Verification of Finite-State Machines: A Distributed Approach

Roberto Gorrieri
roberto.gorrieri@unibo.it

Dipartimento di Informatica — Scienza e Ingegneria
Università di Bologna,
Mura A. Zamboni, 7, 40127 Bologna, Italy

Abstract. Finite-state machines, a simple class of finite Petri nets, are equipped with a truly concurrent, bisimulation-based, behavioral equivalence, called *team equivalence*, which conservatively extends classic bisimulation equivalence over labeled transition systems and which is checked in a distributed manner, without necessarily building a global model of the overall behavior. An associated distributed modal logic, called *basic team modal logic* (BTML, for short), is presented and shown to be coherent with team equivalence: two markings are team equivalent if and only if they satisfy the same BTML formulae.

1 Introduction

A finite-state machine (FSM, for short) is a simple type of finite Petri net [17, 29, 33] whose transitions have singleton pre-set and singleton, or empty, post-set; therefore, they are very similar to finite-state, labeled transition systems (LTSs, for short) [22], a class of models that are suitable for describing sequential, nondeterministic systems, and are also widely used as a semantic model for process algebras (see, e.g., [16]). On this class of models, there is widespread agreement that a very natural and convenient equivalence relation is bisimulation equivalence [28, 25], an equivalence relation that can be verified efficiently for finite-state LTSs; more precisely, if m is the number of transitions and n is the number of states of the LTS, checking whether two states are bisimilar can be done in $O(m \log n)$ time [31].

Even if FSMs are the simplest distributed model of computation, the equivalence checking problem may be not easy. For instance, if we want to check if two markings m_1 and m_2 are *interleaving bisimilar*, (see, e.g., [17]), we have first to build two LTSs, one rooted in m_1 and the other rooted in m_2, usually called the *interleaving marking graphs*, and then to check whether these two rooted LTSs are bisimilar. However, such LTSs have a number of states that can grow exponentially with the size of the marked net, in particular w.r.t. the size of the involved markings, so that the equivalence checking problem is exponential, in general. This problem is shared by essentially all the equivalences that have been proposed in the literature for FSMs (see, e.g., [7, 30, 15, 17]), because all these equivalences are defined directly over the markings of the net.

Our main goal is to define a new equivalence relation that can be computed in a distributed manner, without resorting to a global model of the overall behavior of the analyzed marked net. The initial observation is that a place in an FSM represents a sequential process type and the number of tokens in that place represents the number of currently available instances of that sequential process type. Since an FSM is so similar to an LTS, we propose to define bisimulation equivalence [28, 25] directly over the set of places of the *unmarked* net. The advantage is that bisimulation equivalence is a relation on places, rather than on markings, and so much more easily computable; more precisely, if m is the number of net transitions and n is the number of places, checking whether two places are bisimilar can be done in $O(m \log (n+1))$ time, by adapting the algorithm in [31]. Moreover, the resulting notion of bisimilarity enjoys the same properties of bisimulation over LTSs, i.e., it is coinductive and equipped with a fixed-point characterization [25, 34, 16].

After the bisimulation equivalence over the set of places has been computed once and for all, we can define, in a purely structural way, that two markings m_1 and m_2 are *team equivalent* if they have the same cardinality, say $|m_1| = k = |m_2|$, and there is a bisimulation-preserving, bijective mapping between the two markings, so that each of the k pairs of places (s_1, s_2), with $s_1 \in m_1$ and $s_2 \in m_2$, is such that s_1 and s_2 are bisimilar. Team equivalence is a truly concurrent behavioral equivalence as it is sensitive to the size of the distributed state; as a matter of fact, it relates markings of the same size, only. Therefore, a sequential finite-state machine, i.e., an FSM with a singleton initial marking, can never be equated to a concurrent finite-state machine, i.e., an FSM with a non-singleton initial marking. The name *team equivalence* reminds us that two distributed systems, composed of a team of non-cooperating, sequential processes, are equivalent if it is possible to match each sequential component of the first system with one bisimulation-equivalent, sequential component of the other one, as in any sports where two competing (distributed) teams have the same number of (sequential) players. Once bisimilarity has been computed, checking whether two markings of size k are team equivalent can be computed in $O(k^2)$ time.

Note that to check whether two markings are team equivalent we need not construct an LTS describing the global behavior of the whole system, but only find a suitable, bisimulation-preserving match among the local, sequential states (i.e., the elements of the markings); in other words, we consider a collection of LTSs for the local, sequential states only, and try to match them through bisimilarity. Nonetheless, we will prove that team equivalence is coherent with the global behavior of the net. More precisely, we will show that team equivalence is finer than interleaving bisimilarity (so it respects the token game), actually it coincides with *strong place bisimilarity* [4, 5] (and so it respects the causal semantics of nets). Since we need not to construct the global behavior of the net under scrutiny, if we need to check whether other two markings of the same net, say m_1' and m_2', are team equivalent, we can reuse the already computed bisimulation equivalence over places, and so such a verification will take only $O(k^2)$ time, if k is the size of m_1' and m_2'.

The second part of the paper approaches the problem of finding a modal characterization of team equivalence over FSMs, in line of what Hennessy and Milner proved for standard bisimulation equivalence over LTSs [20]. The basic modal logic we start with is Hennessy-Milner Logic (HML) [20, 3], which is here slightly extended in order to distinguish between successful and unsuccessful termination; the resulting modal logic is called HMT. We prove a *basic coherence theorem* comparing model checking and equivalence checking: two places of an FSM are bisimilar if and only if they satisfy the same HMT formulae. Basic team modal logic (BTML, for short) is a proper, conservative extension of HMT, with an additional operator of parallel composition $\_ \otimes \_$ of formulae, to be used at the top level only. Also in this case, we prove a *full coherence theorem*: two markings are team equivalent if and only if they satisfy the same BTML formulae.

The paper is organized as follows. Section 2 introduces the basic definitions about finite-state machines and two behavioral equivalences: interleaving bisimilarity and strong place bisimilarity [4, 5]; the latter is quite interesting, as we will prove that team equivalence coincides with strong place bisimilarity for FSMs. Section 3 copes with the equivalence checking problem; first, bisimulation over places of an unmarked net is defined, showing that the classic results of bisimulation over LTSs also hold in this case; then, team equivalence is introduced and some examples discussing its pros and cons are presented; moreover, the minimization of an FSM w.r.t. bisimilarity is defined. Section 4 describes first HMT (the new variant of HML), its syntax and semantics, and shows the basic coherence theorem. Then, the new modal logic BTML is introduced and the full coherence theorem is proved. Finally, Section 5 discusses related literature, some future research and open problems.

2 Basic Definitions and Behavioral Equivalences

Definition 1. (Multiset) *Let $\mathbb{N}$ be the set of natural numbers. Given a finite set S, a* multiset *over S is a function $m : S \to \mathbb{N}$. The* support *set $dom(m)$ of a marking m is the set $\{s \in S \mid m(s) \neq 0\}$. The set of all multisets over S, denoted by $\mathscr{M}(S)$, is ranged over by m, possibly indexed. We write $s \in m$ if $m(s) > 0$. The* multiplicity *of s in m is given by the number $m(s)$. The* cardinality *of m, denoted by $|m|$, is the number $\sum_{s \in S} m(s)$, i.e., the total number of its elements. A multiset m such that $dom(m) = \emptyset$ is called* empty *and is denoted by θ. We write $m \subseteq m'$ if $m(s) \leq m'(s)$ for all $s \in S$.*

Multiset union $\_ \oplus \_$ *is defined as follows: $(m \oplus m')(s) = m(s) + m'(s)$; the operation $\oplus$ is commutative, associative and has θ as neutral element. If $m_2 \subseteq m_1$, then we can define* multiset difference $\_ \ominus \_$ *as follows: $(m_1 \ominus m_2)(s) = m_1(s) - m_2(s)$. The* scalar product *of a number j with m is the multiset $j \cdot m$ defined as $(j \cdot m)(s) = j \cdot (m(s))$.*

A multiset m over $S = \{s_1, \ldots, s_n\}$ can be represented as $k_1 \cdot s_1 \oplus k_2 \cdot s_2 \oplus \ldots \oplus k_n \cdot s_n$, where $k_j = m(s_j) \geq 0$ for $j = 1, \ldots, n$. □

Definition 2. (Finite-state machine) *A labeled* finite-state machine *(FSM, for short) is a tuple $N = (S, A, T)$, where*

- *S is the finite set of* places*, ranged over by s (possibly indexed),*
- *A is the finite set of* labels*, ranged over by ℓ (possibly indexed), and*
- *$T \subseteq S \times A \times (S \cup \{\theta\})$ is the finite set of* transitions*, ranged over by t (possibly indexed), such that, for each $\ell \in A$, there exists a transition $t \in T$ of the form (s, ℓ, m).*

Given a transition $t = (s, \ell, m)$, we use the notation:

- *${}^\bullet t$ to denote its* pre-set *s (which is a single place) of tokens to be consumed;*

- $l(t)$ *for its* label ℓ*, and*
- $t^\bullet$ *to denote its* post-set m *(which is a place or the empty multiset* θ*) of tokens to be produced.*

Hence, transition t can be also represented as ${}^\bullet t \xrightarrow{l(t)} t^\bullet$. □

Remark 1. **(Constraints on the definition of an FSM)** Our definition of T as a set of triples ensures that the net is *transition simple*, i.e., for all $t_1, t_2 \in T$, if ${}^\bullet t_1 = {}^\bullet t_2$ and $t_1^\bullet = t_2^\bullet$ and $l(t_1) = l(t_2)$, then $t_1 = t_2$. Note also that we are assuming that each transition has a nonempty pre-set: for our interpretation of net models, where a transition can only be performed by some sequential process, this requirement is strictly necessary. These are the only two constraints we impose over the definition of an FSM, as discussed in [17]. The additional condition that the set A of labels is covered by T (i.e., for each $\ell \in A$ there exists $t \in T$ with label ℓ) is just for economy. □

Graphically, a place is represented by a little circle, a transition by a little box, which is connected by a directed arc from the place in its pre-set and to the place in its post-set, if any.

