Review of Process Algebra and Non-interference

Riccardo Pucella

February 28, 2001

Ryan and Schneider analyze the problem of non-interference in a context where the system under consideration is implemented by a process algebraic program. Unfortunately for me, they use Hoare's CSP, instead of Milner's CCS, which I grok a bit better. The results should carry over. Recall, non-interference means that in essence, the high level user is prevented from influencing the low level user; also I'm talking possibilistic non-interference, not probabilistic. Another way of stating the property is to say: "for any two pair of behaviours of the system that differ only in the high-level user's behavior, the low-level user's behavior cannot distinguish the two behaviors." You can model this in a process algebra by viewing the high-level user's behavior as the internal actions of a program, not visible from the outside. Now, everything hinges on the definition of "distinguishing two behaviors". In process algebra terms, this is process equivalence: if I have two processes, when can I consider them equivalent? The concurrency community has come up with a slew of interpretations of this equivalence, each suitable to certains situations. They get names like 'strong bisimulation' (two process are equivalent if they can do the same actions (including internal ones) in the same contexts, and moreover they make the same nondeterministic choices at the same places), or 'testing equivalence', etc. Ryan and Scheider show that the known notions of non-interference, which specify when a low-level user cannot distinguish behaviors, correspond pleasingly with known notions of process equivalence. The point being, there is not agreed to notion of process equivalence which is suitable for all contexts.

Hence, the problem of coming up with the "right" definition of non-interference can be handled by invoking something akin to invoking NP-completeness: "sorry I couldn't come up with a polynomial time alg for your problem; but I proved it was NP-complete, and there's good evidence that I probably won't be able to come up with a polynomial time alg." Similarly here, "sorry I couldn't come up with a version of non-interference that works in all contexts, but hey, neither could these concurrency theorists, and they've been thinking about it more deeply than I've been."

Of course, what *I* take out of this paper is that whatever intelligent thing we can say about information flow security, we may end up being able to say something to concurrency theorists about what they're doing; a pleasant prospect :)

Counterpoint to the tone of the paper, it's fair to say that I've heard the same kind of criticism we directed at the information flow security folks directed at the concurrency community, and from good concurrency theorists: that there is an underlying unity and intuition in process algebraic equivalences that is just not understood by the majority of the people in the field. Hence the quest for a suitable low level uniform "semantic model" for process algebraic concurrency; anyone interested in attempting to recast Ryan and Schneider's results in the context of Milner's event structures?, he asks half-seriously.