Bisimilarity Congruences for Open Terms and Term Graphs via Tile Logic*

R. Bruni[1], D. de Frutos-Escrig[2], N. Martí-Oliet[2], and U. Montanari[1]

[1] Dipartimento di Informatica, Università di Pisa, Italia.
[2] Dept. Sistemas Informáticos y Programación, Univ. Complutense de Madrid, Spain.
bruni@di.unipi.it, {defrutos,narciso}@sip.ucm.es, ugo@di.unipi.it

Abstract The definition of SOS formats ensuring that bisimilarity on closed terms is a congruence has received much attention in the last two decades. For dealing with open terms, the congruence is usually lifted from closed terms by instantiating the free variables in all possible ways; the only alternatives considered in the literature are Larsen and Xinxin's *context systems* and Rensink's *conditional transition systems*. We propose an approach based on *tile logic*, where closed and open terms are managed uniformly, and study the 'bisimilarity as congruence' property for several tile formats, accomplishing different concepts of open system.

Introduction

The semantics of many languages can be conveniently expressed via *labelled transition systems* (LTSs) whose states are terms over a certain signature and whose labels give some abstract information about system evolution. If such information is sufficient to model the interactions between the various (sub)systems when they are composed through the operators of the language, then the specification can be assembled compositionally. Plotkin's *structural operational semantics* (SOS) [18] is surely one of the most successful frameworks for LTS descriptions, where transitions are defined by induction on the structure of states; then, one is interested in equating all those states with the same behavior. In this paper we rely on *bisimulation* equivalence [17]: two states are equivalent if every action of the one can be simulated by the other, still ending in equivalent states. The need of a compositional framework, where each subcomponent of a state can be safely replaced by any equivalent subcomponent without affecting the overall behavior, motivated a lot of efforts in the definition of SOS formats (e.g., *De Simone* [8], GSOS [1], `tyft/tyxt` [12]) whose syntax guarantees that *bisimilarity is a congruence* — bisimilarity meaning the largest bisimulation. However, the congruence is usually defined on closed terms only; it can be extended to *contexts* (terms with variables) by letting two contexts be equivalent if they are equivalent under all possible instantiations of their arguments. This

* Research supported by CNR Integrated Project *Progettazione e Verifica di Sistemi Eterogenei*, Esprit WG *CONFER2* and *COORDINA*, MURST project *TOSCA*, and CICYT project *Desarrollo Formal de Sistemas Distribuidos* (TIC97–0669–C03–01).

yields the coarsest conservative extension of the equivalence that is preserved by instantiation and, to some extent, is analogous to reducing an equivalence on closed terms to the congruence respected by all contexts.

When designing *open systems*, the preservation of a given equivalence under all possible instantiations should of course be a necessary property; however, it is not very practical to define it by universal quantification on all possible instantiations, since proofs of properties will become difficult. It would be preferable to extend the bisimulation game from closed states to contexts, providing a uniform framework. Thus, the variables of a context C can be viewed as forming its *input interface*, used by C for communicating with its arguments. Then, the label of a transition with source C must include some information about the moves accomplished by each argument for allowing the context to evolve. Moreover, for nonlinear contexts, the degree of subterm sharing may make a difference. In fact, for partially specified components it is convenient to distinguish between the situation in which different copies of a nondeterministic resource are available, which can evolve independently, and the situation in which the resource is shared between many components (and thus the resource must offer the same behavior to all its users). Conceptually, this originates two kinds of open systems that we call *incomplete systems* and *coordinators*.

A first situation would be the case in which variables represent software components possibly used by several processes. For example, one can take a CCS-like context $C[x]|D[x]$, with x a process variable, where the same protocol specification should be employed by both C and D to provide a certain service. In such a case, the two instances of x in $C[x]|D[x]$ must be instantiated with two copies of the same pattern (a suitable agent), which can progress independently.

On the other hand, if the process $C[x]|D[x]$ represents a coordinator which regulates the execution of the argument x then, when x is instantiated to a process q, replication must be avoided: both C and D become connected to *the same* agent q. In this case the two occurrences of q must evolve in the same way and at the same time. Notationally this situation can be expressed by writing $\mathbf{let}\ x = q\ \mathbf{in}\ C[x]|D[x]$ instead of $C[q]|D[q]$. A concrete example for this situation is when several consumers receive information from a single producer by means of possibly replicated handshaking channels: We desire all the consumers to receive the same information through all their channels at the same time.

Ordinary SOS formats do not seem suitable to deal with this latter view in the correct way. For example, let us consider the well-known SOS specification for Milner's finite CCS [17], whose rules are in De Simone format, and take the contexts $C[x] = x\backslash_\alpha|x\backslash_\alpha$ and $D[x] = (x|x)\backslash_\alpha$. Then, the two contexts cannot react as coordinators to any move of x and thus they are equivalent according to bisimilarity (since x is shared, each move of x is replicated along its two occurrences and it is not possible to have complementary actions). However, if we instantiate x with the process $p = \alpha.nil + \bar{\alpha}.nil$, then $C[p]$ is no longer bisimilar to $D[p]$, because the former is stuck and the second can make a τ.

These two cases do not exhaust all the possible ways to treat nonlinear contexts, but they give us a motivation to study different ways to cope with the problem: (1) considering only linear contexts; (2) allowing free duplication of shared components, as if, e.g., the CCS context $(x|x)\backslash_\alpha$ could make a transition to $(x_1|x_2)\backslash_\alpha$, disconnecting the shared resource, and observing that its argument x has been duplicated; (3) employing *term graphs* [9] instead of terms to describe states. The first approach avoids sharing at all. The second proposal enriches the observation algebra. The third proposal is especially attractive for the specification and analysis of distributed systems. In fact, term graphs are finer than terms and allow one to express the sharing of closed subterms, hence distinguishing, e.g., the coordinated system **let** $x = p$ **in** $(x|x)\backslash_\alpha$ from $(p|p)\backslash_\alpha$.

As far as we know, the extension of good SOS formats to incomplete systems has been addressed only recently by Rensink [19], who exploited a previous idea of De Simone. Rensink formalized several extensions to open terms of the bisimilarity on closed terms, based on the notion of *conditional transition systems*. We propose instead the use of *tile systems*, where the extensions (1)–(3) discussed above can be straightforwardly handled, at the levels of both specifications and computational models. The *tile model* [11] is a formalism for modeling open compositional systems. It relies on certain rewrite rules with side effects, called *basic tiles*, which are reminiscent of both SOS rules and *context systems* [14]. It collects intuitions coming from *structured transition systems* [7] and *rewriting logic* (RL) [15], and by analogy with RL, the tile model has a logical presentation, called *tile logic*, where tiles are seen as sequents subject to certain inference rules. Tiles are written $\alpha\colon s \xrightarrow[b]{a} t$, where s and t can be open terms, stating that the *initial configuration* s evolves to the *final configuration* t producing the *effect* b. However, such a step is allowed only if the subcomponents supplied as arguments to s evolve to the subcomponents of t, producing the observation a, which is the *trigger* of the tile α. Triggers and effects are called *observations*.

