Tile Bisimilarity Congruences for Open Terms and Term Graphs*

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 system specifications, the congruence is usually lifted from closed terms to open terms by instantiating the free variables in all possible ways; the only alternatives considered in the literature relying on Larsen and Xinxin's *context systems* and Rensink's *conditional transition systems*. We propose a different approach based on *tile logic*, where both closed and open terms are managed analogously. In particular, we analyze the 'bisimilarity as congruence' property for several tile formats that accomplish different concepts of subterm sharing.

Keywords: Bisimulation, tile logic, SOS formats, term graphs.

Introduction

The semantics of many languages can be conveniently expressed via labelled transition systems whose states are terms over a certain signature and whose labels give some abstract information about the performed transition. If such information is sufficient to model the possible 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) [21] is surely one of the most successful frameworks, where the transitions a system state can perform are defined by induction on the structure of states. Of course, once the transition system has been formalized, one is interested in equating those states that yield the same behavior.

In this paper we rely on *bisimulation* equivalence [20,19]: two states are equivalent if every action of the one can be simulated by the other, still ending in equivalent states. Within this setting, it is essential to have a compositional framework, where each subcomponent of a state can be safely replaced by any equivalent subcomponent without affecting the overall behavior. This motivated a lot of efforts in the definition of SOS formats (e.g., *De Simone* [10], GSOS [2],

* Research supported by CNR Integrated Project *Progettazione e Verifica di Sistemi Eterogenei Connessi mediante Reti*, Esprit Working Groups *CONFER2* and *COORDINA*, MURST project *TOSCa: Tipi, Ordine Superiore e Concorrenza*, and CICYT project *Desarrollo Formal de Sistemas Distribuidos* (TIC97–0669–C03–01).

`tyft/tyxt` [14,1]) whose syntactic constraints guarantee that *bisimilarity is a congruence*, where bisimilarity means the largest bisimulation. In fact those formats can serve as a guide in planning good system specifications. However the congruence is usually defined on closed terms only. It can be of course extended to *contexts* (i.e., terms with variables) by saying that two contexts are equivalent if they are equivalent under all possible instantiations of their free variables; this yields the coarsest conservative extension of the equivalence on closed terms that is preserved by instantiation and, to some extent, is analogous to reducing an equivalence on closed terms to the congruence respected by all possible 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 its *input interface* that is 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 may be convenient to distinguish between the situation in which different copies of a nondeterministic resource are available, which can evolve independently, from 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 give rise to 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 [19], 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 $D[p]$, since the former is stuck and the second can make a τ, i.e., the sharing information on x is lost after instantiation.

It is clear that 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) always 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* [11] instead of terms to describe states. The first solution avoids sharing at all. The second proposal requires an enrichment of the observation algebra. The third proposal is especially attractive for the specification and analysis of distributed systems. In fact, term graphs offer a finer model than terms, allowing one to express the sharing of closed subterms, and hence to distinguish, e.g., the coordinator **let** $x = p$ **in** $(x|x)\backslash_\alpha$ from $(p|p)\backslash_\alpha$.

As far as we know, the extension of good specification formats to incomplete systems has been addressed only recently by Rensink [22], who exploited a previous idea of De Simone. Rensink formalized several extensions to open terms of the bisimilarity defined just on closed terms, based on the notion of *conditional transition systems*. Some of these extensions turned out to be finer than instantiation closure. In this paper, we propose instead the use of *tile systems*, where the extensions (1)–(3) discussed above can be straightforwardly handled, both at the levels of specifications and of computational models. The *tile model* [13] is a formalism for describing the dynamic evolutions of open compositional systems. It relies on certain rewrite rules with side effects, called *basic tiles*, which are reminiscent of both SOS rules and Larsen and Xinxin's *context systems* [16]. It collects intuitions coming from *structured transition systems* [9] and *rewriting logic* [17], and by analogy with rewriting logic, the tile model has a logical presentation, called *tile logic*, where tiles are seen as sequents subject to certain inference rules. Tile logic extends (unconditional) rewriting logic by modeling state changes with side effects and rewriting synchronization.

Tiles are written $\alpha\colon s \xrightarrow[b]{a} t$, stating that the *initial configuration* s of the system evolves to the *final configuration* t producing the *effect* b, which can be observed by the rest of the system. 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*. The vertices of the tile are called *interfaces*. Tiles can be composed horizontally (synchronizing an effect with a trigger), vertically (computational evolutions of a certain component), and in parallel (concurrent steps) to generate larger steps. Given a set of basic tiles, the associated tile logic is obtained by adding some 'auxiliary' tiles — this may be necessary to represent consistent horizontal and vertical rearrangements of interfaces — and then freely composing in all possible ways both auxiliary and basic tiles.

In this paper, we want to stress that tiles are designed for open systems, (either incomplete or coordinators). In particular, the notion of bisimulation can be generalized to that of tile bisimulation, which operates over contexts rather than just over terms. More precisely, we investigate tile formats for basic tiles guaranteeing that tile bisimilarity is a congruence.

