Improved Question-Guided Stubborn Set Methods for State Properties

Lars Michael Kristensen[1] and Antti Valmari[2]

[1] University of Aarhus, Department of Computer Science
DK-8000 Aarhus C., DENMARK, Email: kris@daimi.au.dk
[2] Tampere University of Technology, Software Systems Laboratory
PO Box 553, FIN-33101 Tampere, FINLAND, Email: ava@cs.tut.fi

Abstract. We present two new question-guided stubborn set methods for state properties. The first method makes it possible to determine whether a marking is reachable in which a given state property holds. It generalises the results on stubborn sets for state properties recently suggested by Schmidt in the sense that that stubborn set method can be seen as an implementation of our more general method. We propose also alternative, more powerful implementations that have the potential of leading to better reduction results. This potential is demonstrated on some practical case studies.

As an extension of the first method, we present a second method which makes it possible to determine if from all reachable markings it is possible to reach a marking where a given state property holds. The novelty of this method is that it does not rely on ensuring that no transition is ignored in the reduced state space. Again, the benefit is in the potential for better reduction results.

Topics: System design and verification using nets, Analysis and synthesis of nets, Computer tools for nets.

1 Introduction

State space methods have proven powerful in the analysis and verification of concurrent systems. Unfortunately, the state spaces of systems tend to grow very rapidly when systems become bigger. This well-known phenomenon is referred to as *state explosion*, and it is a serious problem for the use of state space methods in the analysis of real-life systems.

Many techniques for alleviating the state explosion problem have been suggested, such as the *stubborn set methods* [7, 10]. They comprise a subgroup of rather similar methods first suggested in the late 80's and early 90's [3–5]. These methods are based on the fact that the total effect of a set of concurrent transitions is independent of the order in which the transitions occurs. Therefore, it often suffices to investigate only one or some orderings in order to reason about the behaviour of the system.

This paper presents two new stubborn set methods which make it possible to reason about *state properties*. A state property is a property that talks about only one marking. For instance, $M(p) \leq 10$ is a state property, whereas $\exists M' \in [M\rangle : M'(p) > M(p)$ is not.

The first stubborn set method makes it possible to answer the following question: "is it possible to reach a marking where a given state property holds?" The method is *question-guided*, i.e., it takes a state property as input and generates a reduced state space. This reduced state space will contain a marking where the property holds if and only if there exists a reachable marking in which the state property holds. This method is important, because with it one can, e.g., find place bounds, and check reachability of a (perhaps incompletely specified) marking, more efficiently than with existing stubborn set methods [6, 8, 9]. The method presented is based on the ideas in [6], but tries to compute better stubborn sets. This can potentially lead to better reduction results.

The second question-guided method makes it possible to answer the question: "is it possible from all reachable markings to reach a marking where a given state property holds?" This method can for instance be used to check liveness of a single transition with better reduction results then an earlier methods [9] that check liveness of all transitions simultaneously. It can also be used to check whether a given (perhaps incompletely specified) marking is a home marking more efficiently than with the technique described in [6].

The paper is organised as follows. Section 2 recalls the basic facts of Place/-Transition Nets (PT-nets), state spaces, and stubborn sets used in the rest of this paper. Section 3 gives an informal introduction to the first stubborn set method by means of a small example. Sections 4-7 formally develop the new stubborn set methods, and Sect. 8 considers their implementation. Section 9 discusses applications of the first method to boundedness properties of PT-nets. Section 10 gives some numerical data on the performance of the first method on some case studies. Finally, we sum up the conclusions in Sect. 11.

2 Background

This section briefly summarises the basic facts and notation of PT-nets, state spaces, and stubborn sets used in the rest of the paper. We assume that the reader is familiar with the dynamic behaviour of PT-nets and the basic ideas of state spaces (also called occurrence graphs or reachability graphs/tress).

Definition 1. *A* **Place/Transition Net** *is tuple $PTN = (P, T, A, W, M_I)$, where P is a finite set of places, T is a finite set of transitions such that $P \cap T = \emptyset$, $A \subseteq (P \times T) \cup (T \times P)$ is a set of arcs, $W : A \to \mathbb{N}_+$ is an arc weight function, and $M_I : P \to \mathbb{N}_0$ is the initial marking.* □

We use M_I as the initial marking instead of the more conventional M_0. This allows us to use M_0 as the first marking of *occurrence sequences* which does not necessarily start in the initial marking. If a transition t is *enabled* in a marking M_1 (denoted $M_1[t\rangle$), then t may *occur* in M_1 yielding some marking M_2. This is written $M_1[t\rangle M_2$. Extending this notation, an occurrence sequence is denoted $M_0[t_1\rangle M_1 \cdots M_{n-1}[t_n\rangle M_n$ and satisfies $M_{i-1}[t_i\rangle M_i$ for $1 \leq i \leq n$.

When the intermediate markings in an occurrence sequence are not important we will write it as $M_0[t_1t_2\cdots t_n\rangle M_n$. A *reachable marking* is a marking which can be obtained (reached) by an occurrence sequence starting in the initial marking. By $[M\rangle$ we denote the set of markings reachable from a marking M. For a place (transition) x, $\bullet x$ denotes the set of input transitions (places) of x, and $x\bullet$ is a similar notation for output transitions (places). The notation is extended to sets by taking the union of $\bullet x$ $(x\bullet)$ over each member x of the set. In a marking M, the marking of a place p is denoted $M(p)$.

Definition 2. *The* **Full State Space** *of a PT-net is a directed graph* $SG = (V, E)$*, where* $V = [M_I\rangle$ *and* $E = \{\ (M_1, t, M_2) \in V \times T \times V \mid M_1[t\rangle M_2\ \}$. □

In the rest of this paper we assume that a PT-net (P, T, A, W, M_I) with a *finite* full state space $SG = (V, E)$ is given. For some of the stubborn set algorithms presented in this paper we will exploit the *strongly connected components*. A strongly connected component (SCC) is a non-empty set C of reachable markings such that if $M \in C$ then $C = \{\ M' \mid M' \in [M\rangle \wedge M \in [M'\rangle\ \}$. An SCC is said to be a *terminal strongly connected component* iff $M \in C$ implies $[M\rangle \subseteq C$.

State space construction with stubborn sets follows the same procedure as the construction of the full state space of a PT-net, with one exception. When processing a marking, a set of transitions, the so-called *stubborn set*, is constructed. Only the enabled transitions in the stubborn set are used to construct successor markings. This means that only a subset of the relation $M[t\rangle M'$ is used for the construction of the reduced state space. We denote this subset by $M[t\rangle_{SSG}M'$, and define $M[t_1, \cdots, t_n\rangle_{SSG}M'$, $[M\rangle_{SSG}$ as for the full state space but now based on the relation $M[t\rangle_{SSG}M'$. The stubborn set reduced state space (from now on called the *SS state space*) can be defined as a directed graph $SSG = (V_{SSG}, E_{SSG})$ based on the relation $M[t\rangle_{SSG}M'$ in a similar way as the full state space. We define the (terminal) SCCs for the SS state space analogously to the case for the full state space.

The choice of stubborn sets depends on the properties that are being analysed or verified of the system. Many stubborn set algorithms are surveyed in [10]. They all assume that the stubborn sets used in each marking satisfy certain conditions, and stubborn set methods for different properties are obtained by using different conditions. However, it is common to almost all of them that the conditions listed below should hold. Below $T_{\mathsf{s}}(M)$ denotes the stubborn set used in the marking M.

