The Label Splitting Problem

Josep Carmona

Universitat Politècnica de Catalunya, Spain
jcarmona@lsi.upc.edu

Abstract. The theory of regions was introduced by Ehrenfeucht and Rozenberg in the early nineties to explain how to derive (synthesize) an event-based model (e.g., a Petri net) from an automaton. To be applicable, the theory relies on stringent conditions on the input automaton. Although some relaxation on these restrictions was done in the last decade, in general not every automaton can be synthesized while preserving its behavior. A crucial step for a non-synthesizable automaton is to transform it in order to satisfy the synthesis conditions. This paper revisits label splitting, a technique to satisfy the synthesis conditions through renaming of problematic labels. For the first time, the problem is formally characterized and its optimality addressed. Some extensions and applications of the label splitting are presented to illustrate the significance of this technique.

1 Introduction

The use of *formal models* as mathematical descriptions of software or hardware systems opens the door for the adoption of formal methods in many phases of the design of a complex system. This in turn allows incorporating computer-aided techniques in the process, including verification and performance evaluation, disciplines that nowadays resort heavily in simulation. Formal models can be either state-based (i.e., automata-like) or event-based (e.g., Petri nets), and the correspondence between models is a challenging problem that we address in this paper.

The *synthesis problem* [9] consists in building a Petri net [12,14,15] that has a behavior equivalent to a given automaton (*transition system*). The problem was first addressed by Ehrenfeucht and Rozenberg [10] introducing *regions* to model the sets of states that characterize places in the Petri net. The theory is applicable to *elementary transition systems*, a proper subclass of transition systems where additional conditions are required, and for which the synthesis produces a Petri net with isomorphic behavior. These restrictions were significantly relaxed in [7], introducing the subclass of *excitation-closed transition systems*, where not isomorphism but *bisimilarity* is guaranteed. In an excitation-closed transition system, for every event the set of states where the event is enabled should be equal to the intersection of *pre-regions* (regions where the event exits) of the event. The theory of this paper relates to the subclass of excitation-closed transition systems.

K. Jensen et al. (Eds.): ToPNoC VI, LNCS 7400, pp. 1–23, 2012.

When synthesis conditions do not hold, the Petri net derived might have a behavior in general incomparable to the initial behavior [5], and therefore any faithful use of such Petri net may be hampered. To overcome this problem, one might force the synthesis conditions by transforming the initial transition system. The work in [7] was the first one in addressing this problem, introducing *label splitting* as a technique that can be applied when excitation-closure is not satisfied. The technique is based on renaming the labels of a particular event e in the transition system: given the whole set of occurrences of the event, these are relabeled into different labels $e_1, \dots, e_k$, thus preserving the event name but considering each new copy as a new event with respect to the synthesis conditions.

The new events produced by the label splitting technique increase the complexity of the Petri net derived: each new copy will be transformed into a transition, and hence the *label splitting problem* is to find a sequence of splittings that induces the minimal number of transitions in the derived Petri net. The motivation for this minimization is twofold: first, in many applications the Petri net derived is a valuable graphical description of (part of) a system, and therefore its visualization will benefit from having the minimal number of nodes. Second, by deriving a simpler model, the complexity of algorithms that take this model as input may be alleviated when the size is minimal. The technique presented in [7] only presented the label splitting technique as a heuristic to progress into excitation-closure, i.e., it never considered the optimal application of the technique.

The label splitting technique presented in this paper is a particular one: it is defined on the sets of states computed when searching for regions in state-based synthesis methods [5,7]. These sets, called *essential*, are the building blocks used in this paper to decide which labels to split. The methods for label splitting in the aforementioned papers also use the essential sets for label splitting, but as described previously, only in a heuristic manner.

In summary, this paper presents a novel view on the label splitting technique. First, we show how label splitting for excitation closure is nothing else than coloring a graph using a minimal number (the *chromatic number* of the graph) of colors. Second, we characterize the conditions under which an optimal label splitting can be derived to accomplish excitation closure. Finally, we present an algorithm that can be used when excitation closure cannot be attained by a single application of the label splitting technique presented in this paper. This algorithm is based on a relaxation of the label splitting problem that can be mapped into the *weighted set cover* problem.

For the sake of clarity, the theory of this paper will be presented for the class of safe (1-bounded) Petri nets. The contribution can be extended with no substantial change for the class of general (k-bounded) Petri nets.

The organization of the paper is the following: in Section 2 we provide the reader the necessary background to understand the contents of this paper. Then in Section 3 and Section 4 the core contributions of the paper are presented, describing techniques for obtaining regions and forcing (if possible) the synthesis conditions, respectively. In Section 5 a technique to iteratively apply the methods described in the previous sections and thus guaranteeing always a solution are described. Then, in Section 6 we provide applications and extensions of the presented technique.

2 Preliminaries

In this section we describe the basic elements necessary to understand the theory of this paper.

2.1 Finite Transition Systems and Petri Nets

Definition 1 (Transition system). *A* transition system *is a tuple* (S, E, A, s_{in})*, where* S *is a set of* states*,* E *is an alphabet of* events*, such that* $S \cap E = \emptyset$*,* $A \subseteq S \times E \times S$ *is a set of* (labeled) transitions*, and* $s_{in} \in S$ *is the* initial state*.*

We use $s \xrightarrow{e} s'$ as a shorthand for $(s, e, s') \in A$, and we write $s \xrightarrow{*} s'$ if $s = s'$ or there exist a path of labeled transitions $(s_{i-1}, e_i, s_i) \in A$ where $1 \le i \le n$, for some $n \ge 1$, $s_0 = s$, $s_n = s'$. Let $\mathsf{TS} = (S, E, A, s_{in})$ be a transition system. We consider connected transition systems that satisfy the following axioms: i) S and E are finite sets, ii) every event has an occurrence: $\forall e \in E\ \exists s, s' \in S : (s, e, s') \in A$, and iii) every state is reachable from the initial state: $\forall s \in S : s_{in} \xrightarrow{*} s$.

In some parts of the paper it will be required to compare transitions systems. The following definition formalizes a well-known relation between transition systems.

Definition 2 (Simulation, Bisimulation [1]). *Let* $\mathsf{TS}_1 = (S_1, E, A_1, s_{in_1})$ *and* $\mathsf{TS}_2 = (S_2, E, A_2, s_{in_2})$ *be two* TS*s with the same set of events. A* simulation *of* TS_1 *by* TS_2 *is a relation* π *between* S_1 *and* S_2 *such that* $s_1 \pi s_2$ *and*

- *for every* $s_1 \in S_1$*, there exists* $s_2 \in S_2$ *such that* $s_1 \pi s_2$*.*
- *for every* $(s_1, e, s_1') \in A_1$ *and for every* $s_2 \in S_2$ *such that* $s_1 \pi s_2$*, there exists* $(s_2, e, s_2') \in A_2$ *such that* $s_1' \pi s_2'$*.*

When TS_1 is simulated by TS_2 with relation π, and vice versa with relation π^{-1}, TS_1 and TS_2 are *bisimilar* [1].

Definition 3 (Petri net [12, 14, 15]). *A Petri net is a tuple* $\mathsf{PN} = (P, T, F, M_0)$ *where* P *and* T *represent finite disjoint sets of places and transitions, respectively,* $F \subseteq (P \times T) \cup (T \times P)$ *is the flow relation. Let markings be functions* $M : P \to \mathbb{N}$ *assigning a number of tokens to each place. Marking* M_0 *defines the initial state of the system.*

For a node n (place or transition) of a Petri net, $\bullet n = \{n' | (n', n) \in F\}$, and $n\bullet = \{n' | (n, n') \in F\}$, are called the predecessor and successor set of n in F, respectively. A transition $t \in T$ is *enabled* at marking M if $\forall p \in \bullet t : M(p) \geq 1$. The *firing* of an enabled transition t at M results in a marking M' (denoted $M[t\rangle M'$) such that for each place p:

$$M'(p) = \begin{cases} M(p) - 1 & if\, p \in \bullet t - t\bullet \\ M(p) + 1 & if\, p \in t\bullet - \bullet t \\ M(p) & \text{otherwise.} \end{cases}$$

A marking M' is *reachable* from M if there is a sequence of firings $\sigma = t_1 t_2 \ldots t_n$ that transforms M into M', denoted by $M[\sigma\rangle M'$. A sequence of transitions $\sigma = t_1 t_2 \ldots t_n$ is a *feasible sequence* if $M_0[\sigma\rangle M$, for some M. The set of all markings reachable from the initial marking M_0 is called its Reachability Set. The *Reachability Graph* of a Petri net PN ($\mathsf{RG}(\mathsf{PN})$) is a transition system in which the set of states is the Reachability Set, the events are the transitions of the net and a transition (M_1, t, M_2) exists if and only if $M_1[t\rangle M_2$. The initial state of the Reachability Graph is M_0. PN is k-bounded if $M(p) \leq k$ for all places p and for all reachable markings M.

