A Locally Sequential Globally Asynchronous Net from Maximality-Based Labelled Transition System

Adel Benamira and Djamel-Eddine Saidouni

Abstract Given a maximality-based labelled transition system, in this paper we show that such system can be decomposed and considered as distributed components, where each component is a sequential behaviour. In a distributed context, the synchronisation between components is interpreted as an asynchronous interaction. Hence, sequential maximality-based labelled transition systems are represented as locally sequential globally asynchronous nets.

Keywords Maximality semantics · Bisimulation relation · Distributed systems · Petri nets · LSGA nets

1 Introduction

In [6, 7], distributed systems have been defined as a system which consists of sequential components that reside on different locations. These components evolve concurrently. The interactions between components are asynchronous communications.

Nowadays formal methods are frequently used in different areas during the development of concurrent applications. Their use allows the verification of application properties before their implementation. In general verification processes are based on centralized algorithms. However these applications may be implemented on a distributed system where the synchronization between the different components are implemented as asynchronous communication. Hence the following questions have been emerged: which specifications may be implemented on a distributed system ?

A. Benamira (✉) · D.-E. Saidouni
MISC Laboratory, Abdelhamid Mehri University, 25000 Constantine, Algeria
e-mail: benamira.adel@univ-guelma.dz

D.-E. Saidouni
e-mail: djamel.saidouni@univ-constantine2.dz

A. Benamira
Computer Science Department, University of 08 May 45, 24000 Guelma, Algeria

S. Chikhi et al. (eds.), *Modelling and Implementation of Complex Systems*, Lecture Notes in Networks and Systems 1,
DOI 10.1007/978-3-319-33410-3_22

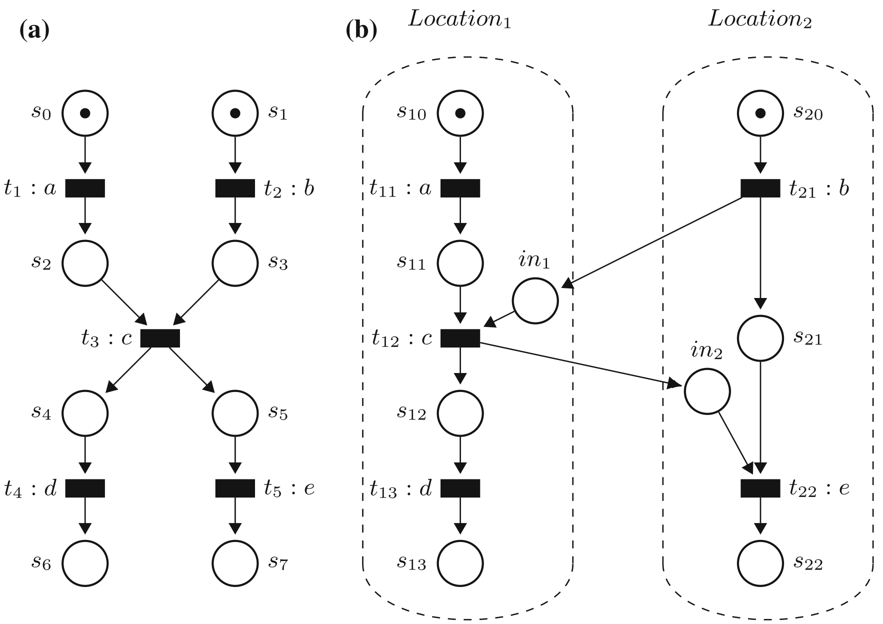


Fig. 1 LSGA net of Petri net. **a** Petri net N. **b** LSGA net of N

and what is the suitable equivalence relation to compare the behaviour of a centralised applications with their distributed implementations?

In the Petri nets framework, Glabbeek et al. [6, 7] gave a precise characterisation of distributable nets and their definition by corresponding class of Petri nets, called LSGA nets (Locally Sequential Globally Asynchronous nets). The ST-bisimulation relation has been proved the suitable equivalence relation between the Petri net specification and their LSGA nets [5, 11, 12].

Figure 1 gives an example of a Petri net with one among its distributed implementations.

Remark that the proposed result is closed to Petri nets model, the use of another specification model requires the definition of a new approach (see Fig. 2) for the generation of distributed implementations (LSGA) from a given specifications. To generalize the result to any input specification model we define a distributed implementation from a semantics[1] model rather than a specification model.

The ST-semantics is originally defined in [11] over Petri nets. In this semantics, non atomic actions are split into starts and ends sub actions. In the literature, the ST-semantics has been applied to process algebras [1, 8]. Another concurrency semantics model, named Maximality-based Labeled Transition System (MLTS), has been defined and used for expressing the semantics of process algebras and P/T Petri nets

[1]Which compatible to the ST-idea, indeed the validation of a distributed implementation is based on the ST-bisimulation.