Definition 3. (Marking, FSM net system, token game) *A multiset over S is called a* marking*. Given a marking m and a place s, we say that the place s contains* $m(s)$ tokens*, graphically represented by* $m(s)$ *bullets inside place s. An* FSM net system $N(m_0)$ *is a tuple* (S, A, T, m_0)*, where* (S, A, T) *is an FSM and* m_0 *is a marking over S, called the* initial marking*. We also say that* $N(m_0)$ *is a* marked *net. An FSM net system* $N(m_0)$ *is* sequential *if* m_0 *is a singleton, i.e.,* $|m_0| = 1$*; while it is* concurrent *if* m_0 *is arbitrary.*

A transition t is enabled *at marking m, denoted by* $m[t\rangle$*, if* ${}^\bullet t \subseteq m$*. The execution (or* firing*) of t enabled at m produces the marking* $m' = (m \ominus {}^\bullet t) \oplus t^\bullet$*. This is written* $m[t\rangle m'$*. This procedure is called the* token game*. A marking m is* stuck *if it does not enable any transition.* □

Example 1. By using the usual drawing convention for Petri nets, Figure 1 shows in (a) a sequential FSM, which performs a, possibly empty, sequence of a's and b's, until it performs one c and then stops *successfully* (the token disappears in the end). Note that a sequential SFM is such that any reachable marking is a singleton or empty. Hence, a sequential FSM is a *safe* (or 1-bounded) net: each place in any reachable marking can hold one token at most.

In (b), a concurrent FSM is depicted, that can perform a forever, interleaved with two occurrences of b, only: the two tokens in s_4 will eventually reach s_5, which is a place representing unsuccessful termination (deadlock). Note that a concurrent FSM is a *k-bounded* net, where k is the size of the initial marking: each place in any reachable marking can hold k tokens at most. □

Now we recall two well-known behavioral equivalences over Petri nets: interleaving bisimilarity and strong place bisimilarity.

Definition 4. (Interleaving Bisimulation) *Let* $N = (S, A, T)$ *be an FSM. An* interleaving bisimulation *is a relation* $R \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$ *such that if* $(m_1, m_2) \in R$ *then*

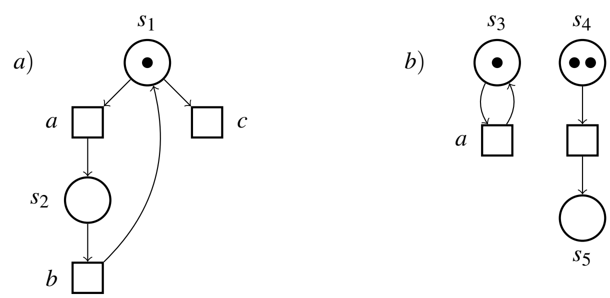

Fig. 1. A sequential finite-state machine in (a), and a concurrent finite-state machine in (b)

- *$\forall t_1$ such that $m_1[t_1\rangle m_1'$, $\exists t_2$ such that $m_2[t_2\rangle m_2'$ with $l(t_1) = l(t_2)$ and $(m_1', m_2') \in R$,*
- *$\forall t_2$ such that $m_2[t_2\rangle m_2'$, $\exists t_1$ such that $m_1[t_1\rangle m_1'$ with $l(t_1) = l(t_2)$ and $(m_1', m_2') \in R$.*

Two markings m_1 and m_2 are interleaving bisimilar *(or* interleaving bisimulation equivalent*), denoted by $m_1 \sim_{int} m_2$, if there exists an interleaving bisimulation R such that $(m_1, m_2) \in R$.* □

Interleaving bisimilarity $\sim_{int}$, which is defined as the union of all the interleaving bisimulations, is the largest interleaving bisimulation and also an equivalence relation.

Remark 2. **(Interleaving bisimulation between two nets)** The definition above covers also the case of an interleaving bisimulation between two FSMs, say, $N_1 = (S_1, A, T_1)$ and $N_2 = (S_2, A, T_2)$ with $S_1 \cap S_2 = \emptyset$, because we may consider just one single FSM $N = (S_1 \cup S_2, A, T_1 \cup T_2)$: An interleaving bisimulation $R \subseteq \mathscr{M}(S_1) \times \mathscr{M}(S_2)$ is also an interleaving bisimulation on $\mathscr{M}(S_1 \cup S_2) \times \mathscr{M}(S_1 \cup S_2)$. Similar considerations hold for all the bisimulation-like definitions we propose in the following, which will be defined on a single net only. □

Remark 3. **(Comparing two marked nets)** The definition above of interleaving bisimulation is defined over an *unmarked* FSM, i.e., a net without the specification of an initial marking m_0. Of course, if one desires to compare two marked nets, then it is enough to find an interleaving bisimulation (over the union of the two nets, as discussed in the previous remark), containing the pair composed of the respective initial markings. This approach is also followed for the other bisimulation-like definitions we propose in the following. □

We now introduce strong place bisimulation equivalence, proposed in [4, 5] as an improvement over *strong bisimilarity*, a place-based, bisimulation-like behavioral relation defined by Olderog in [27] over safe nets (that is not an equivalence relation). Our definition is formulated in a slightly different way, but it is coherent with the original one. An auxiliary definition first.

Definition 5. (Additive closure) *Given an FSM net $N = (S, A, T)$ and a* place relation *$R \subseteq S \times S$, we define a* marking relation *$R^{\oplus} \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$, called the* additive closure *of R, as the least relation induced by the following axiom and rule.*

$$\frac{}{(\theta,\theta)\in R^{\oplus}} \qquad \frac{(s_1,s_2)\in R \quad (m_1,m_2)\in R^{\oplus}}{(s_1\oplus m_1, s_2\oplus m_2)\in R^{\oplus}}$$

□

Note that, by definition, two markings are related by $R^{\oplus}$ only if they have the same size; in fact, the axiom states that the empty marking is related to itself, while the rule, assuming by induction that m_1 and m_2 have the same size, ensures that $s_1 \oplus m_1$ and $s_2 \oplus m_2$ have the same size. An alternative way to define that two markings m_1 and m_2 are related by $R^{\oplus}$ is to state that m_1 can be represented as $s_1 \oplus s_2 \oplus \ldots \oplus s_k$, m_2 can be represented as $s'_1 \oplus s'_2 \oplus \ldots \oplus s'_k$ and $(s_i, s'_i) \in R$ for $i = 1, \ldots, k$.

Definition 6. (Strong Place Bisimulation) *Let $N = (S, A, T)$ be an FSM. A* strong place bisimulation *is a relation $R \subseteq S \times S$ such that if $(m_1, m_2) \in R^{\oplus}$ then*

- *$\forall t_1$ such that $m_1[t_1\rangle m'_1$, $\exists t_2$ such that $m_2[t_2\rangle m'_2$ with $(^\bullet t_1, ^\bullet t_2) \in R^{\oplus}$, $l(t_1) = l(t_2)$, $(t_1^\bullet, t_2^\bullet) \in R^{\oplus}$ and $(m'_1, m'_2) \in R^{\oplus}$,*
- *$\forall t_2$ such that $m_2[t_2\rangle m'_2$, $\exists t_1$ such that $m_1[t_1\rangle m'_1$ with $(^\bullet t_1, ^\bullet t_2) \in R^{\oplus}$, $l(t_1) = l(t_2)$, $(t_1^\bullet, t_2^\bullet) \in R^{\oplus}$ and $(m'_1, m'_2) \in R^{\oplus}$.*

Two markings m_1 and m_2 are strong place bisimilar *(or* strong place bisimulation equivalent*), denoted by $m_1 \sim_p m_2$, if there exists a strong place bisimulation R such that $(m_1, m_2) \in R^{\oplus}$.* □

Strong place bisimilarity $\sim_p$ is an equivalence relation [4]. Its definition, however, is not completely coinductive, as the union of strong place bisimulations may be not a place bisimulation [4], at least for P/T nets. Nonetheless, $\sim_p$ has been characterized as the union of all the *reflexive* strong place bisimulations [4].

Of course, strong place bisimilarity is finer than interleaving bisimilarity, because a strong place bisimulation R is such that $R^{\oplus}$ is an interleaving bisimulation. This will be illustrated in the following, also by means of examples.

3 A Distributed Approach to Equivalence Checking

3.1 Bisimulation on Places

We recall the definition of bisimulation, originally defined in [28, 25] over LTSs, adapted here for unmarked FSMs. Because of the shape of FSMs transitions, in the definition below the post-sets m_1 and m_2 can be either the empty marking θ or a single place.

Definition 7. (Bisimulation) *Let $N = (S, A, T)$ be an FSM. A* bisimulation *is a relation $R \subseteq S \times S$ such that if $(s_1, s_2) \in R$ then for all $\ell \in A$*

- *$\forall m_1$ such that $s_1 \xrightarrow{\ell} m_1$, $\exists m_2$ such that $s_2 \xrightarrow{\ell} m_2$ and either $m_1 = \theta = m_2$ or $(m_1, m_2) \in R$,*
- *$\forall m_2$ such that $s_2 \xrightarrow{\ell} m_2$, $\exists m_1$ such that $s_1 \xrightarrow{\ell} m_1$ and either $m_1 = \theta = m_2$ or $(m_1, m_2) \in R$.*

Two places s and s' are bisimilar *(or* bisimulation equivalent*), denoted $s \sim s'$, if there exists a bisimulation R such that $(s,s') \in R$.* $\square$

We now list some useful properties of bisimulation relations on net places.

Proposition 1. *For each FSM $N = (S,A,T)$, the following hold:*

1. *the identity relation $\mathscr{I} = \{(s,s) \mid s \in S\}$ is a bisimulation;*
2. *the inverse relation $R^{-1} = \{(s',s) \mid (s,s') \in R\}$ of a bisimulation R is a bisimulation;*
3. *the relational composition $R_1 \circ R_2 = \{(s,s'') \mid \exists s'.(s,s') \in R_1 \wedge (s',s'') \in R_2\}$ of two bisimulations R_1 and R_2 is a bisimulation;*
4. *the union $\bigcup_{i \in I} R_i$ of bisimulations R_i is a bisimulation.*

Proof. *The proof is a slight adaptation of the analogous results for standard bisimulation on LTSs [25, 34, 16], and so omitted.* $\square$

Remember that $s \sim s'$ if there exists a bisimulation containing the pair (s,s'). This means that $\sim$ is the union of all bisimulations, i.e.,

$$\sim = \bigcup \{R \subseteq S \times S \mid R \text{ is a bisimulation}\}.$$

By Proposition 1(4), $\sim$ is also a bisimulation, hence the largest such relation.

Proposition 2. *For each FSM $N = (S,A,T)$, relation $\sim \subseteq S \times S$ is the largest bisimulation relation.* $\square$

Observe that a bisimulation relation need not be reflexive, symmetric, or transitive. Nonetheless, the largest bisimulation relation $\sim$ is an equivalence relation. As a matter of fact, as the identity relation $\mathscr{I}$ is a bisimulation by Proposition 1(1), we have that $\mathscr{I} \subseteq \sim$, and so $\sim$ is reflexive. Symmetry derives from the following argument. For any $(s,s') \in \sim$, there exists a bisimulation R such that $(s,s') \in R$; by Proposition 1(2), relation R^{-1} is a bisimulation containing the pair (s',s); hence, $(s',s) \in \sim$ because $R^{-1} \subseteq \sim$. Transitivity also holds for $\sim$. Assume $(s,s') \in \sim$ and $(s',s'') \in \sim$; hence, there exist two bisimulations R_1 and R_2 such that $(s,s') \in R_1$ and $(s',s'') \in R_2$; by Proposition 1(3), relation $R_1 \circ R_2$ is a bisimulation containing the pair (s,s''); hence, $(s,s'') \in \sim$, because $R_1 \circ R_2 \subseteq \sim$. Summing up, we have the following.