D1 If $t \in T_{\mathsf{s}}(M_0)$, $t_1, \ldots, t_n \notin T_{\mathsf{s}}(M_0)$, $M_0[t_1t_2\cdots t_n\rangle M_n$, and $M_n[t\rangle M_n'$, then there is M_0' such that $M_0[t\rangle M_0'$ and $M_0'[t_1t_2\cdots t_n\rangle M_n'$.

D2 If M_0 has an enabled transition, then there is at least one transition $t_{\mathsf{k}} \in T_{\mathsf{s}}(M_0)$ such that if $t_1, \ldots, t_n \notin T_{\mathsf{s}}(M_0)$ and $M_0[t_1t_2\cdots t_n\rangle M_n$, then $M_n\,[t_{\mathsf{k}}\rangle$. Any transition with this property is called a *key transition* of $T_{\mathsf{s}}(M_0)$.

The conditions D1 and D2 as such are not suited for constructing stubborn sets since they refer to occurrence sequences. Therefore, the construction of stubborn sets is in practice implemented by relying on rules that refer only to the structure of the PT-net and the current marking, and which express sufficient conditions to make D1 and D2 hold. The tutorial [10] lists a number of such. Below we give a simple proposition which guarantees that D1 and D2 hold. The proposition analyses the dependencies between transitions at a rather coarse level, and it is not optimal in the sense of yielding smallest possible stubborn sets and smallest SS state spaces. We will use it only for illustration purposes.

Proposition 1. *The conditions D1 and D2 hold if the following hold for every* $t \in T_{\mathsf{s}}(M)$*:*

1. *If* $\exists t_1 \in T : M\ [t_1\rangle$*, then* $\exists t_2 \in T_{\mathsf{s}}(M) : M\ [t_2\rangle$.
2. *If* $\neg M\ [t\rangle$*, then* $\exists p \in {\bullet t} : M(p) < W(p,t)\ \wedge\ {\bullet p} \subseteq T_{\mathsf{s}}(M)$.
3. *If* $M\ [t\rangle$*, then* $({\bullet t}){\bullet} \subseteq T_{\mathsf{s}}(M)$. □

The important aspect of Prop. 1 is that the three items can be read as rules. Item 1 specifies that if there is an enabled transition, then an enabled transition has to be in the stubborn set. Item 2 specifies that if a disabled transition t has been included in the stubborn set, some place p in the preset of t which does not contain enough tokens for t to be enabled must be chosen and its preset included. Finally, item 3 specifies that if an enabled transition t has been included then the postset of the preset of t must be included. A number of algorithms for constructing stubborn sets based on propositions like Prop. 1 are given in [10].

3 An Example

In this section we introduce the first of our improved stubborn set methods in an informal way using the simple PT-net shown in Fig. 1. Figure 2 shows the full state space of this PT-net. Node 1 corresponds to the initial marking. Each arc has an associated label giving the name of the transition to which it corresponds. For a node n we denote the corresponding marking by M_n.

Suppose that we want to check that there exists a reachable marking in which the place p_{10} contains at least two tokens. This can be expressed as the *state property* $\phi \equiv M(p_{10}) \geq 2$. M_9 is the only such marking.

The stubborn set method in [6], in the following referred to as the *attractor set method*, would define an *attractor set* in M_1, denoted $A_\phi(M_1)$, for the *atomic state proposition* $M(p_{10}) \geq 2$. The role of the attractor set is to ensure that in each step of the SS state space construction, progress is made towards a marking where the property holds. The attractor set in M_1 would consist of the transitions which can add tokens to p_{10}. Hence $A_\phi(M_1) = \{t_5, t_6\}$. The attractor set method requires the attractor set to be a subset of the stubborn set in each marking. If we apply Prop. 1, then the stubborn set in M_1 will be $\{t_1, t_2, t_3, t_4, t_5, t_6\}$. Hence both enabled transitions (t_1 and t_2) are in the stubborn set in M_1.

If we consider the marking M_2 then the attractor set remains the same as in M_1 and Prop. 1 gives us $\{t_2, t_4, t_5, t_6\}$ as the stubborn set. Again, all enabled transitions are included in the stubborn set. The situation in M_3 is symmetric to M_2. In M_4, the transition t_2 will be in the stubborn set. The situation in M_6

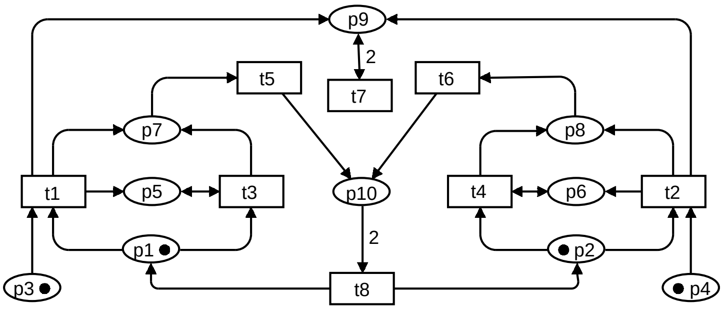

Fig. 1. Example PT-net.

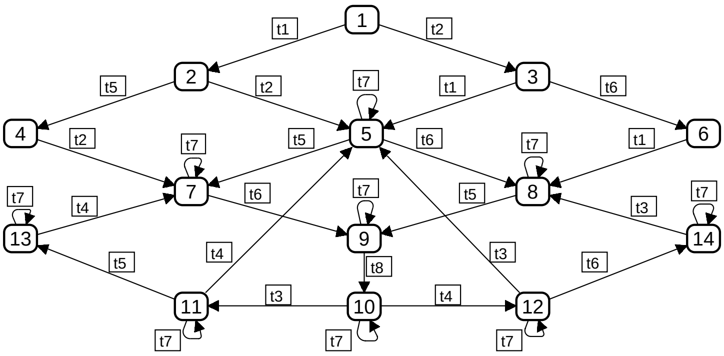

Fig. 2. Full state space for the PT-net in Fig. 1.

is symmetric to M_4, and in M_5, the transitions t_5 and t_6 will be in the stubborn set. In M_7 and M_8, the transition t_6 and t_5, respectively, will be in the stubborn set. In conclusion this means that the attractor set method yields an SS state space consisting of markings M_1 to M_9.

It can however be observed that it is possible to select stubborn sets during the construction of an SS state space with fewer enabled transitions than those required by the attractor set method. This could potentially lead to more reduction. The basic idea in our new method is to relax the requirement that the attractor set must *always* be contained in the stubborn set.

Suppose that the requirement imposed by the attractor set were totally removed. From Prop. 1 it follows that in M_1 we can select $\{t_2, t_4\}$ or $\{t_1, t_3\}$ as the stubborn set. Suppose that we select $\{t_1, t_3\}$. Proposition 1 implies that it is possible to select $\{t_5\}$ or $\{t_2, t_4\}$ as the stubborn set in M_2. If in M_2 we select the latter, then in M_5 we can select $\{t_5\}$, $\{t_6\}$, or $\{t_7\}$ as the stubborn set.

If in M_5 we select the stubborn set consisting of $\{t_7\}$ only, then the construction of the SS state space will terminate at this point, since M_5 is already included in it. This means that we would wrongly conclude that there does not exist a marking in which ϕ holds. The problem is that we have not ensured *progress* towards such a marking. The attractor set method ensures progress in each marking of the SS state space by *always* including the attractor set in the stubborn set. Instead of this strong requirement we will ensure that from each marking in the SS state space *eventually* progress can be made, i.e., a marking is reachable in the SS state space in which progress is made.