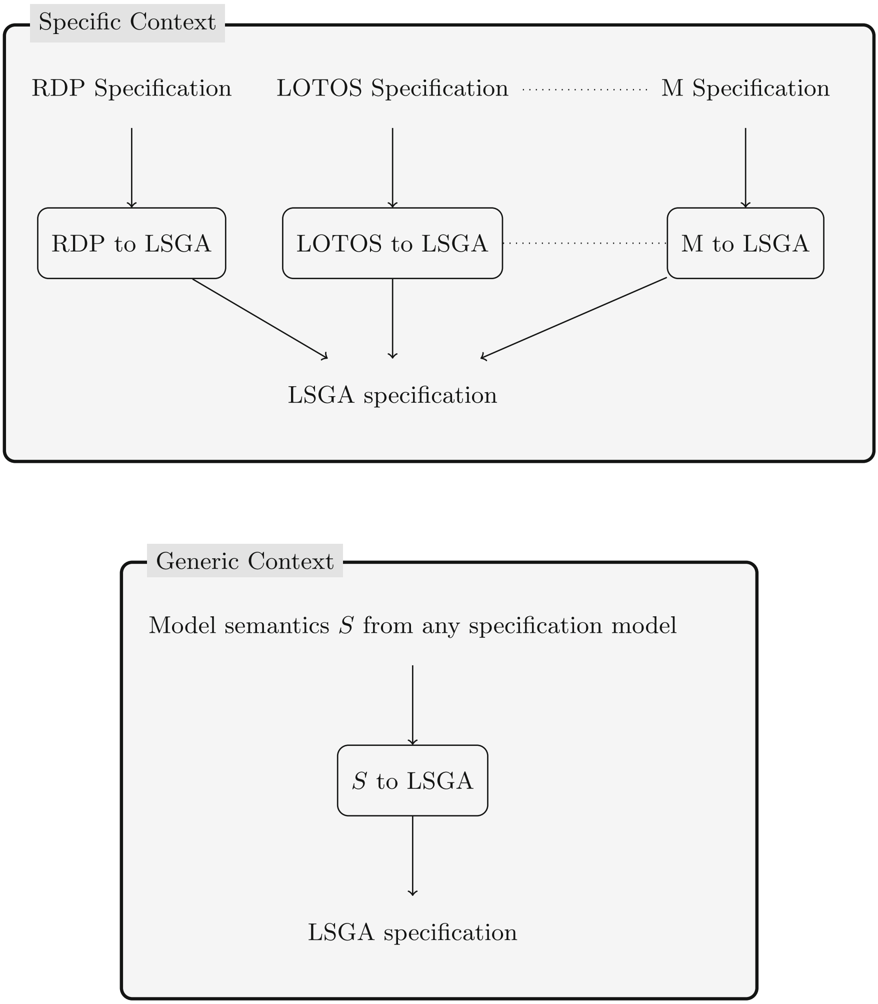
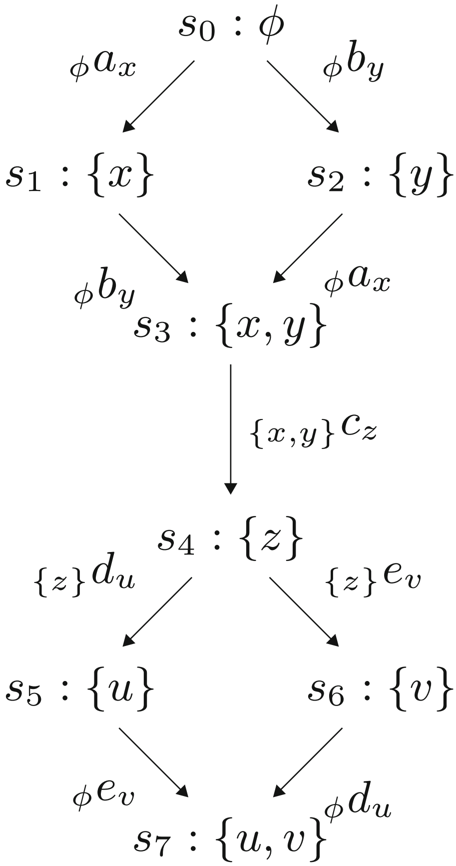


Fig. 2 Specific and generic generation of LSGA specification

with the hypothesis that actions are not necessarily atomic [2–4, 9, 10], i.e. actions are abstractions of finite processes and may elapse in time. The main interest of maximality-based labelled transition system model is that it can be implemented and used for verifying correctness properties without splitting actions into starts and ends sub actions. In this paper, we describe how a MLTS may be seen as a distributed components (sub-MLTS) where synchronizations between components are asynchronous as for LSGA.

Consider the Petri net of Fig. 1a. By applying the approach of [9], the corresponding maximality-based labelled transition system of this Petri net is given by Fig. 3.

Fig. 3 A MLTS



At first, from [4] we can recall that a maximality-based labelled transition system is given by a graph labelled on both states and transitions. Each state is labelled by a set of event names. Each event name identifies the start of execution of an action (eventually under execution) which occurred before this state. This action is said to be potentially under execution in this state. A transition between two states s_i and s_j is labelled by a 3-uple (M, a, x) (denoted $_M a_x$) where x is the event name identifying the start of execution of the action a and M denotes the set of event names representing some causes of the action a. Elements of M belong to state s_i. Occurrence of this transition terminates actions identified by M, thus, the set of event names corresponding to state s_j is that of s_i from which we subtract the set M and add the event name x. Formal definition of a maximality-based labelled transition system will be given in Sect. 2.2.

In the initial state (state s_0) of the maximality-based transition system of Fig. 3, no action is running, from where the association of the empty set with this state. From state s_0, actions a and b can start their execution independently, their starts are respectively identified by event names x and y. a and b can be launched in any order.

The set $\{x\}$ (resp. $\{y\}$) in state s_1 (resp. s_2) stipulates that the action a (resp. b) are potentially under execution in this state. The set $\{x, y\}$ in s_3 shows that actions a and b can be executed simultaneously.

Note that when the system is in state s_1, while the action a has not been terminated yet, the only evolution concerns the start of b. However,when a and b terminate, we can start the action c caused by a and b since the action c which is dependent from the end of a and b. When c terminates, we can start the action d or e. Resulting states are respectively s_5 and s_6. We can observe that from state s_5(resp. s_6), the start of e(resp. d) is always possible. The set $\{u, v\}$ in s_7 shows that actions d and e can be executed simultaneously.

We proceed by defining basic notions of LSGA nets and MLTSs in Sect. 2. In Sect. 3, we show how to decompose a MLTS in set of sequential components such that their interaction defines the initial MLTS, from which we have a direct transformation to LSGA net. This paper is ended by some conclusions of this work.

2 Preliminaries

2.1 Distributed Systems

From [6, 7], a distributed system is defined as follow:

- A distributed system consists of components residing on different locations.
- Components work concurrently.
- Components only allow sequential behaviour.
- Interactions between components are only possible by explicit communications.
- Communication between components is time consuming and asynchronous.

Asynchronous communication is the only interaction mechanism in a distributed system for exchanging signals or information.

- The sending of a message happens always strictly before its receipt (there is a causal relation between sending and receiving a message).
- A sending component sends without regarding the state of the receiver; in particular there is no need to synchronise with a receiving component. After sending the sender continues its behaviour independently of receipt of the message.

In the next, the formal definition of distributed systems in terms of Petri nets [6, 7] is introduced with given the precise characterisation of distributed Petri net.

Definition 1 A (labelled, marked) Petri net is a tuple $N = (S, T, F, I, L)$ where:

- S and T are disjoint sets (of places and transitions),
- $F : (S \times T \cup T \times S) \rightarrow \mathbb{N}$ (the flow relation including arc weights),
- $I : S \rightarrow \mathbb{N}$ (the initial marking), and
- $L : T \longrightarrow A$, for A a set of actions, the labelling function.

