This regular paper was presented as part of the main technical program at IEEE ETFA'2011

On the deadlock analysis of multithreaded control software*

Juan-Pablo López-Grao
Dpt. of Computer Science and Systems Eng.
University of Zaragoza, Spain
jpablo@unizar.es

José-Manuel Colom
Aragon Institute of Engineering Research (I3A)
University of Zaragoza, Spain
jm@unizar.es

Abstract

The long interest in finding efficient solutions to deadlock occurrence induced by resource sharing is persistent in the context of concurrent control software production. Petri net-based correction techniques which were traditionally applied in the context of flexible manufacturing systems (FMS) constitute a promising new approach. In this vein, Gadara nets were introduced as an attempt to import the strengths of these techniques into the software domain. In this paper, we prove that these Petri nets are close to a subclass of S^4PR (a widely-exploited class in the context of FMS) and provide some related equivalence results. Some limitations which Gadara nets present for the modelling and automated correction of software are also unveiled. Last but no least, we present formal proofs of the theorems characterising non-liveness in Gadara nets. To our knowledge, no such proofs were published before.

1. Introduction. Modelling control software

Finantial, spatial, technical. Whatever the reason, resource scarceness is a traditional scenario in diverse systems engineering disciplines. Consequently, available resources are often shared among concurrent processes, which must compete in order to be granted their allocation. Discrete event systems of this kind are named *Resource Allocation Systems (RAS)*. Deadlocks arise when a set of processes is indefinitely waiting for resources that are already held by other processes of the same set [1].

Formal methods-based techniques, and specifically those based on Petri nets [11], constitute a fertile ground to deal with such deadlocks. Many of these real-world RAS can be abstracted into a conceptualization constructed around two entities: processes and resources.

Petri nets feature a simple, orthogonal syntax and an appealing graphical representation for modelling these abstractions [2]. Besides, there exist powerful structural results for certain subclasses of Petri nets for RAS which enable powerful analysis and synthesis techniques for identifying and fixing potential or factual deadlocks [4, 12, 15]. The life cycle is closed with the deployment of the corrections computed for the model over the real-world system.

This methodology has been successfully applied to flexible manufacturing systems (FMS) where processes follow predefined production plans and resources can be artifacts such as robots, machines or conveyor belts, or passive elements such as storage area. Diverse classes of Petri net models, such as S^3PR [4], S^4PR [12, 15] and many others [6, 17] were defined for this aim, with specific attributes for modelling different configurations.

However, all of them prove insufficient for modelling the control software driving the evolution of a FMS, which can also incur in deadlocks [10]. Indeed, the complexity of multithreaded control software is ever-increasing due to technological advances and an eagerness for configuration versatility, sublimated by the emergence of agile automation systems [7]. In this context, the access to physical resources can be encapsulated through software *virtual* resources, such as mutexes or semaphores [3].

Nevertheless, the structure of this category of RAS introduces new challenges due to the particularities of programming languages. First, the control flow of the processes (threads) can contain internal cycles, in the vein of recirculations, due to iterative programming. Second, *release* operations occassionally precede *allocation* operations on semaphores, albeit being used in a conservative way. These and other issues are tackled in [10] from the perspective of general-purpose multithreaded software.

Gadara nets [8, 16] are a new class of Petri nets for RAS modelling in software. The main goal is adding support for internal cycles to the control flow of the processes. However, they were apparently defined trying to retain a structure-based liveness characterization. Unfortunately, this is too ambitious in the general case [10]. Therefore, the authors applied some syntactic restrictions to the class.

In section 2, we review Gadara nets and some of their limitations for modelling multithreaded control software. In section 3, a formal proof of the existence of a structural liveness characterization for Gadara nets is presented. In section 4, we prove the equivalence between Gadara nets and a restricted subclass of S^4PR with respect to their correction through net state equation-based structural methods, e.g., [15]. Section 5 summarizes the conclusions.

*This work has been partially supported by the European Community's Seventh Framework Programme under Project DISC (Grant Agreement n. INFSO-ICT-224498).

978-1-4577-0018-7/11/$26.00

2. The Gadara approach

Gadara nets belong to the family of Petri nets conceived for modelling RAS. They are modular nets that generalize the S^{4}PR class in allowing general state machines but constrain the S^{4}PR class in forbidding the allocation of resources in conflicting transitions inside the state machines (i.e., there is no inclusion relation between these two net classes). A more technical constraint is related to the weights of the minimal p-semiflows associated to resources, which are equal to one. This means that an active thread at most uses one copy per type of resource.

The formal definition, as presented in [16], follows. The reader can find the basic notation of Petri nets in [13].

Definition 1 *[16] Let $I_{\mathcal{N}}$ be a finite set of indices. A Gadara net is a connected ordinary pure P/T net $\mathcal{N} = \langle P, T, F\rangle$ where:*

1. *$P = P_0 \cup P_S \cup P_R$ is a partition of P such that:*
   (a) [idle places] *$P_0 = \{p_{0_1}, ..., p_{0_{|I_{\mathcal{N}}|}}\}$;*
   (b) [process places] *$P_S = P_1 \cup ... \cup P_{|I_{\mathcal{N}}|}$, where $\forall i \in I_{\mathcal{N}}$: $P_i \neq \emptyset$ and $\forall i, j \in I_{\mathcal{N}}, i \neq j$: $P_i \cap P_j = \emptyset$;*
   (c) [resource places] *$P_R = \{r_1, ..., r_n\}, n > 0$.*
2. *$T = T_1 \cup ... \cup T_{|I_{\mathcal{N}}|}$, where $\forall i \in I_{\mathcal{N}}$: $T_i \neq \emptyset$, and $\forall i, j \in I_{\mathcal{N}}, i \neq j$: $T_i \cap T_j = \emptyset$.*
3. *For all $i \in I_{\mathcal{N}}$ the subnet generated by restricting $\mathcal{N}$ to $\langle \{p_{0_i}\} \cup P_i, T_i\rangle$ is a strongly connected state machine. This is called the i-th* process subnet.
4. *For all $p \in P_S$: if $|p^\bullet| > 1$, then $^\bullet(p^\bullet) = \{p\}$.*
5. *For each $r \in P_R$, there exists a unique minimal p-semiflow associated to r, Y_r, fulfilling: $\|Y_r\| \cap P_R = \{r\}, \|Y_r\| \cap P_0 = \emptyset$, $\|Y_r\| \cap P_S \neq \emptyset$ and $\mathbf{0} \lneq Y_r \lneq \mathbf{1}$.*
6. *$P_S = \bigcup_{r \in P_R}(\|Y_r\| \setminus \{r\})$.*

The next definition is included as an extra condition to definition 1 in [16]. For coherence reasons with our previous works, we have extracted it, neatly separating the net structure and marking. Note that this definition presents the other fundamental difference with the class of S^{4}PR systems: in Gadara systems, resource places are binary.