For this purpose we introduce the notion of *up sets*. An up set is a set of transitions chosen such that at least one transition in it has to occur in order to make the state property hold. Hence the up sets are similar to attractor sets. However, unlike the attractor set method, we will not require that the up set is always contained in the stubborn set. Moreover, we will additionally exploit that the state properties which we consider are growing Boolean functions. This makes it possible to ensure progress towards the property by either reducing the length of an occurrence sequence leading to a marking where the state property holds, or by increasing the number of atomic state propositions which are satisfied. This requirement will ensure that t_5 or t_6 is in the stubborn set in M_5. Similarly, it will ensure that t_6 is in the stubborn set in M_7, and that t_5 is in the stubborn set in M_8.

Ensuring eventual progress is however not sufficient for preserving state properties. As a simple example, suppose that we want to show that a marking is reachable in which $M(p_3) = 0$ and $M(p_4) = 1$. This corresponds to showing that M_2 or M_4 is reachable. If $\{t_2, t_4\}$ is selected as the stubborn set in M_1, then neither of M_2 and M_4 will be in the SS state space. The problem is that in M_1 the only enabled transition in the stubborn set is t_2, and an occurrence of this transition can change the value of the state property from True to False. To account for this we introduce the notion of *down sets*. A down set is a set of transitions chosen such that a transition in the down set has to occur in order to make the property not hold. We will ensure that if an enabled transition which is in the down set is in the stubborn set, then the transitions in the up set are also in the stubborn set. This will ensure that if t_2 is in the stubborn set in M_1 then also t_1 is.

4 State Properties

We consider state properties expressed as formulas that are composed of so-called *atomic state propositions* using only the logical operators "$\wedge$" and "$\vee$" and parentheses "(" and ")". For a state property ϕ we denote its atomic state propositions by $\varphi_1, \varphi_2, \ldots, \varphi_n$, and let $I = \{1, 2, \ldots, n\}$ denote the set of indices of the atomic state propositions. The atomic state propositions and state properties are interpreted on the markings of the PT-net, and the resulting truth values are denoted by $\varphi_i(M)$ and $\phi(M)$. The atomic state propositions are defined according to the following syntax, where p, p_1, and p_2 denote arbitrary places and k is an integer constant.

$$\begin{aligned}\varphi_i \ ::= \ & M(p) \geq k \mid M(p_1) \geq M(p_2) \mid M(p) = k \mid M(p_1) = M(p_2) \mid \\ & M(p) \leq k \mid M(p_1) > M(p_2) \mid M(p) \neq k \mid M(p_1) \neq M(p_2)\end{aligned}$$

We have not included $M(p) > k$ and $M(p) < k$ as atomic state propositions since they can be expressed as $M(p) \geq k+1$ and $M(p) \leq k-1$, respectively. The set of atomic state propositions could be extended provided that the corresponding up and down sets to be defined in Sect. 5 are implemented properly.

Above only conjunction and disjunction were allowed as the Boolean operators. However, the atomic state propositions are closed under negation (p_1 and p_2 may be swapped when needed), so formulas which use negation can always be re-written to a form allowed by the above syntax using *De Morgan's equivalences* (i.e., $\neg(\phi_1 \vee \phi_2) = \neg\phi_1 \wedge \neg\phi_2$ and $\neg(\phi_1 \wedge \phi_2) = \neg\phi_1 \vee \neg\phi_2$). Therefore, the syntax does not restrict generality. It is however important for the correctness of the later algorithms that formulas are given in a negation-free form, i.e., that they have been preprocessed before being provided as input to the algorithms.

Definition 3. *Let M be a marking and ϕ a state property constructed from the atomic state propositions $\{ \varphi_i \mid i \in I \}$. The set indices of the atomic state propositions which are satisfied in M is denoted $\mathsf{on}_\phi(M)$. The set of the indices of the atomic state propositions which are not satisfied in M is denoted $\mathsf{off}_\phi(M)$. Formally:*

$$\mathsf{on}_\phi(M) = \{ i \in I \mid \varphi_i(M) \} \text{ and } \mathsf{off}_\phi(M) = \{ i \in I \mid \neg\varphi_i(M) \} \qquad \Box$$

If we let $\mathbb{B} = \{\mathsf{True}, \mathsf{False}\}$, treat the φ_i's as argument symbols, and define $\mathsf{False} \leq \mathsf{True}$, then a state property formula ϕ determines a monotonically increasing Boolean function from $\mathbb{B}^n$ to $\mathbb{B}$.

The following proposition lists important properties of the state property formulas which will be exploited later.

Proposition 2. *Let M and M' be markings and ϕ a state property constructed from the atomic state propositions $\{ \varphi_i \mid i \in I \}$. Then the following holds:*

1. $\forall i \in I : \varphi_i(M) \leq \varphi_i(M') \Rightarrow \phi(M) \leq \phi(M')$.
2. $\phi(M) \wedge \neg\phi(M') \Rightarrow \exists i \in I : \varphi_i(M) \wedge \neg\varphi_i(M')$. □

Item 1 states that ϕ is a monotonically increasing Boolean function. Item 2 states that if ϕ is satisfied in M but not in M' then there exists at least one atomic state proposition which is satisfied in M but not satisfied in M'. Item 2 is a consequence of item 1.

5 Up/Down and Satisfiability Sets

To describe the required properties of the stubborn sets, we define two sets of transitions related to a state property ϕ: an *up set* and a *down set*. The *up set* of ϕ in a marking M is a set of transitions chosen such that if ϕ does not hold in M then at least one transition in the up set must occur before ϕ can start to hold. The *down set* of ϕ is a set of transitions chosen such that it contains at least all transitions whose occurrence can change the value of some atomic state proposition φ_i of ϕ from True to False. In addition to these two sets we define the *satisfiability set* of ϕ in M as a set of indices of the atomic state propositions such that at least one atomic state proposition that has its index in the set has to change its value from False to True in order to make the state property hold.

The implementation of concrete up and down sets will be determined from the atomic state propositions and Boolean combinators. However, the properties of up and down sets are general concepts and not tied to the specific set of state properties considered in this paper. Therefore, we define up and down sets as properties of a set of transitions. A similar remark applies to satisfiability sets.

Definition 4. *Let ϕ be a state property constructed from the atomic state propositions $\{ \varphi_i \mid i \in I \}$ and let $M_0 \in [M_I\rangle$. A set of transitions $T' \subseteq T$ has the* **up set property** *in M_0 with respect to ϕ iff the following holds for all occurrence sequences $M_0[t_1 t_2 \cdots t_n\rangle M_n$ starting in M_0:*

$$\neg\phi(M_0) \wedge \phi(M_n) \Rightarrow \exists j :\ 1 \leq j \leq n \wedge t_j \in T'$$

A set of transitions $T' \subseteq T$ has the **down set property** *with respect to ϕ iff the following holds for all markings $M, M' \in [M_I\rangle$, all $t \in T$, and all $i \in I$:*

$$M\ [t\rangle\ M' \wedge \varphi_i(M) \wedge \neg\varphi_i(M') \Rightarrow t \in T'$$

A set of indices $J \subseteq I$ has the **satisfiability set property** *in M_0 with respect to ϕ iff the following holds for all occurrence sequences $M_0[t_1 t_2 \cdots t_n\rangle M_n$:*

$$\neg\phi(M_0) \wedge \phi(M_n) \Rightarrow \exists i \in J : \neg\varphi_i(M_0) \wedge \varphi_i(M_n) \qquad \square$$

The properties of up set and satisfiability set are relative to the current marking whereas the down set property is not. This is deliberate and due to the way our methods will later use these sets. It is worth observing that the definition of up (down) set property allows approximations of the up (down) sets to be used: if $T' \subseteq T''$ and T' has the up (down) set property then also T'' has the up (down) set property. A similar remark applies to indices and the satisfiability set property. This will be exploited later once we show how to construct such sets. Moreover, if a marking in which ϕ holds is reachable from a marking M then a satisfiability set in M exists and it is non-empty because of Prop. 2. From now on we will assume that we have an algorithm that given a state property ϕ produces some down set down_ϕ, and additionally given a marking produces some up set $\mathsf{up}_\phi(M)$ and some satisfiability set $\mathsf{sat}_\phi(M)$. We will give such an algorithm in Sect. 8.