Definition 2 A multiset over a set S is a function $M : S \longrightarrow \mathbb{N}$, i.e. $M \in \mathbb{N}^S$. For multisets M and N over S write $M \leq N$ if $M(s) \leq N(s)$ for all $s \in S$. $M + N \in \mathbb{N}^S$ is the multiset with $(M + N)(s) = M(s) + N(s)$, and $M - N$ is the function given by $(M - N)(s) = M(s) - N(s)$ (it is not always a multiset). The function $0 : S \longrightarrow \mathbb{N}$ given by $0(s) = 0$ for all $s \in S$ is the empty multiset. A multiset $M \in \mathbb{N}^S$ with $M(s) \leq 1$ for all $s \in S$ is identified with the set $\{s \in S | M(s) = 1\}$. A multiset M over S is finite if $\{s \in S | M(s) > 1\}$ is finite. Let $\mathcal{M}(S)$ denote the collection of finite multisets over S.

Definition 3 For a finite multiset $U : T \longrightarrow \mathbb{N}$ of transitions in a Petri net, let ${}^{\bullet}U, U^{\bullet} : S \longrightarrow \mathbb{N}$ be the multisets of input and output places of U, given by ${}^{\bullet}U(s) = \Sigma_{t \in T} F(s, t).U(t)$ and $U^{\bullet}(s) = \Sigma_{t \in T} U(t).F(t, s)$ for all $s \in S$.

U is enabled under a marking M if ${}^{\bullet}U \leq M$. In that case U can fire under M, yielding the marking $M' = M -{}^{\bullet} U + U^{\bullet}$, written $M \overset{U}{\longrightarrow} M'$ or $M[U\rangle M'$.

Definition 4 The concurrency relation $\smile \subseteq T^2$ is given by $t \smile u \Leftrightarrow \exists M \in [M_0\rangle.M [\{t\}[\,]\{u\}\rangle$ such that $[\,]$ is a conflict relation. N is a structural conflict net iff for all $t, u \in T$ with $t \smile u$ we have ${}^{\bullet}t \cap^{\bullet} u = \phi$.

For example, the net of Fig. 4 [6, 7], has not a structural conflict net because $[\,] = \{(t_1, t_2); (t_2, t_3)\}$. In the other hand, the net of Fig. 1a is it ($[\,] = \phi$).

A distributed Petri net is a Petri net in which a transition and all its input places reside on the same location and location actions can only occur sequentially. The function $D : S \cup T \rightarrow Loc$ (Loc a set of locations) is defined to associate localities to the elements of a net.

The system of Fig. 5 is a distributed Petri net with $Loc = \{1, 2\}$ and $D = \{(s_1, 1), (t_1, 1), (s_3, 1), (s_4, 1), (t_3, 1), (s_6, 1), (s_2, 2), (t_2, 2), (s_5, 2), (s_7, 2), (t_4, 2), (s_8, 2)\}$.

Definition 5 A Petri net $N = (S, T, F, I, L)$ is distributed iff there exists a distribution D such that:

1. $\forall s \in S, t \in T.s \in^{\bullet} t \Rightarrow D(t) = D(s)$,
2. $\forall t, u \in T.t \smile u \Rightarrow D(t) \neq D(u)$.

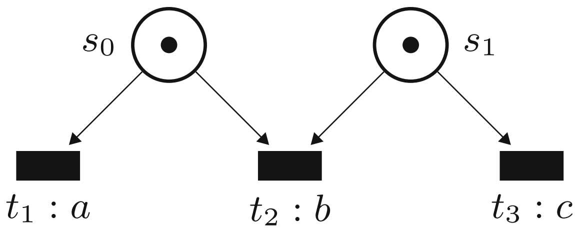


Fig. 4 N has not a structural conflict

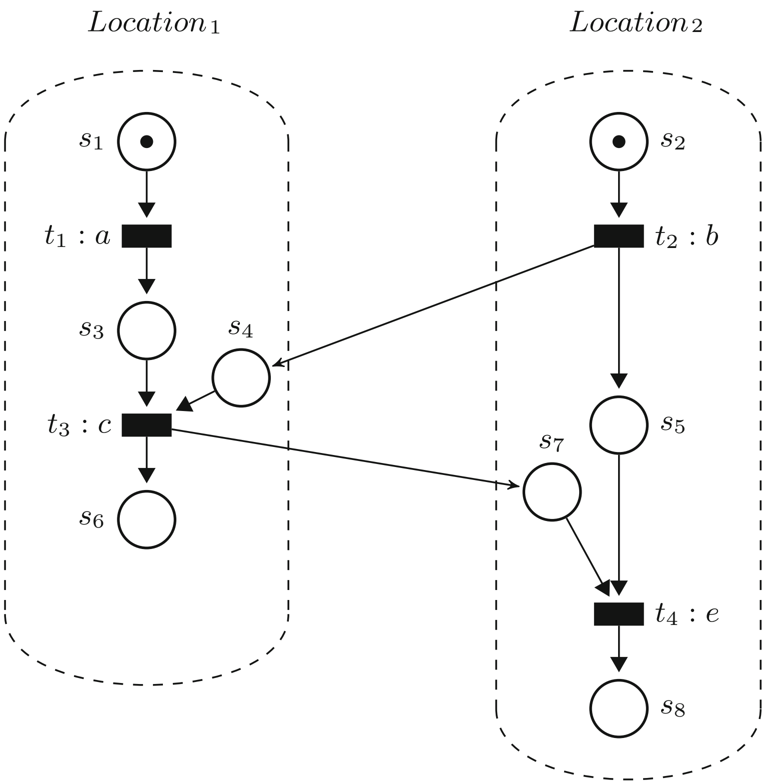


Fig. 5 Distributed Petri net

Proposition 1 *Every distributed Petri net is a structural conflict net.*

Definition 6 Let $N = (S, T, F, M_0, L)$ be a net, $I, O \subseteq S$, $I \cap O = \phi$ and $O^\bullet = \phi$.

1. (N, I, O) is a component with interface (I, O).
2. (N, I, O) is a sequential component with interface (I, O) iff $\exists Q \subseteq S \setminus (I \cup O)$ with $\forall t \in T.|{}^\bullet t \restriction Q| = 1 \wedge |t^\bullet \restriction Q| = 1$ and $|M_0 \restriction Q| = 1$. $A \restriction Y$ denotes the signed multiset over Y defined by $(A \restriction Y)(x) = A(x)$ for all $x \in Y$.

$C = (N, I, O)$ can be regarded as a component of distributed system equipped with a mailbox I and an address O outside C, the first is introduced to receive messages and the second to send messages.

Definition 7 Let $\mathfrak{K}$ be an index set.