The first tile format appeared in the literature is the *algebraic tile format* (ATF) [12,13]. It has a cartesian structure of configurations and a monoidal structure of observations.[1] However, we show that the ATF is not completely satisfactory for defining bisimilarity congruences, due to the different structure of configurations and observations. We focus instead on three tile formats where configurations and observations reflects the approaches — linearity, free duplication and explicit sharing — previously discussed: (1) the *monoidal tile format* (MTF) [18], whose configurations and observations have a monoidal structure; (2) the *term tile format* (TTF) [5,6], whose configurations and observations have a cartesian structure; (3) the *gs-monoidal tile format* (GSTF), whose configurations have a gs-monoidal structure and observations have a monoidal structure. The MTF corresponds to abandoning sharing of subterms in configurations. The TTF extends the cartesian structure of configurations to observations, allowing, e.g., duplication of computations. Using the GSTF one can express sharing of closed subterms, as when employing term graphs.

By imposing the so called *basic source* constraint (i.e., the initial configurations of basic tiles must consist of a single basic operator) on the ATF, one can recover a slightly more general format than De Simone format (final configurations can be nonlinear) but stricter than (positive) GSOS. In this case, tile bisimilarity recovers the ordinary congruences for closed terms, but not for contexts. The analogous constraint on MTF and TTF yields respectively De Simone and a format more general than (positive) GSOS, both guaranteeing that tile bisimilarity is a congruence also for terms with variables. The GSTF cannot be easily recasted in existing SOS formats, due to its treatment of subterm sharing. We are indeed confident that GSTF can find a meaningful application for modeling (coordinator) systems with shared resources. Moreover, the syntactic 'basic source' constraint guarantees again that tile bisimilarity is a congruence.

Outline. In Section 1 we fix the notation and recall the notions of bisimulation, tiles, and tile bisimulation, while in Section 2 we review the most common SOS formats and several tile formats. 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. Formal proofs are moved into the appendix. Section 6 compares our approach with Rensink's proposal in [22]. We conclude by mentioning several directions for further work.

[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.

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. We recall that 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 function from three arguments to one, which is the projection on the second argument.

Bisimulation. A *labelled transition system* (LTS) is a triple $L = (S, \Lambda, \to)$, where S is a set of *states*, Λ is a set of *labels*, and $\to \subseteq S \times \Lambda \times S$. We let $s, t, \ldots$ range over S and $a, b, c, \ldots$ range over Λ. For $\langle s, a, t\rangle \in \to$ 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 $L = (S, \Lambda, \to)$, *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 extended to contexts by letting $C[x_1, \ldots, x_n] \simeq D[x_1, \ldots, x_n]$ iff for all closed terms $t_1, \ldots, t_n$ we have $C[t_1, \ldots, t_n] \simeq D[t_1, \ldots, t_n]$, i.e., the two contexts are equivalent if they are equivalent when instantiated w.r.t. all closed terms. This is not very practical for designing open systems, because of the universal quantification on all possible instantiations.

Tile systems and tile bisimulation. The point of view of the tile model [13] is that bisimulation can be defined uniformly on both closed terms and contexts,

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

Figure 1.

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., [13,3].

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 $R\colon N \to \mathcal{H} \times \mathcal{V} \times \mathcal{V} \times \mathcal{H}$ is a function such that for all $\alpha \in N$, if $R(\alpha) = \langle s, a, b, t \rangle$ then we have $s\colon x \to y$, $a\colon x \to z$, $b\colon y \to w$, and $t\colon z \to w$ for suitable objects x, y, z and w. We will write such rule α either as $\alpha\colon s \xrightarrow[b]{a} t$, 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.

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

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* [13]. 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.

$$\text{(a)}\ \frac{s_1 \xrightarrow{a_1} t_1 \ \ldots \ s_n \xrightarrow{a_n} t_n}{s \xrightarrow{a} t} \qquad \frac{\{x_i \xrightarrow{a_i} y_i \mid i \in I\}}{f(x_1, \ldots, x_n) \xrightarrow{a} D[y_1, \ldots, y_n]}\ \text{(b)}$$

Figure 2.

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 compositional systems. 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. [13]). *If* $\mathcal{R}$ *enjoys decomposition, then* $\simeq_{\mathrm{t}}$ *is a congruence.*

2 Relating SOS and Tile Formats

SOS formats. LTSs over the closed terms of a given signature Σ and label alphabet Λ can be more conveniently specified by means of 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 range over $\mathbb{T}_\Sigma(X)$ and the a_i, a range over Λ. Transitions in the upper part of the rule are called *premises* and the transition in the lower part is called *conclusion*. The LTS associated to a TSS 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 [21] 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 on the associated LTS is a congruence. Several SOS formats have been defined to ensure this important property For example, in the De Simone format (DSF) [10] 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 argument x_i can be used at most once in the premises, and if used cannot appear in D. The positive GSOS format [2] (we do not consider negative premises) extends the DSF in several ways (e.g., the same x_i can appear more than once in the premises and also in the target D of the conclusion, which can be nonlinear). The `tyft/tyxt` format [14] generalizes GSOS by allowing generic terms t_i as sources of the transition in 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 operator f applied to $ar(f)$ different variables. This fact is crucial in the proof that bisimilarity is a congruence: Conclusions of the form $C[x_1, \ldots, x_n] \xrightarrow{a} t$ can make the behavior of certain states be dependent on the context in which they are inserted.