6 Preserving Reachability of State Properties

This section presents the new stubborn set method for determining whether a reachable marking exists in which a given state property holds. The method consists of obeying the D1 condition from Sect. 2 and two additional conditions formulated in the following definition. An explanation of the definition will be given below.

Definition 5. *Let M be a marking and ϕ a state property constructed from the atomic state propositions $\{\ \varphi_i \mid i \in I\ \}$. A set $T_\mathsf{s}(M) \subseteq T$ is* **Reachability of a State Property Preserving (RSPP) stubborn** *in M, iff the following hold:*

D1 *If $t_1, \ldots, t_n \notin T_\mathsf{s}(M)$, $t \in T_\mathsf{s}(M)$, $M[t_1 t_2 \cdots t_n\rangle M_n$, and $M_n[t\rangle M'_n$, then there is M' such that $M[t\rangle M'$ and $M'[t_1 t_2 \cdots t_n\rangle M'_n$.*

SPP1 *If $\neg\phi(M)$ and $\exists t : M\,[t\rangle \wedge t \in \mathsf{down}_\phi \wedge t \in T_\mathsf{s}(M)$ then $\mathsf{up}_\phi(M) \subseteq T_\mathsf{s}(M)$.*

SPP2 *For every $i \in \mathsf{sat}_\phi(M)$ there is an occurrence sequence $M_0[t_1\rangle M_1[t_2\rangle \cdots [t_n\rangle M_n$ such that $M = M_0$, t_j is a key transition of $T_\mathsf{s}(M_{j-1})$ for $1 \leq j \leq n$, and $\phi(M_n) \vee \mathsf{up}_{\varphi_i}(M_n) \subseteq \bigcup_{j=0}^{n} T_\mathsf{s}(M_j)$.* □

The intuitive purpose of SPP1 is to ensure that a next step in the SS state space can be taken in such a way that we do not get further away from a marking where ϕ holds. SPP1 requires that if we have taken an enabled transition in the down set, then we have also included the transitions in the up set. The latter transitions represent a step towards a marking where ϕ holds, since we know that a transition in the up set has to occur in order to make ϕ hold. Therefore if one transition makes regress then there is another transition that makes progress. It is also possible that no enabled transition makes regress or progress.

SPP2, on the other hand, is there to *ensure* progress – to ensure that we will eventually get closer to a marking where ϕ holds. If ϕ holds in M then SPP2 holds trivially since we can then choose $n = 0$. If SPP2 does not take us directly to a marking where ϕ holds, then it ensures that there is a path in the SS state space where we eventually try every transition in the up set of some atomic state proposition which has to change its value. This represents progress, since such an additional atomic state proposition has to be satisfied in order to make the state property hold. SPP2 states its requirement to every element $i \in \mathsf{sat}_\phi(M)$ because $\mathsf{sat}_\phi(M)$ is an upper approximation and we do not necessarily know which member is important.

We now turn to the correctness of the RSPP-stubborn set method. The key to establishing correctness is the following lemma.

Lemma 1. *Let ϕ be a state property, $SG = (V, E)$ the full state space, and $SSG = (V_{SSG}, E_{SSG})$ an SS state space constructed using RSPP-stubborn sets. Let $M_0 \in V_{SSG}$ be a marking such that $\neg\phi(M_0)$ and for which there exists an occurrence sequence $M_0[t_1\rangle M_1[t_2\rangle \cdots [t_n\rangle M_n$ such that $\phi(M_n)$ holds. Then there is a marking $M'_0 \in [M_0\rangle_{SSG}$ such that the following holds.*

1. *There are transitions $t'_1, t'_2, \ldots, t'_m$ and markings $M'_1, M'_2, \ldots, M'_m$ such that $M'_0[t'_1\rangle M'_1[t'_2\rangle \cdots [t'_m\rangle M'_m$ and $\phi(M'_m)$ holds.*
2. *The occurrence sequence in item 1 leading to a marking where ϕ holds is no longer than the original occurrence sequence, i.e., $m \leq n$.*
3. *The length of the occurrence sequence in item 1 has decreased, i.e., $m < n$, or the set of the atomic state propositions of ϕ which are satisfied has grown, i.e., $\mathsf{on}_\phi(M_0) \subset \mathsf{on}_\phi(M'_0)$.* □

Proof of Lemma 1. Since $\neg\phi(M_0)$ and $\phi(M_n)$ then $\mathsf{sat}_\phi(M_0)$ contains an i such that $\neg\varphi_i(M_0)$ and $\varphi_i(M_n)$. Since $\neg\phi(M_0)$ and $T_{\mathsf{s}}(M_0)$ (the RSPP-stubborn set in M_0) satisfies the condition SPP2 there exist key transitions $\bar{t}_1,\ldots,\bar{t}_k$ and markings $\bar{M}_0,\ldots,\bar{M}_k \in V_{SSG}$ such that $M_0 = \bar{M}_0[\bar{t}_1\rangle_{SSG}\bar{M}_1[\bar{t}_2\rangle_{SSG}\cdots[\bar{t}_k\rangle_{SSG}\bar{M}_k$, and $\phi(\bar{M}_k) \vee \mathsf{up}_{\varphi_i}(\bar{M}_k) \subseteq \bigcup_{j=0}^{k} T_{\mathsf{s}}(\bar{M}_j)$.

If $\phi(\bar{M}_j)$ holds for some $0 \leq j \leq k$ then the claim holds by choosing $M_0' = \bar{M}_j$ and $m = 0$. In this case $m < n$ since $\neg\phi(M_0)$ and $\phi(M_n)$. From now on we may therefore assume that $\forall j, 0 \leq j \leq k : \neg\phi(\bar{M}_j)$. In the rest of the proof the following fact is needed:

(1) If $\{t_1,\ldots,t_n\} \cap T_{\mathsf{s}}(\bar{M}_h) = \emptyset$ for every $0 \leq h < l \leq k$, then, due to the key transition property of $\bar{t}_1,\ldots,\bar{t}_k$ and D1, there are markings $\hat{M}_0,\ldots,\hat{M}_l$ such that $M_n = \hat{M}_0[\bar{t}_1\cdots\bar{t}_l\rangle\hat{M}_l$ and $\bar{M}_l[t_1\cdots t_n\rangle\hat{M}_l$. Furthermore, if we assume that there exists a smallest index h such that $\bar{t}_{h+1} \in \mathsf{down}_\phi$ then $\phi(\hat{M}_0),\ldots,\phi(\hat{M}_h)$ hold due to the down set property. Thus, $\neg\phi(\bar{M}_h)$ and $\phi(\hat{M}_h)$ and SPP1 implies that $\emptyset \neq \{t_1,\ldots,t_n\} \cap \mathsf{up}_\phi(\bar{M}_h) \subseteq T_{\mathsf{s}}(\bar{M}_h)$ contrary to our assumption. Consequently, $\bar{t}_1,\ldots,\bar{t}_l \notin \mathsf{down}_\phi$ and we therefore have $\mathsf{on}_\phi(\bar{M}_0) \subseteq \mathsf{on}_\phi(\bar{M}_1) \subseteq \cdots \subseteq \mathsf{on}_\phi(\bar{M}_l)$, $\mathsf{on}_\phi(\hat{M}_0) \subseteq \mathsf{on}_\phi(\hat{M}_1) \subseteq \cdots \subseteq \mathsf{on}_\phi(\hat{M}_l)$, and $\phi(\hat{M}_0),\ldots,\phi(\hat{M}_l)$ hold.