Let $((S_k, T_k, F_k, M_{0K}, L_k), I_k, O_k)$ with $k \in \mathfrak{K}$ be components with interface such that $(S_k \cup T_k) \cap (S_l \cup T_l) = (I_k \cup O_k) \cap (I_l \cup O_l)$ for all $k, l \in \mathfrak{K}$ with $k \neq l$ and $I_k \cap I_l = \phi$ for all $k, l \in \mathfrak{K}$ with $k \neq l$. Then the asynchronous parallel of these components is defined by

$$\|_{k\in\mathfrak{K}} ((S_k, T_k, F_k, M_{0K}, L_k), I_k, O_k) = ((S, T, F, M_0, L), I, O)$$

with $S = \cup_{k\in\mathfrak{K}} S_k, T = \cup_{k\in\mathfrak{K}} T_k, F = \cup_{k\in\mathfrak{K}} F_k, M_0 = \cup_{k\in\mathfrak{K}} M_{0k}, l = \cup_{k\in\mathfrak{K}} l_k, I = \cup_{k\in\mathfrak{K}} I_k$ and $O = \cup_{k\in\mathfrak{K}} O_k \ \cup_{k\in\mathfrak{K}} I_k$.

Definition 8 A Petri net N is an LSGA net iff there exists an index set $\mathfrak{K}$ and sequential components with interface $C_k, k \in \mathfrak{K}$, such that $(N, I, O) = \|_{i\in\mathfrak{K}} C_k$ for some I and O.

We can see that the net of Fig. 5 as an LSGA net with two sequential components C_1 and C_2 such that $C_1 = (N_1, \{s_4\}, \{s_7\})$ and $C_2 = (N_2, \{s_7\}, \{s_4\})$ with:

- $N_1 = (S_1, T_1, MS_1, LT_1)$ such that $S_1 = \{s_1, s_3, s_4, s_6, s_7\}$ and $T_1 = \{t_1, t_3\}$.
- $N_2 = (S_2, T_2, MS_2, LT_2)$ such that $S_2 = \{s_2, s_4, s_5, s_7, s_8\}$ and $T_2 = \{t_2, t_4\}$.

From [7], every LSGA net is distributed net and every LSGA net is a structural conflict net.

2.2 Maximality-Based Labeled Transition Systems

A maximality-based labelled transition system is given by a graph labelled on both states and transitions. Each state is labelled by a set of event names. Each event name identifies the start of execution of an action (eventually under execution) which occurred before this state. This action is said to be potentially under execution in this state. A transition between two states s_i and s_j is labelled by a 3-uple (M, a, x) (denoted ${}_Ma_x$) where x is the event name identifying the start of execution of the action a and M denotes the set of event names representing some causes of the action a. Elements of M belong to state s_i. Occurrence of this transition terminates actions identified by M, thus, the set of event names corresponding to state s_j is that of s_i from which we subtract the set M and add the event name x.

Definition 9 Let $\mathcal{M}$ be a countable set of event names, a maximality-based labeled transition system of support $\mathcal{M}$ is a tuple $(\Omega, \lambda, \mu, \xi, \psi)$ with:

(a)
- $\Omega = (S, T, \alpha, \beta, s_0)$ is a transition system such that:
  - S is the set of states in which the system can be found, this set can be finite or infinite.
  - T is the set of transitions indicating state switch that the system can achieve, this set can be finite or infinite.
  - α and β are two applications of T in S such that for all transition t we have: $\alpha(t)$ is the origin of the transition and $\beta(t)$ its goal.
  - s_0 is the initial state of the transition system Ω.
- (Ω, λ) is a transition system labeled by the function λ on an alphabet Act called support of (Ω, λ). In the other words $\lambda : T \to Act$.

- $\psi : S \rightarrow 2^{\mathcal{M}}$ is a function which associates to each state the finite set of maximal event names present in this state.[2]
- $\mu : T \rightarrow 2^{\mathcal{M}}$ is a function which associates to each transition the finite set of event names corresponding to actions that have already begun their execution and the end of their executions enables this transition.
- $\xi : T \rightarrow \mathcal{M}$ is a function which associates to each transition the event name identifying its occurrence.

(b) such that $\psi(s_0) = \phi$ and for all transition t, $\mu(t) \subseteq \psi(\alpha(t))$, $\xi(t) \notin \psi(\alpha(t)) - \mu(t)$ and $\psi(\beta(t)) = (\psi(\alpha(t)) - \mu(t)) \cup \xi(t)$

In what follows, we use the following assumptions:

- In this present paper we suppose the uniqueness of event name.
- Let $mlts = (\Omega, \lambda, \mu, \xi, \psi)$ a maximality-based labeled transition system such that $\Omega = \langle S, T, \alpha, \beta, s_0 \rangle$. $t \in T$ is a transition for which $\alpha(t) = s$, $\beta(t) = s'$, $\lambda(t) = a$, $\mu(t) = E$ and $\xi(t) = x$. The transition t will be noted $s \xrightarrow{{}_E a_x} s'$.
- The set of Maximality-based labelled transition systems is noted $\mathfrak{Mlts}$.

3 LSGA Net from MLTS

In this section, we assume a given $mlts = (\Omega, \lambda, \mu, \xi, \psi)$ to be a maximality-based labelled transition system over $\mathcal{M}$ such that $\Omega = \langle S, T, \alpha, \beta, s_0 \rangle$.

Firstly, we define a partition of $\mathcal{M}$ such that the only interaction between theirs elements is the synchronous interaction and each element represents a sequential behaviour.[3] In the other words, we decompose $mlts$ into a set of sequential MLTSs so that their parallel composition is a MLTS that is the initial MLTS.

Secondly, for each sequential MLTS we define a component net with interface. The asynchronous parallel composition of the all component nets, which associated to the initial MLTS, defines a LSGA net.

3.1 Generation of Sequential MLTSs Set

In the following, we define two fundamental relations with which the $\mathcal{M}$ is structured in the way that the sequential behaviour is clearly deduced from global behaviour.

Definition 10

- The direct causality relation $\leq \subseteq \mathcal{M}^2$ is given by $x \leq y$ if and only if $\exists s \xrightarrow{{}_E a_y} s'$ such that $x \in E$.
- the independence relation $\| \subseteq \mathcal{M}^2$ is given by $x \| y$ if and only if $\exists s \in S$ such that $x, y \in \psi(s)$.

[2] $2^{\mathcal{M}}$ denotes the part sets of $\mathcal{M}$.

[3] Which equivalent to the notion of a sequential component of distributed system.