In this paper, we want to stress that tiles are designed for open systems, and in fact, the notion of bisimulation can be generalized to that of *tile bisimulation*, which directly operates over contexts. More precisely, we investigate tile formats guaranteeing that tile bisimilarity is a congruence. The first tile format appeared in the literature is the *algebraic tile format* (ATF) [10,11]. It has a cartesian structure of configurations and a monoidal structure of observations.[1] We show that the ATF is not completely satisfactory for our aims, due to the different structure of configurations and observations. We focus instead on three tile formats that reflect the approaches — linearity, free duplication and explicit sharing — discussed above: (1) the *monoidal tile format* (MTF) [16], whose configurations and observations have a monoidal structure; (2) the *term tile format* (TTF) [4], whose configurations and observations have a cartesian structure; (3)

[1] Essentially, cartesian configurations can be seen as substitutions over a given signature (e.g., substitution $[t_1(x_1,\ldots,x_n)/y_1,\cdots,t_m(x_1,\ldots,x_n)/y_m]$ corresponds to a configuration $t\colon n \to m$). In monoidal configurations terms $t_1,\ldots,t_m$ are linear in their variables. In gs-monoidal configurations, sharing of subterms is possible, but it cannot be eliminated by copying, i.e., **let** $x = t_1$ **in** t_2 is different from $t_2[t_1/x]$ whenever x does not occur, or occurs more than once, in t_2.

the *gs-monoidal tile format* (GSTF, from graph substitution), whose configurations have a gs-monoidal structure and observations have a monoidal structure.

By imposing the so called *basic source* constraint (initial configurations of basic tiles must consist of a basic operator) on the ATF, one can recover a slightly more general format than De Simone but stricter than GSOS. In this case, tile bisimilarity recovers the ordinary congruences for closed terms, but not for contexts. The basic source on MTF and TTF yields respectively De Simone and a format more general than (positive) GSOS. In both cases, tile bisimilarity is a congruence also for open terms. The GSTF is hard to recast in existing SOS formats, due to its treatment of subterm sharing. We are indeed confident that GSTF can find a meaningful application in the modeling of systems with shared resources. Again, the basic source guarantees that tile bisimilarity is a congruence.

Outline. In Section 1 we fix the notation and recall the notions of bisimulation and tile bisimulation, while in Section 2 we review the most common SOS formats and several tile formats. In the presentation, category theory and algebraic theories are employed in a mild way. Sections 3, 4 and 5 deal with MTF, TTF and GSTF respectively, showing the main results of the paper, namely that for all of them the basic source implies that tile bisimilarity is a congruence. Due to space limitation, we omit all proofs, which can be found in the technical report [3]. Section 6 compares our approach with Rensink's proposal in [19].

1 Tiles and Bisimulation

Notation. To ease the presentation we will consider only one-sorted signatures, though our results extend to the many-sorted case. A *one-sorted signature* is a set of *operators* Σ together with an arity function $ar\colon \Sigma \to \mathbb{N}$. For $n \in \mathbb{N}$, we let $\Sigma_n = \{f \in \Sigma \mid ar(f) = n\}$. Operators in Σ_0 are called *constants*. We denote by $\mathbb{T}_\Sigma(X)$ the term algebra over Σ and variables in X (disjoint from Σ), with $\mathbb{T}_\Sigma = \mathbb{T}_\Sigma(\varnothing)$. For $t \in \mathbb{T}_\Sigma(X)$ we denote by $var(t)$ the set of variables that appear in t. If $var(t) = \varnothing$ then t is called *closed*, otherwise *open*. A term is *linear* if each variable occurs at most once in it.

A *substitution* is a mapping $\sigma\colon X \to \mathbb{T}_\Sigma(X)$. It is closed if each variable is mapped into a closed term. Substitutions uniquely extend to mappings from terms to terms as usual: $\sigma(t)$ is the term obtained by simultaneously replacing all occurrences of variables x in t by $\sigma(x)$. The substitution mapping x_i to t_i for $i \in [1,n]$ is denoted by $[t_1/x_1,\ldots,t_n/x_n]$, and it is *linear* if each t_i is linear and $var(t_i) \cap var(t_j) = \varnothing$ for $i \neq j$. A substitution σ' can be applied elementwise to $\sigma = [t_1/x_1,\ldots,t_n/x_n]$ yielding $\sigma;\sigma' = [\sigma'(t_1)/x_1,\ldots,\sigma'(t_n)/x_n]$.

A *context* $t = C[x_1,\ldots,x_n]$ denotes a term in which at most the distinct variables $x_1,\ldots,x_n$ appear. The term $C[t_1,\ldots,t_n]$ is then obtained by applying the substitution $[t_1/x_1,\ldots,t_n/x_n]$ to the context $C[x_1,\ldots,x_n]$, which can thus be regarded as a function from n arguments to 1. Note that the x_i may as well not appear in $C[x_1,\ldots,x_n]$. For example, the context $x_2[x_1,x_2,x_3]$ is a substitution from three arguments to one, which is the projection on the second argument.

Bisimulation. A *labelled transition system* (LTS) is a triple $L = (S, \Lambda, \rightarrow)$, where S is a set of *states*, Λ is a set of *labels*, and $\rightarrow \subseteq S \times \Lambda \times S$. We let $s, t, \ldots$ range over S and $a, b, c, \ldots$ range over Λ. For $\langle s, a, t\rangle \in \rightarrow$ we say that s is the *source*, t is the *target* and a is the *observable*, and use the notation $s \xrightarrow{a} t$.

Definition 1. *Given an* LTS $L = (S, \Lambda, \rightarrow)$*, a* bisimulation *on* L *is a symmetric, reflexive relation* $\sim \subseteq S \times S$ *such that if* $s \sim t$ *then for any transition* $s \xrightarrow{a} s'$ *there exists a transition* $t \xrightarrow{a} t'$ *such that* $s' \sim t'$*. We denote by* $\simeq$ *the largest bisimulation, called* bisimilarity*, and we say that two states* s *and* t *are* bisimilar *whenever* $s \simeq t$*, i.e., whenever there exists a bisimulation* $\sim$ *such that* $s \sim t$*.*

When $S = \mathbb{T}_\Sigma$, an essential property of bisimilarity is *compositionality* w.r.t. operators in Σ, guaranteeing that operationally equivalent subsystems can be safely replaced in any context. This amounts to requiring that $\simeq$ is a congruence, i.e., that for all $f \in \Sigma$, $s_i \simeq t_i$ for $i \in [1, ar(f)]$ implies $f(s_1, \ldots, s_{ar(f)}) \simeq f(t_1, \ldots, t_{ar(f)})$. Bisimilarity can be lifted to contexts by letting $C[x_1, \ldots, x_n] \simeq D[x_1, \ldots, x_n]$ if for all $t_1, \ldots, t_n \in \mathbb{T}_\Sigma$ we have $C[t_1, \ldots, t_n] \simeq D[t_1, \ldots, t_n]$.

Tile bisimulation. The point of view of the tile model [11] is that bisimulation can be defined uniformly on both closed terms and contexts, without resorting to instantiation closure. In the following we use monoidal categories for providing an abstract presentation of tile systems. As a matter of notation, sequential composition and monoidal tensor product are denoted by $\_;\_$ and $\_ \otimes \_$, respectively. The unfamiliar reader may consult, e.g., [11,2].