Proposition 3. *For each FSM $N = (S,A,T)$, relation $\sim \subseteq S \times S$ is an equivalence relation.* $\square$

Bisimulation equivalence over a finite-state LTS with n states and m transitions can be computed in $O(m \log n)$ time [31]. The same partition refinement algorithm can be easily adapted also for FSMs: it is enough to consider the empty marking θ as an additional, special place which is bisimilar to itself only. Hence, if N has n places and m transitions, $\sim$ can be computed in $O(m \log (n+1))$ time. It is also possible to characterize $\sim$ as the greatest fixed point of a suitable functional over binary relations, as done over LTSs in [25, 34, 16].

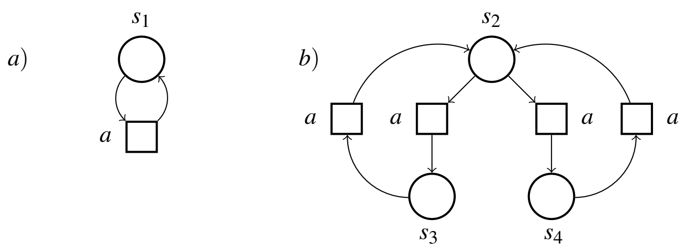

Fig. 2. Two bisimilar FSMs

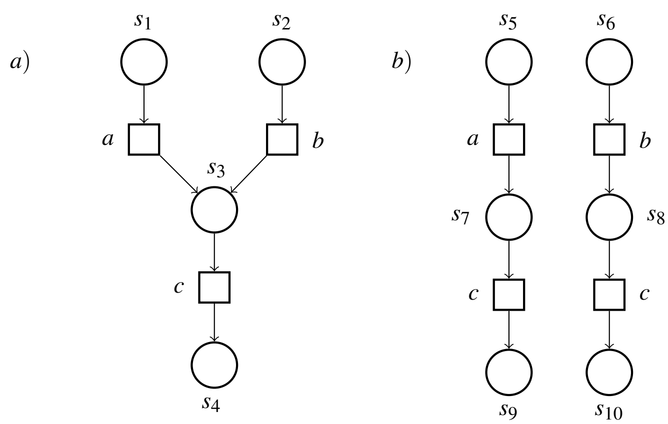

Fig. 3. Other two bisimilar FSMs

Example 2. Consider the nets in Figure 2, where the net in (b) is obtained from the net in (a) by unfolding once the cycle and also by duplicating it. It is not difficult to realize that relation $R = \{(s_1,s_2),(s_1,s_3),(s_1,s_4)\}$ is a bisimulation. For instance, let us consider the pair (s_1,s_2); we have to prove that for any move of s_1, s_2 can reply, and vice versa; the only transition from s_1 is $s_1 \xrightarrow{a} s_1$, which is matched by one of the two transitions from s_2, e.g., $s_2 \xrightarrow{a} s_3$, with $(s_1,s_3) \in R$, as required; symmetrically, for any move of s_2, s_1 can reply, and the reached pairs are (s_1,s_3) and (s_1,s_4), which are both in R, as required. As a matter of fact, any pair of places belongs to relation $\sim$, i.e., $\sim = S \times S$, with $S = \{s_1,s_2,s_3,s_4\}$. □

Example 3. Consider the nets in Figure 3. A possible bisimulation is $R = \{(s_1,s_5),(s_2,s_6),(s_3,s_7),(s_3,s_8),(s_4,s_9),(s_4,s_{10})\}$. This example show that fusion of places is possible. □

Example 4. Consider the nets in Figure 4. It is not difficult to realize that relation $R = \{(s_1,s_5),(s_2,s_6),(s_2,s_7),(s_3,s_8),(s_4,s_8)\}$ is a bisimulation. □

Example 5. Consider the nets in Figure 5. It is not difficult to realize that $s_1 \nsim s_3$. In fact, s_1 may reach s_2 by performing a; s_3 can reply to this move in two different ways

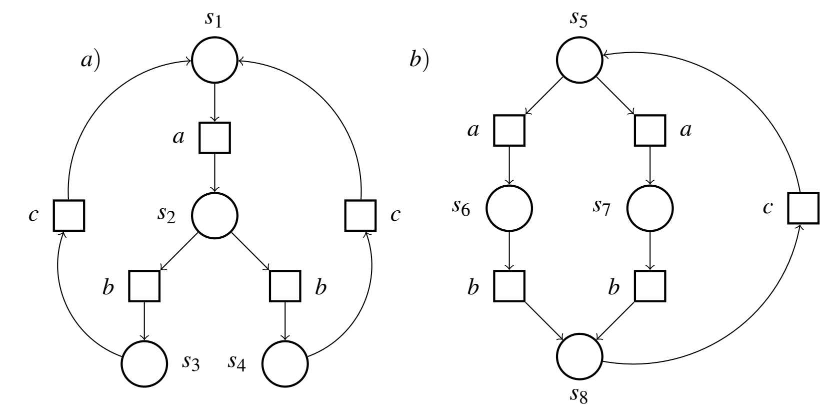

Fig. 4. Two more complex bisimilar FSMs

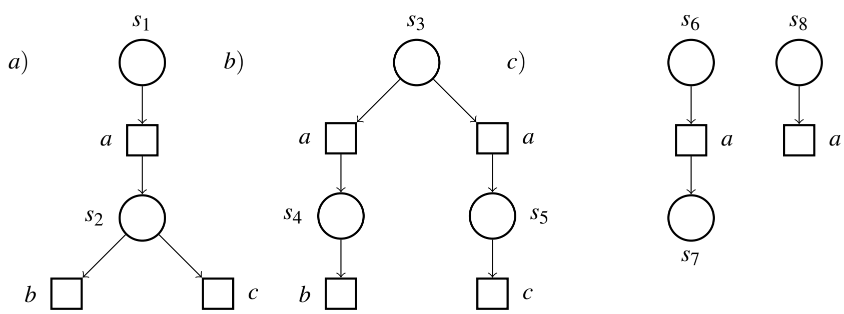

Fig. 5. Some non-bisimilar FSMs

by reaching either s_4 or s_5; however, while s_2 offers both b and c, s_4 may perform only b and s_5 only c; hence, s_4 and s_5 are not bisimilar to s_2 and so also s_1 is not bisimilar to s_3. This example shows that bisimulation equivalence is sensitive to the timing of choice.

Furthermore, also s_6 and s_8 are not bisimilar. In fact, s_6 can reach s_7 by performing a, while s_8 can reply by reaching the empty marking, but $\theta \nsim s_7$. This example shows that bisimulation equivalence is sensitive to the kind of termination of a process: even if s_7 is stuck, it is not equivalent to θ because the latter is the marking of a properly terminated process, while s_7 denotes a deadlock situation. $\square$

3.2 Team Equivalence

Starting from bisimilarity over an unmarked FSM, we can define team equivalence $\circeq$ over its markings in a structural, distributed way.

Definition 8. (Team equivalence) *For any FSM $N = (S, A, T)$, we define team equivalence $\circeq \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$ as the least relation induced by the following axiom and rule.*

$$\frac{}{\theta \mathrel{\mathring{=}} \theta} \qquad \frac{s_1 \sim s_2 \quad m_1 \mathrel{\mathring{=}} m_2}{s_1 \oplus m_1 \mathrel{\mathring{=}} s_2 \oplus m_2}$$

□

In other words, team equivalence $\mathrel{\mathring{=}}$ is the additive closure of $\sim$, i.e., $\mathrel{\mathring{=}} = \sim^{\oplus}$. Hence, team equivalent markings have the same size.

Proposition 4. *For any FSM $N = (S, A, T)$, if $m_1 \mathrel{\mathring{=}} m_2$, then $|m_1| = |m_2|$.* □

This proposition has the pleasing consequence that the proofs about team equivalence can be done by induction on the size of the involved markings, as illustrated in the following proposition stating that $\mathrel{\mathring{=}}$ is an equivalence relation.

Proposition 5. *For any FSM $N = (S, A, T)$, relation $\mathrel{\mathring{=}} \subseteq \mathscr{M}(S) \times \mathscr{M}(S)$ is an equivalence relation.*

Proof. *By induction on the cardinality of the involved markings. The base case is for size 0. The only marking of size 0 is the empty marking θ, which can be related only to itself; hence, on markings of size 0, the relation is an equivalence relation. Now assume that relation $\mathrel{\mathring{=}}$ over markings of size k is an equivalence relation; we will prove that $\mathrel{\mathring{=}}$ is an equivalence relation also on markings of size $k+1$.*

Reflexivity*: $s_1 \oplus m_1 \mathrel{\mathring{=}} s_1 \oplus m_1$ is possible only if $s_1 \sim s_1$ and $m_1 \mathrel{\mathring{=}} m_1$; this is true because $\sim$ is reflexive and m_1 is a marking of smaller size and so, by induction, $\mathrel{\mathring{=}}$ is reflexive on such markings.*

Symmetry*: Assume $s_1 \oplus m_1 \mathrel{\mathring{=}} s_2 \oplus m_2$ due to $s_1 \sim s_2$ and $m_1 \mathrel{\mathring{=}} m_2$; since $\sim$ is symmetric, also $s_2 \sim s_1$; by induction, $m_2 \mathrel{\mathring{=}} m_1$; and so, by rule application in Definition 8, also $s_2 \oplus m_2 \mathrel{\mathring{=}} s_1 \oplus m_1$ holds.*

Transitivity*: Assume $s_1 \oplus m_1 \mathrel{\mathring{=}} s_2 \oplus m_2$ due to $s_1 \sim s_2$ and $m_1 \mathrel{\mathring{=}} m_2$; assume also that $s_2 \oplus m_2 \mathrel{\mathring{=}} s_3 \oplus m_3$ due to $s_2 \sim s_3$ and $m_2 \mathrel{\mathring{=}} m_3$. Since $\sim$ is transitive, then $s_1 \sim s_3$; moreover, by induction, we also have that $m_1 \mathrel{\mathring{=}} m_3$; therefore, by rule application in Definition 8, $s_1 \oplus m_1 \mathrel{\mathring{=}} s_3 \oplus m_3$, as required.* □

The following theorem provides a characterization of team equivalence as a suitable bisimulation-like relation over markings, i.e., over a global model of the overall behavior. It is interesting to observe that this characterization gives a dynamic interpretation of team equivalence, while Definition 8 gives a structural definition of team equivalence.