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

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

Figure 3.

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* [15]. 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 $x_1, \ldots, x_n$, i.e., a standard naming is assumed. This allows us to denote a substitution $[t_1/x_1, \ldots, t_n/x_n]$ just by the tuple $\langle t_1, \ldots, t_n \rangle$, since names of substituted variables are implicit. We omit angle brackets when 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 each operator $f \in \Sigma_n$. The rule `id` yields identity substitutions $\langle x_1, \ldots, x_n \rangle$. The rule `seq` represents the 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 $x_{m+1}, \ldots, x_{m+l}$). Then some 'auxiliary' operators (i.e., not dependent on the signature) are introduced that recover the cartesian structure: *symmetries*, *duplicators*, and *dischargers*. The symmetry $\gamma_{\underline{n},\underline{m}}$ (rule `sym`) corresponds to the permutation $\langle x_{n+1}, \ldots, x_{n+m}, x_1, \ldots, x_n \rangle$. The duplicator $\nabla_{\underline{n}}$ (rule `dup`) introduces variable sharing and therefore nonlinear substitutions; it corresponds to $\langle x_1, \ldots, x_n, x_1, \ldots, x_n \rangle$. The discharger $!_{\underline{n}}$ (rule `dis`) recovers cartesian projections and corresponds to the empty substitution on $x_1, \ldots, x_n$. We shal not discuss the axioms in Table 1 (see, e.g., [3,5] for explanations).

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).

Term graphs [11] are in some sense situated between linear and cartesian terms, because they allow for explicit sharing and discharging, in such a way that

Table 1.

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

(a)
$$\begin{tikzcd} \underline{n} \arrow[r, "s"] \arrow[d, "a_1\otimes\cdots\otimes a_n"'] & \underline{1} \arrow[d, "a"] \\ \underline{n} \arrow[r, "t"'] & \underline{1} \end{tikzcd}$$

(b)
$$\frac{\{x_i \xrightarrow{a_i} y_i \mid i\in I\}}{C[x_1,\ldots,x_n]\xrightarrow{a} D[y_1,\ldots,y_n]}$$

Figure 4.

this information is preserved by composition. In fact, Corradini and Gadducci [8] have shown that by abandoning the naturality of ∇ and ! in the construction of $\mathbf{Th}[\Sigma]$, one obtains a category (called *gs-monoidal*, from graph substitution) $\mathbf{GS}[\Sigma]$ 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 C is linear.

The tile format proposed in the original presentation of tiles [13] is the so-called *algebraic tile format* (ATF) that recollected 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 $s,t\in\mathbb{T}_\Sigma(\{x_1,\ldots,x_n\})$. 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 [13] for more details. The ATF corresponds to SOS rules of the form 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 closed terms s and t and for any label a, $\mathcal{R}\vdash s\xrightarrow[a]{id_{\underline{0}}}t$ holds if and only if the LTS associated to the SOS specification includes the transition $s\xrightarrow{a}t$.

In the ATF, $\mathcal{H}$ is cartesian, whereas $\mathcal{V}$ is only monoidal. We will show that this combination can create some difficulties in dealing with tile bisimilarity as a congruence. In particular, it will be convenient either to consider the *term tile*

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

Figure 5.

format (TTF) [5], where also $\mathcal{V}$ is cartesian, or renounce the horizontal cartesian structure. This latter option can be achieved (at least) in two ways: either considering the *monoidal tile format* (MTF) [18] where also $\mathcal{H}$ is monoidal, or resorting to the *gs-monoidal tile format* (GSTF), where a faithful account of subterm sharing is provided by abandoning the naturality of ∇ and !.

The MTF deals only with linear terms and has no auxiliary tiles. Its basic tiles are similar to those of the ATF, but initial and final configurations must be linear. Although the auxiliary tiles of the GSTF are the same as those of the ATF, the former deals with term graphs rather than terms and therefore when composing tiles the naturality axioms of duplicators and dischargers cannot be applied, yielding a more concrete framework.

According to the TTF basic tiles have the form:

$$\begin{tikzcd} \underline{n} \arrow[r, "h"] \arrow[d, "v"'] & \underline{m} \arrow[d, "u"] \\ \underline{k} \arrow[r, "g"'] & \underline{1} \end{tikzcd}$$

with $h, v \in \mathbf{Th}[\Sigma]$ and $v, u \in \mathbf{Th}[\Lambda]$, where Σ is the horizontal signature of configurations and Λ is the vertical signature of observations. Of course, if Λ contains only elementary actions, regarded as unary operators, then $m = 1$. We present term tiles more concisely as logic sequents $n \triangleleft\ h \xrightarrow[u]{v} g$, where the number of variables in the 'upper-left' corner of the tile is made explicit.