A *tile system* is a tuple $\mathcal{R} = (\mathcal{H}, \mathcal{V}, N, R)$ where $\mathcal{H}$ and $\mathcal{V}$ are monoidal categories with the same set of objects $O_\mathcal{H} = O_\mathcal{V}$, N is the set of rule names and $R: N \rightarrow \mathcal{H} \times \mathcal{V} \times \mathcal{V} \times \mathcal{H}$ is a function such that for all $\alpha \in N$, if $R(\alpha) = \langle s, a, b, t\rangle$ then $s: x \rightarrow y$, $a: x \rightarrow z$, $b: y \rightarrow w$, and $t: z \rightarrow w$ for suitable objects x, y, z and w. We will write such rule α either as $\alpha: s \xrightarrow[b]{a} t$, or as the tile

$$\begin{tikzcd} x \arrow[r, "s"] \arrow[d, "a"'] \arrow[dr, phantom, "\alpha"] & y \arrow[d, "b"] \\ z \arrow[r, "t"'] & w \end{tikzcd}$$

The category $\mathcal{H}$ is called *horizontal* and its arrows are called *configurations*. The category $\mathcal{V}$ is called *vertical* and its arrows are called *observations*. The objects of $\mathcal{H}$ and $\mathcal{V}$ are called *interfaces*. Starting from the basic tiles, more complex tiles can be constructed by means of horizontal, vertical and parallel composition. Moreover, the horizontal and vertical identities are always added and composed together with the basic tiles. All this is illustrated in Figure 1.

Depending on the chosen tile format (see Section 2), $\mathcal{H}$ and $\mathcal{V}$ can be specialized (e.g., to cartesian categories) and suitable *auxiliary tiles* are added and composed with basic tiles and identities in all the possible ways. The set of resulting tiles (also called *flat sequents*) define the *flat tile logic* associated to $\mathcal{R}$. We say that $s \xrightarrow[b]{a} t$ is *entailed* by the logic, written $\mathcal{R} \vdash s \xrightarrow[b]{a} t$, if the sequent $s \xrightarrow[b]{a} t$ can be expressed as the composition of basic and auxiliary tiles.

$$\frac{s \xrightarrow[b]{a} t \quad h \xrightarrow[c]{b} f}{s; h \xrightarrow[c]{a} t; f} \qquad \frac{s \xrightarrow[b]{a} t \quad t \xrightarrow[d]{c} h}{s \xrightarrow[b;d]{a;c} h} \qquad \frac{s \xrightarrow[b]{a} t \quad h \xrightarrow[d]{c} f}{s \otimes h \xrightarrow[b \otimes d]{a \otimes c} t \otimes f} \qquad \frac{t: x \to y \in \mathcal{H}}{t \xrightarrow[y]{x} t} \qquad \frac{a: x \to z \in \mathcal{V}}{x \xrightarrow[a]{a} z}$$

Figure 1. Composition and identity rules for tile logic.

$$\text{(a)} \ \frac{\{s_i \xrightarrow{a_i} t_i \mid i \in I\}}{s \xrightarrow{a} t} \qquad\qquad \frac{\{x_i \xrightarrow{a_i} y_i \mid i \in I\}}{f(x_1, \ldots, x_n) \xrightarrow{a} D[y_1, \ldots, y_n]} \ \text{(b)}$$

Figure 2. SOS schema (a), and De Simone format (b).

Definition 2. *Let $\mathcal{R} = (\mathcal{H}, \mathcal{V}, N, R)$ be a tile system. A symmetric relation $\sim_t$ on configurations is called* tile bisimulation *if whenever $s \sim_t t$ and $\mathcal{R} \vdash s \xrightarrow[b]{a} s'$, then there exists t' such that $\mathcal{R} \vdash t \xrightarrow[b]{a} t'$ and $s' \sim_t t'$. The maximal tile bisimulation is called* tile bisimilarity *and denoted by $\simeq_t$.*

An interesting question concerns suitable conditions under which tile bisimilarity yields a congruence (w.r.t. the operations of the underlying horizontal structure); two main properties have been investigated: *basic source* and *tile decomposition* [11]. The former is strongly related to tile formats and will be discussed later. Tile decomposition has a completely abstract formulation that applies to all tile systems, and it will be very useful in our congruence proofs.

Definition 3. *A tile system $\mathcal{R} = (\mathcal{H}, \mathcal{V}, N, R)$ enjoys the* decomposition property *if for all arrows $s \in \mathcal{H}$ and for all sequents $s \xrightarrow[b]{a} t$ entailed by $\mathcal{R}$, then: (1) if $s = s_1; s_2$ then $\exists c \in \mathcal{V}$, $t_1, t_2 \in \mathcal{H}$ such that $\mathcal{R} \vdash s_1 \xrightarrow[c]{a} t_1$, $\mathcal{R} \vdash s_2 \xrightarrow[b]{c} t_2$ and $t = t_1; t_2$; (2) if $s = s_1 \otimes s_2$ then $\exists a_1, a_2, b_1, b_2 \in \mathcal{V}$, $t_1, t_2 \in \mathcal{H}$ such that $\mathcal{R} \vdash s_1 \xrightarrow[b_1]{a_1} t_1$, $\mathcal{R} \vdash s_2 \xrightarrow[b_2]{a_2} t_2$, $a = a_1 \otimes a_2$, $b = b_1 \otimes b_2$ and $t = t_1 \otimes t_2$.*

This property characterizes compositionality: It amounts to saying that if a system s can undergo a transition α, then for every subsystem s_1 of s there exists some transition α', such that α can be obtained by composing α' with a transition of the rest.

Proposition 1 (cf. [11]). *If $\mathcal{R}$ enjoys decomposition, then $\simeq_t$ is a congruence.*

2 Relating SOS and Tile Formats

SOS formats. LTSs over $\mathbb{T}_\Sigma$ and Λ can be more conveniently specified via a collection of inductive proof rules, called *transition system specification* (TSS). Such rules have the form in Figure 2(a), where the s_i, t_i, s and t are in $\mathbb{T}_\Sigma(X)$ and the a_i, a are in Λ. The generated LTS has set of states $\mathbb{T}_\Sigma$ and all transitions $s \xrightarrow{a} t$ that can be proved by using the TSS rules. In the SOS approach [18] the behavior of s is defined in terms of the behaviors of its subterms, but in general this is not enough to guarantee that bisimilarity is a congruence. To ensure this important property, suitable rule formats have been defined. For example, in the De Simone format (DSF) [8] rules have the form in Figure 2(b), where $f \in \Sigma_n$, $I \subseteq \{1, \ldots, n\}$, the context D is linear and the variables x_i and y_i are distinct, except for $y_i = x_i$ if $i \notin I$. Hence each x_i can be used at most once in the premises, and if used cannot appear in D. The (positive) GSOS format [1] extends the DSF in several ways (e.g., the same x_i can appear more than once in the premises and also in the D, which can be nonlinear). The `tyft/tyxt` format [12] generalizes GSOS by allowing generic terms t_i as sources of the premises. However, in all such formats, the main requirement is that rule conclusions must have the form $f(x_1, \ldots, x_n) \xrightarrow{a} t$, i.e., their sources must consist of a single $f \in \Sigma_n$ applied to n different variables, a crucial fact in the proof that bisimilarity is a congruence.