We now split the proof in two cases.

Case A: $\{t_1,\ldots,t_n\} \cap T_{\mathsf{s}}(\bar{M}_j) \neq \emptyset$ for some $0 \leq j \leq k$. In this case we can pick the smallest such j and apply (1) for $l = j$. Since $T_{\mathsf{s}}(\bar{M}_j)$ contains at least one of the transitions $t_1, t_2, \ldots, t_n$, then we can pick the first such transition t_h and apply D1 on $\bar{M}_j[t_1\cdots t_{h-1}t_h t_{h+1}\cdots t_n\rangle\hat{M}_j$ to obtain a marking M'' such that $\bar{M}_j[t_h\rangle M''[t_1\cdots t_{h-1}t_{h+1}\cdots t_n\rangle\hat{M}_j$. The claim now holds with $M_0' = M''$ and $m = n-1$.

Case B: $\{t_1,\ldots,t_n\} \cap T_{\mathsf{s}}(\bar{M}_j) = \emptyset$ for every $0 \leq j \leq k$. In this case (1) gives us that $\mathsf{on}_\phi(\bar{M}_0) \subseteq \mathsf{on}_\phi(\bar{M}_k)$ and $\mathsf{on}_\phi(\hat{M}_0) \subseteq \mathsf{on}_\phi(\hat{M}_k)$. If $\mathsf{on}_\phi(\bar{M}_0) = \mathsf{on}_\phi(\bar{M}_k)$ then we must have one of the $t_1, t_2, \ldots, t_n$ in $\mathsf{up}_{\varphi_i}(\bar{M}_k)$ and by SPP2 also in $\bigcup_{j=0}^{k} T_{\mathsf{s}}(\bar{M}_j)$ which contradicts the assumption that $\{t_1,\ldots,t_n\} \cap T_{\mathsf{s}}(M_j) = \emptyset$ for every $0 \leq j \leq k$. Therefore $\mathsf{on}_\phi(\bar{M}_0) \subset \mathsf{on}_\phi(\bar{M}_k)$ and the claim holds with $M_0' = \bar{M}_k$ and $m = n$. □

The following theorem states that if there exists a marking in the SS state space from which it is possible to reach a marking where the state property holds then the SS state space also contains a marking in which the state property holds. The correctness of the RSPP stubborn set method follows immediately from the theorem by letting $M_0 = M_I$.

Theorem 1. *Let ϕ be a state property, $SG = (V, E)$ be the full state space, $SSG = (V_{SSG}, E_{SSG})$ an SS state space constructed using RSPP-stubborn sets, and let $M_0 \in V_{SSG}$. Then:*

$$\exists M \in [M_0\rangle : \phi(M) \Leftrightarrow \exists M' \in [M_0\rangle_{SSG} : \phi(M') \qquad \square$$

Proof of Thm. 1. The $\Leftarrow$ direction follows from the fact that $V_{SSG} \subseteq V$ and $E_{SSG} \subseteq E$. For establishing the $\Rightarrow$ direction we apply Lemma 1 inductively to obtain $M_1 \in [M_0\rangle_{SSG}, M_2 \in [M_0\rangle_{SSG}, \ldots$, until we find an $M_n \in [M_0\rangle_{SSG}$ such that $\phi(M_n)$ holds. The induction hypothesis is that there is a marking M_ϕ^i and an occurrence sequence σ_i such that $M_i[\sigma_i\rangle M_\phi^i$ and $\phi(M_\phi^i)$ holds. When $i = 0$ this holds with $M_\phi^0 = M$.

Define the *distance* $\Delta(M', \sigma, \phi)$ between a marking $M' \in V_{SSG}$ and a marking $M_\phi \in V$ which satisfies ϕ and which can be reached from M' by the occurrence sequence σ as follows ($|\sigma|$ denotes the length of the occurrence sequence σ):

$$\Delta(M', \sigma, \phi) = (|I| + 1) \cdot |\sigma| + |\mathsf{off}_\phi(M')|$$

The reason for the rather complicated definition of distance is that if at a marking we choose a transition which decreases the length of the occurrence sequence leading to a marking where ϕ holds we may at the same time switch some of the atomic state propositions off.

If ϕ does not hold in M_{i-1}, then Lemma 1 gives a marking $M_i \in [M_{i-1}\rangle_{SSG}$, an occurrence sequence σ_i, and a marking M_ϕ^i such that $M_i[\sigma_i\rangle M_\phi^i$ and ϕ holds in M_ϕ^i. Items 2 and 3 of Lemma 1 ensure that $\Delta(M_i, \sigma_i, \phi) < \Delta(M_{i-1}, \sigma_{i-1}, \phi)$. Clearly $0 \leq \Delta(M_i, \sigma_i, \phi) < \infty$, so eventually this process terminates in a marking $M_n \in [M_0\rangle_{SSG}$ in which ϕ holds. □

7 Preserving Home State Properties

This section presents a new stubborn set method for determining whether from all reachable markings it is possible to reach a marking where a given state property holds. This can be formally expressed as determining whether $\forall M \in [M_I\rangle : \exists M' \in [M\rangle : \phi(M')$. The method presented is based on the observation that by negation this is the same as determining whether a reachable marking exists from which it is *not* possible to make the given state property hold. This can be expressed as $\exists M \in [M_I\rangle : \forall M' \in [M\rangle : \neg\phi(M')$. We will use "$\phi \in [M\rangle$" as an abbreviation of $\exists M' \in [M\rangle : \phi(M')$ (from M a marking M' can be reached where ϕ holds), and "$\phi \in [M\rangle_{SSG}$" as an abbreviation of $\exists M' \in [M\rangle_{SSG} : \phi(M')$.

The method consists of obeying the conditions from the RSPP-stubborn set method from Sect. 6 and two additional conditions formulated in the following definition.

Definition 6. *Let M be a marking and ϕ a state property. A set $T_\mathsf{s}(M) \subseteq T$ is* **Home State Property Preserving (HSPP) Stubborn** *in M, iff $T_\mathsf{s}(M)$ is RSPP stubborn in M and the following hold:*

D2′ *If $t_1, \ldots, t_n \notin T_\mathsf{s}(M)$, $t \in T_\mathsf{s}(M)$, $M[t_1 t_2 \cdots t_n\rangle M_n$, and $M\,[t\rangle$, then $M_n\,[t\rangle$.*

SPP3 *For every $t \in \mathsf{down}_\phi$ there is an occurrence sequence $M_0\,[t_1\rangle \cdots [t_n\rangle\, M_n$ such that $M = M_0$, $t_j \in T_\mathsf{s}(M_{j-1})$ for $1 \leq j \leq n$, and $t \in T_\mathsf{s}(M_n)$.* □

The intuitive purpose of $T_{\mathsf{s}}(M)$ being RSPP stubborn in the context of this method is to ensure that we from any marking in the SS state space always attempt to make ϕ hold (recall that we are trying to show that there exists a reachable marking from which ϕ cannot be made to hold). The D2′ condition is like the D2 condition from Sect. 2, except that it requires all enabled transitions in the stubborn set to be key transitions and allows $T_{\mathsf{s}}(M) = \emptyset$. Together with the D1 condition inherited from RSPP this implies that HSPP stubborn sets are *strong stubborn sets* [10]. SPP3 is there to *ensure* progress, to ensure that we eventually get closer to a marking from which ϕ cannot be made to hold. This is formulated in terms of transitions in the down set since such a transition has to occur in order to make ϕ not hold.