2.2 Regions and Synthesis

The theory of regions provides a path beween transition systems and Petri nets. We now review this theory (the interested reader can refer to [5, 7, 9, 10, 13] for a complete overview). Given two states $s, s' \in S$ and a subset of states $S' \subseteq S$ of a transition system (S, E, A, s_{in}), if $s \notin S'$ and $s' \in S'$, then we say that transition (s, e, s') *enters* S'. If $s \in S'$ and $s' \notin S'$, then transition (s, e, s') *exits* S'. Otherwise, transition (s, e, s') *does not cross* S'.

Definition 4. *Let* $\mathsf{TS} = (S, E, A, s_{in})$ *be a transition system. Let* $S' \subseteq S$ *be a subset of states and* $e \in E$ *be an event. The following conditions (in the form of predicates) are defined for* S' *and* e*:*

$$\begin{aligned} \mathtt{in}(e, S') &\equiv \exists (s, e, s') \in A : s, s' \in S' \\ \mathtt{out}(e, S') &\equiv \exists (s, e, s') \in A : s, s' \notin S' \\ \mathtt{nocross}(e, S') &\equiv \exists (s, e, s') \in A : s \in S' \Leftrightarrow s' \in S' \\ \mathtt{enter}(e, S') &\equiv \exists (s, e, s') \in A : s \notin S' \wedge s' \in S' \\ \mathtt{exit}(e, S') &\equiv \exists (s, e, s') \in A : s \in S' \wedge s' \notin S' \end{aligned}$$

Note that $\mathtt{nocross}(e, S') = \mathtt{in}(e, S') \vee \mathtt{out}(e, S')$. We will abuse the notation and will use $\mathtt{nocross}(e, S', (s, e, s'))$ to denote a transition (s, e, s') that makes the predicate $\mathtt{nocross}(e, S')$ to hold, and the same for the rest of predicates.

The notion of a *region* is central for the synthesis of Petri nets. Intuitively, each region is a set of states and corresponds to a place in the synthesized Petri net.

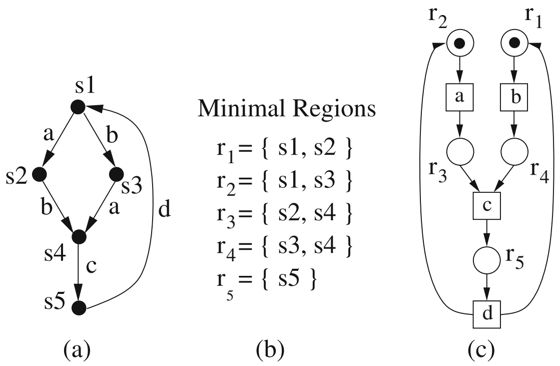

Fig. 1. (a) transition system, (b) its minimal regions, (c) Petri net obtained applying Algorithm of Figure 2

Definition 5 (Region). *A set of states* $r \subseteq S$ *in the transition system* $\mathsf{TS} = (S, E, A, s_{in})$ *is called a* region *if the following two conditions are satisfied for each event* $e \in E$*:*

- *(i)* $\mathtt{enter}(e, r) \;\Rightarrow\; \neg\mathtt{nocross}(e, r) \;\wedge\; \neg\mathtt{exit}(e, r)$
- *(ii)* $\mathtt{exit}(e, r) \;\Rightarrow\; \neg\mathtt{nocross}(e, r) \;\wedge\; \neg\mathtt{enter}(e, r)$

A region is a subset of states in which for *all* events, all transitions labeled with that event have exactly the same "entry/exit" relation. This relation will become the successor/predecessor relation in the Petri net. The event may always be either an *enter* event for the region (case (i) in the definition above), or always be an *exit* event (case (ii)), or never "cross" the region's boundaries, i.e. each transition labeled with e is *internal* or *external* to the region, when the antecedents of neither (i) nor (ii) hold. The transition corresponding to the event will be predecessor, successor or unrelated with the corresponding place respectively. Examples of regions are shown in Figure 1: from the transition system of Figure 1(a), some regions are enumerated in Figure 1(b). For instance, for region r_2, event a is an exit event, event d is an entry event while the rest of events do not cross the region. Let r and r' be regions of a transition system. A region r' is said to be a *subregion* of r if $r' \subset r$. A region r is a *minimal* region if there is no other non-empty region r' which is a subregion of r. Going back to the example of Figure 1, the regions shown in Figure 1(b) are all minimal regions of the transition system of Figure 1(a). On the other hand, the region $\{s1, s2, s3, s4\}$ is non-minimal. Each transition system $\mathsf{TS} = (S, E, A, s_{in})$ has two *trivial regions*: the set of all states, S, and the empty set. The set of non-trivial regions of TS will be denoted by R_{TS}.

A region r is a *pre-region* of event e if there is a transition labeled with e which exits r. A region r is a *post-region* of event e if there is a transition labeled with e which enters r. The sets of all pre-regions and post-regions of e are denoted ${}^\circ e$ and e°, respectively. By definition it follows that if $r \in {}^\circ e$,

Algorithm: Petri net synthesis

- For each event $e \in E$ generate a transition labeled with e in the Petri net;
- For each minimal region $r \in R_{\mathsf{TS}}$ generate a place $\hat{r}$;
- Place $\hat{r}$ contains a token in the initial marking iff the corresponding region r contains the initial state of TS s_{in};
- The flow relation is as follows: $e \in \hat{r}\bullet$ iff r is a pre-region of e and $e \in \bullet\hat{r}$ iff r is a post-region of e, i.e.,

$$F_{\mathsf{TS}} \stackrel{def}{=} \{(\hat{r}, e) | r \in R_{\mathsf{TS}} \;\wedge\; e \in E \;\wedge\; r \in {}^{\circ}e\} \\ \cup \{(e, \hat{r}) | r \in R_{\mathsf{TS}} \;\wedge\; e \in E \;\wedge\; r \in e^{\circ}\}$$

Fig. 2. Algorithm for Petri net synthesis from [13]

then all transitions labeled with e exit r. Similarly, if $r \in e^{\circ}$, then all transitions labeled with e enter r.

The procedure given by [13] to synthesize a Petri net, $N_{\mathsf{TS}} = (R_{\mathsf{TS}}, E, F_{\mathsf{TS}}, R_{s_{in}})$, from a transition system $\mathsf{TS} = (S, E, A, s_{in})$ is illustrated in Figure 2. Notice that only minimal regions are required in the algorithm [9]. Depending on the class of transition systems considered, the algorithm provides different guarantees: for an *elementary transition system*[1], the algorithm derives a Petri net with behavior isomorphic to the initial transition system. For *excitation-closed transition systems* (see Definition 8 below), the algorithm derives a Petri net with behavior bisimilar to the initial transition system. For this latter class, the algorithm generates 1-bounded Petri nets without self-loops [7]. The generalization of the synthesis algorithm to k-bounded Petri nets can be found in [5]. An example of the application of the algorithm is shown in Figure 1. The initial transition system and the set of its minimal regions is given in Figures 1(a) and (b), respectively. The synthesized Petri net is shown in Figure 1(c).

The computation of the minimal regions is crucial for the synthesis methods in [5, 7]. It is based on the notion of *excitation region* [11].

Definition 6 (Excitation region). *The excitation region of an event e, $\mathsf{ER}(e)$, is the set of states at which e is enabled, i.e.*

$$\mathsf{ER}(e) = \{s \mid \exists s' : (s, e, s') \in A\}$$

In Figure 1(a), the set $\mathsf{ER}(c) = \{s4\}$ is an example of an excitation region[2]. The set of minimal regions, needed in the synthesis algorithm of Figure 2, can be

[1] Elementary transition systems are a proper subclass of the transition systems considered in this paper, where additional conditions to the ones presented in Section 2.1 are required.