Auxiliary tiles for term tile systems consist of all tiles that perform consistent rearrangements in the two dimensions, i.e., term tiles $k \triangleleft\ s \xrightarrow[u]{v} t$ such that $s; u = v; t$ with either (1) s, t, u and v are terms over the empty signature (and therefore also in $\mathbf{Th}[\Sigma] \cap \mathbf{Th}[\Lambda]$); or (2) $s, t \in \mathbf{Th}[\Sigma]$ and u and v are terms over the empty signature; or (3) $u, v \in \mathbf{Th}[\Lambda]$ and s and t are terms over the empty signature. 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 consistently 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 consistently 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 [5,3] for more details). Table 2 summarizes the differences between the tile formats we consider.

Table 2.

	$\mathcal{H}$	$\mathcal{V}$	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, $!_t$, $\sigma_{\underline{1},\underline{1}}$, $\tau_{\underline{1}}$, $\pi_{\underline{1}}$,...

Our aim is to extend the theory of SOS formats to tile formats for defining syntactic constraints on basic tiles that enforce the 'tile bisimilarity as congruence' property. One such constraint, already noticed in [13], is called *basic source*, which is reminiscent of analogous restrictions required by most good SOS formats: A tile system enjoys the basic source property if the initial configuration of each basic tile consists of a single operator, rather than a generic context.

Proposition 2 (cf. [13]). *If an algebraic tile system $\mathcal{R}$ enjoys the basic source property, then tile bisimilarity on closed terms is a congruence.*

3 The Monoidal Tile Format

If the monoidal tile format is employed, the basic source requirement establishes a bijective correspondence between basic tiles and rules in De Simone format (as noticed in [13]). Therefore it is trivial to show that tile bisimilarity on closed terms exactly corresponds to bisimilarity on the LTS defined by the associated De Simone presentation of the tile system. However, we can extend this result to contexts in a different way from ordinary 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 $\simeq_{\mathrm{t}} \subseteq \simeq$. It can be shown that $\simeq_{\mathrm{t}}$ is in general finer than $\simeq$ on contexts. To see this, it suffices to introduce a basic tile whose trigger cannot be generated by any closed instance of the associated argument. We propose the following nontrivial example, inspired by [22].

Example 1. Let us consider finite CCS (see Appendix A) 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 De Simone rule

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

whose corresponding MTF tile is $do^n_\mu(x_1) \xrightarrow[\mu]{\mu} do^{n-1}_\mu(x_1)$ (see Appendix A for the other basic tiles). 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 $C_2[x_1] = \beta.\alpha.nil + \beta.nil$ (already considered in [22]) are bisimilar but not tile bisimilar. In fact, if p is a closed 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 such that $p \xrightarrow{\alpha} q$. Therefore $C_2[p] = \beta.\alpha.nil + \beta.nil$ is always able to match the β move, i.e., $C_1[p] \simeq C_2[p]$ for all closed p. On the other hand, we have the tile $C_1[x_1] \xrightarrow[\beta]{id} do^1_\alpha(x_1)$ that cannot be matched by $C_2[x_1]$ ending in tile bisimilar states. In fact, if $C_2[x_1] \xrightarrow[\beta]{id} \alpha.nil$ then the tile $\alpha.nil \xrightarrow[\alpha]{id} nil$ cannot be matched by $do^1_\alpha(x_1)$ (the trigger α would be required). 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 rather than $nil[x_1] = nil \otimes !(x_1)$, where the discarded variable is made explicit as required by the 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 algebraic tile system for finite CCS (see Appendix A) given in [13]. It is straightforward that on closed terms tile bisimilarity $\simeq_{\mathrm{t}}$ coincides with bisimilarity $\simeq$ on the ordinary LTS of CCS. Moreover, since the TSS is De Simone, then bisimilarity 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 tile bisimilarity is not a congruence in the algebra of contexts. Finite CCS admits also a presentation in TTF [5,3], which is recalled in Appendix A. Again we have that on closed terms tile bisimilarity $\simeq_{\mathrm{t}}$ coincides with $\simeq$. But now, thanks to the auxiliary tiles of TTF we can prove that $C_1[x_1] \not\simeq_{\mathrm{t}} C_2[x_1]$. In fact 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 6, noticing 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 of the

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

Figure 6.

variables with closed terms. In fact it is completely analogous and consistent with that of closed terms and moreover takes care of the specification constraints (initial configurations of basic tiles can be seen, e.g., as incomplete open software modules) rather than just of the effective realizations (closed systems). As for MTF, it can be shown that $\simeq_{\mathrm{t}}$ for TTF is in general finer than the corresponding instantiation closure $\simeq$. A nontrivial example is given, e.g., by considering again the family of $do^n_\mu(_)$ operators.

