First order and higher order tile models for open and mobile systems*

Roberto Bruni and Ugo Montanari†

Dipartimento di Informatica, Università di Pisa, Italia

There are two principal ways in which the dynamics of many calculi can be naturally modeled: (1) by means of *labeled transition systems* (LTS) where states are terms in the calculus and transition labels express some observation on the evolution step; or (2) by defining a reduction semantics over terms up to a suitable structural congruence (in the style of the *chemical abstract machine* [1]). While the first approach comes straightforwardly equipped with a variety of abstract semantics (e.g. bisimilarity, trace equivalence) whose compositionality can be guaranteed by specifying the LTS accordingly to suitable syntactic formats (e.g. [8, 10, 2]), the second approach is often easier to apply but provides less informative abstract equivalences w.r.t the dynamics and interactions of components [16, 12] (e.g. reduction to the same normal form). In both cases, it is enviable that the abstract semantics be a congruence, otherwise suitable techniques should be applied for extracting a congruence out of it (e.g. closing w.r.t. all contexts, using dynamic bisimilarity [14]). Another issue is related to the study of calculi with name hiding, passing and creation, whose features often require ad-hoc techniques when defining observational congruences (e.g. early vs. late bisimilarity). Moreover, in several cases, the abstract semantics is defined on ground terms only (representing closed systems), and it is then extended to open terms (i.e. contexts) by instantiating w.r.t. all possible ground instances.

Tile logic [9, 3] attempts at reconciling the most interesting features of (1) and (2) and, at the same time, at dealing with ground and open terms in a uniform way. To this aim, transition labels become pairs, whose components are called *triggers* (expressing the interaction of a context with its arguments) and *effect* (representing the behavior offered to the rest of the system, i.e. a possible context). Tiles can be represented as rectangles where the horizontal dimension is devoted to the assembling of states and the vertical dimension is dedicated to the evolution of components. Thus, triggers and effects form the left and right sides of tiles, respectively. The vertices of tiles are called *interfaces*, connecting the input and output observations to the *initial* (before the step) and *final* (after the step) *configurations*. Thanks to the abstract notions of configuration and observation, tiles allow us to develop a theoretical framework parametric in such structures (e.g. graphs or hypergraphs or trees or λ-terms rather than terms), and able to capture analogies in the structures by means of suitable auxiliary tiles (e.g.

*This research has been partially supported by MURST project *TOSCA* as part of theme 3, task Tiles and theme 7, task TilesSuperiore.

†We warmfully thank our co-authors David de Frutos-Escrig, Narciso Martí-Oliet, Francesca Rossi, and Vladimiro Sassone for their contribution to the results presented here.

representing isomorphic transformations of interfaces or consistent abstractions and applications of functional terms). Reduction semantics and (unconditional) rewriting logic [13] are embedded in the framework as the special cases where no observation is provided. LTS are generalized by the usage of triggers, which allow to define observational equivalences (called *tile trace equivalence* and *tile bisimilarity*) directly on open terms, analogously to [11, 15].

During this talk we illustrate some of the recent progresses in the foundations of tiles. In particular, we discuss (i) suitable specification formats guaranteeing that certain observational equivalences are congruences, distinguishing between *incomplete systems* and *coordinators* [4]; (ii) the usage of auxiliary tiles for finitary presentations of context closure when bisimilarity is not a congruence, with application to open ended systems [7]; (iii) the duality between context and instantiation closure, which is evident in the categorical models of tiles and is well exemplified by an application to logic programming [6]; (iv) the introduction of higher order versions of the framework for an automatic treatment of name passing and name creation in mobile calculi [5].

References

[1] G. Berry and G. Boudol. The chemical abstract machine. *Theoret. Comput. Sci.*, 96(1):217–248, 1992.

[2] B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, 1995.

[3] R. Bruni. *Tile Logic for Synchronized Rewriting of Concurrent Systems*. PhD thesis, Computer Science Department, University of Pisa, 1999. `http://www.di.unipi.it/~bruni/publications/PhDThesis.ps.gz`

[4] R. Bruni, D. de Frutos-Escrig, N. Martì-Oliet, and U. Montanari. Bisimilarity congruences for open terms and term graphs via tile logic. *Proc. CONCUR 2000*, *LNCS* 1877, pp. 259–274. Springer, 2000. `http://www.di.unipi.it/~bruni/publications/concur2000.ps.gz`

[5] R. Bruni and U. Montanari. Cartesian closed double categories, their lambda-notation and the pi-calculus. *Proc. LICS'99*, pp.246–265. IEEE, 1999. `http://www.di.unipi.it/~ugo/LICS99.ps`

[6] R. Bruni, U. Montanari, and F. Rossi. An interactive semantics of logic programming. Submitted. `http://www.di.unipi.it/~bruni/publications/tplp.ps.gz`

[7] R. Bruni, U. Montanari, and V. Sassone. Open ended systems, dynamic bisimulation and tile logic. *Proc. IFIP TCS 2000*, *LNCS* 1872, pp. 440–456. Springer, 2000. `http://www.di.unipi.it/~bruni/publications/ifiptcs2000.ps.gz`

[8] R. De Simone. Higher level synchronizing devices in MEIJE–SCCS. *Theoret. Comput. Sci.*, 37:245–267, 1985.

[9] F. Gadducci and U. Montanari. The tile model. *Proof, Language and Interaction*. MIT Press, 2000. `http://www.di.unipi.it/~ugo/festschrift.ps`

[10] J.F. Groote and F. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Inform. and Comput.*, 100:202–260, 1992.

[11] K.G. Larsen and L. Xinxin. Compositionality through an operational semantics of contexts. *Proc. ICALP'90*, *LNCS* 443, pp. 526–539. Springer, 1990.

[12] J.J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. *Proc. CONCUR 2000*, *LNCS* 1877, pp. 243–258. Springer, 2000.

[13] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoret. Comput. Sci.*, 96:73–155, 1992.

[14] U. Montanari and V. Sassone. Dynamic congruence vs. progressing bisimulation for CCS. *Fundamenta Informaticae*, 16:171–196, 1992.

[15] A. Rensink. Bisimilarity of open terms. *Inform. and Comput.*, 156:345–385, 2000.

[16] P. Sewell. From rewrite rules to bisimulation congruences. *Proc. CONCUR'98*, *LNCS* 1466, pp. 269–284. Springer, 1998.