[2] Excitation regions are not regions in the terms of Definition 5. The term is used for historical reasons. For instance, $\mathsf{ER}(c)$ is not a region.

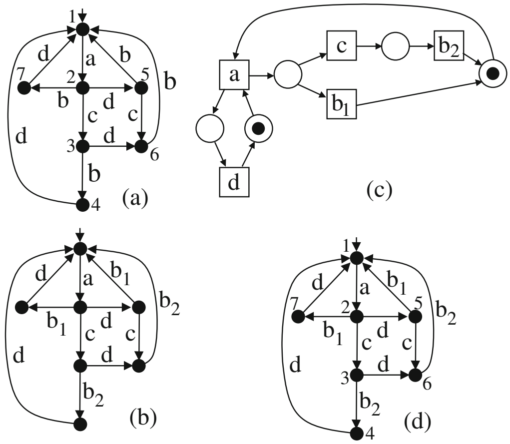

Fig. 3. (a) transition system (only numbers of states are shown) with minimal set of regions $\{s1, s4, s7\}$, $\{s1, s5, s6\}$, $\{s2, s3, s4, s7\}$ and $\{s2, s3, s5, s6\}$ (b) ECTS by label-splitting, (c) synthesized Petri net, (d) reachability graph of the Petri net

generated from the ERs of the events in a transition system in the following way: starting from the ER of each event, set expansion is performed on those events that violate the region condition. A pseudo-code of the expansion algorithm is given at the end of this subsection (as Algorithm 1).

The following lemma from [7] characterizes the states to be added in the expansion of ERs:

Lemma 1 (Set of states to become a region [7]). *Let* $\mathsf{TS} = (S, E, A, s_{in})$ *be a transition system. Let* $r \subset S$ *be a set of states such that* r *is not a region. Let* $r' \subseteq S$ *be a region such that* $r \subset r'$. *Let* $e \in E$ *be an event that violates some of the conditions for* r *to be a region. The following predicates hold for the sets* r *and* r':

1. $\mathtt{in}(e, r) \wedge \big(\mathtt{enter}(e, r) \vee \mathtt{exit}(e, r)\big) \implies$
   $\{s | \exists s' \in r : (s, e, s') \in A \vee (s', e, s) \in A\} \subseteq r'$
2. $\mathtt{enter}(e, r) \wedge \mathtt{exit}(e, r) \implies$
   $\{s | \exists s' \in r : (s, e, s') \in A \vee (s', e, s) \in A\} \subseteq r'$
3. $\mathtt{out}(e, r) \wedge \mathtt{enter}(e, r) \implies$
   $\big(\{s | \exists s' \in r : (s, e, s') \in A\} \subseteq r'\big) \vee \big(\{s | \exists s' \notin r : (s', e, s) \in A\} \subseteq r'\big)$
4. $\mathtt{out}(e, r) \wedge \mathtt{exit}(e, r) \implies$
   $\big(\{s | \exists s' \in r : (s', e, s) \in A\} \subseteq r'\big) \vee \big(\{s | \exists s' \notin r : (s, e, s') \in A\} \subseteq r'\big)$

In cases 1 and 2 above, the violating event e is converted into a `nocross` event, where only one way of expanding r is possible. However, in case 3 (4) there are two possibilities for expansion, depending on whether the violating event will be converted into a `nocross` or `enter` (`nocross` or `exit`) event.

Lemma 1 provides the basis for the derivation of the set of essential sets of states that will be the basis for the theory of this paper. Algorithm 1 presents

Algorithm 1. Expand_States [7]

Input: r is the set of states to be expanded,
explored is the set of expansions already generated
Output: R collects all regions

1 **begin**
2 **if** $r \in$ *explored* **then** return; `// avoid repeating computations`
3 **if** r *is a region* **then**
4 $R = R$ - $\{r_i | r \subset r_i\}$; `// remove supersets (non-minimal) of` r
5 **if** $\neg \exists r_j \in R : r_j \subseteq r$ **then** $R = R \cup \{r\}$; `//` r `added if minimal`
6 return;
7 **end**
8 find $e \in E$ violating some region condition in r;
9 $r' = r \cup \{$set of states to legalize $e\}$; `// Lemma 1: all cases`
10 Expand_States (r',explored,R); `// Recursion`
11 explored = explored $\cup \{r'\}$;
12 **if** *another expansion is needed* **then** `// Case 3 or 4 in Lemma 1`
13 $r' = r \cup \{$another set of states to legalize $e\}$
14 Expand_States (r',explored,R); `// Recursion`
15 explored = explored $\cup \{r'\}$
16 **end**
17 **end**

the pseudo-code for this procedure. The idea of the procedure is, starting from a given set of states r (in our setting it will be the excitation region of an event, as formalized below), if r is a region then update the region set found so far (R) to delete superset regions of r, and insert r into R if it is minimal according to R (lines 1–1). If r is not a region, the algorithm expands r by adding states that are listed in some of the cases that hold from Lemma 1 (lines 1–1). For cases 3 or 4 of Lemma 1, the algorithm performs the other expansion needed, in lines 1–1. Since initially, the set of states to start with is not a region, at least one case will provide a set of states for the expansion. The new set obtained is recursively processed in the same manner, to derive a new set and so on. Moreover, since in cases 3 and 4 there are two possibilities for expansion, this recursive procedure branches in each option. Importantly, the procedure generates all the minimal pre-regions of an event [7].

In summary, set expansion to legalize violating events in a set of states generates a binary exploration tree, whose leafs are the regions found and internal nodes are non-regions. An example of such tree can be found in Figure 4. The following definition formalizes the notion of essential set:

Definition 7 (Essential set of an event). *Let* $\mathsf{TS} = (S, E, A, s_{in})$ *be a transition system, and let* $e \in E$. $Essential(e, \mathsf{TS})$ *is the set of sets of states found by Algorithm 1, with initial set* $\mathsf{ER}(e)$*. Formally:*

– $\mathsf{ER}(e) \in Essential(e, \mathsf{TS})$,
– *Let* $B \subseteq S$ *with* $B \in Essential(e, \mathsf{TS})$*. Then every set* B' *constructed from* $r = B$ *as described in Lemma 1 is included in* $Essential(e, \mathsf{TS})$.

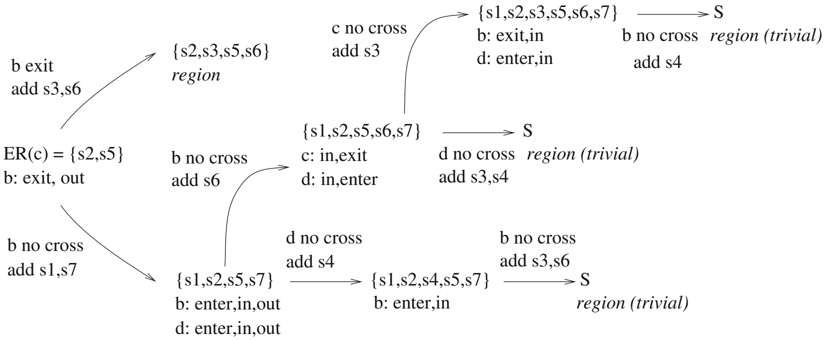

Fig. 4. Computation of essential sets ($S = \{s1, s2, s3, s4, s5, s6, s7\}$)

Notice that[3] $Essential(e, \mathsf{TS}) \subseteq \mathcal{P}(S)$, and $\forall S' \in Essential(e, \mathsf{TS}) : \mathsf{ER}(e) \subseteq S'$. For instance, in Figure 4 we show the computation of $Essential(c, \mathsf{TS})$ for the TS of Figure 3(a). Hence, $Essential(c, \mathsf{TS}) = \{\{s2, s5\}, \{s1, s2, s5, s7\}, \{s2, s3, s5, s6\}, \{s1, s2, s4, s5, s7\}, \{s1, s2, s5, s6, s7\}, \{s1, s2, s3, s5, s6, s7\}, \{s1, s2, s3, s4, s5, s6, s7\}\}$. For instance, the three sets of states $\{s1, s2, s5, s7\}$, $\{s1, s2, s5, s6, s7\}$ and $\{s2, s3, s5, s6\}$ in $Essential(c, \mathsf{TS})$ are computed according to case 4 in Lemma 1 on event b, while the set $\{s1, s2, s4, s5, s7\}$ is computed from set $\{s1, s2, s5, s7\}$ to deal with the violation caused by event d (case 3 in Lemma 1). Notice that the essential sets found are partitioned into regions and non-regions. Regions are in turn partitioned into minimal and non-minimal regions. In the example above, the sets $\{s2, s5\}$, $\{s1, s2, s5, s7\}$, $\{s1, s2, s4, s5, s7\}$ and $\{s1, s2, s3, s5, s6, s7\}$ are essential sets that are not regions, whereas the sets $\{s2, s3, s5, s6\}$, and $\{s1, s2, s3, s4, s5, s6, s7\}$ (the leafs of the tree shown in Figure 4) are essential sets that are regions, being the last one the trivial region.