Definition 2 *[16] Let $\mathcal{N} = \langle P, T, F\rangle$ be a Gadara net. An initial marking m_0 is acceptable for $\mathcal{N}$ iff $m_0[P_0] \geq \mathbf{1}, m_0[P_S] = \mathbf{0}, m_0[P_R] = \mathbf{1}$.*

Figure 1 depicts a Gadara net with an acceptable initial marking. As we will see later, the non-liveness of a Gadara net is characterized by the existence of a structural artifact, a *bad siphon*, that eventually gets *insufficiently marked* or empty. This can be prevented by inserting a monitor place which restricts the system behaviour:

Definition 3 *[16] Let $\mathcal{N} = \langle P, T, F\rangle$ be a Gadara net. A controlled Gadara net is a connected generalized pure P/T net $\mathcal{N}_c = \langle P \cup P_C, T, F \cup F_c, W_c\rangle$ such that, in addition to all conditions in Definition 1 for $\mathcal{N}$, we have:*

7. *For each $p_c \in P_C$, there exists a unique minimal p-semiflow associated to p_c, $Y_{p_c} \in \mathbb{N}^{|P \cup P_C|}$, fulfilling: $\|Y_{p_c}\| \cap P_C = \{p_c\}, \|Y_{p_c}\| \cap P_R = \emptyset, \|Y_{p_c}\| \cap P_0 = \emptyset, \|Y_{p_c}\| \cap P_S \neq \emptyset$ and $Y_{p_c}[p_c] = 1$.*

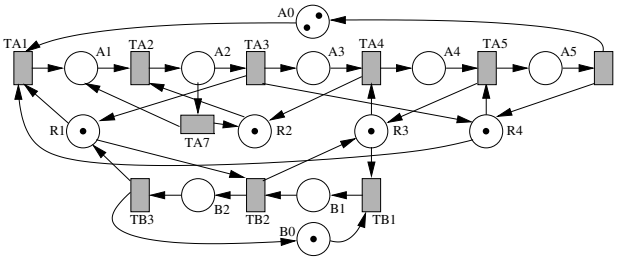

Figure 1. A (non-live) Gadara net

Definition 4 *[16] Let $\mathcal{N}_c = \langle P_0 \cup P_S \cup P_R \cup P_C, T, F \cup F_c, W_c\rangle$ be a controlled Gadara net. An initial marking m_0 is acceptable for $\mathcal{N}_c$ iff $m_0[P_0] \geq \mathbf{1}, m_0[P_S] = \mathbf{0}, m_0[P_R] = \mathbf{1}$ and for every $p_c \in P_C, p \in P_S$: $m_0[p_c] \geq Y_{p_c}[p]$.*

The net of figure 1 has three bad siphons. The minimal siphon $D = \{R1, R2, R3, R4, A2, A5, B2\}$ is empty at the reachable marking $m = \{A1, B1, A3\}$. This siphon can be controlled by aggregating a control place p_c which would have arcs from $TA1$ and $TA2$ with the following non-unitary weights: $C[p_c, TA2] = -C[p_c, TA1] = 2$. Those non-unitary arc weights are due to the fact that $A1$ belongs to the support of the minimal p-semiflow of two different resource places, Y_{R1} and Y_{R4}. Out of curiosity, there exists another minimal siphon, $D' = \{R1, R2, R3, A2, A4, B2\}$ which is also empty at m. If we control D' then we obtain a control place with only unitary arcs. This, of course, does not always happen. As a result, $Y_r \in \{0, 1\}^{|P \cup P_C|}$ but $Y_{p_c} \in \mathbb{N}^{|P \cup P_C|}$, in general.

Please note that, from now onwards, we will use the term Gadara nets for referring to controlled Gadara nets.

As discussed in [10], very complex phenomena can appear when internal cycles are allowed in the control flow of the processes. This is true even in safe nets with no resource lending [10] or overlapped (i.e., not nested) internal cycles, as the net system in figure 2 reveals. In this case, no *bad* siphon ever becomes insufficiently marked, even when the net is non-live. Thus, the classic structural characterization [15] does not work in the general context.

The "good behaviour" of Gadara nets originates from the fact that conflicts induced by process places are free-choice. This seems to approximate these models to the kind of systems with linear processes, such as the L-S^{3}PR class [5]. This modelling assumption can however be overrestrictive for modelling software systems: some kind of software cannot be modelled with Gadara nets, due to the usage of non-blocking allocation primitives, which are supported by (e.g.) POSIX locks. A similar argument can be applied when conditional statement expressions must be evaluated atomically. Additionally, general, non-binary semaphores are not supported, and the case of `signal` operations preceding `wait` operations is neither considered. These uncovered aspects in the modelling of real software restrain an automated translator to Petri nets from working, unless we constrain the kind of programs that the engineer can construct.

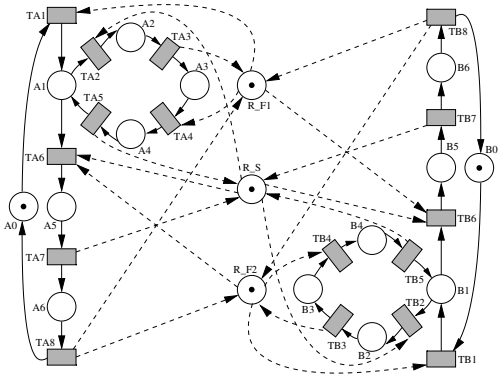

Figure 2. A S^{5}PR with no bad siphon ever becoming insufficiently marked

3. The liveness characterization

In [16], the authors enunciate a liveness characterization for Gadara nets based on the existence of an insufficiently marked siphon at a reachable marking (there captured by the equivalent concept of *resource-induced deadly marked siphon*). Unfortunately, we have been unable to find any formal proof of this claim in the existing literature. Instead, it is stated that the same proof strategy to that followed for S^{4}PR can be extended for Gadara nets [16], albeit this is a dubious claim. Please mind that conflicts induced by process places are not free-choice, in general, for S^{4}PR nets. This is a restriction imposed by Gadara nets that must be taken into account in the proof: otherwise, it would be also generalizable for nets like the one in figure 2. Hence, the liveness theorem needs further proof in order to be unequivocally validated.

We will prove the liveness theorem over a superclass of controlled Gadara nets which we will define next.

Definition 5 *An extended Gadara (e-Gadara) net is a connected* generalized *pure P/T net $\mathcal{N} = \langle P, T, F, W\rangle$ (or, equivalently, $\mathcal{N} = \langle P, T, C\rangle$) following definition 1 except for condition 5, which is generalized as follows:*

5. *For each $r \in P_R$, there exists a unique minimal p-semiflow associated to r, $Y_r \in \mathbb{N}^{|P|}$, fulfilling: $\|Y_r\| \cap P_R = \{r\}, \|Y_r\| \cap P_0 = \emptyset, \|Y_r\| \cap P_S \neq \emptyset$ and $Y_r[r] = 1$.*

Definition 6 *Let $\mathcal{N} = \langle P, T, C\rangle$ be an e-Gadara net. An initial marking m_0 is acceptable for $\mathcal{N}$ iff $m_0[P_0] \geq \mathbf{1}$, $m_0[P_S] = \mathbf{0}$ and $\forall r \in P_R, p \in P_S : m_0[r] \geq Y_r[p]$.*

Please note that the control places are included in P_R in definition 5 (therefore, no subset P_C is defined). As a result, definition 4 is consistent with definition 6.

Some more definitions follow which will be instrumental both for the liveness theorems enunciations and proofs.

Definition 7 *Let $\mathcal{N} = \langle P, T, C\rangle$ be an e-Gadara net. The* set of holders *of $r \in P_R$ is the support of the minimal p-semiflow Y_r without the place r: $\mathcal{H}_r = ||Y_r|| \setminus \{r\}$. This definition can be extended to sets of resources $A \subseteq P_R$ in the following way: $\mathcal{H}_A = \cup_{r\in A}\mathcal{H}_r$.*

Definition 8 *Given a marking m in an e-Gadara net, a transition t is said to be m-process-enabled (m-process-disabled) iff its input process place is (not) marked, and m-resource-enabled (m-resource-disabled) iff all (some) input resource places have (not) enough tokens to fire it, i.e., $m[P_R, t] \geq Pre[P_R, t]$ ($m[P_R, t] \ngeq Pre[P_R, t]$).*

Before proceeding with liveness theorems 13 and 14, we will deal with four instrumental and easy lemmas.