Tile formats. Since we are particularly interested in considering tile systems where configurations and observations are *freely generated* by the horizontal signature Σ and by (the signature associated to) the set of labels Λ, in what follows we shall present tile systems as tuples of the form $\mathcal{R} = (\Sigma, \Lambda, N, R)$. In particular, we employ categories of substitutions on Σ and Λ. In fact, substitutions and their composition $\_;\_$ form a cartesian category $\mathbf{Subs}_\Sigma$, with linear substitutions forming a monoidal subcategory. An alternative presentation of $\mathbf{Subs}_\Sigma$ can be obtained resorting to *algebraic theories* [13]. The free algebraic theory associated to a signature Σ is the category $\mathbf{Th}[\Sigma]$: Its objects are 'underlined' natural numbers, the arrows from $\underline{m}$ to $\underline{n}$ are n-tuples of terms in the free Σ-algebra with (at most) m variables, and composition of arrows is term substitution. The arrows of $\mathbf{Th}[\Sigma]$ are generated from Σ by the inference rules in Figure 3, modulo the axioms in Table 1. It is folklore that $\mathbf{Th}[\Sigma]$ is isomorphic to $\mathbf{Subs}_\Sigma$, and the arrows from $\underline{0}$ to $\underline{1}$ are in bijective correspondence with the closed terms over Σ.

An object $\underline{n}$ (interface) can be thought of as representing the n (ordered) variables $x_1, \ldots, x_n$. This allows us to denote $[t_1/x_1, \ldots, t_n/x_n]$ just by the tuple $\langle t_1, \ldots, t_n \rangle$, since a standard naming of substituted variables is assumed. We omit angle brackets if no confusion can arise. The rule `op` defines basic substitutions $[f(x_1, \ldots, x_n)/x_1] = f(x_1, \ldots, x_n)$ for all $f \in \Sigma_n$. The rule `id` yields identity substitutions $\langle x_1, \ldots, x_n \rangle$. The rule `seq` represents application of α to β. The rule `mon` composes substitutions in parallel (in $\alpha \otimes \beta$, α goes from $x_1, \ldots, x_n$ to $x_1, \ldots, x_m$, while β goes from $x_{n+1}, \ldots, x_{n+k}$ to $x_{m+1}, \ldots, x_{m+l}$). Three 'auxiliary' operators (i.e., not dependent on Σ) are introduced that recover the cartesian structure (rules `sym`, `dup` and `dis`). The *symmetry* $\gamma_{\underline{n},\underline{m}}$ is the permutation $\langle x_{n+1}, \ldots, x_{n+m}, x_1, \ldots, x_n \rangle$. The *duplicator* $\nabla_{\underline{n}} = \langle x_1, \ldots, x_n, x_1, \ldots, x_n \rangle$ introduces sharing and hence nonlinear substitutions. The *discharger* $!_{\underline{n}}$ is the empty substitution on $x_1, \ldots, x_n$, recovering cartesian projections. Due to space limitations we cannot detail the axioms in Table 1 (consult, e.g., [2,4]).

Monoidal theories relate to algebraic theories as linear substitutions relate to generic substitutions. This originates a subcategory $\mathbf{M}[\Sigma]$ of $\mathbf{Th}[\Sigma]$ whose arrows are those generated by rules `op`, `id`, `seq`, and `mon` in Figure 3 modulo the axioms of category and tensor product (first and second rows in Table 1).

$$\texttt{op}\,\frac{f \in \Sigma_n}{f\colon \underline{n} \to \underline{1}} \qquad \texttt{id}\,\frac{n \in \mathbb{N}}{id_{\underline{n}}\colon \underline{n} \to \underline{n}} \qquad \texttt{seq}\,\frac{\alpha\colon \underline{n} \to \underline{m} \quad \beta\colon \underline{m} \to \underline{k}}{\alpha;\beta\colon \underline{n} \to \underline{k}} \qquad \texttt{mon}\,\frac{\alpha\colon \underline{n} \to \underline{m} \quad \beta\colon \underline{k} \to \underline{l}}{\alpha \otimes \beta\colon \underline{n+k} \to \underline{m+l}}$$

$$\texttt{sym}\,\frac{n, m \in \mathbb{N}}{\gamma_{\underline{n},\underline{m}}\colon \underline{n+m} \to \underline{m+n}} \qquad \texttt{dup}\,\frac{n \in \mathbb{N}}{\nabla_{\underline{n}}\colon \underline{n} \to \underline{n+n}} \qquad \texttt{dis}\,\frac{n \in \mathbb{N}}{!_{\underline{n}}\colon \underline{n} \to \underline{0}}$$

Figure 3. The inference rules for the generation of $\mathbf{Th}[\Sigma]$.

category	$\alpha;(\beta;\delta) = (\alpha;\beta);\delta$		$\alpha; id_{\underline{m}} = \alpha = id_{\underline{n}};\alpha$
tensor	$(\alpha;\alpha') \otimes (\beta;\beta') = (\alpha \otimes \beta);(\alpha' \otimes \beta')$		$id_{\underline{n+m}} = id_{\underline{n}} \otimes id_{\underline{m}}$
product	$\alpha \otimes (\beta \otimes \delta) = (\alpha \otimes \beta) \otimes \delta$		$\alpha \otimes id_{\underline{0}} = \alpha = id_{\underline{0}} \otimes \alpha$
symmetries	$\gamma_{\underline{n},\underline{m+k}} = (\gamma_{\underline{n},\underline{m}} \otimes id_{\underline{k}});(id_{\underline{m}} \otimes \gamma_{\underline{n},\underline{k}})$	$\gamma_{\underline{n},\underline{0}} = id_{\underline{n}}$	$\gamma_{\underline{n},\underline{m}};\gamma_{\underline{m},\underline{n}} = id_{\underline{n+m}}$
duplicators	$\nabla_{\underline{n+m}} = (\nabla_{\underline{n}} \otimes \nabla_{\underline{m}});(id_{\underline{n}} \otimes \gamma_{\underline{n},\underline{m}} \otimes id_{\underline{m}})$		$\nabla_{\underline{0}} = id_{\underline{0}}$
	$\nabla_{\underline{n}};(id_{\underline{n}} \otimes \nabla_{\underline{n}}) = \nabla_{\underline{n}};(\nabla_{\underline{n}} \otimes id_{\underline{n}})$		$\nabla_{\underline{n}};\gamma_{\underline{n},\underline{n}} = \nabla_{\underline{n}}$
discharger	$!_{\underline{n+m}} = !_{\underline{n}} \otimes !_{\underline{m}}$	$!_{\underline{0}} = id_{\underline{0}}$	$\nabla_{\underline{n}};(id_{\underline{n}} \otimes !_{\underline{n}}) = id_{\underline{n}}$
naturality	$(\alpha \otimes \beta);\gamma_{\underline{m},\underline{l}} = \gamma_{\underline{n},\underline{k}};(\beta \otimes \alpha)$	$\alpha;\nabla_{\underline{m}} = \nabla_{\underline{n}};(\alpha \otimes \alpha)$	$\alpha;!_{\underline{m}} = !_{\underline{n}}$

Table 1. Axiomatization of $\mathbf{Th}[\Sigma]$.

Term graphs [9] are in some sense situated between linear and cartesian terms, because they allow for explicit sharing and discharging, in such a way that this information is preserved by composition. In fact, it has been shown in [6] that by abandoning the naturality of ∇ and !, one obtains a *gs-monoidal* category $\mathbf{GS}[\Sigma]$ which is isomorphic to the category of (ranked) term graphs on Σ. For example, in $\mathbf{GS}[\Sigma]$ the composition $[t_1/x_1];C[x_1]$ can be written as **let** $x_1 = t_1$ **in** $C[x_1]$, with the convention that it evaluates to $C[t_1]$ when x_1 occurs exactly once in C.