5 The Gs-Monoidal Format for Explicit Sharing

It should be evident that 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. The general setting of 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 (where ∇ and ! are not natural). Although the auxiliary and basic tiles of GSTF are much like those of ATF, the gs-monoidal (rather than cartesian) structure is lifted from configurations to (horizontal composition of) tiles.

Example 3. The behavior of the term graph representation

$$\textbf{let } x_1 = \alpha.nil + \bar{\alpha}.nil \textbf{ in } (x_1|x_1)\backslash_\alpha$$

of the CCS agent $((\alpha.nil + \bar{\alpha}.nil)|(\alpha.nil + \bar{\alpha}.nil))\backslash_\alpha$ must be evaluated 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 7(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 7(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, like coordinators discussed in the Introduction.

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).*

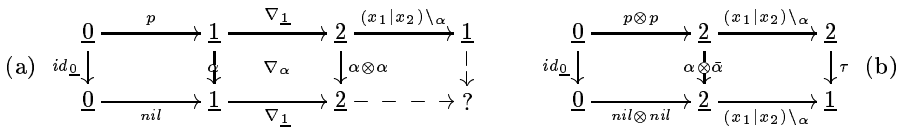

Figure 7.

6 Related Work: Conditional Transition Systems

The problem of defining formats for bisimilarity congruences on open terms has been previously addressed by Rensink in [22], exploiting *conditional transition systems* (CTS). The basic ingredients of 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 $\{x_1 \xrightarrow{a_1} y_1, ..., x_n \xrightarrow{a_n} y_n\}$ (with variable names as nodes) that involves suitable assumptions about the variables in s and t, defining the hypotheses under which the open transition exists. 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 environments used along the consecutive steps of a computation are persistent, and thus they can be reused when necessary.

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), whose rule premises and conclusions have the form $\Gamma_i \vdash s_i \xrightarrow{a_i} t_i$, and come equipped with three administrative rules (called *variable*, *weakening* and *substitution*) that operate on the basic conditional transitions of any specification. The latter have a special format designed for expressing the behavior of basic symbols of the signature. CTSSs also include binders, although Rensink investigates only *recursion based* CTSSs.

The main theorem of [22] 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 (algebraic) tile 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, while vertical composition of tiles has no administrative counterpart in CTSS.

The presence of nonlinear terms (that are dealt with differently in ATF, TTF and GSTF) also introduces transitions that are unexpected in Rensink's framework. For instance, the tile $x_1|x_1 \xrightarrow[\alpha;\alpha]{\alpha} x_1|x_1$ appears in all the tile systems for CCS that we have considered (except MTF, where duplication is not allowed); this corresponds to using the same assumption to prove 'atomically' two consecutive steps, which can have some interpretation in terms of $\simeq^{\mathrm{hp}}$, but not of $\simeq^{\mathrm{fh}}$.

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 relies on global linearity of the modeled system. 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, via explicit sharing of closed subterms, which was missing completely in the literature. Currently we are studying several applications that employs specifications in GSTF. In many such cases the congruence proofs (via decomposition property) can be carried out at the pictorial level as tile pastings (see Appendix B).

At a first look, open systems are just natural extensions of the notion of closed system, but when we study the point with more attention we immediately find different classes of open systems. An initial classification would distinguish incomplete systems, which define the main 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 of systems 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 (i.e., an incomplete system by itself). In another recent paper [7], 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 for (external) contexts, practically without any format limitation. It seems rather interesting to try to unify the advances in both directions in order to obtain a uniform treatment of both internal and external contexts in the framework of dynamic bisimilarity.

Another related topic which we are working at is to restrict the tiles to be considered when defining bisimulation for open terms, trying to remove those tiles containing triggers which in fact cannot be generated after instantiating the variables in any possible way. The idea is to obtain a weaker notion, closer to the one obtained by instantiating variables with closed terms.

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.