Lemma 9 *[9] Every e-Gadara net is consistent.*

Proof:
The process subnets of $\mathcal{N}$ are strongly connected state machines and therefore each one is consistent, i.e., every transition t of $\mathcal{N}$ is covered by at least a t-semiflow of the state machine containing t. We prove that these t-semiflows are also t-semiflows of the net $\mathcal{N}$. Indeed, if X is a t-semiflow of $\mathcal{N}$ without resources it is enough to prove that $\forall r \in P_R : C[r, T] \cdot X = 0$. Taking into account definition 5.5, $C[r, T] = -\sum_{p\in\|Y_r\|\setminus\{r\}} Y_r[p] \cdot C[p, T]$, and therefore, $C[r, T] \cdot X = -(\sum_{p\in\|Y_r\|\setminus\{r\}} Y_r[p] \cdot C[p, T]) \cdot X = -\sum_{p\in\|Y_r\|\setminus\{r\}} Y_r[p] \cdot C[p, T] \cdot X = 0$. Therefore, $\mathcal{N}$ is consistent. ♢

Lemma 10 *Let $\langle \mathcal{N}, m_0\rangle$ be an e-Gadara net with an acceptable initial marking. Then, for every $t \in T$, there exists a t-semiflow containing t being realizable from m_0.*

Proof:
We will prove that a single token can be extracted from any idle place at m_0 and be freely moved in isolation through its corresponding state machine. Let M_1 be the subset of reachable markings such that one and only one process place is (mono-)marked, i.e., $M_1 = \{m \in RS(\mathcal{N}, m_0) \mid \exists |p \in P_S : m[p] = 1, ||m|| \cap P_S = \{p\}\}$.

First, every $t \in P_0{}^\bullet$ is enabled at m_0 since ${}^\bullet t \subseteq P_0 \cup P_R$ and, by the definition of acceptable initial marking, $P_0 \subset ||m_0||$ and $\forall r \in P_R : m_0[r] \geq Y_r[q] = Pre[r, t]$, with $q = t^\bullet \cap P_S$. By firing t a marking of M_1 is reached.

Without loss of generality, let $m \in M_1$. We prove that every m-process-enabled transition t is enabled. If $t^\bullet \cap P_S = \emptyset$ then $m[P_R] \geq \mathbf{0} = Pre[P_R, t]$. Thus, $m \xrightarrow{t} m_0$. Otherwise, let $\{p\} = {}^\bullet t \cap P_S$ and $\{q\} = t^\bullet \cap P_S$. Then $\forall r \in P_R : Pre[r, t] = max(Y_r[q] - Y_r[p], 0)$. By definition 6, $\forall r \in P_R : m_0[r] \geq Y_r[q]$. Then $m[r] = m_0[r] - Y_r[p] \geq Y_r[q] - Y_r[p]$. Also, $m[r] \geq 0$. Thus, $m[P_R] \geq Pre[P_R, t]$; i.e. $m \xrightarrow{t} m'$, $m' \in M_1$.

We have proven that an isolated token can be carried from $m_0[P_0]$ to any arbitrary $p \in P$. If p belongs to a circuit, we can take that token and make it travel around the circuit. Since every t-semiflow corresponds to a circuit in a state machine (the dual is proven in [11]), and e-Gadara nets are consistent by lemma 9, the new lemma holds. ♢

Since a token in a strongly connected state machine can be moved in isolation to any other arbitrary place, the next lemma is obvious. Thus, the proof is omitted, yet provided in [9]. Note that $\boldsymbol{\sigma}$ denotes the firing count vector of σ.

Lemma 11 *[9] Let $\langle \mathcal{N}, m\rangle$, $\mathcal{N} = \langle P, T, C\rangle$, be a set of isolated marked strongly connected state machines, and let $P_0 \subset P$ be an arbitrary subset of places such that P_0 contains one and only one place of each strongly connected state machine. Then there exists at least one firing sequence σ, $m \xrightarrow{\sigma} m'$, such that there exists no t-semiflow $X \neq \mathbf{0}$ of $\mathcal{N}$, with $\boldsymbol{\sigma} - X \geq \mathbf{0}$, and $\|m'\| = P_0$.*

Lemma 12 *Let $\langle \mathcal{N}, m\rangle$, $\mathcal{N} = \langle P, T, C\rangle$, be a set of isolated marked strongly connected state machines. Let $p \in P$ be a marked place at m, $m[p] > 0$, and let σ be a firing sequence, $m \xrightarrow{\sigma} m'$, such that $m'[p] = 0$. Then there also exists a firable sequence σ', $m \xrightarrow{\sigma'} m'$, with $\boldsymbol{\sigma}' = \boldsymbol{\sigma}$ and $\exists t \in p^\bullet, \sigma'' \in T^* : \sigma' = t\,\sigma''$.*

Proof:
Since $p \in \|m\| \setminus \|m'\|$, there exists at least one transition in $p^\bullet$ which appears once or more times in σ. Let σ be defined as $\sigma = u\,t\,v$, where $u \in (T \setminus p^\bullet)^*$, $t \in p^\bullet$ and $v \in T^*$; i.e., u is the maximal prefix before the first firing of a transition in $p^\bullet$. We prove that $\sigma' = t\,u\,v$ is firable from m. It is enough to prove that $t\,u$ is firable, since it implies that a marking m_2 is reached from which v is firable (because it is the same marking m_2 reached when fired the prefix $u\,t$ of σ). Since $m[p] > 0$, we can fire t from m and we reach m_1, with $m_1[p'] \geq m[p']$, $\forall p' \in P \setminus \{p\}$. Since $m \xrightarrow{u}$ and every transition t that appears in u holds $p \notin {}^\bullet t$ then u must also be firable from m_1. $\diamondsuit$

Theorem 13 *Let $\langle \mathcal{N}, m_0\rangle$ be an e-Gadara net with an acceptable initial marking. $\langle \mathcal{N}, m_0\rangle$ is non-live iff $\exists m \in RS(\mathcal{N}, m_0)$ such that the set of m-process-enabled transitions is non-empty and each one of these transitions is m-resource-disabled.*

Proof:
$\Rightarrow$) Let m' be a reachable marking such that at least one transition t in $\mathcal{N}$ is dead. Let $\mathcal{N}^{P_S}$ be the net $\mathcal{N}$ without the resource places, and $m'_{|P_S}$ ($m_{0|P_S}$) denote the marking m' (m_0) restricted to the places of $\mathcal{N}^{P_S}$. Let $\Sigma = \{\sigma \mid m'_{|P_S} \xrightarrow{\sigma} m_{0|P_S}$ and there is no t-semiflow X with $\boldsymbol{\sigma} - X \geq \mathbf{0}\}$. By lemma 11, the set Σ is non-empty. Besides, since the unitary vector of dimension $|T|$ is a t-semiflow of $\mathcal{N}^{P_S}$, every $\sigma \in \Sigma$ holds $|\sigma| < K \cdot |T|$, where $K = \sum_{p \in P_S} m[p]$. Consequently, the set Σ is finite.

Let σ_1 be the sequence of Σ which has the longest prefix u, $\sigma_1 = u\,v$, such that $m' \xrightarrow{u}$ in $\mathcal{N}$. If $u = \sigma_1$, $m' \xrightarrow{u} m_0$. But t would be eventually firable by lemma 10, contradicting the hypothesis that t is dead at m'. Therefore $u \neq \sigma_1$, and $m' \xrightarrow{u} m$, $m \neq m_0$. Thus, $m[P_S] \neq \mathbf{0}$. The set of m-process-enabled transitions is non-empty.