The tile format proposed in the original presentation of tiles [11] is the so-called *algebraic tile format* (ATF) that recollects the perspective of most TSS: configurations are terms, and observations are the arrows of the monoidal category freely generated by labels (regarded as unary operators). Auxiliary tiles lift the horizontal cartesian structure to the horizontal composition of tiles. In the ATF basic tiles have the form in Figure 4(a), where the a_i and a can be either labels (viewed as arrows from $\underline{1}$ to $\underline{1}$) or identities and $s, t \in \mathbb{T}_\Sigma(\{x_1, \ldots, x_n\})$. The ATF corresponds to SOS rules as in Figure 4(b), where $I \subseteq \{1, \ldots, n\}$, C and D are contexts (that correspond to s and t in the tile), and all the y_i and x_i are different if $i \in I$, but $y_k = x_k$ whenever $k \notin I$. The correspondence follows since for all $s, t \in \mathbb{T}_\Sigma$ and for all $a \in \Lambda$, $\mathcal{R} \vdash s \xrightarrow[a]{id_{\underline{0}}} t$ holds if and only if $s \xrightarrow{a} t$ in the LTS associated to the SOS specification. Typical auxiliary tiles for the ATF are those in Figure 5: ∇_a duplicates the observation a (trigger of the tile) propagating it to two instances of the unique variable in the initial interface, while $\gamma_{a,b}$ swaps the subcomponents in the initial interface, together with their observations, and $!_a$ discharges the initial interface and its move a. We refer to [11] for more details.

In the ATF, $\mathcal{H}$ is cartesian, whereas $\mathcal{V}$ is only monoidal. We will show in Section 4 that this combination can compromise the tile bisimilarity as congru-

$$\text{(a)}\quad \begin{tikzcd} \underline{n} \arrow[r, "s"] \arrow[d, "a_1\otimes\cdots\otimes a_n"'] & \underline{1} \arrow[d, "a"] \\ \underline{n} \arrow[r, "t"'] & \underline{1} \end{tikzcd} \qquad\qquad \frac{\{x_i \xrightarrow{a_i} y_i \mid i \in I\}}{C[x_1,\ldots,x_n] \xrightarrow{a} D[y_1,\ldots,y_n]}\quad \text{(b)}$$

Figure 4. A generic ATF tile (a) and its SOS counterpart (b).

$$\begin{tikzcd} \underline{1} \arrow[r, "\langle x_1,x_1\rangle"] \arrow[d, "a"'] \arrow[dr, phantom, "\nabla_a"] & \underline{2} \arrow[d, "a\otimes a"] \\ \underline{1} \arrow[r, "\langle x_1,x_1\rangle"'] & \underline{2} \end{tikzcd} \qquad \begin{tikzcd} \underline{2} \arrow[r, "\langle x_2,x_1\rangle"] \arrow[d, "a\otimes b"'] \arrow[dr, phantom, "\gamma_{a,b}"] & \underline{2} \arrow[d, "b\otimes a"] \\ \underline{2} \arrow[r, "\langle x_2,x_1\rangle"'] & \underline{2} \end{tikzcd} \qquad \begin{tikzcd} \underline{1} \arrow[r, "!_{\underline{1}}"] \arrow[d, "a"'] \arrow[dr, phantom, "!_a"] & \underline{0} \arrow[d, "id_{\underline{0}}"] \\ \underline{1} \arrow[r, "!_{\underline{1}}"'] & \underline{0} \end{tikzcd}$$

Figure 5. Auxiliary tiles for ATF.

ence property. In particular, it will be convenient either to consider the *term tile format* (TTF) [4], where also $\mathcal{V}$ is cartesian, or renounce the horizontal cartesian structure. This latter option can be achieved in (at least) two ways: either using the *monoidal tile format* (MTF) [16] where also $\mathcal{H}$ is monoidal, or resorting to the *gs-monoidal tile format* (GSTF), where a faithful account of subterm sharing is provided. The MTF deals only with linear terms and has no auxiliary tiles. Its basic tiles are similar to those of the ATF (but configurations must be linear). Though the auxiliary tiles of the GSTF are the same as those of the ATF, the former deals with term graphs rather than terms and thus the naturality axioms of ∇ and ! are not valid. Basic TTF tiles have the form:

$$\begin{tikzcd} \underline{n} \arrow[r, "s"] \arrow[d, "v"'] & \underline{m} \arrow[d, "u"] \\ \underline{k} \arrow[r, "t"'] & \underline{1} \end{tikzcd}$$

with $s, t \in \mathbf{Th}[\Sigma]$ and $v, u \in \mathbf{Th}[\Lambda]$. If labels in Λ are regarded as unary operators, then $m = 1$. We present term tiles more concisely as sequents $n \triangleleft\ s \xrightarrow[u]{v} t$, where the number of variables in the 'upper-left' interface is explicit. Auxiliary term tiles consist of all tiles that perform consistent rearrangements in the two dimensions, i.e., tiles $n \triangleleft\ s \xrightarrow[u]{v} t$ such that $s; u = v; t$ with either (1) s, t, u and v are terms over the empty signature $\varnothing$ (hence $s, t, v, u \in \mathbf{Th}[\Sigma] \cap \mathbf{Th}[\Lambda]$); or (2) $s, t \in \mathbf{Th}[\Sigma]$ and $u, v \in \mathbf{Th}[\varnothing]$; or (3) $u, v \in \mathbf{Th}[\Lambda]$ and $s, t \in \mathbf{Th}[\varnothing]$. Typical auxiliary term tiles are $\pi_{\underline{1}} = 1 \triangleleft\ x_1 \xrightarrow[x_1,x_1]{x_1} x_1, x_1$ and $\tau_{\underline{1}} = 1 \triangleleft\ x_1, x_1 \xrightarrow[x_1,x_2]{x_1,x_1} x_1, x_2$ that duplicate the unary interface, $\sigma_{\underline{1},\underline{1}} = 2 \triangleleft\ x_2, x_1 \xrightarrow[x_1,x_2]{x_2,x_1} x_1, x_2$ that swaps the two components of the initial interface, and $\nabla_f = 2 \triangleleft\ f(x_1, x_2) \xrightarrow[\nabla_{\underline{1}}]{\nabla_{\underline{2}}} \langle f(x_1, x_2), f(x_3, x_4)\rangle$ that vertically duplicates the configuration $f(x_1, x_2)$ (see [4,2] for more details). Table 2 summarizes the differences between the tile formats we consider.

	$\mathcal{H}$	$\mathcal{V}$	auxiliary tiles
monoidal tile format	$\mathbf{M}[\Sigma]$	$\mathbf{M}[\Lambda]$	none
gs-monoidal tile format	$\mathbf{GS}[\Sigma]$	$\mathbf{M}[\Lambda]$	$\gamma_{a,b}$, ∇_a, $!_a$
algebraic format	$\mathbf{Th}[\Sigma]$	$\mathbf{M}[\Lambda]$	$\gamma_{a,b}$, ∇_a, $!_a$
term tile format	$\mathbf{Th}[\Sigma]$	$\mathbf{Th}[\Lambda]$	$\gamma_{a,b}$, ∇_a, $!_a$, $\gamma_{s,t}$, ∇_t, $!_t$, $\sigma_{\underline{1},\underline{1}}$, $\tau_{\underline{1}}$, $\pi_{\underline{1}}$,...

Table 2. Features of MTF, GSTF, ATF and TTF.