Theorem 1. *Let $N = (S, A, T)$ be an FSM. Two markings m_1 and m_2 are team equivalent, $m_1 \mathrel{\mathring{=}} m_2$, if and only if $|m_1| = |m_2|$ and*

- *$\forall t_1$ such that $m_1[t_1\rangle m_1'$, $\exists t_2$ such that ${}^\bullet t_1 \sim {}^\bullet t_2$, $l(t_1) = l(t_2)$, $t_1^\bullet \mathrel{\mathring{=}} t_2^\bullet$, $m_2[t_2\rangle m_2'$ and $m_1' \mathrel{\mathring{=}} m_2'$,*
- *$\forall t_2$ such that $m_2[t_2\rangle m_2'$, $\exists t_1$ such that ${}^\bullet t_1 \sim {}^\bullet t_2$, $l(t_1) = l(t_2)$, $t_1^\bullet \mathrel{\mathring{=}} t_2^\bullet$, $m_1[t_1\rangle m_1'$ and $m_1' \mathrel{\mathring{=}} m_2'$.*

Proof. *($\Rightarrow$) If $m_1 \mathrel{\mathring{=}} m_2$, then $|m_1| = |m_2|$ by Proposition 4. Moreover, for any t_1 such that $m_1[t_1\rangle m_1'$, we have that $m_1 = s_1 \oplus \overline{m}_1$, where $s_1 = {}^\bullet t_1$. As $m_1 \mathrel{\mathring{=}} m_2$, by Definition 8, it follows that there exist s_2 and $\overline{m}_2$ such that $m_2 = s_2 \oplus \overline{m}_2$, $s_1 \sim s_2$ and $\overline{m}_1 \mathrel{\mathring{=}} \overline{m}_2$. Since $s_1 \sim s_2$, by Definition 7, there exists a transition t_2 such that ${}^\bullet t_2 = s_2$, $l(t_2) = l(t_1)$ and either $t_1^\bullet = \theta = t_2^\bullet$ or $t_1^\bullet \sim t_2^\bullet$. In the former case, $m_1' = \overline{m}_1$ and $m_2' = \overline{m}_2$, and so $m_1' \mathrel{\mathring{=}} m_2'$ because $\overline{m}_1 \mathrel{\mathring{=}} \overline{m}_2$; in the latter case, $m_1' = t_1^\bullet \oplus \overline{m}_1$ and $m_2' = t_2^\bullet \oplus \overline{m}_2$, and so $m_1' \mathrel{\mathring{=}} m_2'$ by Definition 8. The case when m_2 moves first is symmetric and hence omitted.*

($\Leftarrow$) Let us assume that $|m_1| = |m_2|$ and that the two bisimulation-like conditions hold; then, we prove that $m_1 \mathrel{\mathring{=}} m_2$. First of all, assume that no transition t_1 is enabled at m_1; in such a case, also no transition t_2 can be enabled at m_2; in fact, if $m_2[t_2\rangle m_2'$, then, by the second condition, a transition t_1 must be executable at m_1, contradicting the assumption that no transition is enabled at m_1. If each place in m_1 is a deadlock, and similarly each place in m_2 is a deadlock, then all the places in m_1 and m_2 are pairwise bisimilar; hence, the condition $|m_1| = |m_2|$ is enough to ensure that $m_1 \mathrel{\mathring{=}} m_2$. Now, assume that $m_1[t_1\rangle m_1'$ for some t_1; the first condition ensures that there exists t_2 such that ${}^\bullet t_1 \sim {}^\bullet t_2$, $l(t_1) = l(t_2)$, $t_1^\bullet \mathrel{\mathring{=}} t_2^\bullet$, $m_2[t_2\rangle m_2'$ and $m_1' \mathrel{\mathring{=}} m_2'$. Note that $t_1^\bullet \mathrel{\mathring{=}} t_2^\bullet$ holds if and only if either $t_1^\bullet = \theta = t_2^\bullet$ or $t_1^\bullet \sim t_2^\bullet$. In the former case, we have that $m_1 = {}^\bullet t_1 \oplus m_1'$and $m_2 = {}^\bullet t_2 \oplus m_2'$, and so $m_1 \mathrel{\mathring{=}} m_2$ by Definition 8. In the latter case, we have that $m_1' = t_1^\bullet \oplus \overline{m}_1$, $m_2' = t_2^\bullet \oplus \overline{m}_2$, $m_1 = {}^\bullet t_1 \oplus \overline{m}_1$, $m_2 = {}^\bullet t_2 \oplus \overline{m}_2$. Since $m_1' \mathrel{\mathring{=}} m_2'$ and $t_1^\bullet \sim t_2^\bullet$, it follows that $\overline{m}_1 \mathrel{\mathring{=}} \overline{m}_2$, and so $m_1 \mathrel{\mathring{=}} m_2$, because ${}^\bullet t_1 \sim {}^\bullet t_2$. Symmetrically, if we start from a transition t_2 enabled at m_2. □

Corollary 1. (Strong place bisimilarity and team equivalence coincide) *Let $N = (S, A, T)$ be an FSM. Two markings m_1 and m_2 are team equivalent, $m_1 \mathrel{\mathring{=}} m_2$, if and only if they are strong place bisimilar, $m_1 \sim_p m_2$.*

Proof. *By Theorem 1, it is clear that bisimulation equivalence $\sim$ is a strong place bisimulation, because $\mathring{=} = \sim^\oplus$. Moreover, $\sim$ is the largest strong place bisimulation, hence $\sim^\oplus = \sim_p$. In fact, suppose, towards a contradiction, that there exists a place bisimulation R such that $\sim \subset R$, i.e., R contains a pair (s_1, s_2) such that $s_1 \nsim s_2$. Since $(s_1, s_2) \in R^\oplus$, by Definition 6, these two places are to be bisimilar, which contradicts our assumption.* □

Hence, bisimulation equivalence $\sim$ is the largest strong place bisimulation. Therefore, our characterization of strong place bisimilarity is very appealing because it is based on the basic definition of *local* bisimulation on the places of the unmarked net, and, moreover, offers a very efficient algorithm to check if two markings are strong place bisimilar.

In fact, once $\sim$ is computed in $O(m \log (n+1))$ time (and implemented as an adjacency matrix), then the complexity of checking if two markings m_1 and m_2 (represented as a list of places with multiplicities) are team equivalent is $O(k^2)$, where k is the size of the markings, since the problem is essentially that of finding for each element s_1 of m_1 a matching, bisimilar element s_2 of m_2, as described by the algorithm in Table 1.

Corollary 2. (Team equivalence is finer than interleaving bisimilarity) *Let $N = (S, A, T)$ be an FSM. If $m_1 \mathrel{\mathring{=}} m_2$ then $m_1 \sim_{int} m_2$.*

Proof. *By Theorem 1, it is clear that $\mathring{=}$ is an interleaving bisimulation.* □

Let $N = (S, A, T)$ be an FSM.
Let $\sim \subseteq S \times S$ be the bisimulation equivalence, computed by the algorithm in [31].
Let m_1 and m_2 be two markings on S such that $|m_1| = k = |m_2|$.

1. Let R be the set of currently matched bisimilar places, initialized to $\emptyset$;
2. Let i be an integer variable, initialized to 1;
3. Let b be a boolean variable, initialized to *true*;
4. **while** ($i \leq k$ **and** b) **do**:
   a) select an element s'_i from m_1;
   b) **if** there exists $s''_i \in m_2$ such that $s'_i \sim s''_i$
      then
      1) add (s'_i, s''_i) to R;
      2) $m_1 := m_1 \ominus s'_i$;
      3) $m_2 := m_2 \ominus s''_i$;
      4) $i := i + 1$;
      else $b :=$ *false*;
5. **if** $b =$ *true* **then** R is the set of matched bisimilar places, justifying that m_1 and m_2 are team bisimilar.

Table 1. Algorithm for checking whether two markings of equal size are team equivalent

Example 6. Team equivalence is a truly concurrent equivalence. According to [17], the sequential FSM in Figure 6(a) denotes the net for the sequential CCS [25, 16] process term $a.b.\mathbf{0} + b.a.\mathbf{0}$, which can perform the two actions a and b in either order. On the contrary, the concurrent FSM in (b) denotes the net for the parallel CCS process term $a.\mathbf{0}|b.\mathbf{0}$. Note that s_1 is not team equivalent to $s_4 \oplus s_5$, because the two markings have different size. Nonetheless, s_1 and $s_4 \oplus s_5$ are interleaving bisimilar. □

Example 7. If two markings m_1 and m_2 are interleaving bisimilar and have the same size, then they may be not team equivalent. For instance, consider Figure 6(c), which, according to [17], denotes the net for the CCS process term $a.C\,|\,b.\mathbf{0}$, where C is a constant with empty body, i.e., $C \doteq \mathbf{0}$. Markings $s_4 \oplus s_5$ and $s_6 \oplus s_7$ have the same size, they are interleaving bisimilar (actually, they are even fully concurrent bisimilar [7]), but they are not team equivalent. In fact, if $s_6 \oplus s_7 \xrightarrow{a} s_7 \oplus s_8$, then $s_4 \oplus s_5$ may try to respond with $s_4 \oplus s_5 \xrightarrow{a} s_5$, but $s_7 \oplus s_8$ and s_5 are not team bisimilar, as they have different size. □

Example 8. Continuing Example 2 about Figure 2, it is clear that, for instance, $2 \cdot s_1$ is team bisimilar to any marking obtained with a distribution of two tokens over the places of the net in (b); for instance, $2 \cdot s_1 \mathrel{\mathring{=}} s_2 \oplus s_3$ or $2 \cdot s_1 \mathrel{\mathring{=}} 2 \cdot s_4$. Moreover, $3 \cdot s_1 \mathrel{\mathring{=}} s_2 \oplus 2 \cdot s_4$, as well as $s_1 \oplus s_3 \mathrel{\mathring{=}} s_2 \oplus s_4$. □

Example 9. Continuing Example 4 about Figure 4, we have, for instance, that the marking $s_1 \oplus 2 \cdot s_2 \oplus s_3 \oplus s_4$ is team equivalent to any marking with one token in s_5, two