Now we prove that every transition in $(\|m\| \cap P_S)^\bullet$ is disabled at m. Without loss of generality, we take an arbitrary $p \in \|m\| \cap P_S$. Let $m_{|P_S}$ denote the marking m restricted to $\mathcal{N}^{P_S}$. By lemma 12, there exists $t \in p^\bullet, v'' \in T^*$ such that $v' = t\,v''$ is firable from $\langle N^{P_S}, m_{|P_S}\rangle$ with $\mathbf{v} = \mathbf{v}'$. Then the sequence $\sigma_2 = u\,t\,v''$ is firable from $\langle N^{P_S}, m'_{|P_S}\rangle$ and belongs to Σ, since $\boldsymbol{\sigma}_2 = \boldsymbol{\sigma}_1$. But $u\,t$ is not firable from m' since otherwise u would not be the longest fireable prefix of every sequence in Σ. Since t is m-process-enabled, t must be m-resource-disabled, with ${}^\bullet t \cap P_R \neq \emptyset$. Then, by definition 1, point 4, $|p^\bullet| = 1$. Thus every transition in $p^\bullet$ is m-process-enabled, m-resource-disabled, and so is every transition in $(\|m\| \cap P_S)^\bullet$.

$\Leftarrow$) Let $t \in (||m|| \cap P_S)^\bullet$. In order to fire t some more tokens are needed in some places belonging to $P_R \cap {}^\bullet t$. Since tokens in the process places cannot progress at m, we can only change the marking of such resources by activating some idle processes. Let ET be the set of m-process-enabled transitions, let $AP = {}^\bullet ET \cap P_S$, and let $m \xrightarrow{\sigma} m'$. We are going to prove, by induction over the length of σ that: (i) $||\sigma|| \cap ET = \emptyset$, and (ii) $\forall p \in AP : m'[p] \geq m[p]$.

Doing so, and since $m[P_S \setminus AP] = \mathbf{0}$, it can be deduced that $\forall p \in P_S : m'[P_S] \geq m[P_S]$. But then $\forall r \in P_R : m'[r] = m_0[r] - \sum_{p \in P_S} m'[p] \cdot Y_r[p] \leq m_0[r] - \sum_{p \in P_S} m[p] \cdot Y_r[p] = m[r]$. Therefore, no transition of ET can be m'-resource-enabled.

- *Case σ= t.* Since no transition of ET is enabled at m, then $t \in P_0{}^\bullet$ and then $t \notin ET$. On the other hand, if $t \notin {}^\bullet AP$, $\forall p \in AP : m'[p] = m[p]$. If $t \in {}^\bullet AP$, let $t^\bullet \cap P_S = \{q\} \in AP$. In this case, $m'[q] = m[q] + 1$ and $m'[p] = m[p]$ for every $p \in AP \setminus \{q\}$.
- *General case.* $m \xrightarrow{\sigma''} m'' \xrightarrow{t} m'$, where σ'', m'' verify the induction hypothesys. But since $\forall p \in AP : m''[p] \geq m[p]$ then $\forall r \in P_R : m''[r] \leq m[r]$, so every transition of ET is m''-resource disabled, and $t \notin ET$. Therefore, $\forall p \in AP : m'[p] \geq m''[p] \geq m[p]$, and we can conclude. $\diamondsuit$

It is worth mentioning that the second half of the proof of theorem 13 is almost literally that presented in [14] for S^{4}PR nets. This is also true for the next theorem:

Theorem 14 *Let $\langle \mathcal{N}, m_0\rangle$ be an e-Gadara net with an acceptable initial marking. $\langle \mathcal{N}, m_0\rangle$ is non-live iff $\exists m \in RS(\mathcal{N}, m_0)$ and a siphon D such that $m[P_S] > 0$ and the firing of each m-process-enabled transition is prevented by a set of resource places belonging to D.*

Proof:
$\Leftarrow$) Each m-process-enabled transition is m-resource-disabled and $m[P_S] > \mathbf{0}$. Hence $\langle \mathcal{N}, m_0\rangle$ is non-live.

$\Rightarrow$) Let m be a marking such that the set of m-process-enabled transitions is non-empty and each m-process-enabled transition is m-resource-disabled. We construct D, with $D = D_R \cup D_S$, as follows: (i) $D_R = D \cap P_R = \{r \in P_R \mid \exists t \in r^\bullet : m[r] < Pre[r,t] \wedge m[{}^\bullet t \cap P_S] > \mathbf{0}\}$, and (ii) $D_S = D \cap P_S = \{p \in \mathcal{H}_{D_R} \mid m[p] = 0\}$.

We are going to prove that $D_S \neq \emptyset$ and $D_S \subset \mathcal{H}_{D_R}$. Let us suppose that $D_S = \emptyset$. Let F be a directed path defined as $F = p_0 t_0 p_1 t_1 ... p_k t_k$ such that $\forall i \in \{1, ..., k\} : p_i \in {}^\bullet t_i \cap P_S$, $p_0 \in {}^\bullet t_0 \cap P_0$ and $\exists j \in \{1, ..., k\} : t_j \cap {}^\bullet D_R \neq \emptyset$. Such a path must exist since the process nets are strongly connected state machines: thus, for every $i \in I_{\mathcal{N}}$, $t_j \in T_i$, exists a circuit containing p_{0_i} and t_j such that p_{0_i} and t_j appear only once.

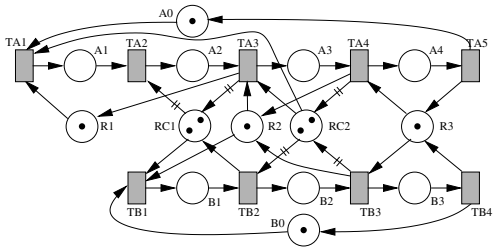

Figure 3. A net belonging to the controlled Gadara class with no minimal siphon becoming insufficiently marked

Let t be the last transition in the directed path F such that $t \in {}^\bullet D_R$. Let $r \in t^\bullet \cap D_R$. Since $P_S \cap {}^\bullet t \in \mathcal{H}_r$ and $D_S = \emptyset$, $m[P_S \cap {}^\bullet t] > 0$, i.e. t is m-process-enabled. Since $t \notin D_R{}^\bullet$ (if $t \in D_R{}^\bullet$, and taking into account that the net is self-loop free and $P_0 \cap \cup_{r \in P_R} ||Y_r|| = \emptyset$, t could not be the last one), then t is also m-resource-enabled and therefore t can fire contradicting the hypothesys that from m only transitions in $P_0{}^\bullet$ can occur.

If $D_S = \mathcal{H}_{D_R}$, since $m[D_S] = \mathbf{0}$, $\forall r \in D_R : m[r] = m_0[r]$ which makes impossible for r to prevent the firing of any transition (m_0 is acceptable). Then, $D_S \subset \mathcal{H}_{D_R}$.

Let us now prove that $D = D_R \cup D_S$ is a siphon. Let $t \in {}^\bullet D$; two cases must be checked.

First case ($t \in {}^\bullet D_R$). Let $r \in D_R$ be such that $t \in {}^\bullet r$. Let $p \in \mathcal{H}_r \cap {}^\bullet t$ (there exists such p because there is an arc from t to r). If $m[p] = 0$, then $p \in D_S$, and $t \in D_S{}^\bullet$. Otherwise, since t is disabled, $\exists r' \in (P_R \cap {}^\bullet t) : m[r'] < Pre[r', t]$, i.e. disabling it. Then $r' \in D_R$, and in consequence, $t \in D_R{}^\bullet$.