Our aim is to find syntactic constraints on basic tiles that enforce the 'tile bisimilarity as congruence' property. One such constraint, already noted in [11], is called *basic source* (it is reminiscent of analogous restrictions required by most well-behaved SOS formats): A tile system enjoys the basic source property if the initial configuration of each basic tile consists of a single operator $f\colon \underline{ar(f)} \to \underline{1}$.

Proposition 2 (cf. [11]). *If an algebraic tile system $\mathcal{R}$ enjoys the basic source property, then tile bisimilarity on closed terms is a congruence.*

2.1 Tiles and CCS

In this section, we recall the operational semantics of Milner's *calculus for communicating systems* (CCS) [17] and the tile systems proposed in the literature for recovering the behavior of (finite) agents. We let Δ (ranged over by α) be the set of *basic actions*, and $\bar{\Delta}$ the set of *complementary actions* (with $\bar{(\_)}$ an involutive function such that $\Delta = \bar{\bar{\Delta}}$ and $\Delta \cap \bar{\Delta} = \varnothing$). We denote $\Delta \cup \bar{\Delta}$ by Λ (ranged over by λ), let $\tau \notin \Lambda$ be a *distinguished* action, and let $Act = \Lambda \cup \{\tau\}$ (ranged over by μ) be the set of CCS actions. Then, CCS *processes* (also *agents*) are generated by the grammar: $P ::= nil \mid \mu.P \mid P\backslash_\alpha \mid P + P \mid P|P$.

Assuming the reader familiar with the notation, we just recall that the operations above correspond to the *inactive* process, *action prefix*, *restriction*, *nondeterministic sum* and *parallel composition*. We let P, Q, R, ... range over the set **Proc** of CCS processes. The SOS system for CCS processes is defined by the set of De Simone rules in Figure 6 (the obvious symmetric rules for sum and asynchronous communication are omitted).

The basic ATF tiles for CCS proposed in [11] are given in Figure 7. The basic MTF tiles for CCS proposed in [16] differ from those in the ATF only in the treatment of nondeterministic sum. This is because the MTF has no horizontal discharger and therefore an explicit unary operator $!(\_)$ must be introduced for garbaging discarded arguments. Thus, the MTF tile for (left choice in) nondeterministic sum is $\langle +_\mu : x_1 + x_2 \xrightarrow[\mu]{\mu \otimes id} x_1 \otimes !(x_2)$. There are no tiles with source $!(x_1)$ and therefore the operator $!(\_)$ locks its argument. The basic GSTF tiles for CCS have not been presented in the literature, but they are essentially the same as those of the MTF case, where the operator $!(\_)$ is the discharger $!_{\underline{1}}$. The basic TTF tiles for CCS proposed in [4] are given in Figure 8. Apart from the notation, the main difference w.r.t. the ATF tiles resides again in the rule for sum (the unused argument can be discarded at the level of observations).

$$\frac{}{\mu.P \xrightarrow{\mu} P} \qquad \frac{P \xrightarrow{\mu} Q}{P\backslash_\alpha \xrightarrow{\mu} Q\backslash_\alpha} \text{ (if } \mu \notin \{\alpha, \bar{\alpha}\})$$

$$\frac{P \xrightarrow{\mu} Q}{P + R \xrightarrow{\mu} Q} \qquad \frac{P \xrightarrow{\mu} Q}{P|R \xrightarrow{\mu} Q|R} \qquad \frac{P \xrightarrow{\lambda} Q,\ P' \xrightarrow{\bar{\lambda}} Q'}{P|P' \xrightarrow{\tau} Q|Q'}$$

Figure 6. De Simone rules for finite CCS.

$$\mathtt{act}_\mu : \mu.x_1 \xrightarrow[\mu]{id} x_1 \qquad \mathtt{res}_{\mu,\alpha} : x_1\backslash_\alpha \xrightarrow[\mu]{\mu} x_1\backslash_\alpha \text{ (if } \mu \notin \{\alpha, \bar{\alpha}\})$$

$$\langle +_\mu : x_1 + x_2 \xrightarrow[\mu]{\mu \otimes id} x_1 \qquad \rfloor_\mu : x_1|x_2 \xrightarrow[\mu]{\mu \otimes id} x_1|x_2 \qquad \|_\lambda : x_1|x_2 \xrightarrow[\tau]{\lambda \otimes \bar{\lambda}} x_1|x_2$$

Figure 7. ATF tiles for finite CCS.

$$\mathtt{act}_\mu : 1 \lhd\ \mu.x_1 \xrightarrow[\mu(x_1)]{x_1} x_1 \qquad \mathtt{res}_{\mu,\alpha} : 1 \lhd\ x_1\backslash_\alpha \xrightarrow[\mu(x_1)]{\mu(x_1)} x_1\backslash_\alpha \text{ (if } \mu \notin \{\alpha, \bar{\alpha}\})$$

$$\langle +_\mu : 2 \lhd\ x_1 + x_2 \xrightarrow[\mu(x_1)]{\mu(x_1)} x_1 \qquad \rfloor_\mu : 2 \lhd\ x_1|x_2 \xrightarrow[\mu(x_1)]{\mu(x_1), x_2} x_1|x_2 \qquad \|_\lambda : 2 \lhd\ x_1|x_2 \xrightarrow[\tau(x_1)]{\lambda(x_1), \bar{\lambda}(x_2)} x_1|x_2$$

Figure 8. TTF tiles for finite CCS.

3 The Monoidal Tile Format

If the MTF is employed, the basic source property establishes a bijective correspondence between basic tiles and rules in DSF (as noted in [11]). Hence it is easy to show that tile bisimilarity on closed terms coincides with bisimilarity on the LTS generated by the TTS associated to the tile system. However, we can extend the congruence to contexts in a different way from instantiation closure.

Theorem 1. *If a monoidal tile system $\mathcal{R} = (\Sigma, \Lambda, N, R)$ enjoys the basic source property, then tile bisimilarity defines a congruence also on contexts.*

Of course a congruence on contexts is closed under instantiations, and therefore $\simeq_t \subseteq \simeq$. It can be shown that $\simeq_t$ is in general finer than $\simeq$ on contexts. We propose the following example, inspired by [19].

Example 1. Let us consider finite CCS (see Section 2.1) extended with the family of unary operators $do^n_\mu(\_)$ with $n \in \mathbb{N}$ and μ an action. For all $n > 0$, the behaviour of $do^n_\mu(\_)$ is described by the DSF rule in Figure 9(a), whose corresponding MTF tile is $do^n_\mu(x_1) \xrightarrow[\mu]{\mu} do^{n-1}_\mu(x_1)$. There are no rules for $do^0_\mu(\_)$. Then the contexts $C_1[x_1] = \beta.do^1_\alpha(x_1) + \beta.\alpha.nil + \beta.nil$ and $C_2[x_1] = \beta.\alpha.nil + \beta.nil$ are bisimilar but not tile bisimilar. In fact, if p is a process, then $C_1[p] \xrightarrow{\beta} do^1_\alpha(p)$, and $do^1_\alpha(p)$ has either no transition (when p cannot do α) or a transition leading to the deadlocked state $do^0_\alpha(q)$ for some q with $p \xrightarrow{\alpha} q$. Thus $C_2[p] = \beta.\alpha.nil + \beta.nil$

$$\text{(a)}\quad \frac{P \xrightarrow{\mu} Q}{do^n_\mu(P) \xrightarrow{\mu} do^{n-1}_\mu(Q)}$$

$$\begin{tikzcd}
\underline{1} \arrow[r, "\nabla_{\underline{1}}"] \arrow[d, "\nabla_{\underline{1}}"'] \arrow[dr, phantom, "\tau_{\underline{1}}"] & \underline{2} \arrow[r, "(x_1|x_2)\backslash_\alpha"] \arrow[d, dashed, "id"] & \underline{1} \arrow[d, "id"] \\
\underline{2} \arrow[r, "id"] \arrow[d, "{\alpha(x_1),\bar{\alpha}(x_2)}"'] & \underline{2} \arrow[r, "(x_1|x_2)\backslash_\alpha"] \arrow[d, dashed, "{\alpha(x_1),\bar{\alpha}(x_2)}"] & \underline{1} \arrow[d, "\tau"] \\
\underline{2} \arrow[r, "id"] & \underline{2} \arrow[r, "(x_1|x_2)\backslash_\alpha"] & \underline{1}
\end{tikzcd}\quad \text{(b)}$$