The correctness of the HSPP stubborn set method follows immediately from the following theorem by letting $M_0 = M_I$.

Theorem 2. *Let ϕ be a state property, $SG = (V, E)$ be the full state space, $SSG = (V_{SSG}, E_{SSG})$ an SS state space constructed using HSPP stubborn sets, and let $M_0 \in V_{SSG}$. Then:*

$$\exists M \in [M_0\rangle : \phi \notin [M\rangle \Leftrightarrow \exists M_{SSG} \in [M_0\rangle_{SSG} : \phi \notin [M_{SSG}\rangle_{SSG} \qquad \Box$$

Proof of Thm. 2. The $\Leftarrow$ direction follows immediately from Thm. 1. We prove the $\Rightarrow$ direction by showing the following for a strictly decreasing sequence of values of n:

(1) There are $M_0^n, t_1^n, \ldots, t_n^n$ and M^n such that $M_0^n \in [M_0\rangle_{SSG}$, $M_0^n\ [t_1^n t_2^n \cdots t_n^n\rangle\ M^n$ and $\phi \notin [M^n\rangle$.

Initially we get (1) for some finite non-negative value of n from the left hand side of the theorem if we choose $M_0^n = M_0$. If $\phi \notin [M_0^n\rangle$, then $\phi \notin [M_0^n\rangle_{SSG}$, so M_0^n can be chosen as the M_{SSG} and the right hand side of the theorem holds. This happens at the latest when $n = 0$. Therefore, we get our result by showing that if (1) holds for some n and $\phi \in [M_0^n\rangle$, then (1) holds also for some m such that $0 \leq m < n$.

So we assume that $\phi \in [M_0^n\rangle$ holds. Thm. 1 asserts the existence of $t_1'', \ldots, t_h''$ and $M_0'', \ldots, M_h''$ such that $M_0^n = M_0''$, $M_0''[t_1''\rangle_{SSG} M_1''[t_2''\rangle_{SSG} \cdots [t_h''\rangle_{SSG} M_h''$ and $\phi(M_h'')$. We first establish the existence of an occurrence sequence $\bar{M}_0[\bar{t}_1\rangle_{SSG}\bar{M}_1$ $[\bar{t}_2\rangle_{SSG} \cdots [\bar{t}_k\rangle_{SSG}\bar{M}_k$ such that $M_0^n = \bar{M}_0$ and $\{t_1^n, \ldots, t_n^n\} \cap T_{\mathsf{s}}(\bar{M}_j) \neq \emptyset$ for some $1 \leq j \leq k$. We split the proof in two cases.

Case A: If at least one $t_1^n, \ldots, t_n^n$ belongs to $T_{\mathsf{s}}(M_0'') \cup \cdots \cup T_{\mathsf{s}}(M_h'')$, then we just choose $k = h$ and $\bar{t}_i = t_i''$ and $\bar{M}_i = M_i''$ for $1 \leq i \leq h$.

Case B: If none of $t_1^n, \ldots, t_n^n$ is in $T_{\mathsf{s}}(M_0'') \cup \cdots \cup T_{\mathsf{s}}(M_h'')$, then due to D1 and D2′ there is M_h' such that $M_h''[t_1^n \cdots t_n^n\rangle M_h'$ and $M^n[t_1'' \cdots t_h''\rangle M_h'$. Because of $\phi \notin [M^n\rangle$ we know that $\neg\phi(M_h')$. Because $\phi(M_h'')$ holds, there is $t_i^n \in \{t_1^n, \ldots, t_n^n\}$ such that $t_i^n \in \mathsf{down}_\phi$. Therefore, SPP3 gives the desired occurrence sequence.

We can now choose the smallest j such that $\{t_1^n, \ldots, t_n^n\} \cap T_{\mathsf{s}}(\bar{M}_j) \neq \emptyset$. Let i be the smallest number such that $t_i^n \in T_{\mathsf{s}}(\bar{M}_j)$. Due to D1 and D2′ there is M^{n-1} such that $\bar{M}_j[t_1^n \cdots t_n^n\rangle M^{n-1}$ and $M^n[\bar{t}_1 \cdots \bar{t}_j\rangle M^{n-1}$. We have $\phi \notin [M^{n-1}\rangle$, because otherwise $\phi \notin [M^n\rangle$ would not hold. Due to D1 there is M_0^{n-1} such that $\bar{M}_j[t_i^n\rangle_{SSG} M_0^{n-1}[t_1^n \cdots t_{i-1}^n t_{i+1}^n \cdots t_n^n\rangle M^{n-1}$. We thus have (1) for $n-1$. □

8 Implementation

We now consider the implementation of the RSPP and HSPP stubborn set methods presented in the previous sections. In Sect. 8.1 we show how to construct up, down, and satisfiability sets. In Sect. 8.2 we discuss different ways of implementing the conditions D1, D2′, and SPP1-3.

8.1 Implementation of Up/Down and Satisfiability Sets

In this section we show how to define up, down and satisfiability sets for the state properties considered in this paper. The construction is in all three cases specified inductively using the syntactical structure of the state properties. We end the section with a proposition which states that the defined up, down, and satisfiability sets posses the up, down, and satisfiability set properties as defined in Def. 4. First we give the definition of up sets.

Definition 7. *Let M be a marking and ϕ a state property. The up set $\mathsf{up}_\phi(M)$ in M is defined as follows. If ϕ holds in M we define $\mathsf{up}_\phi(M) = \emptyset$. If ϕ does not hold in M we define $\mathsf{up}_\phi(M)$ according to the following cases:*

Case $\phi \equiv M(p) \leq k$ *: $\mathsf{up}_\phi(M)$ consists of the transitions which can remove tokens from p and which add at most k tokens to p:*

$$\mathsf{up}_\phi(M) = \{\, t \in T \mid W(p,t) > W(t,p) \wedge W(t,p) \leq k \,\}$$

Case $\phi \equiv M(p) = k$ *: If p contains too few tokens then $\mathsf{up}_\phi(M)$ consists of the transitions which can add tokens and do not require additional tokens to be present on p, and if p contains too many tokens then $\mathsf{up}_\phi(M)$ consists of the transitions which can remove tokens from p and which add at most k tokens:*
$\mathsf{up}_\phi(M) =$
$$\begin{cases} \{\, t \in T \mid W(p,t) < W(t,p) \wedge W(p,t) \leq M(p) \,\} & \textit{if } M(p) < k \\ \{\, t \in T \mid W(p,t) > W(t,p) \wedge W(t,p) \leq k \,\} & \textit{if } M(p) > k \end{cases}$$

Case $\phi \equiv M(p) \geq k$ *: $\mathsf{up}_\phi(M)$ consists of the transitions which can add tokens and which do not require additional tokens to be present on p:*

$$\mathsf{up}_\phi(M) = \{\, t \in T \mid W(p,t) < W(t,p) \wedge W(p,t) \leq M(p) \,\}$$

Case $\phi \equiv M(p) \neq k$ *: $\mathsf{up}_\phi(M)$ consists of the transitions which can change the marking of p and which do not require additional tokens to be present on p:*

$$\mathsf{up}_\phi(M) = \{\, t \in T \mid W(p,t) \neq W(t,p) \wedge W(p,t) \leq k \,\}$$

Case $\phi \equiv M(p_1) > M(p_2)$ **or** $\phi \equiv M(p_1) \geq M(p_2)$ *:*

$$\mathsf{up}_\phi(M) = \{\, t \in T \mid W(t,p_1) - W(p_1,t) > W(t,p_2) - W(p_2,t) \,\}$$