2.3 Excitation-Closed Transition Systems

In this section we define formally the class of transition systems that will be considered in this paper.

Definition 8 (Excitation-closed transition systems). *A transition system* $\mathsf{TS} = (S, E, A, s_{in})$ *is an* excitation-closed *transition system (*ECTS*) if it satisfies the following two axioms:*

- *Excitation closure (*EC*): For each event e:* $\bigcap_{r \in {}^\circ e} r = \mathsf{ER}(e)$
- *Event effectiveness (*EE*): For each event e:* ${}^\circ e \neq \emptyset$

ECTSs include the class of elementary transition systems [7]. Interestingly, the excitation closure axiom (EC) for ECTSs is equivalent to the *forward closure* condition [10, 13] required for elementary transition systems.

[3] $\mathcal{P}(S)$ denotes the set of all subsets of S.

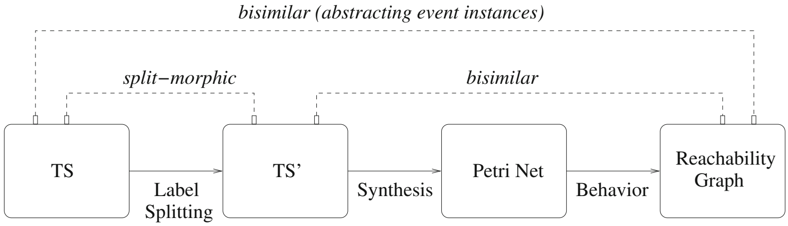

Fig. 5. Relationship between the different objects if label splitting is applied

The synthesis algorithm in Figure 2 applied to an ECTS produces a Petri net with reachability graph *bisimilar* to the initial transition system [7]. When the transition system is not excitation-closed, then it must be transformed to enforce that property. One possible strategy is to represent every event by multiple transitions labeled with different instances of the same label. This technique is called *label splitting.* Figure 3 illustrates the technique. The initial transition system, shown in Figure 3(a), is not an ECTS: the event c does not satisfy the EE axiom nor the EC axiom, since no pre-regions of c exist. In order for the EC/EE axioms on c to be satisfied, one can force the set of states $\{s2, s5\}$ (the excitation region of event c) to be a region by applying label splitting. The set $\{s2, s5\}$ is not a region because event b violates the region condition (some b-transitions `exit` and some others `nocross`). Hence, following this partition on the b-transitions with respect to $\{s2, s5\}$, the transition system is transformed by splitting the event b into the events b_1 and b_2, as shown in Figure 3(b), resulting in an ECTS. The synthesized Petri net, with two transitions for event b is shown in Figure 3(c). Transition system of Figure 3(b) is *split-morphic* [7] to the transition system of Figure 3(a): there exists a surjective mapping between the sets of events, where different instances (a_1, a_2, $\ldots$) of the single event a are mapped to a. The reachability graph of the Petri net of Figure 3(c), shown in Figure 3(d), is bisimilar to transition system of Figure 3(b). Moreover, if we abstract away the label indexes in the reachability graph of Figure 3(d), the equivalence relation between the transition system of Figure 3(a) and the reachability graph is bisimilarity: the relation $\pi = (si, si)$, for $1 \leq i \leq 7$ is a bisimulation if event instances are abstracted. Figure 5 shows the relationships between the original transition system, the transformed one obtained through label splitting, and the reachability graph of the synthesized Petri net.

Hence in Petri net synthesis label splitting might be crucial for the existence of a Petri net with bisimilar behavior. The following definition describes the general application of label splitting:

Definition 9 (Label splitting). *Let* $\mathsf{TS} = (S, E, A, s_{in})$ *be a transition system. The splitting of event* $e \in E$ *produces a transition system* $\mathsf{TS}' = (S, E', A', s_{in})$, *with $E' = E - \{e\} \cup \{e_1, \ldots, e_n\}$, and such that every transition $(s_1, e, s_2) \in A$ corresponds to exactly one transition (s_1, e_i, s_2), and the rest of transitions for events different from e in A are preserved in A'.*

Label splitting is a powerful transformation which always guarantees excitation closure: any TS can be converted into one where every transition has a different label. By definition, the obtained TS is ECTS but the size of the derived Petri net is equal to the size of the obtained ECTS. In this paper we aim at reducing the number of labels, thus reducing the size of the Petri net derived.

The work presented in this paper considers a particular application of the label splitting technique which is based on converting a set of states into a region, described in the next section.

3 Optimal Label Splitting to Obtain a Region

In this section the following problem is addressed: given a transition system $\mathsf{TS} = (S, E, A, s_{in})$ and a set of states $S' \subseteq S$ which is not a region, determine the minimal number of label splittings to be applied in order that S' becomes a region. This is a crucial step for the technique presented in the following section to satisfy the ECTS property for a transition system. The main contribution of this section is to show that the problem might be reduced to computing the *chromatic number* of a graph [17].

First, let us introduce the concept of *gradient graph*:

Definition 10 (Gradient Graph). *Given a transition system* $\mathsf{TS} = (S, E, A, s_{in})$*, a set $S' \subseteq S$ and an event $e \in E$, the* gradient graph *of e with respect to S' in* TS*, denoted as $\mathcal{GG}(e, S') = (A_e, M)$ is an undirected graph defined as:*

- $A_e = \{(s, x, s') | (s, x, s') \in A \ \wedge \ x = e\}$*, is the set of nodes, and*
- $M = \{\{v, v'\} | v, v' \in A_e \ \wedge$
  $[(\mathtt{enter}(e, S', v) \ \wedge \ (\mathtt{nocross}(e, S', v') \ \vee \ \ \mathtt{exit}(e, S', v'))) \ \vee$
  $(\mathtt{exit}(e, S', v) \wedge (\mathtt{nocross}(e, S', v') \vee \mathtt{enter}(e, S', v')))]\}$ *is the set of edges.*

Informally, the gradient graph contains as nodes the transitions of an event e, and an edge exists between two nodes which satisfy different predicates on set S', like one transition `enter` S' and the other does not, or one transition `exit` S' and the other does not. For instance, the gradient graph on event b and set of states $S' = \{s_2, s_5\}$ in the transition system of Figure 3(a) is shown in Figure 6(b) (for the sake of clarity we show in Figure 6(a) only the transitions on event b from Figure 3(a)).

A graph $G = (V, E)$ is *k-colourable* if there exists an assignment $\alpha : V \to \{1, 2, \ldots k\}$ for which any pair of nodes $v, v' \in V$ such that $\{v, v'\} \in E$ satisfy $\alpha(v) \neq \alpha(v')$. The *chromatic number*, $\chi(G)$, of a graph G is the minimum k for which G is k-colourable [17]. The rest of the section shows the relation between the chromatic number and the optimal label splitting to obtain a region.