Second case ($t \notin {}^\bullet D_R$). Then $\exists p \in D_S : t \in {}^\bullet p$ and $\exists r' \in D_R : p \in \mathcal{H}_{r'}$. If $\exists r \in ({}^\bullet t \cap D_R)$, $t \in D^\bullet$ and we can conclude. Let us now suppose that ${}^\bullet t \cap D_R = \emptyset$. In this case, t cannot be m-process-enabled; if it was, by theorem 13, t has to be m-resource-disabled, and then, there would exist $r \in {}^\bullet t \cap D_R$. Let $\{q\} = {}^\bullet t \cap P_S$ (this place exists because $p \in \mathcal{H}_{r'}$ and ${}^\bullet t \cap D_R = \emptyset$). Since t is not m-process-enabled, $m[q] = 0$. Moreover, since $p \in \mathcal{H}_{r'}$, p belongs to a minimal p-semiflow containing r' in its support and since $r' \notin {}^\bullet t$, q is also in the support of such p-semiflow, which implies that $q \in \mathcal{H}_{r'}$. Therefore, $q \in D_S$ (q is not marked), and $t \in D_S{}^\bullet$.

By construction, the firing of each m-process-enabled transition is prevented by some resource places in D. $\diamondsuit$

A siphon that holds the condition of theorem 14 is said to be a *bad siphon* that becomes *insufficiently marked at* m. Note that minimal siphons are insufficient to characterize non-liveness for controlled Gadara nets. The net in figure 3 is non-live: the siphon $D = \{RC1, RC2, A3, B2\}$ becomes insufficiently marked at $m = A1 + B1 + RC1 + RC2 + R3$, but it is not minimal, since it contains the minimal siphon $D' = \{RC2, A3, B2\}$. D' is not insufficiently marked for any reachable marking. It is worth noting that no siphon, be it minimal or not, is ever fully emptied.

4. Approaching Gadara by means of CPR

Gadara nets can be transformed into CPR nets (a restricted subclass of S^{4}PR) so that controlling a Gadara net through net state equation-based structural methods [15] can alternatively be conducted in the space of the transformed net: as we will prove onwards, both classes are equivalent at that level.

Paradoxically, the syntactic restriction enforced to retain a structural characterization places Gadara nets into an instrumental role from the angle of structural liveness analysis and synthesis: the maturity of the techniques introduced for S^{4}PR nets [12, 15] suggests working in the transformed space.

We will start by introducing the subclass of CPR nets.

Definition 15 *Let $I_\mathcal{N}$ be a finite set of indices. A net of Confluent Processes with Resources (CPR net) is a connected* generalized *pure P/T net $\mathcal{N} = \langle P, T, F, W \rangle$ (or, equivalently, $\mathcal{N} = \langle P, T, C \rangle$) defined with the same conditions of definition 1 except conditions 4 and 5, which are redefined as follows:*

4. *For all $p \in P_S$: $|p^\bullet| = 1$.*
5. *For each $r \in P_R$, there exists a unique minimal p-semiflow associated to r, $Y_r \in \mathbb{N}^{|P|}$, fulfilling: $\{r\} = \|Y_r\| \cap P_R, \|Y_r\| \cap P_0 = \emptyset, \|Y_r\| \cap P_S \neq \emptyset$ and $Y_r[r] = 1$.*

Clearly, CPR nets are a subclass of e-Gadara nets. The corresponding definition of acceptable initial marking is consistent with definition 6 (indeed the conditions are identical) and has been omitted for space considerations.

Also, a CPR net is an S^{4}PR such that there is no conflict induced by a process place, i.e. $\forall p \in P_S$: $|p^\bullet| = 1$. Again, it must be noticed that the concept of acceptable initial marking for CPR nets is consistent with that provided for the superclass S^{4}PR [15].

In the same vein, the rest of definitions are inherited from the e-Gadara superclass. In all cases, these definitions collapse perfectly with those given for S^{4}PR nets.

Next, we will introduce a rule to transform Gadara nets into CPR nets. The free choice constraint in the process subnets of Gadara nets makes that, from the point of view of the allocation of resources, a process first decides the computation path and, after that, the allocation of resources is deterministic. In other words, resources do not participate in the internal choices of the processes. Choices on resources only happen in the competition relations between processes for the resources.

We take advantage of this behaviour and introduce a transformation for Gadara nets such th at this *a priori* decision about the internal computation path to be carried out when choices appear is dealt from the initial state.

Definition 16 *Let $\mathcal{N} = \langle P, T, C \rangle$, $P = P_0 \cup P_S \cup P_R$, be an e-Gadara net such that $\exists p \in P_S : |p^\bullet| > 1$. Let p_{0_i} be the idle place of the process subnet to which p belongs. The net $\mathcal{N}_e = \langle P, T_e \cup \{t\}, C_e \rangle$ is said to be a* conflict expansion of p in $\mathcal{N}$*, where:*

- $T_e = T \setminus (p^\bullet \cap {}^\bullet P_0)$.
- $\forall t' \in T \setminus p^\bullet : C_e[P,t'] = C[P,t']$.
- $\forall t' \in p^\bullet \setminus {}^\bullet P_0, p' \in P \setminus (P_R \cup \{p, p_{0_i}\}), r \in P_R :$
  $C_e[p',t'] = C[p',t']$ *(thus: $C_e[p',t'] \geq 0$),*
  $C_e[r,t'] = C[r,t'] - Y_r[p]$ *(thus: $C_e[r,t'] \leq 0$),*
  $C_e[p,t'] = 0$, $C_e[p_{0_i},t'] = -1$.
- $\forall p' \in P \setminus (P_R \cup \{p, p_{0_i}\}), r \in P_R :$
  $C_e[p',t] = 0$,
  $C_e[r,t] = Y_r[p]$ *(thus: $C_e[r,t] \geq 0$),*
  $C_e[p_{0_i},t] = -C_e[p,t] = 1$.

Corollary 17 *$\mathcal{N}_e$ is an e-Gadara net and for every $r \in P_R$ its associated minimal p-semiflow Y_r^e holds $Y_r^e = Y_r$.*

Corollary 18 *If there exist no more conflicts in the process subnets of $\mathcal{N}_e$, then $\mathcal{N}_e$ is a CPR net.*

The proof of corollary 17 (which is intuitive but cumbersome to prove) is left to the reader. Corollary 18 is straightforward. A consequence of these corollaries is that, starting from an e-Gadara net, we can always obtain a CPR net by way of sucessively expanding its conflicts.

Next, we will prove an interesting result regarding (non-)liveness preservation after the net transformation. Since the set of places of $\mathcal{N}$ is equal to the set of places of $\mathcal{N}_e$, markings over $\mathcal{N}$ will be trivially mapped over $\mathcal{N}_e$, and viceversa. This will be assumed for the rest of the paper, and transitively extended to nets obtained by way of a sucession of conflict expansions starting from $\mathcal{N}$.