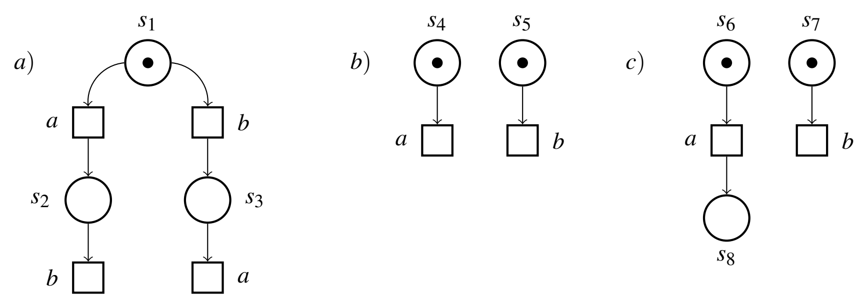

Fig. 6. Three non-team equivalent net systems: $a.b.\mathbf{0} + b.a.\mathbf{0}$, $a.\mathbf{0} \,|\, b.\mathbf{0}$ and $a.C \,|\, b.\mathbf{0}$ (with $C \doteq \mathbf{0}$)

tokens distributed over s_6 and s_7, and two tokens in s_8; e.g., $s_5 \oplus s_6 \oplus s_7 \oplus 2 \cdot s_8$ or $s_5 \oplus 2 \cdot s_7 \oplus 2 \cdot s_8$. Note that $m_1 = s_1 \oplus 2 \cdot s_2$ has the same size of $m_2 = 2 \cdot s_5 \oplus s_6$, but the two are not team equivalent; in fact, the algorithm in Table 1 would first match s_1 with one instance of s_5; then, one instance of s_2 with s_6; but, now, it is unable to find a match for the second instance of s_2, because the only element left in m_2 is s_5, and s_2 is not bisimilar to s_5. □

Example 10. Consider again the net in Figure 5(c). Even if s_7 is stuck, its addition to any marking m makes the new marking $m \oplus s_7$ not team equivalent to m. For instance, $s_6 \oplus s_7 \not\sim^{\oplus} s_6$. This example also shows that team equivalence is sensitive to the size of the marking. □

The examples above and the algorithm in Table 1 make clear that two markings m_1 and m_2 are *not* team bisimilar if they have different size, or if the algorithm in Table 1 ends with b holding *false*, i.e., by singling out a place s'_i in the residual of m_1 which has no matching bisimilar place in the residual of m_2.

3.3 Minimizing Nets

In the theory of deterministic finite automata (DFAs, for short; see, e.g., [21]), two language equivalent states can be merged to obtain a language equivalent, smaller DFA; in fact, it is possible to get the least DFA, whose states are language equivalence classes of the states of the original DFA. Similarly, in the theory of labeled transition systems (LTSs, for short), two bisimilar states can be merged to get a smaller, behaviorally equivalent LTS; in fact, it is possible to get the least LTS, whose states are bisimulation equivalence classes of the states of the original LTS (see, e.g., [16]).

The situation is not very different for finite-state machines, where the bisimulation equivalence relation $\sim$ over places can be used to obtain reduced nets. Also in this case, two bisimilar places can be safely merged; hence, given an FSM N, we can get its behaviorally equivalent, reduced FSM net N', whose places are bisimulation equivalence classes of places of N.

Definition 9. (Reduced net) *Let $N = (S, A, T)$ be an FSM and let $\sim$ be the bisimulation equivalence relation over its places. The* reduced *net $N' = (S', A, T')$ is defined as follows:*

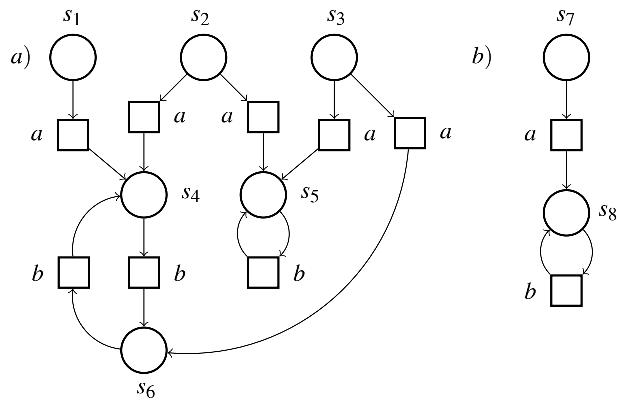

Fig. 7. An FSM in (a) and its reduced net in (b)

- $S' = \{[s]_\sim \mid s \in S\}$, *where* $[s]_\sim = \{s' \in S \mid s \sim s'\}$*;*
- $T' = \{([s]_\sim, a, [m]_\sim) \mid (s,a,m) \in T\}$*,*

where $[m]_\sim$ *is defined as follows:* $[\theta]_\sim = \theta$ *and* $[m_1 \oplus m_2]_\sim = [m_1]_\sim \oplus [m_2]_\sim$*. If the net* N *has initial marking* $m_0 = k_1 \cdot s_1 \oplus \ldots \oplus k_n \cdot s_n$*, then* N' *has initial marking* $[m_0]_\sim = k_1 \cdot [s_1]_\sim \oplus \ldots \oplus k_n \cdot [s_n]_\sim$*.* □

Lemma 1. *Let* $N = (S,A,T)$ *be an FSM and let* $N' = (S',A,T')$ *be its reduced net w.r.t.* $\sim$*. Relation* $R = \{(s,[s]_\sim) \mid s \in S\}$ *is a bisimulation.*

Proof. If $s \xrightarrow{a} m$*, then also* $[s]_\sim \xrightarrow{a} [m]_\sim$ *by definition of* T'*; if* $m = \theta$*, then also* $[\theta]_\sim = \theta$ *and so the bisimulation condition is satisfied; otherwise, if* $m = s'$*, then* $(s',[s']_\sim) \in R$*, as required. The case when* $[s]_\sim$ *moves first is slightly more complex for the freedom in choosing the representative in an equivalence class. Transition* $[s]_\sim \xrightarrow{a} [m]_\sim$ *is possible, by Definition of* T'*, if there exists* $m' \in [m]_\sim$ *such that* $s \xrightarrow{a} m'$*; if* $m = \theta$*, then also* $m' = \theta$ *and the bisimulation condition is satisfied; otherwise, if* $m' = s'$*, then* $m = s''$ *with* $s' \sim s''$*, so that* $[s']_\sim = [s'']_\sim$*; hence,* $(s',[s']_\sim) \in R$*, as required.* □

Theorem 2. *Let* $N = (S,A,T)$ *be an FSM and let* $N' = (S',A,T')$ *be its reduced net w.r.t.* $\sim$*. For every* $m \in \mathscr{M}(S)$*, we have that* $m \mathrel{\dot{=}} [m]_\sim$*.*

Proof. By induction on the size of m. If $m = \theta$*, then* $[m]_\sim = \theta$ *and the thesis follows trivially. If* $m = s \oplus m'$*, then* $[m]_\sim = [s]_\sim \oplus [m']_\sim$*; by Lemma 1,* $s \sim [s]_\sim$ *and, by induction,* $m' \mathrel{\dot{=}} [m']_\sim$*; therefore, by the rule in Definition 8,* $m \mathrel{\dot{=}} [m]_\sim$*.* □

As a consequence of this theorem, we would like to point out that the reduced net w.r.t. $\sim$ is indeed the *least* net offering the same team behavior as the original net: no further fusion of places can be done, as there are not two places in the reduced net which are bisimilar.

Example 11. Continuing Example 2, consider the net in Figure 2(b). Since $\sim = S \times S$, with $S = \{s_2, s_3, s_4\}$, it follows that its reduced net is isomorphic to the net in Figure 2(a), where s_1 stands for $[s_2]_\sim = \{s_2, s_3, s_4\}$. □

Example 12. Continuing Example 4 about Figure 4, we have that $\sim = R \cup R^{-1} \cup \mathscr{I}$, where $R = \{(s_1,s_5),(s_2,s_6),(s_2,s_7),(s_6,s_7),(s_3,s_8),(s_4,s_8),(s_3,s_4)\}$. Hence, the reduced net is composed of three places only, namely $[s_1]_\sim = \{s_1,s_5\}$, $[s_2]_\sim = \{s_2,s_6,s_7\}$ and $[s_3]_\sim = \{s_3,s_4,s_8\}$, and three transitions only, namely $([s_1]_\sim, a, [s_2]_\sim), ([s_2]_\sim, b, [s_3]_\sim)$ and $([s_3]_\sim, c, [s_1]_\sim)$. □

Example 13. Consider the net in Figure 7(a). It is not difficult to realize that the equivalence classes of $\sim$ are $\{s_1,s_2,s_3\}$ and $\{s_4,s_5,s_6\}$. Hence, the reduced net is isomorphic to the net in Figure 7(b). If the initial marking of the net in (a) is $s_1 \oplus s_3$, then the initial marking of the reduced net is $2 \cdot s_7$. Of course, the two initial markings are team equivalent. □

4 Modal Characterization

In this section we propose a new modal logic, called BTML (basic team modal logic), based on a variation of Hennessy-Milner Logic [20, 3], called HMT, which is sensitive to the kind of termination. We will prove that model checking is coherent with equivalence checking: two markings are team bisimilar if and only if they satisfy the same BTML formulae.

4.1 HMT: Hennessy-Milner Logic with Termination

Here we propose a slight variation of Hennessy-Milner Logic [20, 3], which is sensitive to the kind of termination (see Examples 1 and 5). The HMT *formulae* are generated from the finite set A of actions by the following abstract syntax:

$$F \ ::= \ nn \mid vv \mid F \wedge F \mid F \vee F \mid \langle a \rangle F \mid [a]F$$

where a is any action in A, nn and vv denote two atomic propositions (which are not *true* and *false*), $\wedge$ is the operator of logical conjunction, $\vee$ is disjunction, $\langle a \rangle F$ denotes *possibility* (it is possible to do a and then reach a place/marking where F holds), $[a]F$ denotes *necessity* (by doing a, only places/markings where F holds can be reached). We denote by $\mathscr{F}_A$ the set of all HMT formulae, built from the actions in A. The logical values tt for *true* and $f\!f$ for *false* can be defined as derived operators:

$$tt = nn \vee vv \qquad f\!f = nn \wedge vv.$$

We sometimes use some useful abbreviations: if $B = \{a_1, a_2, \ldots, a_k\} \subseteq A$, $k \geq 1$, then $\langle B \rangle F$ stands for $\langle a_1 \rangle F \vee \langle a_2 \rangle F \vee \ldots \vee \langle a_k \rangle F$, and $[B]F$ stands for $[a_1]F \wedge [a_2]F \wedge \ldots \wedge [a_k]F$. In case $B = \emptyset$, then $\langle B \rangle F = f\!f$ and $[B]F = nn$; the reason for the latter will be clear in the following.