We note that the relation $\leq$ is not transitive: let $s \xrightarrow{E_1 a_x} s_1 \cdots \xrightarrow{E_p a_y} s_p \cdots \xrightarrow{E_n a_z} s_n$. The assertion $(x \in E_p) \wedge (x \leq y) \wedge (x, y \in E_n) \wedge (y \leq z)$ does never satisfied indeed $x \notin \psi(s_p)$ (see Definition 9(b)), hence $x \leq y \wedge y \leq z \not\Rightarrow x \leq z$.

The conflict relation $[\,] \subseteq \mathcal{M}^2$ is given by $x[]y$ if and only if $x \nleq y$ and $y \nleq x$ and $x \not\parallel y$, in the other words, the conflict has been deduced from basic relations $\leq$ and $\|$. We tell that $(\mathcal{M}, \leq, \|)$ is a set of events $\mathcal{M}$ which is structured by $\leq$ and $\|$.

mlts is a structural conflict if and only if $\nexists x, y \in \mathcal{M} : x[\,]y$. Throughout the rest of this section, we restrict[4] our study to structural conflict MLTS.

Next, we define a set of concepts with which we characterise the structure $(\mathcal{M}, \leq, \|)$.

Let $x \in \mathcal{M}$, event x is a synchronous point if and only if:

- $\exists t : s \xrightarrow{E a_x} s' \in T$ such that $|E| \geq 2$ or
- $|\{s \mid \forall t : s' \xrightarrow{E a_y} s \in T \wedge x \in E\}| \geq 2$.

Let $\mathbb{S} : \mathcal{M} \longrightarrow 2^{\mathcal{M}}$, the notation $\mathbb{S}_{\mathcal{M}}$ is the set of all synchronous and branch-out points in $\mathcal{M}$.

In the system of Fig. 3, the event z is both synchronous and branch-out point, the event z is a synchronous point indeed the action c is dependent to the end of a and b, and it is a branch-out point as the end of c causes the execution of d or e.

Let $\sigma \subseteq \mathcal{M}$ such that $x_1 \leq x_2 \leq \cdots \leq x_n$, σ is a sequence if and only if $\sigma \cap \mathbb{S}_{\mathcal{M}} = \phi$. Let $\sigma_1 = \{x_{11}, x_{12}, \ldots, x_{1n}\}$ and $\sigma_2 = \{x_{21}, x_{22}, \ldots, x_{2m}\}$ two not empty sequences, the order $\sigma_1 \leq \sigma_2$ is defined if and only if $x_{1n} \leq x_{21}$.

The sequence σ is a full sequence if and only if $\exists y \notin \mathbb{S}_{\mathcal{M}} \wedge \exists x \in \sigma$ such that $x \leq y$ or $y \leq x$ then $y \in \sigma$. Let $\mathbb{F} : \mathcal{M} \longrightarrow 2^{\mathcal{M} \times \mathcal{M}}$, the notation $\mathbb{F}_{\mathcal{M}}$ is the set of all full sequences in $\mathcal{M}$.

Let $\sigma_1 = \{x_{11}, x_{12}, \ldots, x_{1n}\}$ and $\sigma_2 = \{x_{21}, x_{22}, \ldots, x_{2m}\}$ be two not empty full sequences, the order $\sigma_1 \leq_{\mathbb{S}} \sigma_2$ is defined if and only if $\exists x \in \mathbb{S}_{\mathcal{M}} : x_{1n} \leq x \leq x_{2m}$, the set $(\mathbb{F}_{\mathcal{M}}, \leq_{\mathbb{S}})$ is a partial order. The relation $\sigma_1 \| \sigma_2$ is introduced if and only if $x_{11} \| x_{21}$.

The next lemma says that $\mathcal{M}$ is well structured w.r.t definition of full sequences.

Lemma 1 *The full sequences over $\mathcal{M}$ have the following proprieties.*

1. *For each not empty full sequences $\sigma_1, \sigma_2 \in \mathbb{F}_{\mathcal{M}}$:*

   (a) $\sigma_1 \| \sigma_2 \Rightarrow \forall x \in \sigma_1, \forall y \in \sigma_2 : x \| y$.
   (b) $\sigma_1 \leq_{\mathbb{S}} \sigma_2 \vee \sigma_2 \leq_{\mathbb{S}} \sigma_1 \vee \sigma_1 \| \sigma_2$.
   (c) $(\sigma_1 \leq \{x\} \wedge \sigma_1 \leq \{y\}) \Rightarrow (x = y)$.
   (d) $(\{x\} \leq \sigma_1 \wedge \{y\} \leq \sigma_1) \Rightarrow (x = y)$.

2. *For each not empty full sequences $\sigma_1, \sigma_2, \sigma_3 \in \mathbb{F}_{\mathcal{M}}$:*

   (a) $(\sigma_1 \leq_{\mathbb{S}} \sigma_2 \wedge \sigma_3 \leq_{\mathbb{S}} \sigma_2) \Rightarrow \exists x \in \mathbb{S}_{\mathcal{M}}$ *such that* $\sigma_1 \leq \{x\} \leq \sigma_2 \wedge \sigma_3 \leq \{x\} \leq \sigma_2$.

[4]From the fact that every LSGA net is a structural conflict net.

(b) $(\sigma_1 \leq_{\mathbb{S}} \sigma_2 \wedge \sigma_1 \leq_{\mathbb{S}} \sigma_3) \Rightarrow \exists x \in \mathbb{S}_{\mathcal{M}}$ *such that* $\sigma_1 \leq \{x\} \leq \sigma_2 \wedge \sigma_1 \leq \{x\} \leq \sigma_3$.

Proof Let $\sigma_1 = \{x_{11}, x_{12}, \ldots, x_{1n}\}$ and $\sigma_2 = \{x_{21}, x_{22}, \ldots, x_{2m}\}$.

- For Property 1(a): we proceed by absurd. If $\exists u \in \sigma_1$ and $\exists v \in \sigma_2$ such that $u \leq v$ we have a contradiction. Let $x_{11} \leq x_{12} \leq \cdots x_{1i} \leq u$ with $(i \leq n)$ and $u \leq v$ we have $x_{11} \leq x_{12} \leq \cdots x_{1i} \leq v$. So we have $x_{21} \leq v$ and $x_{11} \leq v$, in other words, v is a synchronous event, which contradicts the definition of a full sequence σ_2. Similar if $v \leq u$, we have u as a synchronous event.
- For Property 1(b): from the fact that we have only one case from $x_{11} \leq x_{21}$, $x_{21} \leq x_{11}$ and $x_{11} \| x_{21}$, and by definition of $\leq$ and $\|$, we have one from $\sigma_1 \leq_{\mathbb{S}} \sigma_2$, $\sigma_2 \leq_{\mathbb{S}} \sigma_1$ and $\sigma_1 \| \sigma_2$.
- For Property 1(c): by absurd, $\exists x, y \in \mathbb{S}_{\mathcal{M}} : x \neq y$ such that $\sigma_1 \leq \{x\}$ and $\sigma_1 \leq \{y\}$ hence $x_{1n} \leq \{x\}$ and $x_{1n} \leq \{y\}$. So x_{1n} is a branch-out point, in the other words, we have a contradiction to the definition of a full sequence σ_1.
- Proof of Property 1(d) is similar to proof of Property 1(c).
- For Property 2(a): holds from Property 1(c) and definition of $\mathbb{F}_{\mathcal{M}}$.
- For Property 2(b): holds from Property 1(d) and definition of $\mathbb{F}_{\mathcal{M}}$.