References

1. K. Bernstein. A congruence theorem for structured operational semantics of higher-order languages. In *Proc. 13th LICS Symposium*. IEEE Press, 1998.
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 TD-99-01, Computer Science Deptartment, University of Pisa, 1999.
4. R. Bruni, F. Gadducci, and U. Montanari. Normal Forms for Partitions and Relations. In *Proc. WADT'98*, vol. 1589 of *LNCS*, pages 31–47. Springer, 1999.
5. R. Bruni, J. Meseguer, and U. Montanari. Process and term tile logic. Technical Report SRI-CSL-98-06, SRI International, 1998.
6. R. Bruni, J. Meseguer, and U. Montanari. Symmetric monoidal and cartesian double categories as a semantic framework for tile logic. *Mathematical Structures in Computer Science*, 1999. To appear.
7. 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.
8. A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via gs-monoidal categories. *Applied Categorical Structures*, 7(4):299–331, 1999.
9. 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.
10. R. De Simone. Higher level synchronizing devices in MEIJE–SCCS. *Theoret. Comput. Sci.*, 37:245–267, 1985.
11. M.C. van Eekele, M.J. Plasmeijer, and M.R. Sleep, editors. *Term Graph Rewriting: Theory and Practice*, Wiley, London, 1993.
12. 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.
13. F. Gadducci and U. Montanari. The tile model. In *Proof, Language and Interaction: Essays in Honour of Robin Milner*. MIT Press, 1999. To appear.
14. J.F. Groote and F. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100:202–260, 1992.
15. F.W. Lawvere. Functorial semantics of algebraic theories. *Proc. National Academy of Science*, 50:869–872, 1963.
16. 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.
17. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoret. Comput. Sci.*, 96:73–155, 1992.
18. J. Meseguer and U. Montanari. Mapping tile logic into rewriting logic. In *Proc. WADT'97*, vol. 1376 of *LNCS*, pages 62–91. Springer, 1998.
19. R. Milner. *A Calculus of Communicating Systems*, vol. 92 of *LNCS* Springer, 1980.
20. D. Park. Concurrency and automata on infinite sequences. In *Proc. 5th G-I Conference*, vol. 104 of *LNCS*, pages 167–183, Springer, 1981.
21. G. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Aarhus University, Computer Science Department, 1981.
22. A. Rensink. Bisimilarity of open terms. In *Proc. EXPRESS'97, Expressiveness in Concurrency*, vol. 7 of *Elect. Notes in Th. Comput. Sci.*, Elsevier Science, 1997.

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

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

Figure 8.

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

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

Figure 9.

A Tiles and CCS

Milner's *Calculus for Communicating Systems* (CCS) [19] is among the better known and studied concurrency models. We recall here its operational semantics and the tile specifications that have been 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* (where $\bar{_}$) is an involutive function such that $\Delta = \bar{\bar{\Delta}}$ and $\Delta \cap \bar{\Delta} = \varnothing$). We denote by Λ (ranged over by λ) the set $\Delta \cup \bar{\Delta}$. 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, a CCS *process* is a term generated by the following grammar:

$$P ::= nil \ \mid \ \mu.P \ \mid \ P\backslash_\alpha \ \mid \ P + P \ \mid \ P|P$$

We let P, Q, R, ... range over the set **Proc** of CCS processes. Assuming the reader familiar with the notation, we give just an informal description of CCS operators: the constant *nil* yields the *inactive* process; process $\mu.P$ can behave like P but only after the execution of action μ; process $P\backslash_\alpha$ is P with actions α and $\bar{\alpha}$ blocked by *restriction* $_\backslash_\alpha$; process $P + Q$ is the *nondeterministic guarded sum* of P and Q; process $P|Q$ is the *parallel composition* of P and Q. The SOS system for CCS processes is defined by the set of De Simone rules in Figure 8 (the obvious symmetric rules for sum and asynchronous communication are omitted).

The basic tiles of the algebraic tile system for CCS proposed in [13] are given in Figure 9.

The basic tiles of the monoidal tile system for CCS proposed in [18] differ from those in the algebraic format only in the treatment of nondeterministic sum. This is because the monoidal format has no horizontal discharger and therefore an explicit unary operator $!(_)$ must be introduced for garbaging discarded arguments. Thus, the MTF tiles for nondeterministic sum are given in Figure 10. There are no tiles with source $!(x_1)$ and therefore the operator $!(_)$ locks its argument.

$$\langle +_\mu : x_1 + x_2 \xrightarrow[\mu]{\mu \otimes id} x_1 \otimes !(x_2) \qquad +\rangle_\mu : x_1 + x_2 \xrightarrow[\mu]{id \otimes \mu} !(x_1) \otimes x_2$$

Figure 10.

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

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

Figure 11.

The basic tiles of the gs-monoidal tile systems for CCS have not been presented in the literature, however, they are essentially the same as those of the MTF case, but where the operator $!(_)$ is the discharger $!_{\underline{1}}$.

The basic tiles of the term tile system for CCS proposed in [5] are given in Figure 11. Apart from the notation, the main difference w.r.t. the ATF tiles resides again in the rule for sum. In fact we can now discard the unused argument at the level of observations (see the trigger of $\langle +_\mu)$.

B Proofs

Proof of Theorem 2

Proof. The proof reduces to demonstrating that for term tile systems the basic source implies the decomposition property, and then to apply Proposition 1. To show this, we proceed by induction on the proof of a generic tile $\alpha = s \xrightarrow[b]{a} t$.

The base cases (identities, auxiliary tiles and basic tiles) are either obvious or implied by the basic source property.

If the proof is obtained as the vertical composition of two tiles $\alpha_1 = s \xrightarrow[b_1]{a_1} t_1$ and $\alpha_2 = t_1 \xrightarrow[b_2]{a_2} t$, then the decomposition property trivially follows from the inductive hypothesis applied to α_1 and α_2.