The semantics of a formula F is the set of places (including the empty marking θ) that satisfy it; hence, the semantic function is parametrized with respect to some given FSM $N = (S, A, T)$. Let $[\![-]\!]_h : \mathscr{F}_A \to \wp(S \cup \{\theta\})$ be the denotational semantics function, defined in Table 2.

$[\![nn]\!]_h = S$	$[\![vv]\!]_h = \{\theta\}$	$[\![tt]\!]_h = S \cup \{\theta\}$	$[\![ff]\!]_h = \emptyset$
$[\![F_1 \wedge F_2]\!]_h = [\![F_1]\!]_h \cap [\![F_2]\!]_h$	$[\![F_1 \vee F_2]\!]_h = [\![F_1]\!]_h \cup [\![F_2]\!]_h$		
$[\![\langle a \rangle F]\!]_h = \{s \in S \mid \exists m. s \xrightarrow{a} m \text{ and } m \in [\![F]\!]_h\}$			
$[\![[a]F]\!]_h = \{s \in S \mid \forall m (s \xrightarrow{a} m \text{ implies } m \in [\![F]\!]_h)\}$			

Table 2. Denotational semantics

Proposition 6. *Let $N = (S, A, T)$ be an FSM. For every $F \in \mathscr{F}_A$, we have that $[\![F]\!]_h$ is a finite set.*

Proof. *By inspecting the cases in Table 2, $[\![F]\!]_h$ is always a subset of the universe $S \cup \{\theta\}$, which is a finite set.* □

Definition 10. **(HMT satisfaction relation)** *We say that a place s* satisfies *a formula F, written $s \models F$, if $s \in [\![F]\!]_h$; similarly, the empty marking θ satisfies F, written $\theta \models F$, if $\theta \in [\![F]\!]_h$.* □

Of course, this definition offers a trivial model-checking algorithm. In order to check whether a place s (or the empty marking θ) satisfies a formula F, just compute the finite set $[\![F]\!]_h$ and then verify whether s (or θ) belongs to this set.

Let us now comment on the semantic definitions in Table 2. The semantics of *nn* is S: all the places satisfy *nn*, but not the empty marking θ; complementarily, the semantics of *vv* is $\{\theta\}$: only the empty marking satisfies *vv*. Note that the semantics of the logical values *tt* and *ff* respects the expected condition that all the places (and the empty marking) satisfy *tt* , while no place (and not even the empty marking) satisfies *ff*. Indeed,

$$[\![tt]\!]_h = [\![nn \vee vv]\!]_h = [\![nn]\!]_h \cup [\![vv]\!]_h = S \cup \{\theta\}, \text{ and}$$
$$[\![ff]\!]_h = [\![nn \wedge vv]\!]_h = [\![nn]\!]_h \cap [\![vv]\!]_h = S \cap \{\theta\} = \emptyset.$$

The logical operator of conjunction $\_ \wedge \_$ is interpreted as intersection $\_ \cap \_$ of the set of places satisfied by the two subformulae; symmetrically, disjunction is interpreted as set union.

The semantics of $\langle a \rangle F$ is the set of all the places that can perform a and, in doing so, reach a place (or the empty marking) satisfying F. Therefore, the formula $\langle a \rangle nn$ is satisfied by any place able to perform a, reaching another place; while $\langle a \rangle vv$ by any place able to perform a, reaching the empty marking. Of course, $\langle a \rangle tt$ is satisfied by any place able to perform a.

The semantics of $[a]F$ is the set of all the places that, by performing a, can only reach places (or the empty marking) satisfying F. Note that a place s, which is unable to perform a altogether, satisfies $[a]F$, for any F, because the universal quantification in the definition of its semantic is vacuously satisfied; however, the empty marking does

$nn^c = vv$	$vv^c = nn$	$(F_1 \wedge F_2)^c = F_1^c \vee F_2^c$
$(F_1 \vee F_2)^c = F_1^c \wedge F_2^c$	$(\langle a \rangle F)^c = [a](F^c) \vee vv$	$([a]F)^c = \langle a \rangle (F^c) \vee vv$

Table 3. Negation as complement

not satisfy $[a]F$ for any F. Therefore, the formula $[a]nn$ is satisfied by any place that, by performing a, can never reach the empty marking; while $[a]vv$ by any place that, by performing a, can never reach a place; hence, $[a]\mathit{ff}$ is satisfied by any place that cannot perform a. To explain why the auxiliary notation $[B]F$, when $B = \emptyset$, is to be interpreted as nn, we have to point out that the semantics of a box formula $[a]F$ is a set of places (not the empty marking), and so all the places, and only the places, satisfy this formula.

Example 14. Let us consider the net in Figure 4. It is not difficult to realize that formula $F = [a][b]\langle c \rangle nn$ is such that $s_1 \models F$ and $s_5 \models F$. Also formula $F' = \langle a \rangle [b]\mathit{ff}$ is such that $s_1 \not\models F'$ and $s_5 \not\models F'$. □

Example 15. Let us consider the net in Figure 5. It is not difficult to realize that formula $F = [a]\langle b \rangle vv$ is such that $s_1 \models F$ while $s_3 \not\models F$. Moreover, formula $F' = \langle a \rangle vv$ is such that $s_8 \models F'$ while $s_6 \not\models F'$. □

Two formulae F_1 and F_2 are equivalent, denoted by $F_1 \simeq F_2$, if $[\![F_1]\!]_h = [\![F_2]\!]_h$ for all FSMs. Several equalities hold over HMT formulae, a few of which are outlined below.

Proposition 7. *The following equalities over HMT formulae hold:*

1. $F_1 \wedge (F_2 \wedge F_3) \simeq (F_1 \wedge F_2) \wedge F_3$	2. $F_1 \vee F_2 \simeq F_2 \vee F_1$
3. $(F_1 \wedge F_2) \vee F_3 \simeq (F_1 \vee F_3) \wedge (F_2 \vee F_3)$	4. $F \vee \mathit{ff} \simeq F$
5. $F_1 \wedge (F_2 \vee F_3) \simeq (F_1 \wedge F_2) \vee (F_1 \wedge F_3)$	6. $F \wedge \mathit{tt} \simeq F$
7. $\langle a \rangle (F_1 \vee F_2) \simeq \langle a \rangle F_1 \vee \langle a \rangle F_2$	8. $[a](F_1 \wedge F_2) \simeq [a]F_1 \wedge [a]F_2$

□

Negation is not included syntactically in HMT, but semantically we can show, as described in [3] for HML, that any formula F has an associated formula F^c, we call the *complement* of F and defined in Table 3, such that the following holds:

$$[\![F^c]\!]_h = \overline{[\![F]\!]_h} = (S \cup \{\varepsilon\}) \setminus [\![F]\!]_h.$$

Note that $\mathit{tt}^{\,c} = (nn \vee vv)^c = nn^c \wedge vv^c = vv \wedge nn = \mathit{ff}$. Similarly, one can see that $\mathit{ff}^{\,c} = \mathit{tt}$.

We would like to comment on a few of them, in particular, $(\langle a \rangle F)^c = [a](F^c) \vee vv$. Remember that the universe is $S \cup \{\theta\}$ and so the negation of $\langle a \rangle F$ is to be computed w.r.t. this universe. Hence, we have to consider not only those places that, by doing a, can only reach some marking not satisfying F (which is the semantics of $[a](F^c)$), but also the empty marking, which in fact satisfies vv. Similarly, $([a]F)^c = \langle a \rangle (F^c) \vee vv$ because, since the semantics of $[a]F$ is a set of places, its complement contains also the empty marking, which is not included in the semantics of $\langle a \rangle (F^c)$.

We are now ready to prove the basic coherence theorem: two places are bisimilar if and only if they satisfy the same HMT formulae. The proof follows the same steps of the original proof in [20, 3] for HML and LTSs.

Proposition 8. *Let $N = (S, A, T)$ be an FSM. If $s_1 \sim s_2$, then s_1 and s_2 satisfy the same HMT formulae, i.e., $\{F_1 \in \mathscr{F}_A \mid s_1 \models F_1\} = \{F_2 \in \mathscr{F}_A \mid s_2 \models F_2\}$.*

Proof. *Let us assume that $s_1 \sim s_2$. We will prove that, for any $F \in \mathscr{F}_A$, if $s_1 \models F$ then also $s_2 \models F$. This is enough because, by a symmetric argument, we can also prove that if $s_2 \models F$ then also $s_1 \models F$, and so s_1 and s_2 satisfy the same HMT formulae.*

The proof, by induction of the structure of F, proceeds by case analysis on the shape of F; the first two cases are the base cases of the induction.

- *$F = nn$: note that $s_2 \models nn$, because s_2 is a place.*
- *$F = vv$: s_1 cannot satisfy vv, because only θ satisfies it; hence, this case is empty.*
- *$F = F_1 \wedge F_2$: since $s_1 \models F_1 \wedge F_2$, it follows that $s_1 \models F_1$ and $s_1 \models F_2$; by induction, we can assume that also $s_2 \models F_1$ and $s_2 \models F_2$; hence, also $s_2 \models F_1 \wedge F_2$, as required.*
- *$F = F_1 \vee F_2$: analogous to the previous one, hence omitted.*
- *$F = \langle a \rangle F'$: since $s_1 \models \langle a \rangle F'$, there exists a marking m_1 such that $s_1 \xrightarrow{a} m_1$ and $m_1 \models F'$; by definition of $\sim$, there exists a marking m_2 such that $s_2 \xrightarrow{a} m_2$ and either $m_1 = \theta = m_2$ or $m_1 \sim m_2$. In the former case, since $m_1 = m_2$, also $m_2 \models F'$ holds; hence, $s_2 \models \langle a \rangle F'$. In the latter case, since $m_1 \sim m_2$ and $m_1 \models F'$, we can apply induction (because F' is a subformula) and conclude that also $m_2 \models F'$; hence, $s_2 \models \langle a \rangle F'$ also in this case.*
- *$F = [a]F'$: since $s_1 \models [a]F'$, for all m_1 such that $s_1 \xrightarrow{a} m_1$, it follows that $m_1 \models F'$. As $s_1 \sim s_2$, for each m_2 such that $s_2 \xrightarrow{a} m_2$, there exists m_1 such that $s_1 \xrightarrow{a} m_1$ and either $m_2 = \theta = m_1$ or $m_1 \sim m_2$. Now, if $m_2 = \theta = m_1$, then also $m_2 \models F'$; otherwise, since $m_1 \sim m_2$ and $m_1 \models F'$, by induction, it follows also that $m_2 \models F'$. Hence, for all m_2 such that $s_2 \xrightarrow{a} m_2$, we have that $m_2 \models F'$; therefore, $s_2 \models [a]F'$, as required.*

As no other cases are possible, the proof is complete. □

Proposition 9. *Let $N = (S, A, T)$ be an FSM. If s_1 and s_2 satisfy the same HMT formulae, i.e., $\{F_1 \in \mathscr{F}_A \mid s_1 \models F_1\} = \{F_2 \in \mathscr{F}_A \mid s_2 \models F_2\}$, then $s_1 \sim s_2$.*

Proof. *We want to prove that $R = \{(s, s') \mid s$ and s' satisfy the same HMT formulae$\}$ is a bisimulation, hence proving that two places that satisfy the same formulae are bisimilar.*

Assume $(s_1, s_2) \in R$ and $s_1 \xrightarrow{a} m_1$. We will prove that there exists some m_2 such that $s_2 \xrightarrow{a} m_2$ and either $m_2 = \theta = m_1$ or $(m_1, m_2) \in R$. Since R is symmetric, this is enough for proving that R is a bisimulation.