In the next, we present, in the first, how generate a maximality-based labeled transition system from a given full sequence. In the second, the synchronous parallel operator of maximality-based labeled transition systems is defined.

Definition 11 Let $\sigma = \{x_1, x_2, \ldots, x_n\}$ sequence in $\mathcal{M}$ such that $x_1 \leq x_2 \leq \cdots \leq x_n$. The construction $\mathcal{G}(\sigma) = (\Omega_\sigma, \lambda_\sigma, \mu_\sigma, \xi_\sigma, \psi_\sigma)$ is a maximality-based labeled transition system such that $\Omega_\sigma = \left\langle S_\sigma, T_\sigma, \alpha_\sigma, \beta_\sigma, s^0_\sigma \right\rangle$ such that:

- $s^0_\sigma \xrightarrow{\phi a_{x_1}} s_1 \in T_\sigma$ which, in the *mlts* of the origin, the beginning of execution of $a \in Act$ is associated to event x_1 .
- $\forall i \in 2 \ldots n : s_{i-1} \xrightarrow{\{x_{i-1}\} a_{x_i}} s_i \in T_\sigma$ which, in the *mlts* of the origin, the beginning of execution of $a \in Act$ is associated to event x_i.

Definition 12 Let $mlts_1 = (\Omega_1, \lambda_1, \mu_1, \xi_1, \psi_1)$ and $mlts_2 = (\Omega_2, \lambda_2, \mu_2, \xi_2, \psi_2)$ be two maximality-based labeled transition systems such that
$\Omega_1 = \left\langle S_1, T_1, \alpha_1, \beta_1, s^1_0 \right\rangle$ and $\Omega_2 = \left\langle S_2, T_2, \alpha_2, \beta_2, s^2_0 \right\rangle$. The synchronous parallel of $mlts_1$ and $mlts_2$ over $L \subseteq \mathcal{M}$ is defined by $mlts_1 |[L]| mlts_2 = (\Omega, \lambda, \mu, \xi, \psi)$ such that $\Omega = \langle S, T, \alpha, \beta, s_0 \rangle$ with:

1. $s_0 = (s^1_0, s^2_0)$.
2. $S = S_1 \times S_2$.
3. $\psi(S) = \psi(S_1) \cup \psi(S_2)$.
4. $\forall s \xrightarrow{E a_x} s' \in T_1$ such that $x \notin L \Rightarrow \forall s'' \in S_2 : (s, s'') \xrightarrow{E a_x} (s', s'') \in T$.
5. $\forall s \xrightarrow{E a_x} s' \in T_2$ such that $x \notin L \Rightarrow \forall s'' \in S_1 : (s'', s) \xrightarrow{E a_x} (s'', s') \in T$.
6. $\forall s_1 \xrightarrow{E_1 a_x} s'_1 \in T_1$ and $\forall s_2 \xrightarrow{E_2 a_x} s'_2 \in T_2$ such that $x \in L \Rightarrow (s_1, s_2) \xrightarrow{E_1 \cup E_2 a_x} (s'_1, s'_2) \in T$.

Lemma 2 *The synchronous parallel operator have the following proprieties.*

1. *For each* $\sigma_1, \sigma_2 \in \mathbb{F}_{\mathcal{M}}$ *such that* $\sigma_1 \| \sigma_2$*:*
   (a) $mlts' = \mathcal{G}(\sigma_1)|[\,]|\mathcal{G}(\sigma_2)$ *such that* $\mathbb{F}_{\mathcal{M}'} = \{\sigma_1, \sigma_2\}$ *and* $\mathbb{S}_{\mathcal{M}'} = \phi$.
   (b) *if* $\sigma_1 \leq \{x\} \wedge \sigma_2 \leq \{x\}$ *then* $mlts' = \mathcal{G}(\sigma_1 \leq \{x\})|[\{x\}]|\mathcal{G}(\sigma_2 \leq \{y\})$ *such that* $\mathbb{F}_{\mathcal{M}'} = \{\sigma_1, \sigma_2\}$ *and* $\mathbb{S}_{\mathcal{M}'} = \{x\}$.
   (c) *if* $\sigma_1 \leq \{x\} \wedge \sigma_2 \leq \{y\}$ *then* $mlts' = \mathcal{G}(\sigma_1 \leq \{x\})|[\{x, y\}]|\mathcal{G}(\sigma_2 \leq \{y\})$ *such that* $\mathbb{F}_{\mathcal{M}'} = \{\sigma_1, \sigma_2\}$ *and* $\mathbb{S}_{\mathcal{M}'} = \{x, y\}$.
2. *For each* $\sigma_1, \sigma_2, \sigma_3 \in \mathbb{F}_{\mathcal{M}}$*:*
   (a) *if* $\sigma_1 \leq \sigma_2 \wedge \sigma_3 \leq \sigma_2$ *which* $\sigma_1 \leq \{x\} \leq \sigma_2 \wedge \sigma_3 \leq \{x\} \leq \sigma_2$ *then* $mlts' = \mathcal{G}(\sigma_1 \leq \{x\})|[\{x\}]|\mathcal{G}(\{x\} \leq \sigma_2)|[\{x\}]|\mathcal{G}(\sigma_3 \leq \{x\})$ *such that* $\mathbb{F}_{\mathcal{M}'} = \{\sigma_1, \sigma_2, \sigma_3\}$ *and* $\mathbb{S}_{\mathcal{M}'} = \{x\}$.
   (b) *if* $\sigma_1 \leq \sigma_2 \wedge \sigma_1 \leq \sigma_3$ *which* $\sigma_1 \leq \{x\} \leq \sigma_2 \wedge \sigma_1 \leq \{x\} \leq \sigma_3$ *then* $mlts' = \mathcal{G}(\sigma_1 \leq \{x\})|[\{x\}]|\mathcal{G}(\{x\} \leq \sigma_2)|[\{x\}]|\mathcal{G}(\{x\} \leq \sigma_3)$ *such that* $\mathbb{F}_{\mathcal{M}'} = \{\sigma_1, \sigma_2, \sigma_3\}$ *and* $\mathbb{S}_{\mathcal{M}'} = \{x\}$.
3. *For each* $C_1 = \sigma_0 \leq \{x_1\} \leq \sigma_1 \leq \{x_2\} \cdots \leq \{x_n\} \leq \sigma_n$ *and*
   $C_2 = \sigma'_0 \leq \{y_1\} \leq \sigma'_1 \leq \{y_2\} \cdots \leq \{y_n\} \leq \sigma'_m$,
   $mlts' = \mathcal{G}(C_1)|[\{x_1 \ldots x_n, y_0 \ldots y_m\}]|\mathcal{G}(C_2)$ *such that*
   $\mathbb{F}_{\mathcal{M}'} = \{\sigma_1 \ldots \sigma_n, \sigma'_0 \ldots \sigma'_m\}$ *and* $\mathbb{S}_{\mathcal{M}'} = \{x_1 \ldots x_n, y_0 \ldots y_m\}$.