Theorem 19 *Let $\langle \mathcal{N}, m_0\rangle$,$\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C\rangle$, be an e-Gadara net with an acceptable initial marking such that $\exists p \in P_S : |p^\bullet| > 1$, and $\mathcal{N}_e = \langle P_0 \cup P_S \cup P_R, T_e \cup \{t\}, C_e\rangle$ be an e-Gadara net being the conflict expansion of p in $\mathcal{N}$. $\langle \mathcal{N}, m_0\rangle$ is non-live $\Rightarrow$ $\langle \mathcal{N}_e, m_0\rangle$ is non-live.*

Proof:
Let $m \in RS(\mathcal{N}, m_0)$ such that $m[P_S] > 0$ and every m-process-enabled transition is m-resource-disabled. Such m must exist by theorem 13. Let σ be a firing sequence of $\mathcal{N}$ such that $m_0 \xrightarrow{\sigma} m$. Let $T' = \{t' \in T \mid C[p,t'] < 0\}$. We will contruct a firing sequence σ_e of $\mathcal{N}_e$ such that $m_0 \xrightarrow{\sigma_e} m$ by copying σ after making some replacements in it, following these two rules: (i) For each occurrence of a transition $u \in T' \setminus T_e$ in σ we replace u per t in σ', and (ii) For each occurrence of a transition $v \in T' \cap T_e$ in σ we replace v per the sequence tv in σ'.

The sequence σ' must also be firable from m_0, since $C[P, T \setminus T'] = C_e[P, T \setminus T']$, and (i) $\forall u \in T' \setminus T_e : C_e[P,t] = C[P,u]$, and (ii) $\forall v \in T' \cap T_e : C_e[P,t] + C_e[P,v] = C[P,v]$ and t must be firable in $\mathcal{N}_e$ whenever v is firable in $\mathcal{N}$, since t has the same input process place than v and no input resource place. Thus, $m_0 \xrightarrow{\sigma_e} m$.

Finally let T_{mpe} be the non-empty set of m-process-enabled transitions of $\mathcal{N}$. For every $u \in T_{mpe}$, u is the unique output transition of its input process place in $\mathcal{N}$. Otherwise, $C[P_R, u] = \mathbf{0}$ and therefore u would not be m-resource-disabled. Thus, p is not the input place of u, $\forall u \in T_{mpe}$, and therefore $T_{mpe} \cap T' = \emptyset$. Then $C_e[P, T_{mpe}] = C[P, T_{mpe}]$. Thus, T_{mpe} is also the set of m-process-enabled transitions of $\mathcal{N}_e$, and every transition in T_{mpe} is m-resource-disabled over $\mathcal{N}_e$. By theorem 13, $\langle \mathcal{N}_e, m_0\rangle$ is non-live. $\diamondsuit$

The reverse of theorem 19 is not true in general, since there may exist killing spurious solutions in a live system $\langle \mathcal{N}, m_0\rangle$ which are reachable deadlocks in $\langle \mathcal{N}_e, m_0\rangle$. Nevertheless, theorem 19 allows us to work over the transformed net in order to enforce liveness, since if $\langle \mathcal{N}_e, m_0\rangle$ is live then $\langle \mathcal{N}, m_0\rangle$ is live. However, this is only reasonable if the number of siphons to be controlled is not severely increased. The next result is related to this:

Lemma 20 *Let $\mathcal{N} = \langle P, T, C\rangle$, $P = P_0 \cup P_S \cup P_R$, be an e-Gadara net such that $\exists p \in P_S : |p^\bullet| > 1$, $\mathcal{N}_e = \langle P, T_e \cup \{t\}, C_e\rangle$ be a conflict expansion of p in $\mathcal{N}$, and $D \subseteq P$. If D is a siphon of $\mathcal{N}_e$ then D is a siphon of $\mathcal{N}$.*

Proof:
Let $T' = \{t' \in T \mid C[p,t'] < 0\}$ and $Prop1(t_1) \equiv [(\exists p_1 \in D : C[p_1,t_1] > 0) \Rightarrow (\exists p_2 \in D : C[p_2,t_1] < 0)]$, for every $t_1 \in T$. We must prove that $\forall t_1 \in T : Prop1(t_1)$. Since $\forall t_3 \in T \setminus T' : C[P,t_3] = C_e[P,t_3]$, it is enough to prove that $\forall t_1 \in T' : Prop1(t_1)$.

Let us prove that $\forall t_1 \in T' \setminus T_e : Prop1(t_1)$. Without loss of generality, let $t_1 \in T' \setminus T_e$. If $\nexists p_1 \in D$ such that $C[p_1,t_1] > 0$, then $Prop1(t_1) \equiv True$. Let $p_1 \in D$ such that $C[p_1,t_1] > 0$. Note that $\forall r \in P_R : C[r,t_1] = Y_r[p]$, $C[p_{0_i},t_1] = 1$, $C[p,t_1] = -1$, and $\forall p_2 \in P \setminus (P_R \cup \{p_{0_i}, p\}) : C[p_2,t_1] = 0$. Thus, $C_e[P,t] = C[P,t_1]$. Then $C_e[p_1,t] = C[p_1,t_1] > 0$. Since $p_1 \in D$ and p is the unique input place of t, then $p \in D$. Since $C[p,t_1] < 0$, $Prop1(t_1) \equiv True$.

We will now prove that $\forall t_1 \in T' \cap T_e : Prop1(t_1)$. Without loss of generality, let $t_1 \in T' \cap T_e$.

Suppose that $\nexists p_1 \in D$ such that $C_e[p_1,t_1] > 0$. Since the unique output place of t_1 in $\mathcal{N}_e$ is a process place, if $C[P_R,t_1] = \mathbf{0}$, then $\nexists p_2 \in D$ such that $C[p_2,t_1] > 0$, and $Prop1(t_1) \equiv True$. If $C[P_R,t_1] \gneq \mathbf{0}$, let r be an arbitrary $r \in P_R$ such that $C[r,t_1] > 0$. Then $Y_r[p] > 0$ and therefore $C_e[r,t] = Y_r[p] > 0$. Since p is the unique input of t, then $p \in D$. Since $C[p,t_1] < 0$, $Prop1(t_1) \equiv True$.

Otherwise, $\exists p_1 \in D$ such that $C_e[p_1,t_1] > 0$. Then $C[p_1,t_1] \geq C_e[p_1,t_1] > 0$ and $\exists p_2 \in D$ such that $C_e[p_2,t_1] < 0$. If $p_2 \in P_0$ then $p \in D$, since p is the unique input place of t and t is an input transition of P_0. Since $C[p,t_1] < 0$, $Prop1(t_1) \equiv True$. If $p_2 \notin P_0$, i.e., $p_2 \in P_R$, then $p \in D$, since $\forall r \in P_R : C_e[r,t] = Y_r[p] \geq -C_e[r,t_1]$ and p is the unique input transition of t. Since $C[p,t_1] < 0$, $Prop1(t_1) \equiv True$. $\diamondsuit$

Corollary 21 *The number of siphons of $\mathcal{N}_e$ is lower than or equal to the number of siphons of $\mathcal{N}$.*

Although the reverse of lemma 20 is not true (i.e., a siphon of $\mathcal{N}$ is not always a siphon of $\mathcal{N}_e$) there exists a close relation between the siphons of both nets. Indeed, for every siphon in $\mathcal{N}$ there exists another siphon in $\mathcal{N}_e$

which contains the same resource places. The next proposition is instrumental; the corresponding lemma follows.

Proposition 22 *Let $\mathcal{N} = \langle P, T, C\rangle$, $P = P_0 \cup P_S \cup P_R$, be an e-Gadara net such that $\exists p \in P_S : |p^\bullet| > 1$, $\mathcal{N}_e = \langle P, T_e \cup \{t\}, C_e\rangle$ be a conflict expansion of p in $\mathcal{N}$, and $D \subseteq P$. Let $\mathcal{N}_{et} = \langle P, T_e, C_{et}\rangle$ be the subnet generated by restricting $\mathcal{N}_e$ to $\langle P, T_e\rangle$. If D is a siphon of $\mathcal{N}$ then D is a siphon of $\mathcal{N}_{et}$.*

Proof:
Let $T' = \{t' \in T \mid C[p, t'] < 0\}$ and $Prop2(t_2) \equiv [(\exists p_1 \in D : C_e[p_1, t_2] > 0) \Rightarrow (\exists p_2 \in D : C_e[p_2, t_2] < 0)]$, for every $t_2 \in T_e$. We must prove that $\forall t_2 \in T_e : Prop2(t_2)$. Since $\forall t_3 \in T \setminus T' : C[P, t_3] = C_e[P, t_3]$, it is enough to prove that $\forall t_2 \in T' \cap T_e : Prop2(t_2)$.