If the proof is obtained as the parallel composition of two tiles $\alpha_1 = s_1 \xrightarrow[b_1]{a_1} t_1$ and $\alpha_2 = s_2 \xrightarrow[b_2]{a_2} t_2$, then $s = s_1 \otimes s_2$ and we have the following cases:

1. let $s = s'_1; s'_2$ for some s'_1 and s'_2, then there exist $s_{1,1}$, $s_{1,2}$, $s_{2,1}$ and $s_{2,2}$ such that $s'_1 = s_{1,1} \otimes s_{2,1}$ and $s'_2 = s_{1,2} \otimes s_{2,2}$, with also $s_1 = s_{1,1}; s_{1,2}$ and $s_2 = s_{2,1}; s_{2,2}$. Then by inductive hypothesis we have that α_1 and α_2 can be decomposed along the $s_{i,j}$ into smaller proofs $\alpha_{i,j}$ for $i, j \in [1, 2]$. Then the parallel compositions of $\alpha_{1,1}$ with $\alpha_{2,1}$ and of $\alpha_{1,2}$ with $\alpha_{2,2}$ provide a horizontal decomposition of α along s'_1 and s'_2;

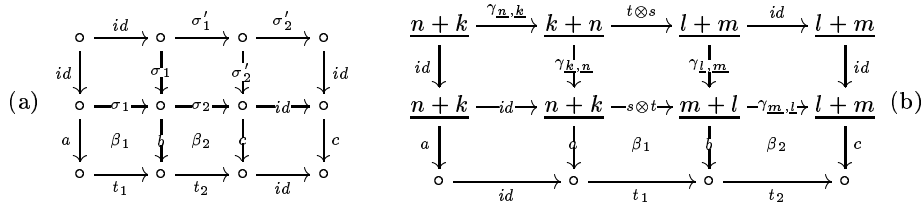

Figure 12.

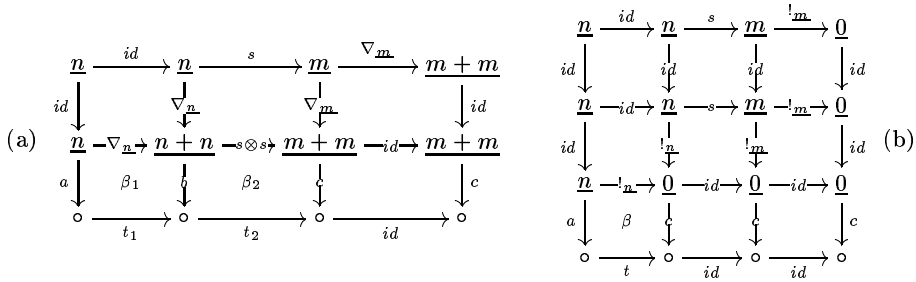

Figure 13.

2. let $s = s'_1 \otimes s'_2$ for some s'_1 and s'_2. If $s_i = s'_i$ for $i \in [1,2]$ then the decomposition is trivial. If $s_1 = s'_1 \otimes s'_{2,1}$ with $s'_2 = s'_{2,1} \otimes s_2$, then we apply the inductive hypothesis to s_1 obtaining a decomposition of α_1 along s'_1 and $s'_{2,1}$ into the parallel composition of α'_1 and $\alpha'_{2,1}$, but then the expected decomposition of α along s'_1 and s'_2 is given by α'_1 and the parallel composition of $\alpha'_{2,1}$ with α_2. Likewise if $s_2 = s'_{1,2} \otimes s'_2$ with $s'_1 = s_1 \otimes s'_{1,2}$.

If the proof is obtained as the horizontal composition of two tiles $\alpha_1 = s_1 \xrightarrow[b_1]{a} t_1$ and $\alpha_2 = s_2 \xrightarrow[b]{b_1} t_2$, then $s = s_1; s_2$ and we have the following cases:

1. let $s = s'_1 \otimes s'_2$ for some s'_1 and s'_2, then there exist $s'_{1,1}$, $s'_{1,2}$, $s'_{2,1}$ and $s'_{2,2}$ such that $s_1 = s'_{1,1} \otimes s'_{2,1}$ and $s_2 = s'_{1,2} \otimes s'_{2,2}$, with also $s'_1 = s'_{1,1}; s'_{1,2}$ and $s'_2 = s'_{2,1}; s'_{2,2}$. Then by inductive hypothesis we have that α_1 and α_2 can be decomposed along the $s'_{i,j}$ into smaller proofs $\alpha'_{i,j}$ for $i, j \in [1,2]$. Then the horizontal compositions of $\alpha'_{1,1}$ with $\alpha'_{1,2}$ and of $\alpha'_{2,1}$ with $\alpha'_{2,2}$ provide a parallel decomposition of α along s'_1 and s'_2;
2. let $s = s'_1; s'_2$ for some s'_1 and s'_2. This is the more intriguing case, where the auxiliary term tiles are exploited, making the difference between the algebraic and the term tile formats. If $s_i = s'_i$ for $i \in [1,2]$ then the decomposition is trivial. If $s_1 = s'_1; s'_{2,1}$ with $s'_2 = s'_{2,1}; s_2$, then we apply the inductive hypothesis to s_1 obtaining a decomposition of α_1 along s'_1 and $s'_{2,1}$ into the sequential composition of α'_1 and $\alpha'_{2,1}$, but then the expected decomposition of α along s'_1 and s'_2 is given by α'_1 and the sequential composition of $\alpha'_{2,1}$ with α_2. Likewise if $s_2 = s'_{1,2}; s'_2$ with $s'_1 = s_1; s'_{1,2}$. However,