It is straightforward to prove Lemma 2 by Definitions 11 and 12.

Lemma 2 means that the synchronous parallel operator is an identity function over $\mathbb{F}_{\mathcal{M}} \times \mathbb{F}_{\mathcal{M}} \times \mathbb{S}_{\mathcal{M}}$. Consequently, the synchronous parallel operator of all paths of $(\mathbb{F}_{\mathcal{M}}, \leq_{\mathbb{S}})$ is the initial MLTS. Therefore, we can take each path with theirs synchronous points as sequential component. Hence, *mlts* has been decomposed into a set of sequential MLTSs so that their parallel composition is the initial MLTS.

In the following, we give a formal definition of a decomposition.

Definition 13 Let $Y = \{\sigma_0, \sigma_1, \ldots, \sigma_n\}$ be a path in $(\mathbb{F}_{\mathcal{M}}, \leq_{\mathbb{S}})$ and $S = \{s_0, s_1, \ldots, s_{n-1}\}$ be a subset of $\mathbb{S}_{\mathcal{M}}$ such that $C = \sigma_0 \leq s_0 \leq \sigma_1 \leq s_1 \cdots, \leq s_{n-1} \leq \sigma_n$. The sequence C is an element of $\mathbb{C}(\mathcal{M})$ if and only if:

$\forall C' \in \mathbb{C}(\mathcal{M}) : C' \cap C \subseteq \mathbb{S}_{\mathcal{M}}$.

An element of $\mathbb{C}(\mathcal{M})$ is an alternative sequence of full sequence and elements of $\mathbb{S}_{\mathcal{M}}$.

For example, given *mlts* of Fig. 3 with $\mathbb{F}_{\mathcal{M}} = \{\{x\}, \{y\}, \{u\}, \{v\}\}$ and $\mathbb{S}_{\mathcal{M}} = \{z\}$. We have a multi-possible decomposition of *mlts*:
$\mathbb{C}(\mathcal{M}) = \{\{x, z, u\}, \{y, z, v\}\}$ or $\mathbb{C}(\mathcal{M}) = \{\{x, z, v\}, \{y, z, u\}\}$.

$\mathcal{G}(\{x, z, u\})|[\{z\}]|\mathcal{G}(\{y, z, v\} = mlts = \mathcal{G}(\{x, z, v\})|[\{z\}]|\mathcal{G}(\{y, z, u\}$

Theorem 1 *Let* $mlts = (\Omega, \lambda, \mu, \xi, \psi)$ *to be a maximality-based labeled transition system such that* $\Omega = \langle S, T, \alpha, \beta, s_0 \rangle$ *and let* $(\mathcal{M}, \leq, \|)$ *associated to mlts.*

$$mlts = \mathcal{SPC}(\mathcal{M})$$

With $\mathcal{SPC}(\mathcal{M}) = \mathcal{G}(C_1)|[\mathbb{S}_{\mathcal{M}}]|\mathcal{G}(C_2)|[\mathbb{S}_{\mathcal{M}}]|\mathcal{G}(C_3)\ldots\mathcal{G}(C_{n-1})|[\mathbb{S}_{\mathcal{M}}]|\mathcal{G}(C_n)$ *which* $C_i \in \mathbb{C}(\mathcal{M})$*,* $\mathcal{SPC}$ *is a synchronous parallel composition of all elements of* $\mathbb{C}(\mathcal{M})$*.*

Proof Holds by the fact that the structure of $\mathcal{M}$ is preserved by the synchronous parallel operator $|[\cdots]|$ (see Lemma 2).

3.2 Generation of LSGA

From the definition of $\mathbb{C}(\mathcal{M})$, we can have a distribution of $\mathcal{M}$ in different localities D such that for each $C_1, C_2 \in \mathbb{C}_{\mathcal{M}}$ and for each events x and y of C_1 and for each event z in C_2 : $D(x) = D(y) \neq D(z)$.[5] In the other words, we can transform each $C \in \mathbb{C}_{\mathcal{M}}$ to a component net with interface, thereafter we have a LSGA net.

To transform the synchronous to asynchronous interaction between $C_1, C_2 \in \mathbb{C}_{\mathcal{M}}$ such that $C_1 \cap C_2 = S \neq \phi$ we must redefine the sequences C_1, C_2 as follow $\forall s \in S \Longrightarrow (s \in C_1 \wedge s \notin C_2) \vee (s \notin C_1 \wedge s \in C_2)$.

In the following, we give a formal definition of the transformation of the synchronous to asynchronous interaction.

Definition 14 Let $\mathbb{A}_{\mathcal{M}}$ be a set of the asynchronous components generated from $\mathbb{C}_{\mathcal{M}}$. We can define $\mathbb{A}_{\mathcal{M}}$ as follow:

- $\forall \mathcal{A}_1, \mathcal{A}_2 \in \mathbb{A}_{\mathcal{M}} \Longrightarrow \mathcal{A}_1 \cap \mathcal{A}_2 = \phi$ and
- $\forall C_1, C_2 \in \mathbb{C}_{\mathcal{M}}$ such that $C_1 \cap C_2 = S \neq \phi$, we have $\mathcal{A}_1 = (C_1 \setminus S) \cup S_1$ and $\mathcal{A}_2 = (C_2 \setminus S) \cup S_2$ such that $S = S_1 \cup S_2$.