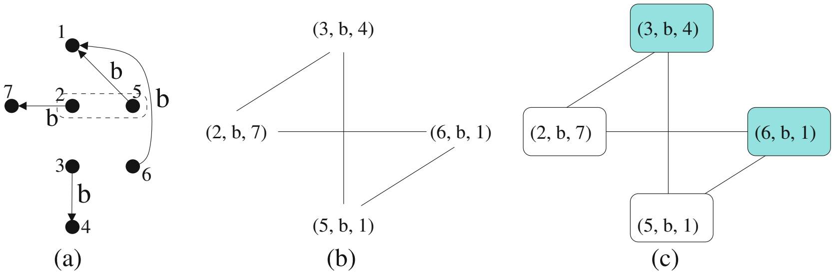

Fig. 6. (a) Projection of the transition system of Figure 3(a) showing only the transitions involving event b with the encircled set of states $\{s_2, s_5\}$, (b) $\mathcal{GG}(b, \{s_2, s_5\})$, (c) coloring

Definition 11 (Label splitting as gradient graph coloring). *Given a transition system* $\mathsf{TS} = (S, E, A, s_{in})$*, the label splitting of event* e *according to a coloring* α *of the gradient graph* $\mathcal{GG}(e, S')$ *produces the transition system* $\mathsf{TS}' = (S, E', A', s_{in})$ *where* $E' = E - \{e\} \cup \{e_1, \ldots, e_n\}$*, with* $\{e_1, \ldots, e_n\}$ *being the colors defined by the coloring of* $\mathcal{GG}(e, S')$*. Every transition* (s, e, s') *of event* e *is transformed into* $(s, e_{\alpha((s,e,s'))}, s')$*, whilst the rest of transitions of events in* $E - \{e\}$ *are preserved in* A'*.*

For instance, the label splitting of the transition system of Figure 3(a) according to the coloring shown on Figure 6(c) of $\mathcal{GG}(b, \{s_2, s_5\})$ is shown on Figure 3(b).

Proposition 1. *Given a transition system* $\mathsf{TS} = (S, E, A, s_{in})$ *and the gradient graph* $\mathcal{GG}(e, S')$*. If event* e *is split in accordance with a* $\chi(\mathcal{GG}(e, S'))$*-coloring of* $\mathcal{GG}(e, S')$ *then the new inserted events* $\{e_1, \ldots, e_{\chi(\mathcal{GG}(e,S'))}\}$ *satisfy the region conditions for* S' *(cf., Definition 5)).*

Proof. By contradiction: assume that there exists $e_i \in \{e_1, \ldots, e_{\chi(\mathcal{GG}(e,S'))}\}$ such that conditions of Definition 5 do not hold. Without loss of generality, we assume that there exist (s_1, e_i, s_2) and (s'_1, e_i, s'_2) for which predicates $\texttt{enter}(e_i, S', (s_1, e_i, s_2))$ and $\texttt{nocross}(e_i, S', (s'_1, e_i, s'_2))$ hold (the other cases can be proven similarly). But then the nodes (s_1, e, s_2) and (s'_1, e, s'_2) are connected by an edge in $\mathcal{GG}(e, S')$, but they are assigned the same color e_i. This is a contradiction. □

Notice that, when the graph $\mathcal{GG}(e, S')$ contains no arcs it means that for the set S' all the e-transitions satisfy the region predicates of Definition 5. This graph can be trivially colored with one color, i.e., $\chi(\mathcal{GG}(e, S')) = 1$. Hence, in that case, according to Definition 11, no increase in the number of labels arises due to the event splitting, e.g., $|E'| = |E - \{e\} \cup \{e_1\}| = |E| - 1 + 1 = |E|$, with E, E' being the alphabets of Definition 11. In this case the transformation is denoted *renaming*.

Definition 12 (Renaming). *If* $\chi(\mathcal{GG}(e, S')) = 1$ *then the application of label splitting transformation on event* e *for* S' *is called* renaming.

Clearly, renaming an event e is a cosmetic transformation which does not change any of the predicates of Definition 5 for the label of the new event on the set of states considered.

The following corollary provides the first main result of this section that relates coloring of the gradient graph to obtaining regions in a transition system.

Corollary 1. *Given a transition system* $\mathsf{TS} = (S, E, A, s_{in})$ *and a set* $S' \subseteq S$*. If every event* e *is split according to the colors required for achieving* $\chi(\mathcal{GG}(e, S'))$*-coloring, then* S' *is a region in the resulting transition system.*

Proof. It follows from iterative application of Proposition 1. □

In the following theorem we abuse the notation and extend the χ operator to sets of states, and define a lower bound to the number of label splittings:

Theorem 1. *Given a transition system* $\mathsf{TS} = (S, E, A, s_{in})$*,* $E = \{e_1, \ldots, e_n\}$ *and a set* $S' \subseteq S$*. Denote* $\chi(S') = \chi(\mathcal{GG}(e_1, S')) + \ldots + \chi(\mathcal{GG}(e_n, S'))$*. To make* S' *a region, the minimum number of labels needed including renaming is* $\chi(S')$*, and the minimum number of labels needed excluding renaming is* $\chi(S') - n$*.*

Proof. By contradiction: if there is a $k < \chi(S')$ such that only k labels are needed to convert S' into region, then there is an event $e \in E$ for which less than $\chi(\mathcal{GG}(e, S'))$ labels are used to split e for satisfying the conditions of Definition 5. This leads to a contradiction to the chromatic number of the graph $\mathcal{GG}(e, S')$. Finally, notice that formally the application of the transformation in Definition 11 removes all the original events (some of them are simply renamed into a new event). Hence it follows that the minimum number of labels excluding renaming is $\chi(S') - n$. □

Hence Theorem 1 establishes that at least $\chi(S') - n$ extra labels are needed in order for S' to become a region. Clearly, if apart from the splittings derived from $\chi(S')$, other splittings are additionally done, S' will still be a region. The theory presented in this section represents the core idea for the label splitting technique of this paper. The next section shows how to apply it to obtain ECTSs.

4 Optimal Label Splitting on Essential Sets for Synthesis

Given a non-ECTS, the following question arises: is there an algorithm to transform it into an ECTS with a minimal number of extra labels? This section addresses this problem, deriving sufficient conditions under which a positive answer can be given. As was done in previous work [7], in this paper we will restrict the theory to a particular application of the label splitting: instead of an arbitrary instantiation of Definition 9 which may split an event of a transition system in an arbitrary way, we will only consider the splittings used to convert essential sets into regions (Definition 11), a technique which has been shown in the previous section.

We tackle the problem in two phases: first, we show how the ECTS conditions for an event e (Definition 8) can be achieved by using the essential sets found in the expansion (see Lemma 1) of the excitation region $\mathsf{ER}(e)$, described in Definition 7. Then we show the conditions under which the strategy can be applied in the general case, i.e., considering all the events not satisfying some of the conditions required in Definition 8.

For the definitions and theorems in this section, we assume in the formalizations a transition system $\mathsf{TS} = (S, E, A, s_{in})$. We use $Witness(e, \mathsf{TS})$ to denote the sets of essential sets of an event e that are not regions and, if they are converted into regions, both the EC and EE axioms from Definition 8 on e will hold. Formally:

Definition 13 (Witness sets of an event). *Let $e \in E$. $Witness(e, \mathsf{TS})$ is defined as follows:*

$$\begin{aligned}\mathcal{C} = \{S_1, \ldots, S_k\} \in Witness(e, \mathsf{TS}) \iff & (\textstyle\bigcap_{q \in ({}^\circ e \,\cup\, \mathcal{C})} q) = \mathsf{ER}(e) \wedge \big(\forall S_i \in \mathcal{C} : \\ & S_i \in Essential(e, \mathsf{TS}) \wedge S_i \notin R_{\mathsf{TS}} \\ & \wedge \neg\mathtt{nocross}(e, S_i) \wedge \neg\mathtt{enter}(e, S_i)\big)\end{aligned}$$

A witness set contains non-regions S_i that are candidates to be pre-regions of the event (since both $\neg\mathtt{nocross}(e, S_i)$ and $\neg\mathtt{enter}(e, S_i)$ hold, and therefore the only possibility for e in S_i is to exit). The intersection of the sets forming a witness, together with the existing pre-regions of the event, is the excitation region of the event. Hence, if event e does not satisfy the EC or EE conditions from Definition 8, the singleton $\{\mathsf{ER}(e)\}$ will always be in $Witness(e, \mathsf{TS})$[4]. For instance, for the TS of Figure 3(a), a witness for event c is $\{\{s2, s5\}\}$. Notice that if an event satisfies both the EC and EE axioms from Definition 8, then its witness set is empty.