Without loss of generality, let $t_2 \in T' \cap T_e$, and note that $\forall p_1 \in P : C_e[p_1, t_2] \leq C[p_1, t_2]$. Suppose that $\nexists p_1 \in D$ such that $C[p_1, t_2] > 0$. Then $\forall p_1 \in D : C_e[p_1, t_2] \leq C[p_1, t_2] \leq 0$, i.e., $\nexists p_2 \in D$ such that $C_e[p_2, t_2] > 0$, and $Prop2(t_2) \equiv True$. Otherwise, $\exists p_2 \in D$ such that $C[p_2, t_2] < 0$. Then $C_e[p_2, t_2] \leq C[p_2, t_2] < 0$, and thus $Prop2(t_2) \equiv True$. ♢

Lemma 23 *[9] Let $\mathcal{N} = \langle P, T, C\rangle$, $P = P_0 \cup P_S \cup P_R$, be an e-Gadara net such that $\exists p \in P_S : |p^\bullet| > 1$, $\mathcal{N}_e = \langle P, T_e \cup \{t\}, C_e\rangle$ be a conflict expansion of p in $\mathcal{N}$, and $D \subseteq P$. If D is a siphon for $\mathcal{N}$, then $\exists D_e \supseteq D$, with $D_e \setminus D \subset P_S$, such that D_e is a siphon of $\mathcal{N}_e$.*

Proof:
Let $i \in I_{\mathcal{N}}$ the index of the process subnet of $\mathcal{N}_e$ to which p belongs. Let $\mathcal{N}_{et} = \langle P, T_e, C_{et}\rangle$ be the subnet generated by restricting $\mathcal{N}_e$ to $\langle P, T_e\rangle$. By proposition 22, D is a siphon of $\mathcal{N}_{et}$. However D is not a siphon of $\mathcal{N}_e$ iff $\exists p_1 \in (\{p_{0_i}\} \cup P_R) \cap D$ such that $C_e[p_1, t] > 0$.

If $p_1 = p_{0_i}$ then $D_e = D \cup P_i$ is a siphon of $\mathcal{N}_e$, since the i-th process subnet is a strongly connected state machine. Otherwise, $r = p_1$ is a resource place, $r \in P_R$, with $Y_r[p] > 0$. Let $D_t = (\mathcal{H}_r \setminus D) \cap P_i$. We will prove that $D_e = D \cup D_t$ is a siphon of D.

Let $T' = \{t \in T \mid \exists p \in D_t \text{ such that } C[p, t] > 0\}$ and $Prop2(t_1) \equiv [(\exists p_1 \in D : C_e[p_1, t_1] > 0) \Rightarrow (\exists p_2 \in D : C_e[p_2, t_1] < 0)]$. We must prove that $\forall t_1 \in T : Prop2(t_1)$. Since for every $t_2 \in T \setminus (T' \cup \{t\}) : C_e[D, t_2] = C_{et}[D, t_2]$, it is enough to prove that $\forall t_1 \in T' : Prop2(t_1) \wedge Prop2(t)$. Since $p \in \mathcal{H}_r$, then $p \in D$ and $Prop2(t) \equiv True$.

Finally, we will prove that $\forall t_1 \in T' : Prop2(t_1)$. Without loss of generality, let $t_1 \in T'$, and let p_1 the output process place of t_1, $p_1 \in D_t$. If $C[r, t] < 0$, then $Prop2(t_1) \equiv True$ since $r \in D$. Otherwise, if $C[r, t] = 0$ then $\exists p_2 \in P_i$ such that $C[p_2, t] < 0$. If $p_2 \in D$ then $Prop2(t_1) \equiv True$. If $p_2 \notin D$ then $p_2 \in \mathcal{H}_r$ (because $C[r, t] < 0$) and thus $p_2 \in D_t$. Summing up, in all cases, $Prop2(t_1) \equiv True$. ♢

Next, we will introduce the complete expansion and reduction rules, based on definition 16. In order to be able to undo a conflict expansion after having enforced liveness,

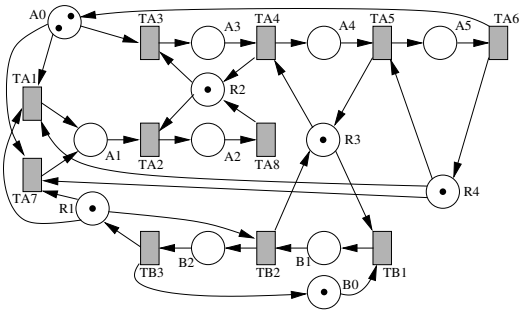

Figure 4. A CPR net which is the conflict expansion of place A2 in the net of figure 1

we will need to keep record of the previous steps. The following definition is instrumental for this aim.

Definition 24 *Let $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C\rangle$ be an e-Gadara net. Its* associated expansion record (AER) *is a duple $\langle T_{\mathcal{N}}, \Psi_{\mathcal{N}}\rangle$, where $T \subseteq T_{\mathcal{N}}$ and $\Psi_{\mathcal{N}}$ is a set of triples in $T_{\mathcal{N}} \times P_S \times \mathcal{P}(T_{\mathcal{N}})$ such that $\forall (t, p, \tau) \in \Psi_{\mathcal{N}}$: (i) $\exists p_0 \in P_0 : t \in {}^\bullet p_0, \tau \in p_0{}^\bullet$; (ii) $\{p\} = {}^\bullet t \cap P_S$; and (iii) $\forall (t', p, \tau') \in \Psi_{\mathcal{N}} : t \neq t', \tau \cap \tau' = \emptyset$.*

$\Psi_{\mathcal{N}}$ registers which conflicts were previously expanded, and how. $T_{\mathcal{N}}$ is a record of the whole set of transitions, including those which were removed or created at past conflict expansions. The transformations are formally defined as follows:

Rule 1 (Expansion Rule)
Input: *An e-Gadara net $\mathcal{N} = \langle P_0 \cup P_S \cup P_R, T, C\rangle$ such that $\exists p \in P_S : |p^\bullet| > 1$, plus its AER $\langle T_{\mathcal{N}}, \Psi_{\mathcal{N}}\rangle$.*
Output: *An e-Gadara net $\mathcal{N}_e = \langle P_0 \cup P_S \cup P_R, T_e \cup \{t\}, C_e\rangle$ which is the conflict expansion of p in $\mathcal{N}$, plus its AER $\langle T_{\mathcal{N}} \cup \{t\}, \Psi_{\mathcal{N}_e}\rangle$, with $\Psi_{\mathcal{N}_e} = \Psi_{\mathcal{N}} \cup \{(t, p, p^\bullet)\}$.*

Rule 2 (Reduction Rule)
Input: *An e-Gadara net $\mathcal{N}_e = \langle P, T_e \cup \{t\}, C_e\rangle$ plus its AER $\langle T_{\mathcal{N}} \cup \{t\}, \Psi_{\mathcal{N}_e}\rangle$, such that there exists $(t, p, \tau) \in \Psi_{\mathcal{N}_e}$ and there also exists an e-Gadara net $\mathcal{N} = \langle P, T, C\rangle$ with $\mathcal{N}_e$ being the conflict expansion of p in $\mathcal{N}$.*
Output: *The e-Gadara net $\mathcal{N}$ plus its AER $\langle T_{\mathcal{N}}, \Psi_{\mathcal{N}}\rangle$, with $\Psi_{\mathcal{N}} = \Psi_{\mathcal{N}_e} \setminus \{(t, p, \tau)\}$.*

Thanks to theorem 19, we can enforce liveness directly over the transformed CPR net. Once enough control places have been aggregated so as to make it live, we can apply the reduction rule to obtain a live e-Gadara net. However, it is worth mentioning that it may be necessary to move carefully the arcs of some control places before. This is due to the fact that some transitions were uncontrollable in the original net: namely, those belonging to a conflict in a process subnet.