Case $\phi \equiv M(p_1) = M(p_2)$ *:* $\mathsf{up}_\phi(M) =$

$$\begin{cases} \{\, t \in T \mid W(t,p_1) - W(p_1,t) > W(t,p_2) - W(p_2,t) \,\} \text{ if } M(p_1) < M(p_2) \\ \{\, t \in T \mid W(t,p_1) - W(p_1,t) < W(t,p_2) - W(p_2,t) \,\} \text{ if } M(p_1) > M(p_2) \end{cases}$$

Case $\phi \equiv M(p_1) \neq M(p_2)$ *:*

$$\mathsf{up}_\phi(M) = \{\, t \in T \mid W(t,p_1) - W(p_1,t) \neq W(t,p_2) - W(p_2,t) \,\}$$

Case $\phi \equiv \phi_1 \wedge \phi_2$ *:* $\mathsf{up}_\phi(M)$ *is the up set of one* ϕ_i *which does not hold in* M*:*

$$\big(\neg\phi_1(M) \wedge \mathsf{up}_\phi(M) = \mathsf{up}_{\phi_1}(M)\big) \ \vee \ \big(\neg\phi_2(M) \wedge \mathsf{up}_\phi(M) = \mathsf{up}_{\phi_2}(M)\big)$$

Case $\phi \equiv \phi_1 \vee \phi_2$ *:* $\mathsf{up}_\phi(M) = \mathsf{up}_{\phi_1}(M) \cup \mathsf{up}_{\phi_2}(M)$ □

Next we give the definition of down sets.

Definition 8. *Let* M *be a marking and* ϕ *a state property. The down set* down_ϕ *is defined as follows:*
Case $\phi \equiv M(p) \leq k$ *:* $\mathsf{down}_\phi = \{\, t \in T \mid W(p,t) < W(t,p) \wedge W(p,t) \leq k \,\}$
Case $\phi \equiv M(p) = k$ *:* $\mathsf{down}_\phi = \{\, t \in T \mid W(p,t) \neq W(t,p) \wedge W(p,t) \leq k \,\}$
Case $\phi \equiv M(p) \neq k$ *:* $\mathsf{down}_\phi = \{\, t \in T \mid W(p,t) \neq W(t,p) \wedge W(t,p) \leq k \,\}$
Case $\phi \equiv M(p) \geq k$ *:* $\mathsf{down}_\phi = \{\, t \in T \mid W(p,t) > W(t,p) \wedge W(t,p) < k \,\}$
Case $\phi \equiv M(p_1) > M(p_2)$ **or** $\phi \equiv M(p_1) \geq M(p_2)$ *:*
$\mathsf{down}_\phi = \{\, t \in T \mid W(p_1,t) - W(t,p_1) > W(p_2,t) - W(t,p_2) \,\}$
Case $\phi \equiv M(p_1) = M(p_2)$ **or** $\phi \equiv M(p_1) \neq M(p_2)$ *:*
$\mathsf{down}_\phi = \{\, t \in T \mid W(t,p_1) - W(p_1,t) \neq W(t,p_2) - W(p_2,t) \,\}$
Case $\phi \equiv \phi_1 \wedge \phi_2$ **or** $\phi \equiv \phi_1 \vee \phi_2$ *:* $\mathsf{down}_\phi = \mathsf{down}_{\phi_1} \cup \mathsf{down}_{\phi_2}$ □

Finally, we give the definition satisfiability sets.

Definition 9. *Let* M *be a marking and* ϕ *a state property constructed from the atomic state propositions* $\{\, \varphi_i \mid i \in I \,\}$*. The satisfiability set* $\mathsf{sat}_\phi(M)$ *in* M *is defined as follows. If* $\phi(M)$ *holds then* $\mathsf{sat}_\phi(M) = \emptyset$*. Otherwise it is defined as follows.*

Case $\phi \equiv \varphi_i$ *:* $\mathsf{sat}_\phi(M) = \{\, i \,\}$
Case $\phi \equiv \phi_1 \vee \phi_2$ *:* $\mathsf{sat}_\phi(M) = \mathsf{sat}_{\phi_1}(M) \cup \mathsf{sat}_{\phi_2}(M)$
Case $\phi \equiv \phi_1 \wedge \phi_2$ *:* $\mathsf{sat}_\phi(M)$ *is the satisfiability set of one* ϕ_i *which does not hold in* M*:*
$(\neg\phi_1(M) \wedge \mathsf{sat}_\phi(M) = \mathsf{sat}_{\phi_1}(M)) \vee (\neg\phi_2(M) \wedge \mathsf{sat}_\phi(M) = \mathsf{sat}_{\phi_2}(M))$ □

The following proposition states that the up, down, and satisfiability sets defined above have the required properties. The proof of the proposition is based on structural induction on the state properties and is not contained in this paper.

Proposition 3. *Let* M *be a marking and assume that* $\mathsf{up}_\phi(M) \subseteq T$*,* $\mathsf{down}_\phi \subseteq T$*, and* $\mathsf{sat}_\phi(M) \subseteq T$ *are constructed according to Def. 7, Def. 8, and Def. 9, respectively. Then the following hold:*

1. $\mathsf{up}_\phi(M)$ *has the up set property in* M *with respect to* ϕ*.*
2. down_ϕ *has the down set property with respect to* ϕ*.*
3. $\mathsf{sat}_\phi(M)$ *has the satisfiability set property in* M *with respect to* ϕ*.* □

8.2 Implementation of RSPP and HSPP Stubborn

We now consider the implementation of D1, D2′, and SPP1-3. The implementation of D1, D2′ and SPP1 is rather straightforward. Techniques for ensuring D1 and D2′ are well-established (see, e.g., [10] for a survey), and SPP1 can be handled with similar techniques. Below we suggest three implementations of SPP2 and SPP3. The more complex implementations has the potential of leading to better reductions of the state space.

Attractor Set. A simple way to implement SPP2 is to ensure that in each marking M encountered during the SS state space generation we have $\mathsf{up}_{\varphi_i}(M) \subseteq T_{\mathsf{s}}(M)$ for every $i \in \mathsf{sat}_{\phi}(M)$. In the case of SPP3 we also ensure that $\mathsf{down}_{\phi} \subseteq T_{\mathsf{s}}(M)$. This guarantees that the n in the formulation of SPP2 and SPP3 can be chosen to be zero. SPP1 is automatically guaranteed since $\bigcup_{\{\ i \in \mathsf{sat}_{\phi}(M)\ \}} \mathsf{up}_{\varphi_i(M)} \subseteq T_{\mathsf{s}}(M)$ has the up set property in M. This implementation of SPP2 coincides with the attractor set method suggested in [6].

Terminal SCC Detection. A more powerful implementation of SPP2 can be obtained by exploiting strongly connected components (SCCs) and the fact that for a directed graph it is always possible to reach the nodes belonging to some terminal SCC. This fact implies that if all enabled transitions in the stubborn sets used are key transitions, then a sufficient condition for SPP2 and SPP3 to hold is that for every i there exists an occurrence sequence satisfying SPP2 and SPP3 in each of the terminal SCCs of the SS state space. Stubborn sets in which all enabled transitions are key transitions are also referred to as *strong stubborn sets*. Strong stubborn sets are already guaranteed in case of HSPP due to D2′. In case of RSPP, we can obtain strong stubborn sets by simply ensuring also D2′ in addition to D1, SPP1, and SPP2.