Finally, if $\mathcal{C} = \{S_1, \ldots, S_k\}$, we abuse the notation and use $\chi(\mathcal{C})$ to denote $\chi(\mathcal{GG}(e_1, S_1) \cup \ldots \cup \mathcal{GG}(e_1, S_k)) + \ldots + \chi(\mathcal{GG}(e_n, S_1) \cup \ldots \cup \mathcal{GG}(e_n, S_k))$, with $E = \{e_1, \ldots, e_n\}$. The union operator on gradient graphs is defined as $\mathcal{GG}(e, S_1) \cup \ldots \cup \mathcal{GG}(e, S_k) = (A_e, M_1 \cup \ldots \cup M_k)$, with A_e and M_i being the nodes and edges of the graph $\mathcal{GG}(e, S_i)$, for $i = 1, \ldots, k$ cf., Definition 10[5]. For the example of Figure 3(a), we have $\chi(\{\{s2, s5\}\}) = \chi(\mathcal{GG}(a, \{s2, s5\}) + \chi(\mathcal{GG}(b, \{s2, s5\}) + \chi(\mathcal{GG}(c, \{s2, s5\}) + \chi(\mathcal{GG}(d, \{s2, s5\}) = 1 + 2 + 1 + 1 = 5$.

First, we start by describing the minimal strategy to make an event satisfy the conditions of Definition 8:

---

[4] There is a special case where some states in $\mathsf{ER}(e)$ are connected through e-transitions, which can then be considered in a generalization of Definiton 13. For the sake of readability, we use this simple version of witness.

[5] We only consider the union of gradient graphs of the same event.

Proposition 2. *Let $\mathcal{C} = \{S_1, \ldots, S_k\} \in Witness(e, \mathsf{TS})$ such that $\chi(\mathcal{C})$ is minimal, i.e., $\forall \mathcal{C}' \in Witness(e, \mathsf{TS}) : \chi(\mathcal{C}) \leq \chi(\mathcal{C}')$. Then, if only label splitting on essential sets (Definition 11) is considered, $\chi(\mathcal{C})$ is the minimal number of labels needed to make the* EC *and* EE *axioms of Definition 8 on e to hold.*

Proof. First, it is clear that after applying label splitting on essential sets from a non-empty witness, the set of pre-regions of e becomes non-empty. We now prove the minimality of $\mathcal{C}$ by contradiction. Assume the EC/EE axioms can hold by the sole application of label splitting on essential sets and with less labels than $\chi(\mathcal{C})$, where $\chi(\mathcal{C})$ is minimal. Then this means that there is a set $\mathcal{C}'$ that can be used instead of $\mathcal{C}$ to make the EC and EE axioms to hold, and which requires fewer label splittings than $\chi(\mathcal{C})$. Since $\mathcal{C}' \notin Witness(e, \mathsf{TS})$ (otherwise $\mathcal{C}'$ will be the minimal witness instead of $\mathcal{C}$), then at least one set $S'_i \in \mathcal{C}'$ satisfies that $S'_i \notin Essential(e, \mathsf{TS})$ or $S'_i \in R_{\mathsf{TS}}$, since for the rest of witness conditions that can be violated (intersection not yielding $\mathsf{ER}(e)$ or e entering/crossing S'_i, S'_i cannot be used to make the EC and EE axioms to hold). In any of the two situations, we reach a contradiction: if $S'_i \notin Essential(e, \mathsf{TS})$, then the label splitting is not applied on essential sets, and if $S'_i \in R_{\mathsf{TS}}$, then $\mathcal{C}' \setminus \{S'_i\} \in Witness(e, \mathsf{TS})$, and clearly $\chi(\mathcal{C}) > \chi(\mathcal{C}' \setminus \{S'_i\})$. □

Given a non-ECTS, the optimal label splitting problem on essential sets is to determine which sets to convert into regions in order to satisfy, for each event, the EE and EC axioms, using the minimal number of labels.

Definition 14 (Optimal label splitting on essential sets). *Given the events $e_1, \ldots, e_k \in E$ violating the* EC *or* EE *axioms from Definition 8, define the universe $\mathcal{U}$ as $Witness(e_1, \mathsf{TS}) \cup \ldots \cup Witness(e_k, \mathsf{TS})$. The optimal label splitting problem is to determine sets $S_1, \ldots, S_n \in \mathcal{U}$ such that $\forall 1 \leq i \leq k : \exists \mathcal{C} \in Witness(e_i, \mathsf{TS}) : \mathcal{C} \subseteq \{S_1, \ldots, S_n\}$ and where $\chi(\{S_1, \ldots, S_n\})$ is minimal.*

Notice that Definition 14 is defined on the set of events violating the EC/EE condition, selecting a set of essential sets which both requires minimal number of labels and ensures the EC/EE axioms to hold for these events. An interesting result guarantees EC/EE preservation for those events that both satisfy initially the EC/EE axioms and were not split:

Proposition 3. *Given an event $e \in E$ such that the* EC/EE *axioms from Definition 8 hold, and e has not been selected for splitting. Then these axioms hold in the new transition system obtained after label splitting.*

Proof. Label splitting preserves regions: the predicates of Definition 5 that hold on each region will also hold if some event is split. Hence, the non-empty set of regions ensuring $\bigcap_{r \in {}^\circ e} r = \mathsf{ER}(e)$ and ${}^\circ e \neq \emptyset$ is still valid after label splitting. □

However, label splitting may split events for which the EC/EE axioms were satisfied or attained. Unfortunately, the new events appearing might not satisfy these axioms, as the following example demonstrates.

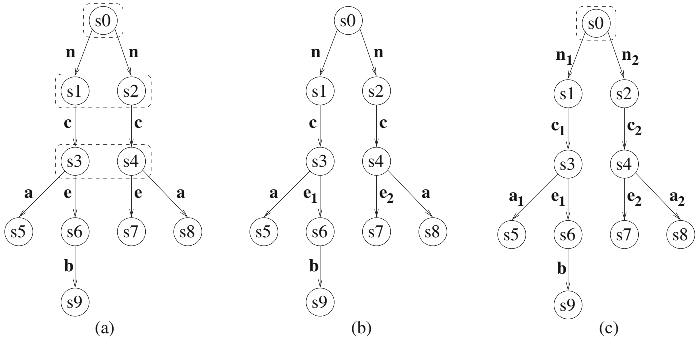

Fig. 7. (a) Initial transition system where minimal regions are drawed within dashed lines: all the events but b satisfy the EC axiom, (b) the splitting of e leads to new events e_1 and e_2 not satisfying the EC axiom, (c) the final ECTS, where all events have been split

Example 1. In the transition system of Figure 7(a) events n, c, a and e satisfy both the EC and EE axioms, since all these events have non-empty set of pre-regions whose intersection is the excitation region of the event. However, b does not satisfy neither the EC nor the EE axioms, and the splitting required for b to satisfy these axioms is done on the essential set $\{s6\}$, which requires to split the event e, resulting in the transition system of Figure 7(b). The new events e_1 and e_2 arising from the splitting of e do not satisfy the EC axiom (but do satisfy the EE axiom). This requires further splittings, which as in the case of b, force the splitting of events resulting in new events that do not satisfy the EC axiom. Four iterations are required to obtain the ECTS shown in Figure 7(c).

This example invalidates any label splitting strategy that aims at reaching excitation closure in just one iteration by only inspecting the violations to the EC/EE axioms: in general, when the splitting of some event is applied, its ER is divided into several ERs for which there might be no set of pre-regions which guarantees the satisfaction of the EC/EE axioms on these new events arising. Importantly, the label splitting technique preserves the regions, but new regions might be necessary for the new events arising from a splitting. Therefore, any label splitting technique that focuses on EC/EE violations must be an iterative method (see next section for such a method). However, if the new labels inserted do not incur violations to the EC/EE axioms, the presented technique guarantees the optimal label splitting:

Theorem 2. *Let* $\mathsf{TS}' = (S, E', A', s_{in})$ *be the transition system reached after splitting labels on events* $e_1, \ldots, e_k$ *that violate the* EC/EE *axioms of Definition 8 in transition system* TS*, using the minimal combination of witness* $S_1, \ldots, S_n$ *from Definition 14. Then, if the new events appearing satisfy the* EC/EE *axioms, the number of splittings performed is minimal and* TS' *is* ECTS.