Assume, towards a contradiction, that there exists no m_2 such that $s_2 \xrightarrow{a} m_2$ and either $m_2 = \theta = m_1$ or $(m_1, m_2) \in R$. In the former case, if $m_1 = \theta$ and $s_2 \not\xrightarrow{a} \theta$, then the formula $\langle a \rangle vv$ is satisfied by s_1 but not by s_2, contradicting the previous assumption that s_1 and s_2 satisfy the same formulae.

In the latter case, since the net is finite, the set $\{m \in S \mid s_2 \xrightarrow{a} m\}$ is finite; let us denote such a set by $\{m'_1, m'_2, \ldots, m'_k\}$, where $k \geq 0$. By assumption, for $j = 1, \ldots, k$, *none of the m'_j is such that $(m_1, m'_j) \in R$, and so none of them satisfies the same formulae of m_1. Therefore, for each $j = 1, \dots, k$, there exists a formula F_j such that $m_1 \models F_j$ while $m'_j \not\models F_j$. Hence, the formula*

$$F = \langle a \rangle (F_1 \wedge F_2 \wedge \dots \wedge F_k)$$

is such that $s_1 \models F$ while $s_2 \not\models F$, contradicting also in this case the previous assumption that s_1 and s_2 satisfy the same formulae. (In case $k = 0$, $F = \langle a \rangle tt$.) □

Theorem 3. (Basic coherence) *Let $N = (S, A, T)$ be a FSM and let $s_1, s_2 \in S$. It holds that $s_1 \sim s_2$ if and only if $\{F_1 \in \mathscr{F}_A \mid s_1 \models F_1\} = \{F_2 \in \mathscr{F}_A \mid s_2 \models F_2\}$.*

Proof. *Direct consequence of Propositions 8 and 9.* □

4.2 BTML: Basic Team Modal Logic

The BTML *formulae* are generated from the HMT formulae, built from the finite set A of actions, by the following abstract syntax:

$$G \ ::= \ F \mid G \otimes G$$

where F is an HMT formula in $\mathscr{F}_A$ and $\_ \otimes \_$ is the parallel composition of formulae. We denote by $\mathscr{G}_A$ the set of all BTML formulae.

Given an FSM $N = (S, A, T)$, let $[\![-]\!]_t : \mathscr{G}_A \to \mathscr{P}(\mathscr{M}(S))$ be the denotational semantics function, defined as follows:

$$[\![F]\!]_t \ = \ [\![F]\!]_h \qquad [\![G_1 \otimes G_2]\!]_t \ = \ [\![G_1]\!]_t \otimes [\![G_2]\!]_t$$

where $M_1 \otimes M_2 = \{m_1 \oplus m_2 \mid m_1 \in M_1, m_2 \in M_2\}$.

Proposition 10. *Let $N = (S, A, T)$ be an FSM. For every $G \in \mathscr{G}_A$, we have that $[\![G]\!]_t$ is a finite set of markings.*

Proof. *By induction on the structure of G. The base case is when G is actually an HMT formula; in such a case, Proposition 6 ensures that its semantics is a finite subset of $S \cup \{\theta\}$. If $G = G_1 \otimes G_2$, then by induction, we can assume that $[\![G_1]\!]_t$ and $[\![G_2]\!]_t$ are finite sets of size k_1 and k_2, respectively; the thesis follows by observing that the size of $[\![G_1]\!]_t \otimes [\![G_2]\!]_t$ is less than, or equal to, $k_1 \times k_2$.* □

Definition 11. (BTML satisfaction relation) *We say that a marking m satisfies formula G, written $m \models G$, if $m \in [\![G]\!]_t$.* □

Of course, this definition offers a trivial model-checking algorithm. In order to check whether a marking m satisfies a formula G, just compute the finite set $[\![G]\!]_t$ and then verify whether m belongs to this set.

Two formulae G_1 and G_2 are equivalent, denoted by $G_1 \simeq G_2$, if $[\![G_1]\!]_t = [\![G_2]\!]_t$ for all FSMs. Some equalities hold over BTML formulae, as outlined below.

Proposition 11. *The following equalities over BTML formulae hold:*

1. $G_1 \otimes (G_2 \otimes G_3) \simeq (G_1 \otimes G_2) \otimes G_3$,
2. $G_1 \otimes G_2 \simeq G_2 \otimes G_1$,
3. $G \otimes vv \simeq G$.

□

The equalities above allows us to consider a restricted class of formulae, where occurrences of the subformula *vv* are always absorbed, and that are right associative, so that any formula $G_1 \otimes G_2$, which is not equivalent to *vv*, can be manipulated (by exploiting associativity and commutativity of $\otimes$ and *vv* absorption) to an equivalent form as $F \otimes G$, where F is an HMT formula. This is implicitly assumed in the proof of Proposition 12.

Example 16. Let us consider the FSM in Figure 4 and the formula $F_1 \otimes F_2$, where $F_1 = [a][b]\langle c\rangle nn$ and $F_2 = [\{a,c\}]ff$. The marking $s_1 \oplus s_2$ satisfies $F_1 \otimes F_2$ because $s_1 \models F_1$ and $s_2 \models F_2$. The semantics of $F_1 \otimes F_2$ is given by $\{s_1, s_5\} \otimes \{s_2, s_6, s_7\} = \{s_1 \oplus s_2, s_1 \oplus s_6, s_1 \oplus s_7, s_5 \oplus s_2, s_5 \oplus s_6, s_5 \oplus s_7\}$. □

Example 17. Let us consider the FSM in Figure 7 and the formula $F \otimes F \otimes F$, where $F = [a][b]\langle b\rangle nn$. The marking $s_1 \oplus s_2 \oplus s_3$ satisfies $F \otimes F \otimes F$ because $s_1 \models F$, $s_2 \models F$ and $s_3 \models F$. The semantics of $F \otimes F \otimes F$ is given by $\{s_1, s_2, s_3, s_7\} \otimes \{s_1, s_2, s_3, s_7\} \otimes \{s_1, s_2, s_3, s_7\}$. For instance, $s_2 \oplus 2 \cdot s_7 \models F \otimes F \otimes F$. □

Proposition 12. *Let $N = (S, A, T)$ be a FSM. If $m_1 \mathrel{\mathring{=}} m_2$, then m_1 and m_2 satisfy the same BTML formulae, i.e., $\{G_1 \in \mathscr{G}_A \mid m_1 \models G_1\} = \{G_2 \in \mathscr{G}_A \mid m_2 \models G_2\}$.*

Proof. *By induction on the size k of the involved, team equivalent markings.*

If $k = 0$, then $m_1 = \theta = m_2$ and so the two markings satisfy the same formulae.

If $k = n+1$, then, by Definition 8, $m_1 = s_1 \oplus m'_1$, $m_2 = s_2 \oplus m'_2$ such that $s_1 \sim s_2$ and $m'_1 \mathrel{\mathring{=}} m'_2$. Note that, by Definition 11, m_1 satisfies all the BTML formulae $F_1 \otimes G_1$ such that $s_1 \models F_1$ and $m'_1 \models G_1$ (and also all the other formulae equivalent to these according to Proposition 11). Symmetrically for m_2.

By Proposition 8, s_1 and s_2 satisfy the same HMT formulae: $\{F_1 \in \mathscr{F}_A \mid s_1 \models F_1\} = \{F_2 \in \mathscr{F}_A \mid s_2 \models F_2\}$. By induction, we also have that $\{G_1 \in \mathscr{G}_A \mid m'_1 \models G_1\} = \{G_2 \in \mathscr{G}_A \mid m'_2 \models G_2\}$. Hence, the thesis follows easily. □

Proposition 13. *Let $N = (S, A, T)$ be a FSM. If m_1 and m_2 satisfy the same BTML formulae, i.e., $\{G_1 \in \mathscr{G}_A \mid m_1 \models G_1\} = \{G_2 \in \mathscr{G}_A \mid m_2 \models G_2\}$, then $m_1 \mathrel{\mathring{=}} m_2$.*

Proof. *We actually prove the contranominal: if two markings are not team equivalent, then they do not satisfy the same BTML formulae. Two markings are not team equivalent if they have not the same size or if the algorithm in Table 1 fails. In the former case, assume that $|m_1| = k > |m_2|$ for some $k \geq 1$. Then, the BTML formula*

$$nn^k = \underbrace{nn \otimes \ldots \otimes nn}_{k \text{ times}}$$

is such that $m_1 \models nn^k$, while $m_2 \not\models nn^k$, and so m_1 and m_2 do not satisfy the same BTML formulae.