it could also be the case that some of the axioms in Table 1 must be applied to prove that $s_1'; s_2' = s_1; s_2$ in such a way that none of the previous correspondence can be established. Hence we must check that the decomposition is not compromised by the application of the various axioms. *Monoidal* and *category* axioms (first and second rows) are directly reflected at the level of tiles and therefore preserve decomposition. *Coherence* axioms can be dealt with very easily by exploiting auxiliary tiles, because they involve just auxiliary operators. In fact let $\sigma_1; \sigma_2 = \sigma_1'; \sigma_2'$ be a generic coherence axiom such that a tile $\beta = \sigma_1; \sigma_2 \xrightarrow[c]{a} t$ can be decomposed as the sequential composition of $\beta_1 = \sigma_1 \xrightarrow[b]{a} t_1$ and $\beta_2 = \sigma_2 \xrightarrow[c]{b} t_2$, then we want to show that it can be decomposed also along σ_1' and σ_2'. The proof is illustrated in Figure 12(a). In fact, since all σ_i and σ_i' are auxiliary, they define both horizontal and vertical arrows, and by the property of auxiliary term tiles we know that the tiles $id \xrightarrow[\sigma_1]{id} \sigma_1$, $\sigma_1' \xrightarrow[\sigma_2']{\sigma_1} \sigma_2$, and $\sigma_2' \xrightarrow[id]{\sigma_2'} id$ are present. Therefore the tiles $\beta_1' = \sigma_1' \xrightarrow[\sigma_2';c]{a} t$ and $\beta_2' = \sigma_2' \xrightarrow[c]{\sigma_2';c} id$ (obtained by the horizontal and vertical pastings of β_1, β_2 and auxiliary tiles) provide a decomposition of β along σ_1' and σ_2'. Notice that each axiom generates two cases to check. For instance, the coherence axiom $\nabla_{\underline{n}}; \gamma_{\underline{n},\underline{n}} = \nabla_{\underline{n}}$ originates two cases: the first with $\sigma_1 = \sigma_1' = \nabla_{\underline{n}}$, $\sigma_2 = \gamma_{\underline{n},\underline{n}}$, $\sigma_2' = id_{\underline{n+n}}$, and the second one with $\sigma_1 = \sigma_1' = \nabla_{\underline{n}}$, $\sigma_2 = id_{\underline{n+n}}$, $\sigma_2' = \gamma_{\underline{n},\underline{n}}$. Finally, the *naturality* axioms must be considered (as for coherence, each axioms originates two cases to check):

$[(s \otimes t); \gamma_{\underline{m},\underline{l}} = \gamma_{\underline{n},\underline{k}}; (t \otimes s)]$: In the first case we assume that a tile β with initial configuration $(s \otimes t); \gamma_{\underline{m},\underline{l}}$ can be decomposed along $(s \otimes t)$ and $\gamma_{\underline{m},\underline{l}}$, then we want to show that the same tile can be decomposed along $\gamma_{\underline{n},\underline{k}}$ and $(t \otimes s)$. Since symmetries are auxiliary operators, the proof relies on the existence of auxiliary tiles $\gamma \xrightarrow[\gamma]{id} id$, $t \otimes s \xrightarrow[\gamma]{\gamma} s \otimes t$ and $id \xrightarrow[id]{\gamma} \gamma$ (subscripts omitted for readability), as illustrated in Figure 12(b). Likewise for the second case.

$[s; \nabla_{\underline{m}} = \nabla_{\underline{n}}; (s \otimes s)]$: Here the simpler case is assuming that a tile β with initial configuration $\nabla_{\underline{n}}; (s \otimes s)$ can be decomposed along $\nabla_{\underline{n}}$ and $(s \otimes s)$ in β_1 and β_2, and then show that there exists a decomposition β_1', β_2' of β along s and $\nabla_{\underline{m}}$. The pictorial proof for this case is illustrated in Figure 13(a). The second case is more complex to illustrate but analogous and thus omitted.

$[s; !_{\underline{m}} = !_{\underline{n}}]$: Here the interesting case is decomposing a tile β with initial configuration $!_{\underline{n}}$ along s and $!_{\underline{m}}$. The proof is illustrated in Figure 13(b). $\square$

The proof of Theorems 1 and 3 are analogous. In particular, the proof of Theorem 1 is much simpler since coherence and naturality axioms are not present. The proof of Theorem 3 is based on the fact that basic observations are all defined as unary arrows. In fact, since not all the auxiliary term tiles needed in the proof of coherence axioms are present, it is necessary to show that symmetries, duplicators and dischargers distribute on observations. On the other hand naturality of duplicators and dischargers cannot be dealt with, but we can avoid it since the horizontal structure is gs-monoidal.