Proof. The minimality of the witness according to Definition 14 ensures that no fewer splittings are possible to make $e_1, \ldots, e_k$ to satisfy the EC/EE axioms: if less splittings were possible by a different set of essential sets, then the essential sets chosen by Definition 14 will be that set. Moreover, the assumption implies that the new events arising from the splitting satisfy the EC/EE axioms. Finally, Proposition 3 guarantees that events that were not split still satisfy the EC/EE axioms. The set of events E' is partitioned into these two sets, and therefore TS' is ECTS. □

Notice that the current characterization of the label splitting problem, as seen from Theorem 2, is dynamic, i.e., the new labels appearing when problematic labels are split must satisfy the axioms required in an ECTS in order to guarantee a minimal number of labels. In contrast, a static characterization will simply impose constraints on the initial transition system in order to determine the minimal number of labels required for reaching an ECTS. Static techniques for solving the label splitting problem may represent an alternative to the techniques described in this paper.

5 A Greedy Algorithm for Iterative Label Splitting

The optimal label splitting problem presented in the previous section considers the minimal application of the label splitting technique to satisfy the EE/EC conditions for the initial set of events. Moreover, it was shown that in general it may not be possible to reach an ECTS by a single application of the technique.

The label splitting problem is similar to the *weighted set cover* (WSC) problem [6]. The WSC problem can be informally described as follows: given a finite set Y and a family $\mathcal{F}$ of subsets of Y, such that every element of Y belongs to at least one subset in $\mathcal{F}$, and each set S_i in $\mathcal{F}$ has an associated weight w_i, determine a minimum-weight cover, i.e. a set $\mathcal{C} \subseteq \mathcal{F}$ that contains all the elements in Y and the sum of the weights of the elements in $\mathcal{C}$ is minimal. The WSC problem is known to be NP-hard, and can be formulated as an *integer linear programming* (ILP) model [6]. This section is devoted to show how to encode in an ILP model the label splitting problem, inspired by the corresponding ILP formulation for the WSC problem. Moreover, an algorithm for label splitting that iterates until excitation closure is achieved is presented.

Clearly, the label splitting problem is akin to the WSC problem: if $e_1, \ldots, e_k$ are the events that do not satisfy the EC or EE axioms, then $\mathcal{F} = Witness(e_1, \mathsf{TS}) \cup \ldots \cup Witness(e_k, \mathsf{TS})$ is the family of subsets to consider, and the problem is to find a set of sets of states $\{S_1, \ldots, S_n\}$ that i) at least one witness is covered for each event e_i, $1 \le i \le k$, and ii) requires the minimal number of labels.

The ILP model for the WSC problem simply minimizes the sum of individual costs of each element S_i while covering the set Y. In our setting, we would like to minimize the number of labels, and hence will use the function χ in the cost function of the ILP model. Then, based on the ILP model for the WSC problem, the resulting ILP model for the label splitting problem is:

$$min \sum_{\mathcal{C} \in \mathcal{W}} \chi(\mathcal{C}) \cdot X_{\mathcal{C}} \quad (1)$$

subject to

$$\forall e \in \{e_1, \dots, e_k\} : \sum_{\mathcal{C} \in Witness(e,\mathsf{TS})} X_{\mathcal{C}} \geq 1$$
$$X_{\mathcal{C}} \in \{0, 1\}$$

where $e_1, \dots e_k, \in E$ are the set of events that do not satisfy the EC/EE axioms, $\mathcal{W} = Witness(e_1, \mathsf{TS}) \cup \dots \cup Witness(e_k, \mathsf{TS})$ and $X_{\mathcal{C}}$ denotes the binary variable that selects witness $\mathcal{C}$ to be or not in the solution. A solution to the ILP model (1) will then minimize the sum of labels needed to convert the witnesses selected for each event violating the EE/EC axioms into regions[6].

Imagine that a solution to model (1) contains a pair of witnesses $\mathcal{C}_1 = \{S_1, S_2, S_3\}$ and $\mathcal{C}_2 = \{S_2, S_3, S_4\}$. Unfortunately, the equality

$$\chi(\mathcal{C}_1 \cup \mathcal{C}_2) = \chi(\mathcal{C}_1) + \chi(\mathcal{C}_2) \quad (2)$$

does not hold since the fact that $\mathcal{C}_1$ and $\mathcal{C}_2$ share some set (e.g., share sets S_2 and S_3) implies that at the right hand side of the equality colors are counted in every gradient graph of an event, while in the left hand side only one gradient graph (which is the union of these gradient graphs) is considered. In that situation one may be counting twice the labels (colors) needed in the right part of the equality of (2). Notice that ILP model (1) minimizes the sum of labels of individual witnesses (as done in the right part of equality (2)), and hence it may provide non-optimal solutions to the label splitting problem. In spite of this, by using an ILP solver to find an optimal solution for model (1), one may have a strategy to proceed into deriving an ECTS with reasonable (although maybe not-optimal) number of label splittings. Hence, our proposed strategy based on the ILP model (1) is greedy.

Algorithm 2 presents the iterative strategy to derive an ECTS. It first computes the minimal regions of the initial transition system. The algorithm iterates when the EE/EC axioms do not hold for at least one event (function ExcitationClosed(TS',$\mathcal{R}$)) will return false). Then the main loop of the technique starts by collecting the witnesses for events violating the EE/EC axioms of the current transition system, which are provided in the set $\mathcal{W}$ (line 5). Then model (1) based on $\mathcal{W}$ is created and an ILP solver is invoked in line 6 that will provide a solution (a set of witnesses that ensure the satisfaction of the EE/EC axioms for the problematic events). Notice that for the sake of clarity of the algorithm, we provide in line 6 the sets that form the cover instead of providing the particular witness selected for each event (i.e., given a solution $\mathcal{C}_1, \dots, \mathcal{C}_k$ of model (1), $\{S_1, \dots, S_n\} = \bigcup_{1 \leq i \leq k} \mathcal{C}_i$). In line 7 the splitting of labels corresponding to $\chi(\{S_1, \dots, S_n\})$ is performed, ensuring that sets $S_1, \dots, S_n$ become regions. The new regions are appended to the regions found so far (which are still

[6] In model (1) it is important to use the $\geq$ in the constraints part since in an optimal solution it may happen that an event is covered by more than one witness.

Algorithm 2. GreedyIterativeSplittingAlgorithm

Input: Transition system $\mathsf{TS} = (S, E, A, s_{in})$
Output: Excitation-closed transition system $\mathsf{TS}' = (S, E', A', s_{in})$ bisimilar to TS

begin
  $\mathsf{TS}' = \mathsf{TS}$
  $\mathcal{R}$ = GenerateMinimalRegions(TS)
  while *not (ExcitationClosed(*TS'*,*$\mathcal{R}$*))* **do**
    $\mathcal{W}$ = CollectWitnesses(TS')
    $(S_1, \ldots, S_n)$ = Solution ILP model (1)
    TS' = SplitLabels(TS', $S_1, \ldots, S_n$)
    $\mathcal{R} = \mathcal{R} \cup \{S_1, \ldots, S_n\}$
  end
end

regions, see proof of Proposition 3), and the excitation closure is re-evaluated to check convergence.

Although in general the presented iterative technique is not guaranteed to provide an ECTS that has the minimal number of labels, in terms of convergence the improvements with respect to the previous (also non-optimal) approaches [5, 7] are:

- the whole set of events violating the EE/EC axioms are considered in every iteration: in previous work only one event is considered at a time, and
- for every event violating the EE/EC axioms, the necessary splittings are applied to attain excitation closure: on the previous work, only one of the splittings was applied.

Hence the *macro* technique presented in this section is meant to speed-up the achievement of the excitation closure, when compared to the *micro* techniques presented in the literature.

6 Extensions and Applications of Label Splitting

The theory presented in this paper may be extended into several directions. Here we informally enumerate some possible extensions.

Extension to k-Bounded Synthesis. For the sake of clarity, we have restricted the theory to safe Petri nets. The extension to k-bounded Petri nets can be done by adapting the notion of gradient graph to use *gradients* instead of the predicates required in Definition 5, and lift the notion of region to multisets instead of sets. Informally, the gradient of an event is an integer value which represents the modification made by the event on the multiplicity of the region (see [2, 5] for the formal definition of gradient). When more than one gradient is assigned to a given event in a multiset, then the multiset is not a region. Given a *multiset* r which is not a region, the essential sets will be those multisets $r' \geq r$

such that the number of gradients for some event has decreased, i.e., r' is a sound step towards deriving a region from multiset r. The excitation closure definition (see Definition 8) for the general case should be the one described in [5]: a multiset which is a region will be a pre-region of an event if its *support* (set of states with multiplicity different than 0) includes the excitation region of the event. The support sets of pre-regions will be intersected and for the event to satisfy the EC/EE axioms this intersection should be equal to the excitation region of the event. The concept of the witness set of an event can be naturally adapted to use multisets instead of sets by using the support of each multiset accordingly.