Finally, lemma 25 introduces a powerful result regarding the potential reachability set (PRS) of the transformed net.

Lemma 25 *Let $\langle \mathcal{N}, m_0\rangle$ be an e-Gadara net with an acceptable initial marking such that $\exists p \in P_S : |p^\bullet| > 1$,*

and $\langle \mathcal{N}_e, m_0 \rangle$ be an e-Gadara net obtained by applying the conflict expansion transformation rule over $\mathcal{N}$. Then $m \in PRS(\mathcal{N}, m_0)$ iff $m \in PRS(\mathcal{N}_e, m_0)$.

Proof:

For every $i \in I_{\mathcal{N}}$, let Y_{S_i} denote the unique minimal p-semiflow of $\mathcal{N}$ induced by the i-th process subnet of $\mathcal{N}$. It is easy to see that Y_{S_i} is also a unique minimal p-semiflow of $\mathcal{N}_e$, induced by the i-th process subnet of $\mathcal{N}_e$. On the other hand, by corollary 17, $Y_r = Y_r^e$, for all $r \in P_R$.

Let B be a matrix of dimensions $(|P_R| + |I_{\mathcal{N}}|) \times |P|$ of integers such that the rows of B are the set of vectors $\{Y_{S_i} \mid i \in I_{\mathcal{N}}\} \cup \{Y_r \mid r \in P_R\}$. Then B is a non-negative canonical basis of p-semiflows both for $\mathcal{N}$ and $\mathcal{N}_e$.

Finally, since a Gadara net is consistent (by lemma 9) and conservative (by construction), a non-negative canonical basis of p-semiflows (B) generates the same solution space than the net state equation. Hence, $PRS(\mathcal{N}, m_0) = PRS(\mathcal{N}_e, m_0)$. ♢

Many efficient structure-based liveness enforcing techniques rely on the net state equation. In [15], one of these is presented for S^{4}PR nets, which is a superclass of CPR. The result of lemma 25 encourages the application of this kind of techniques over the transformed net, since the space solution of the net state equation is equal on both nets (original and transformed).

5. Conclusions

In this paper, we have presented an overview of Gadara nets and its limitations for modelling multithreaded control software. From the structural analysis and synthesis perspective, we have proved that the syntactic restrictions introduced in Gadara nets provoke significant constraints from the point of view of the behaviours allowed in the allocation of resources. This means that we can bridge Gadara nets with a subclass of S^{4}PR in which the allocation of resources internal to a process is deterministic, i.e., resources do not participate in the internal choices. Consequently, we can use liveness enforcing methods based solely on structural information, leaving this class close to the well-studied S^{4}PR class in that context. Unfortunately, state-space exploration and region theory based methods can be too consuming for real-world concurrent control software systems due to their usually huge dimensions. Finally, in [10] we have introduced a more versatile class for modelling multithreaded software systems. Nevertheless, new, more complex phenomena arise, and further study is required to overcome the problems that arise in this new framework.

References

[1] E.-G. Coffman, M. Elphick, and A. Shoshani. System deadlocks. *ACM Computing Surveys*, 3(2):67–78, 1971.

[2] J.-M. Colom. The resource allocation problem in flexible manufacturing systems. In W.-M.-P. van der Aalst and E. Best, editors, *Proc. of the 24th Int. Conf. on Applic. & Theory of Petri Nets*, volume 2679 of *LNCS*, pages 23–35, Eindhoven, Netherlands, 2003. Springer.

[3] E. W. Dijkstra. The structure of the "THE"-multiprogramming system. In *Proc. of the 1st ACM Symposium on Operating System Principles*, SOSP '67, pages 10.1–10.6, New York, NY, USA, 1967. ACM.

[4] J. Ezpeleta, J.-M. Colom, and J. Martínez. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Transactions on Robotics and Automation*, 11(2):173–184, 1995.

[5] J. Ezpeleta, F. García-Valles, and J.-M. Colom. A class of well structured Petri nets for flexible manufacturing systems. In J. Desel and M. Silva, editors, *Proc. of the 19th Int. Conf. on Applic. & Theory of Petri Nets*, volume 1420 of *LNCS*, pages 65–83, Lisbon, Portugal, 1998. Springer.

[6] M.-D. Jeng, X.-L. Xie, and M.-Y. Peng. Process nets with resources for manufacturing modeling and their analysis. *IEEE Transactions on Robotics and Automation*, 18(6):875–889, 2002.

[7] P. T. Kidd. *Agile manufacturing: forging new frontiers*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.

[8] H. Liao, S. Lafortune, S. Reveliotis, Y. Wang, and S. Mahlke. Synthesis of maximally-permissive liveness-enforcing control policies for Gadara Petri nets. In *Proc. of the 49th IEEE Conf. on Decision and Control (CDC 10)*, pages 2797–2804. IEEE, 2010.

[9] J.-P. López-Grao and J.-M. Colom. Synthesis of live multithreaded software: A methodology based on Petri nets. Technical report, Dpt. of Computer Science and Systems Engineering, Univ. Zaragoza, 2011.

[10] J.-P. López-Grao and J.-M. Colom. A Petri net perspective on the resource allocation problem in software engineering. In K. Jensen, S. Donatelli, and J. Kleijn, editors, *Transactions on Petri Nets and Other Models of Concurrency V (ToPNoC V)*, LNCS. Springer, 2011. To appear.

[11] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[12] J. Park and S.-A. Reveliotis. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Transactions on Automatic Control*, 46(10):1572–1583, 2001.

[13] M. Silva, E. Teruel, and J. M. Colom. Linear algebraic and linear programming techniques for the analysis of Place/Transition net systems. In W. Reisig and G. Rozenberg, editors, *Lectures on Petri Nets I: Basic Models*, volume 1491 of *LNCS*, pages 309–373. Springer, 1998.

[14] F. Tricas. *Deadlock analysis, prevention and avoidance in sequential resource allocation systems*. PhD thesis, University of Zaragoza, Zaragoza, 2003.

[15] F. Tricas, F. García-Valles, J.-M. Colom, and J. Ezpeleta. A Petri net structure-based deadlock prevention solution for sequential resource allocation systems. In *Proc. of the 2005 Int. Conf. on Robotics and Automation (ICRA)*, pages 272–278, Barcelona, Spain, 2005. IEEE.

[16] Y. Wang, H. Liao, S. Reveliotis, T. Kelly, S. Mahlke, and S. Lafortune. Gadara nets: Modeling and analyzing lock allocation for deadlock avoidance in multithreaded software. In *Proc. of the Joint 48th IEEE Conf. on Decision and Control and 28th Chinese Control Conf.*, pages 4971–4976, Shanghai, China, 2009. IEEE.

[17] X. Xie and M.-D. Jeng. ERCN-merged nets and their analysis using siphons. *IEEE Transactions on Robotics and Automation*, 29(4):692–703, 1999.