Definition 15 Let $C \in \mathbb{A}(\mathcal{M})$ which $C = c_0 \leq c_1 \leq c_2 \leq \cdots \leq c_n$. Let $\mathfrak{Cio}(C) = (N, I, O)$ be a component net with interface such that $N = (S, T, F, M_0, L)$ be a Petri net and:

- For each $c_j \in \mathbb{F}_{\mathcal{M}}$ which $c_j = x_{j1} \leq x_{j2} \leq \cdots \leq x_{jm}$:
  - $x_{ji} \in T$ with $i \in \{1 \ldots m\}$.
  - $s_{ji} \in S$ with $i \in \{0 \ldots m\}$.
  - $F(x_{ji}, s_{ji}) = 1 \wedge F(s_{j(i-1)}, x_{ji}) = 1$ with $i \in \{1 \ldots m\}$.
- For each synchronous point $c_j \in \mathbb{S}_{\mathcal{M}}$ which $c_j = \{x\}$:
  - $x \in T$,
  - $i_{jk} \in I$ with $k \in \{0 \ldots |Left(x)| - 1\}$ such that $Left(x) = \{y | \forall y \in \mathcal{M} \cdot y \geq x\}$,
  - $o_{jk} \in O$ with $k \in \{1 \ldots |right(x)| - 1\}$ such that $right(x) = \{y | \forall y \in \mathcal{M} \cdot x \geq y\}$,
  - $F(s_{(j-1)p}, x) = 1$ such that $p = |c_{j-1}| + 1$ and $\forall i_{jk} \in I : F(i_{jk}, x) = 1$
  - $F(x, s_{(j+1)p}) = 1$ such that $p = |c_{j+1}| + 1$ and $\forall o_{jk} \in O : F(x, o_{jk}) = 1$.
- $s_{00} = M_0$.

[5] From the fact that for each $C_1, C_2 \in \mathbb{C}_{\mathcal{M}}$ and for each event x of C_1 and event y in C_2 such that $x, y \notin \mathbb{S}_{\mathcal{M}}$:$x \| y$.

Definition 16 The asynchronous parallel composition of all element $C_i \in \mathbb{A}(\mathcal{M})$ is a LSGA i.e. $\mathfrak{L}sga(\mathcal{M})) =\|_{i\in\mathbb{N}} \mathfrak{Cio}(C_i)$ for all $C_i \in \mathbb{A}(\mathcal{M})$.

By Definition 16 and Theorem 1 it follows:

Lemma 3 *Let mlts to be a maximality-based labeled transition system over $\mathcal{M}$. Let mlts′ to be a corresponding maximality-based labeled transition system of $\mathfrak{L}sga(\mathcal{M})$: $mlts = mlts'$.*

4 Conclusions

In this paper, we define a distributed implementation from a semantics model rather than a specification model. We proposed an approach for decomposing a given MLTS to a set of sequential MLTSs related only by synchronous interactions. This decomposition has twofold objectives: Firstly, the behaviour of this decomposition and the initial MLTS are identical; Secondly, the interaction between sequential MLTSs can be seen as an asynchronous interaction. In other words, our decomposition produces a distributed system compatible to the definition of R.V. Glabbeek U. Goltz and J.W. Schicke-Uffmann. For proving this compatibility, we introduced the definition of a component net with interface from a sequential MLTS.

References

1. Aceto, L., Hennessy, M.: Adding action refinement to a finite process algebra. In: Automata, Languages and Programming, 18th International Colloquium, ICALP91, Madrid, Spain, July 8–12, 1991, Proceedings, pp. 506–519 (1991)
2. Benamira, A., Saïdouni, D.: Maximality-based labeled transition systems normal form. In: Modeling Approaches and Algorithms for Advanced Computer Applications, Studies in Computational Intelligence, vol. 488, pp. 337–346. Springer (2013)
3. Bouneb, M., Saïdouni, D., Ilié, J.: A reduced maximality labeled transition system generation for recursive petri nets. Formal Asp. Comput. **27**(5–6), 951–973 (2015). http://dx.doi.org/10.1007/s00165-015-0341-3
4. Courtiat, J.P., Saïdouni, D.E.: Relating maximality-based semantics to action refinement in process algebras. In: FORTE, pp. 293–308 (1994)
5. Glabbeek, R.J.V.: The refinement theorem for ST-bisimulation semantics. In: Proceedings IFIP TC2 Working Conference on Programming Concepts and Methods, Sea of Gallilee, Israel, April 1990, pp. 27–52. North-Holland (1990)
6. Glabbeek, R.J.V., Goltz, U., Schicke, J.: On synchronous and asynchronous interaction in distributed systems. In: Mathematical Foundations of Computer Science, 33rd International Symposium, MFCS 2008, Torun, Poland, August 25-29, Proceedings. LNCS, vol. 5162, pp. 16–35. Springer (2008)
7. Glabbeek, R.J.V., Goltz, U., Schicke-Uffmann, J.: On characterising distributability. Log. Methods Comput. Sci. **9**(3) (2013). http://dx.doi.org/10.2168/LMCS-9(3:17)2013
8. Hennessy, M.: Concurrent testing of processes (extended abstract). In: CONCUR ’92, Third International Conference on Concurrency Theory, Stony Brook, NY, USA, August 24-27, 1992, Proceedings, pp. 94–107 (1992). http://dx.doi.org/10.1007/BFb0084785

9. Saïdouni, D.E., Belala, N., Bouneb, M.: Aggregation of transitions in marking graph generation based on maximality semantics for Petri nets. In: VECoS'2008, University of Leeds, UK. eWiC Series, The British Computer Society (BCS) (July, 2–3rd 2008). ISSN: 1477-9358
10. Saïdouni, D.E., Belala, N., Bouneb, M.: Maximality-based structural operational semantics for Petri nets. In: Proceedings of INTELLIGENT SYSTEMS AND AUTOMATION:(CISA'09), Tunisia. vol. 1107, pp. 269–274. American Institute of Physics (2009). ISBN: 978-0-7354-0642-1
11. van Glabbeek, R.J., Vaandrager, F.W.: Petri net models for algebraic theories of concurrency. In: PARLE, Parallel Architectures and Languages Europe, Volume II: Parallel Languages, Eindhoven, The Netherlands, June 15–19, 1987, Proceedings, pp. 224–242 (1987)
12. Vogler, W.: Bisimulation and action refinement. In: STACS 91, 8th Annual Symposium on Theoretical Aspects of Computer Science, Hamburg, Germany, February 14–16, 1991, Proceedings, pp. 309–321 (1991). http://dx.doi.org/10.1007/BFb0020808