution are not relegated to meta-level—they are explicitly dealt with at object-level. This makes nominal rewriting significantly different from classical frameworks of higher-order rewriting systems based on 'higher-order syntax'.

NoCo (Nominal Confluence tool) is an automated confluence prover for *nominal rewrite systems (NRSs)*. The tool has been developed in Toyama–Aoto group in RIEC, Tohoku University and has been reported in [6]. NoCo is written in Standard ML of New Jersey (SML/NJ). The tool registered to the category of confluence of nominal rewrite systems that has been adopted as one of the demonstration categories in CoCo 2015. Up to our knowledge, it is a first tool that deals with confluence of NRSs.

NoCo proves whether input NRSs are Church-Rosser modulo the α-equivalence (CR$\approx_\alpha$) based on Corollary 40 of [6]. Notions and notations in the following explanation are based on [6]. The corollary provides the following conditions for NRS $\mathcal{R}$ being CR$\approx_\alpha$: (1) $\mathcal{R}$ is orthogonal and (2) $\mathcal{R}$ is abstract skeleton preserving (ASP). It is straightforward to check (2), as the standardness is just a syntactical restriction and $\nabla \vdash a\#X$ is easily checked for any freshness constraint ∇, $a \in \mathcal{A}$ and $X \in \mathcal{X}$. For (1), one has to check (1-a) left-linearity and that (1-b) there's no proper overlaps. The checking of (1-a) is easy. For (1-b), one has to check whether $\nabla_1 \cup \nabla_2^{\pi_2} \cup \{l_1 \approx l_2^{\pi_2}|_p\}$ is unifiable for some permutation π_2, for given $\nabla_1, \nabla_2, l_1, l_2|_p$—this problem is different from nominal unification problems as π_2 is not fixed. Fortunately, this problem is known as an equivariant unification, and has been known to be decidable [1]. From the equivariant unification algorithm in [1], we obtain a constraint of π_2 for unifiability, if the problem is equivariantly unifiable. The system also reports concrete critical pairs generated from this constraint, if there is a proper overlap.

References

[1] J. Cheney. Equivariant unification. *Journal of Automated Reasoning*, 45:267–300, 2010.

[2] M. Fernández and M. J. Gabbay. Nominal rewriting. *Information and Computation*, 205:917–965, 2007.

[3] M. Fernández, M. J. Gabbay, and I. Mackie. Nominal rewriting systems. In *Proceedings of PPDP'04*, pages 108–119. ACM Press, 2004.

[4] M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2002.

[5] A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186:165–193, 2003.

[6] T. Suzuki, K. Kikuchi, T. Aoto, and Y. Toyama. Confluence of orthogonal nominal rewriting systems revisited. In *Proceedings of 26th RTA*, volume 36 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 301–317, 2015.

[7] C. Urban, A. M. Pitts, and M. J. Gabbay. Nominal unification. *Theoretical Computer Science*, 323:473–497, 2004.