Figure 9. DSF rule for do^n_μ (a) and the cell pasting of Example 2 (b).

can always match the β move, i.e., $C_1[p] \simeq C_2[p]$ for all closed p. On the other hand, the tile $C_1[x_1] \xrightarrow[\beta]{id} do^1_\alpha(x_1)$ cannot be bisimulated by $C_2[x_1]$. In fact, if $C_2[x_1] \xrightarrow[\beta]{id} \alpha.nil$, then $\alpha.nil \xrightarrow[\alpha]{id} nil$ cannot be matched by $do^1_\alpha(x_1)$. If instead $C_2[x_1] \xrightarrow[\beta]{id} nil$ then $do^1_\alpha(x_1) \not\simeq_{\mathrm{t}} nil$. Note that we have slightly abused the notation by writing, e.g., nil instead of $nil[x_1] = nil \otimes !(x_1)$ as required by MTF.

4 Algebraic vs. Term Tile Formats

When dealing with nonlinear contexts and tile bisimulation, the ATF can give rise to unexpected results, as the following example illustrates.

Example 2. Let us consider the ATF tiles for finite CCS in Section 2.1. It is straightforward that on closed terms tile bisimilarity $\simeq_{\mathrm{t}}$ coincides with bisimilarity $\simeq$ on the ordinary LTS. Moreover, we know that $\simeq$ is a congruence. However, on open terms $\simeq_{\mathrm{t}}$ is different from $\simeq$. In fact, let us consider the contexts $C_1[x_1] = (x_1|x_1)\backslash_\alpha$ and $C_2[x_1] = (x_1\backslash_\alpha | x_1\backslash_\alpha)$. Then $C_1[x_1] \simeq_{\mathrm{t}} C_2[x_1]$, but $C_1[x_1] \not\simeq C_2[x_1]$, since there exists a process p (e.g., $p = \alpha.nil + \bar{\alpha}.nil$) such that $C_1[p] \xrightarrow{\tau} Q$ and $C_2[p]$ cannot move. But then, of course, $C_1[p] \not\simeq_{\mathrm{t}} C_2[p]$ and therefore $\simeq_{\mathrm{t}}$ is not a congruence on contexts. Finite CCS admits also a presentation in TTF (also recalled in Section 2.1). Again we have that, on closed terms, $\simeq_{\mathrm{t}}$ and $\simeq$ coincide. However, thanks to the auxiliary TTF tiles, $C_1[x_1] \not\simeq_{\mathrm{t}} C_2[x_1]$, because the tile $1 \triangleleft C_1[x_1] \xrightarrow[\tau(x_1)]{\alpha(x_1),\bar{\alpha}(x_1)} (x_1|x_2)\backslash_\alpha$ (see Figure 9(b), noting that $C_1[x_1] = \nabla_{\underline{1}}; (x_1|x_2)\backslash_\alpha$) cannot be mimicked by $C_2[x_1]$.

A more general result demonstrates that the fact that the TTF for CCS behaves better than its ATF counterpart is not a mere coincidence.

Theorem 2. *If a term tile system $\mathcal{R} = (\Sigma, \Lambda, N, R)$ enjoys the basic source property, then tile bisimilarity is a congruence (also for open terms).*

The previous theorem states that if a system can be expressed in TTF with basic source, then we have a more satisfactory equivalence on open terms than the one obtained by closing contexts under all possible instantiations. In fact it is completely analogous and consistent with that of closed terms and takes

$$
\text{(a)}\quad
\begin{tikzcd}
\underline{0} \arrow[r, "p"] \arrow[d, "id_{\underline{0}}"'] & \underline{1} \arrow[r, "\nabla_{\underline{1}}"] \arrow[d, "\alpha"] \arrow[r, phantom, shift right=4ex, "\nabla_\alpha"] & \underline{2} \arrow[r, "(x_1|x_2)\backslash_\alpha"] \arrow[d, "\alpha\otimes\alpha"] & \underline{1} \arrow[d, dashed] \\
\underline{0} \arrow[r, "nil"'] & \underline{1} \arrow[r, "\nabla_{\underline{1}}"'] & \underline{2} \arrow[r, dashed] & ?
\end{tikzcd}
\qquad
\begin{tikzcd}
\underline{0} \arrow[r, "p\otimes p"] \arrow[d, "id_{\underline{0}}"'] & \underline{2} \arrow[r, "(x_1|x_2)\backslash_\alpha"] \arrow[d, "\alpha\otimes\bar{\alpha}"] & \underline{2} \arrow[d, "\tau"] \\
\underline{0} \arrow[r, "nil\otimes nil"'] & \underline{2} \arrow[r, "(x_1|x_2)\backslash_\alpha"'] & \underline{1}
\end{tikzcd}
\quad\text{(b)}
$$

Figure 10. Tile pastings described in Example 3.

care of the specification constraints (initial configurations can be seen, e.g., as incomplete open software modules) rather than just of the effective realizations (closed systems). As for MTF, tile bisimilarity for TTF is in general finer than $\simeq$.

5 The Gs-Monoidal Format for Explicit Sharing

The definition of good SOS formats for process algebras based on term graphs rather than terms is not straightforward, since the interplay between configurations and observations is difficult to manage. Tiles can be used to overcome this inconvenience. The idea is to consider a monoidal category of observations and a gs-monoidal category of configurations. Though the auxiliary and basic tiles of GSTF are much like those of ATF, this time the gs-monoidal (not the cartesian) structure is lifted from configurations to (horizontal composition of) tiles.

Example 3. The term graph representation **let** $x_1 = \alpha.nil + \bar{\alpha}.nil$ **in** $(x_1|x_1)\backslash_\alpha$ of the CCS agent $((\alpha.nil+\bar{\alpha}.nil)|(\alpha.nil+\bar{\alpha}.nil))\backslash_\alpha$ can only evolve by synchronizing the moves that $p = \alpha.nil + \bar{\alpha}.nil$ can perform with the open behavior of the nonlinear context $(x_1|x_1)\backslash_\alpha = \nabla_{\underline{1}};(x_1|x_2)\backslash_\alpha$. Since p can execute either α or $\bar{\alpha}$ but not both at the same time, then no such synchronization is possible (see the incomplete tile pasting in Figure 10(a)). This is not the case of $(p|p)\backslash_\alpha$, where there are two subcomponents p that can perform complementary moves and synchronize (see Figure 10(b), noticing that $p;\nabla_{\underline{1}} \neq \nabla_{\underline{0}};(p\otimes p) = p\otimes p$).