Label Splitting for Petri Net Classes. Maybe one of the strongest points of the theory of regions is the capability to guide the synthesis for particular Petri net classes [4, 7]. The core idea is to select, among the set of all regions, those that satisfy particular conditions. Here we list some interesting classes that may be obtained by restricting the label splitting technique presented in this paper.

- In a *marked graph* [12], every region (place) must have exactly one event satisfying the `enter` predicate (input transition) and one satisfying the `exit` predicate (output transition), and the rest of events should be `nocross`. In general, when the goal is a particular Petri net class defined by conditions on the structure with respect to the connections to the places, one can simply select the essential sets that, if transformed into regions by using the theory of Section 3, will still satisfy these conditions.
- In a *state machine* [12], every transition has exactly one input and one output place. One important feature of the regions of a state machine is that they form a *partitioning* of the state space of the net [3]. Figure 8(a) shows an example, where eight regions form the partitioning. The corresponding state machine is depicted in Figure 8(b). Hence, the selection of essential sets may be guided to construct partitionings of the state space and therefore the derivation of a state machine will be guaranteed. The generalization to *conservative Petri nets* is also possible by lifting the notion of partitioning to *multi-partitioning* [3].
- In a *Free-choice* net [8], if two different transitions share a place of their pre-sets, then their pre-sets should be equal (i.e., ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset \implies {}^\bullet t_1 = {}^\bullet t_2$). When transforming essential sets into regions by label splitting, an extra condition is required for selecting the sets to split: for each pair of regions derived, either none or the whole set of `exit` events should be shared between the two. This crucial restriction will guarantee to derive Free-choice nets.
- In an *Ordinary* Petri net, arcs have always weight one. The extension to k-bounded Petri nets described before can be restricted to only consider gradients in the range $\{-1, 0, +1\}$, while allowing any multiplicity in the multisets derived. This will ensure the derivation of ordinary Petri nets.

Clearly, by restricting the class of Petri nets to be derived, one can no longer ensure bisimilarity between the derived Petri net and the initial transition system. This is sometimes acceptable, specially in the context of *Process Mining*.

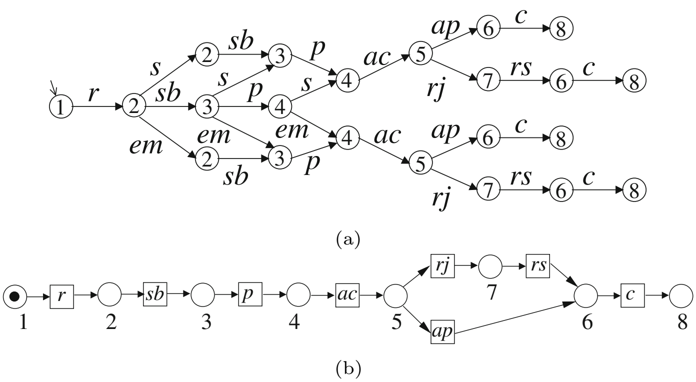

Fig. 8. (a) Example of region partitioning. The eight regions shown in the figure are identified by natural numbers, i.e., the region where label r exits has states with number 1, the region where r enters has states with number 2 (three states), and so on, (b) State machine corresponding to the partitioning shown in (a) (places contain the region number to which they correspond). Notice that this state machine does not contain the transitions em and s, since they are concurrent with some transitions in the net of Figure 8(b).

Application to Process Mining. The *discovery* of a formal process model from a set of executions (called *log*) of a system, the *conformance* of a model in describing a log, and the *enhancement* of a process model are the main disciplines in the novel area of Process Mining [16]. Petri nets are one of the most popular models used in Process Mining. Importantly, the synthesis conditions of this paper are relaxed in the context of Process Mining [5], since the derivation of a *simple* model is often preferred even if some extra traces are possible in the model while not observed in the log (this phenomenon is called *overapproximation*). Simplicity here may have various meanings, but it usually implies that the underlying graph of the derived Petri net can be understood by a human. By requiring the minimal number of labels to be split, the approach presented in this paper may be interesting in the context of Process Mining, since the by-product of this optimization is the derivation of Petri nets with minimal number of transitions, thus being more readable. Hence, in Process Mining the synthesis conditions may be required for particular events in order to control the degree of overapproximation that the derived Petri net will have. When a particular event is forced to satisfy the synthesis conditions, it can be guaranteed with the theory of this paper that the number of labels will be minimal if label splitting on essential sets is applied.

7 Conclusions

This paper has presented a fresh look at the problem of label splitting, by relating it to some well-known NP-complete problems like chromatic number or set covering. In a restricted application of label splitting based on essential sets, optimality is guaranteed if certain conditions hold.

Extensions of the technique have been discussed, motivating the appropriateness of using essential sets for label splitting since it allows to control the derived Petri net: the theory can be adapted to produce k-bounded Petri nets and interesting Petri net classes. Also, one potential application of the technique is presented in the area of Process Mining.

As future work, there are some research lines to follow. First, regarding the technique presented in this paper, it will be very important to study stronger constraints on the transition system that makes the label splitting on essential sets technique not to be an iterative process, as happens with the one presented in this paper. Second, addressing the general problem with unrestricted application of splitting (i.e., not using essential sets but instead arbitrary selections of labels to split) might be an interesting direction to follow. Moreover, incorporating the presented techniques (e.g., Algorithm 2) into our synthesis tool [5] will be considered.

Acknowledgements. We would like to thank anonymous reviewers for their comments that help improving the final version of the paper. This work has been supported by the projects FORMALISM (TIN2007-66523) and TIN2011-22484.

References

1. Arnold, A.: Finite Transition Systems. Prentice Hall (1994)
2. Bernardinello, L., Michelis, G.D., Petruni, K., Vigna, S.: On synchronic structure of transition systems. In: Proceedings of the International Workshop on Structures in Concurrency Theory (STRICT), pp. 69–84 (May 1995)
3. Carmona, J.: Projection approaches to process mining using region-based techniques. Data Min. Knowl. Discov. 24(1), 218–246 (2012)
4. Carmona, J., Cortadella, J., Kishinevsky, M.: A Region-Based Algorithm for Discovering Petri Nets from Event Logs. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 358–373. Springer, Heidelberg (2008)
5. Carmona, J., Cortadella, J., Kishinevsky, M.: New region-based algorithms for deriving bounded Petri nets. IEEE Transactions on Computers 59(3), 371–384 (2009)
6. Cormen, T.H., Stein, C., Rivest, R.L., Leiserson, C.E.: Introduction to Algorithms. McGraw-Hill Higher Education (2001)
7. Cortadella, J., Kishinevsky, M., Lavagno, L., Yakovlev, A.: Deriving Petri nets from finite transition systems. IEEE Transactions on Computers 47(8), 859–882 (1998)
8. Desel, J., Esparza, J.: Free-choice Petri Nets. Cambridge Tracts in Theoretical Computer Science, vol. 40. Cambridge University Press (1995)
9. Desel, J., Reisig, W.: The synthesis problem of Petri nets. Acta Inf. 33(4), 297–315 (1996)
10. Ehrenfeucht, A., Rozenberg, G.: Partial (Set) 2-Structures. Part I, II. Acta Informatica 27, 315–368 (1990)
11. Kishinevsky, M., Kondratyev, A., Taubin, A., Varshavsky, V.: Concurrent Hardware: The Theory and Practice of Self-Timed Design. John Wiley and Sons, London (1993)

12. Murata, T.: Petri Nets: Properties, analysis and applications. Proceedings of the IEEE, 541–580 (April 1989)
13. Nielsen, M., Rozenberg, G., Thiagarajan, P.: Elementary transition systems. Theoretical Computer Science 96, 3–33 (1992)
14. Petri, C.A.: Kommunikation mit Automaten. PhD thesis, Bonn, Institut für Instrumentelle Mathematik (1962) (technical report Schriften des IIM Nr. 3)
15. Reisig, W., Rozenberg, G. (eds.): APN 1998. LNCS, vol. 1491. Springer, Heidelberg (1998)
16. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (May 2011)
17. West, D.B.: Introduction to Graph Theory. Prentice-Hall (1996)