In the latter case, looking at the algorithm in Table 1, let s'_i be the element of the residual of m_1 that has no bisimilar match in the residual of m_2. Assume that $dom(m_2)$ has $k \geq 1$ places which are not bisimilar to s'_i, namely $\{s''_1, \ldots, s''_k\} \subseteq dom(m_2)$. Hence, by (the contranominal of) Proposition 9, for each $s''_j \in m_2$, there is an HMT formula F_j such that $s'_i \models F_j$ and $s''_j \not\models F_j$, for $j = 1, \ldots, k$. Let m'_1 be the marking composed of all *the elements s' in m_1 such that $s' \sim s_i'$; to be precise, any $s \in m_1'$ is such that $s \sim s_i'$, and any $s \in m_1 \ominus m_1'$ is such that $s \nsim s_i'$. Then,*

$$m_1' \models G^h = \underbrace{G \otimes \ldots \otimes G}_{h \textit{ times}},$$

where $G = F_1 \wedge \ldots \wedge F_k$ and $h = |m_1'|$. By Definition 11, also $m_1 \models G^h \otimes nn^l$, where $l = |m_1| - |m_1'|$. On the contrary, $m_2 \not\models G^h \otimes nn^l$ because in m_2 there are less than h elements which are bisimilar to s_i' and any other s_j'' is such that $s_j'' \not\models F_j$ and so $s_j'' \not\models G$. In conclusion, m_1 and m_2 do not satisfy the same BTML formulae. □

Theorem 4. **(Full coherence)** *Let $N = (S, A, T)$ be a FSM and let m_1, m_2 two markings. It holds that $m_1 \mathring{=} m_2$ if and only if m_1 and m_2 satisfy the same BTML formulae, i.e., $\{G_1 \in \mathscr{G}_A \mid m_1 \models G_1\} = \{G_2 \in \mathscr{G}_A \mid m_2 \models G_2\}$.*

Proof. *Direct consequence of Propositions 12 and 13.* □

5 Conclusion, Related Literature and Future Research

Team equivalence is a truly concurrent equivalence which seems the most natural, intuitive and simple extension of LTS bisimulation equivalence to FSMs; it also has a very low complexity, actually the lowest one, to the best of our knowledge, for FMSs. Moreover, it coincides with strong place bisimilarity, proposed by Autant, Belmesk and Schnoebelen in [4, 5], an equivalence relation refining an earlier proposal by Olderog [27], under the name of strong bisimilarity. In [15] van Glabbeek shows that strong place bisimilarity is finer than his *structure-preserving bisimilarity*, an equivalence relation relating markings of the same size only, which is, in turn, finer than *history-preserving bisimilarity* [32], which on nets takes the form of so-called *fully concurrent bisimilarity* [7]. Moreover, in [15] it is proved that structure-preserving bisimilarity (hence also team equivalence, being finer than it) respects the causal semantics of nets: two structure-preserving bisimilar nets have the same same *causal nets* [6, 27].

According to [17], (concurrent) finite-state machines can be represented by a simple CCS [25, 16] subcalculus, called CFM, which comprises the empty process **0**, the action prefixing operator, the choice operator, constants (for recursion) and the asynchronous (i.e., without communication capabilities) parallel operator, to be used at the top level only. We claim that team equivalence is a congruence, i.e., it is preserved by the CFM operators, and so it would be interesting to investigate a (possibly finite) axiomatization of team equivalence over CFM process terms. This axiomatization might be obtained by adding, to the sound and complete set of axioms of strong bisimilarity for finite-state CCS [24] (suitably adapted to consider the peculiar way the empty process **0** is dealt with in [17]), the expected axioms for the parallel operator, stating that it is associative, commutative, with **0** as its identity.

Since CFM is the process algebra representing, up to net isomorphism, all the possible finite-state machines [17], it would also be interesting to compare team equivalence with other non-interleaving equivalences proposed on process algebras, such as *distributed bisimulation* equivalence [11] or *causal bisimuation* equivalence [12] (the latter being equivalent to *history-preserving bisimilarity* [32]). Aceto in [2] shows that these non-interleaving equivalences do coincide on a simple calculus rather similar to CFM. As discussed above, team equivalence coincides with strong place bisimilarity, which is finer than structure-preserving bisimilarity, which is finer than history-preserving bisimilarity; hence, team equivalence is finer than all of these non-interleaving behavioral equivalences. However, team equivalence and history-preserving bisimilarity are close enough, as the difference is essentially in the way they deal with stuck places (i.e., places that do not enable any transition). For instance, if m_1 and m_2 are team equivalent (and so also history-preserving bisimilar) and m is a stuck marking, then m_1 is history-preserving bisimilar to (but not team bisimilar to) $m_2 \oplus m$. This observation may induce to think that if m_1 and m_2 are history-preserving bisimilar and have the same size, then they are team equivalent. This is not true; in fact, Example 7 discusses two CFM processes, whose net semantics is outlined in Figure 6(b) and (c), which are history-preserving bisimilar, have the same size, but are not team equivalent. Nonetheless, we conjecture that a history-preserving (or fully concurrent [7]) bisimulation R on an FSM, with the property that for all pairs $(m_1, m_2) \in R$, m_1 and m_2 have the same size, is actually such that m_1 and m_2 are team equivalent. Hence, we think that team equivalence can be seen as a *resource-aware* strengthening of history-preserving bisimilarity. As a matter of fact, a token in a place of an FSM represents a sequential process, so that one processor is needed to implement it. Therefore, the marking gives a precise information about the number of resources/processors that are needed to implement the system. In this way, an equivalence relation, like team equivalence, which relates markings of the same size only is resource aware, and so more useful from a practical point of view.

Team equivalence is characterized by a very simple and natural modal logic, namely BTML, extending conservatively Hennessy-Milner Logic (HML). BTML parallel composition operator $\_ \otimes \_$ on formulae reminds the spatial operator of Caires' and Cardelli's *spatial logic* [10], also used on spatial transition systems in [1]. More complex modal logics characterizing some non-interleaving equivalences have been proposed in, e.g., [9, 8]. Another possible future work is to extend BTML to become a temporal logic, with least and greatest fixpoint operators, as in Kozen's modal mu-calculus [23].

We are planning to extend team equivalence to FSMs with silent moves [19]. We are also planning to extend our approach to more generous subclasses of Petri nets; in particular, in [18], we are extending this approach to BPP nets (i.e., nets whose transitions have singleton pre-set, but with arbitrary post-set); such extension requires a new definition of bisimulation on places, called *team bisimulation*, and a proper extension of BTML, called TML.

Acknowledgments: The anonymous referees are thanked for their detailed comments and suggestions. Luca Aceto is warmly thanked for his careful proof-reading and for serving as an editor for this paper.

References

1. L. Acciai, M. Boreale, G. Zavattaro, "On the Relationship between Spatial Logics and Behavioral Simulations", in Procs. Fossacs'10, LNCS 6014, 146-160, Springer, 2010.
2. L. Aceto, "Relating Distributed, Temporal and Causal Observations of Simple Processes", *Funda. Info.* 17(4):319-331, 1992.

3. L. Aceto, A. Ingólfsdóttir, K. Larsen, J. Srba, *Reactive Systems: Modelling, Specification and Verification*, Cambridge University Press, 2007.
4. C. Autant, Z. Belmesk, Ph. Schnoebelen, “Strong Bisimilarity on Nets Revisited”, in Procs. PARLE’91, vol. II: Parallel Languages, LNCS 506, 295-312, Springer, 1991.
5. C. Autant, Ph. Schnoebelen, “Place Bisimulations in Petri Nets”, in Procs. Application and Theory of Petri Nets 1992, LNCS 616, 45-61, Springer, 1992.
6. E. Best, R. Devillers, “Sequential and Concurrent Behavior in Petri Net Theory”, *Theoretical Computer Science* 55(1):87-136, 1987.
7. E. Best, R. Devillers, A. Kiehn, L. Pomello, “Concurrent Bisimulations in Petri Nets”, *Acta Inf.* 28(3): 231-264, 1991.
8. P. Baldan, S. Crafa, “A Logic for True Concurrency”, *J. ACM* 61(4): 24:1-24:36, 2014.
9. J.C. Bradfield, S.B. Fröschle, “Independence-Friendly Modal Logic and True Concurrency”. *Nord. J. Comput.* 9(1): 102-117, 2002.
10. L. Caires, L. Cardelli, “A Spatial Logic for Concurrency” (part I). *Inf. Comput.* 186(2): 194-235, 2003.
11. I. Castellani, M. Hennessy, “Distributed Bisimulations”, *J.ACM* 36(4):887-911, 1989.
12. Ph. Darondeau, P. Degano, “Causal Trees”, in Procs. ICALP’89, LNCS 372, 234-248, Springer, 1989.
13. R.J. van Glabbeek, F.W. Vaandrager, “Petri Net Models for Algebraic Theories of Concurrency”, In Proc. PARLE’87, LNCS 259, 224-242, Springer-Verlag, 1987.
14. R.J. van Glabbeek “The Individual and Collective Token Interpretations of Petri Nets”, in Proc. CONCUR 2005, LNCS 3653, 323-337, Springer, 2005.
15. R.J. van Glabbeek, “Structure Preserving Bisimilarity, Supporting an Operational Petri Net Semantics of CCSP”, in [26], 99-130, 2015.
16. R. Gorrieri, C. Versari, *Introduction to Concurrency Theory: Transition Systems and CCS*, EATCS Texts in Theoretical Computer Science, Springer-Verlag, 2015.
17. R. Gorrieri, *Process Algebras for Petri Nets: The Alphabetization of Distributed Systems*, EATCS Monographs in Theoretical Computer Science, Springer-Verlag, 2017.
18. R. Gorrieri, “Team Bisimilarity, and its Associated Modal Logic, for BPP nets”, in preparation, 2017.
19. R. Gorrieri, “Team Equivalences for Finite-State Machines with Silent Moves’, in preparation, 2017.
20. M. Hennessy, R. Milner, “Algebraic Laws for Nondeterminism and Concurrency”, *Journal of the ACM* 32(1):137-161, 1985.
21. J.E. Hopcroft, R. Motwani, J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, 2nd ed., Addison-Wesley, 2001.
22. R. Keller, “Formal Verification of Parallel Programs”, *Comm. of the ACM* 19(7):561-572, 1976.
23. D. Kozen, “Results on the Propositional mu-Calculus”, *Theor. Computer Science* 27:333-354, 1983.
24. R. Milner. “A Complete Inference Systems for a Class of Regular Behaviors”, *J. Comput. System Sci.* 28: 439-466, 1984.
25. R. Milner. *Communication and Concurrency*, Prentice-Hall, 1989.
26. R. Meyer, A. Platzer, H. Wehrheim (Eds.), *Correct System Design* — Symposium in Honor of Ernst-Rüdiger Olderog on the Occasion of His 60th Birthday, LNCS 9360, Springer, 2015.
27. E.R. Olderog, *Nets, Terms and Formulas*, Cambridge Tracts in Theoretical Computer Science 23, Cambridge University Press, 1991.
28. D.M.R. Park, “Concurrency and Automata on Infinite Sequences”, In Proc. 5th GI-Conference on Theoretical Computer Science, LNCS 104, 167-183, Springer-Verlag, 1981.
29. J.L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, 1981.

30. L. Pomello, G. Rozenberg, C. Simone, "A Survey of Equivalence Notions for Net Based Systems", in *Advances in Petri Nets: The DEMON Project*, LNCS 609, 410-472, Springer-Verlag, 1992.
31. R. Paige, R.E. Tarjan, "Three Partition Refinement Algorithms", *SIAM Journal of Computing* 16(6):973-989, 1987.
32. A. Rabinovich, B.A. Trakhtenbrot, "Behavior Structures and Nets", *Fundamenta Informaticae* 11(4):357-404, 1988.
33. W. Reisig, *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*, Springer-Verlag, 2013.
34. D. Sangiorgi, *An Introduction to Bisimulation and Coinduction*, Cambridge University Press, 2012.