Likewise MTF and TTF, also tile bisimilarity for GSTF enjoys a nice congruence property, providing a good format for the specification of resource aware systems.

Theorem 3. *If a gs-monoidal tile system $\mathcal{R} = (\Sigma, \Lambda, N, R)$ enjoys the basic source property, then tile bisimilarity is a congruence (also for open terms).*

6 Related Work: Conditional Transition Systems

The basic ingredients of Rensink's *conditional transition systems* (CTSs) are *conditional transitions* $\Gamma \vdash s \xrightarrow{a} t$, where s and t are open terms, and the *environment* Γ is a finite graph $\{x_1 \xrightarrow{a_1} y_1, \ldots, x_n \xrightarrow{a_n} y_n\}$ that provides suitable assumptions on the arguments of s and t for the open transition to exist. Building on this, Rensink proposes two kinds of bisimilarities called *formal hypothesis* ($\simeq^{\mathrm{fh}}$) and *hypothesis preserving* ($\simeq^{\mathrm{hp}}$). The difference between them is that in the latter the Γ are persistent, and thus can be reused during bisimulation.

Although one could expect that conditional transitions correspond to tiles $s \xrightarrow[a]{\Gamma} t$ for Γ belonging to a suitable category of observations, a comparison with the tile model is not straightforward. One important difference is that in Rensink's approach the environment Γ is observed, which provides all the *potential* triggers of the systems, while in the tile approach the observation of a particular step includes the actual trigger. Moreover, a different operation closure is introduced by Rensink via *conditional transition system specifications* (CTSSs) that come equipped with three administrative rules (called *variable*, *weakening* and *substitution*) that operate on the basic conditional transitions of any specification. The main theorem of [19] states that for (recursion based) CTSSs both $\simeq^{\mathrm{fh}}$ and $\simeq^{\mathrm{hp}}$ are congruences and $\simeq^{\mathrm{fh}} \subset \simeq^{\mathrm{hp}} \subset \simeq$. Rensink also conjectured that in the ATF setting $\simeq_{\mathrm{t}} = \simeq^{\mathrm{fh}} = \simeq^{\mathrm{hp}}$, probably inspired by the fact that tile triggers define acyclic environments, where the assumptions made in the past cannot influence successive steps. The fact that for the ATF the basic source property does not imply that $\simeq_{\mathrm{t}}$ is a congruence gives evidence that the models built via CTSS and tiles can be very different, and indeed a formal correspondence between administrative rules and tile operations is hard to state — informally, variable, substitution and weakening rules correspond respectively to horizontal identities, horizontal composition and parallel composition. Nonlinear terms also introduce tiles such as $x_1|x_1 \xrightarrow[\alpha;\alpha]{\alpha} x_1|x_1$ (it appears in all the tile systems for CCS that we have considered, except MTF); this corresponds to using the same assumption to prove 'atomically' two consecutive steps, which can have some interpretation for $\simeq^{\mathrm{hp}}$, but not for $\simeq^{\mathrm{fh}}$.

Concluding Remarks and Future Work

We have proposed several tile formats for defining bisimilarity congruences directly on both closed and open terms. The simpler MTF is designed for linear systems. The more expressive TTF reflects nonlinearity of configurations at the level of observations. The GSTF provides a sound formal framework for the treatment of resource aware systems, previously missing in the literature. In many such cases the congruence proofs (via the decomposition property) can be carried out at the pictorial level as tile pastings (see details in the technical report [3]).

Though at a first look open systems seem just the natural extensions of closed systems, we have noted an initial classification that would distinguish between incomplete systems, which define the behavior (at the top level) of the system to be refined by providing the corresponding components, and coordinators, which define services to be used by the processes instantiating their free arguments. Thus, the latter class corresponds to bottom-up design and the most usual notion of reusable component. As discussed in the Introduction, it is reasonable to cope with variables in these two classes of open systems in a different way, specially when their interfaces contain repeated occurrences of the same variables.

Another interesting point is the duality between the extension of a process by (partial) instantiation of its variables, and by embedding it in a context (seen as an incomplete system by itself). In [5], the notion of *ground* tile bisimulation has been developed, which thanks to additional transitions labelled by contexts (in the style of dynamic bisimilarity) is a congruence, practically without any format limitation on basic tiles. It seems rather interesting to try to unify the advances in both directions to obtain a uniform treatment of both internal and external contexts. Finally, an interesting open problem is the extension of our results to the adequate notion of weak tile bisimulation.

Acknowledgements. We thank Arend Rensink for some preliminary discussions on the relationship between conditional transition systems and tiles. We also thank the anonymous referees for their helpful comments.

References

1. B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, 1995.
2. R. Bruni, Tile Logic for Synchronized Rewriting of Concurrent Systems, Ph.D. Thesis TD-1/99, Computer Science Department, University of Pisa, 1999.
3. R. Bruni, D. de Frutos-Escrig, N. Martí-Oliet, and U. Montanari. Tile bisimilarity congruences for open terms and term graphs. Technical Report TR-00-06, Computer Science Department, University of Pisa, 2000.
4. R. Bruni, J. Meseguer, and U. Montanari. Process and term tile logic. Technical Report SRI-CSL-98-06, SRI International, 1998.
5. R. Bruni, U. Montanari, and V. Sassone. Open ended systems, dynamic bisimulation and tile logic. In *Proc. IFIP-TCS 2000, LNCS.* Springer, 2000. To appear.
6. A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via gs-monoidal categories. *Applied Categorical Structures*, 7(4):299–331, 1999.
7. A. Corradini and U. Montanari. An algebraic semantics for structured transition systems and its application to logic programs. *Th. Comput. Sci.*, 103:51–106, 1992.
8. R. De Simone. Higher level synchronizing devices in MEIJE–SCCS. *Theoret. Comput. Sci.*, 37:245–267, 1985.
9. M.C. van Eekele, M.J. Plasmeijer, and M.R. Sleep, editors. *Term Graph Rewriting: Theory and Practice*, Wiley, London, 1993.
10. F. Gadducci and U. Montanari. Rewriting rules and CCS. in *Proceeding WRLA'96*, vol. 4 of *Elect. Notes in Th. Comput. Sci.*, Elsevier Science, 1996.
11. F. Gadducci and U. Montanari. The tile model. In *Proof, Language and Interaction: Essays in Honour of Robin Milner.* MIT Press, 1999. To appear.
12. J.F. Groote and F. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100:202–260, 1992.
13. F.W. Lawvere. Functorial semantics of algebraic theories. *Proc. National Academy of Science*, 50:869–872, 1963.
14. K.G. Larsen and L. Xinxin. Compositionality through an operational semantics of contexts. In *Proc. ICALP'90*, vol. 443 of *LNCS*, pages 526–539, Springer, 1990.
15. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoret. Comput. Sci.*, 96:73–155, 1992.
16. J. Meseguer and U. Montanari. Mapping tile logic into rewriting logic. In *Proc. WADT'97*, vol. 1376 of *LNCS*, pages 62–91. Springer, 1998.
17. R. Milner. *A Calculus of Communicating Systems*, vol. 92 of *LNCS* Springer, 1980.
18. G. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Aarhus University, Computer Science Department, 1981.
19. A. Rensink. Bisimilarity of open terms. *Inform. and Comput.* 156:345–385, 2000.