Checking that the terminal SCCs satisfy the requirement formulated above can be done on-the-fly when combining a depth-first generation of the SS state space with generation of SCCs by means of TARJAN's algorithm [2]. If a terminal SCC C is about to be completed and the construction of the SS state space is about to backtrack from the marking M_0 then we check that either $\phi(M)$ holds in some $M \in C$ or for each atomic state proposition φ_i we have that $\mathsf{up}_{\varphi_i}(M_0) \subseteq \bigcup_{\{\ M \in C\ \}} T_{\mathsf{s}}(M)$. If we find an atomic state proposition i such that 1) $\mathsf{up}_{\varphi_i}(M_0) \not\subseteq \bigcup_{\{\ M \in C\ \}} T_{\mathsf{s}}(M)$ and 2) the stubborn set in M_0, $T_{\mathsf{s}}^{\mathsf{up}_i}(M_0)$ containing $\mathsf{up}_{\varphi_i}(M_0) - \bigcup_{\{\ M \in C\ \}} T_{\mathsf{s}}(M)$ contains enabled transitions which are not in $T_{\mathsf{s}}(M_0)$, then we extend the stubborn set used in M_0 with $T_{\mathsf{s}}^{\mathsf{up}_i}(M_0)$. The extension of the stubborn set is simple to implement since the union of two stubborn sets are again a stubborn set. SPP3 requires also the check that for every $t \in \mathsf{down}_{\phi}$, there is a $M \in C$ such that $t \in T_{\mathsf{s}}(M)$.

The use of terminal SCCs was first suggested in [9] for a condition which from an implementation point of view resembles SPP2 and SPP3. The condition was later called "S" in [10]. We refer to [9] for additional details about its implementation.

Cycle Detection. An approximation to ensuring that an occurrence sequence exists satisfying SPP2 and SPP3 in each of the terminal SCCs is to ensure the stronger requirement that such an occurrence sequence exists in each of the SCCs. This can implemented without the use of TARJAN's algorithm, and we can rely on depth-first generation and strong stubborn sets only. The algorithm operates as follows.

Whenever we reach a marking M_1 during the SS state space generation which is on the depth-first search stack, we search backwards in the stack through markings $M_n, M_{n-1}, \ldots, M_1$ and check whether for all i we have that $\mathsf{up}_{\varphi_i}(M_1) \subseteq \bigcup_{j=1}^{n} T_{\mathsf{s}}(M_j)$. For all atomic state proposition i such that $\mathsf{up}_{\varphi_i}(M_1) \not\subseteq \bigcup_{j=1}^{n} T_{\mathsf{s}}(M_j)$ we compute a new stubborn set in M_1, $T_{\mathsf{s}}^{\mathsf{up}_i}(M_1)$ containing $\mathsf{up}_{\varphi_i}(M_1) - \bigcup_{j=1}^{n} T_{\mathsf{s}}(M_j)$, and extend the stubborn set used in M_1 with $T_{\mathsf{s}}^{\mathsf{up}_i}(M)$. Again, SPP3 requires taking also down_{ϕ} into account in the check.

9 Applications

In this section we develop stubborn set methods for *boundedness properties* based on the RSPP stubborn set method. The considered boundedness properties are inspired from how boundedness properties of High-level Petri Nets are interpreted at the level of the equivalent PT-net. The purpose of this section is twofold. Firstly, to develop methods for a general set of boundedness properties as such, and secondly to illustrate how the results of this paper can be applied as a tool for developing stubborn set methods for state properties composed of atomic state propositions beyond those considered in this paper.

Best Upper Bounds. An integer k is an *upper bound* for a set of places $P' \subseteq P$ iff $\forall M \in [M_I\rangle : \sum_{p \in P'} M(p) \leq k$. We are interested in finding the minimal such k, denoted the *best upper bound* of P'. One approach is to check the state properties $\phi^k(M) \equiv \sum_{p \in P'} M(p) \geq k$ starting with $k = 0$ and incrementing k until a k_0 is found for which a marking with $M(p) \geq k_0$ is not reachable. $k_0 - 1$ is then the best upper bound. A problem which has to be solved before this approach works is that we have not allowed $\sum_{p \in P'} M(p) \geq k$ as an atomic state proposition. However, all that is needed to make our stubborn set algorithm work in this case is to define proper up and down sets for this "new" atomic state proposition. The up set for ϕ^k can be defined as the set of transitions which adds tokens to P' and which does not require additional tokens to be present on P'. The down set can be defined as the transitions which can remove tokens from P' and which produces less than k token on P'. Formally:

$$\mathsf{up}_{\phi^k}(M) = \{\ t \in T \mid \sum_{p \in P'} W(p,t) < \sum_{p \in P'} W(t,p) \wedge \sum_{p \in P'} W(p,t) \leq \sum_{p \in P'} M(p)\ \}$$

$$\mathsf{down}_{\phi^k} = \{\ t \in T \mid \sum_{p \in P'} W(p,t) > \sum_{p \in P'} W(t,p) \wedge \sum_{p \in P'} W(t,p) < k\ \}$$

An alternative is to observe that up_{ϕ^k} is independent of k and down_{ϕ^k} can be approximated from above by removing "$\sum_{p \in P'} W(t,p) < k$" from its definition. This means that the stubborn sets used are then independent of k. It therefore suffices to generate just a single SS state space.

Best Lower Bounds. An integer k is a *lower bound* for a set of places $P' \subseteq P$ iff $\forall M \in [M_I\rangle : \sum_{p \in P'} M(p) \geq k$. We are interested in finding the maximal such k, referred to as the *best lower bound* of P'. Similarly to the upper bound case we consider state properties of the form : $\phi^k(M) \equiv \sum_{p \in P'} M(p) \leq k$ starting with $k = 0$ and continuing until the first k_0 is found for which a marking with $M(p) \leq k_0$ is reachable. k_0 is then the best lower bound. The up and down sets for the atomic state proposition $\sum_{p \in P'} M(p) \leq k$ can be defined as shown below. If one is interested in generating only a single SS state space when finding the best lower bound of a set of places, then the dependency of k can be eliminated like for the best upper bound case by approximating the up and down sets to become independent of k.

$$\begin{aligned}
\mathsf{up}_{\phi^k}(M) &= \{\ t \mid \sum_{p \in P'} W(p,t) > \sum_{p \in P'} W(t,p) \wedge \sum_{p \in P'} W(t,p) \leq k\ \} \\
\mathsf{down}_{\phi^k} &= \{\ t \mid \sum_{p \in P'} W(p,t) < \sum_{p \in P'} W(t,p) \wedge \forall p \in P' : W(p,t) \leq k\ \}
\end{aligned}$$

10 Experimental Results

We have implemented the RSPP stubborn set method on top of the state space tool of Design/CPN [1]. The prototype implements the *Attractor Set* and *Cycle Detection* algorithms given in Sect. 8.2. The construction of stubborn sets is based on the *strong component algorithm* described in [10] adapted to take the condition SPP1 into account.

Tables 1 and 2 gives numerical data on the reduction obtained with the two implemented algorithms on some examples. For PETERSON's and HYMAN's mutual exclusion algorithms we consider the two state properties corresponding to mutual exclusion (*Mutual Excl.*), and that each of the two processes can reach the critical section (*Reach. of CS*). For the Reader/Writer protocol we consider three state properties; the writers can get write access (*Reach. of Write*), the three readers can get read access (*Reach. of Read*), and the protocol guarantees exclusive write (*Excl. Write*). For the Reader/Write protocol we consider a configuration with 2 writers and 3 readers. For the Master/Slave protocol we consider two properties; a marking is reachable in which the master has received a response from all slaves which in turn have returned to their idle state (*DoneIdle*), and the master never continues before having received a response from all slaves (*DoneWIdle*). For the Master/Slave protocol we consider configurations with 3,5 and 6 slaves.

Table 1 gives information about the performance of the *Cycle Detection* algorithm. The table contains two main parts. In the *Up set Driven* part the construction of the stubborn set is initiated from the transitions in the up set and it favours stubborn sets containing transitions in the up set. In the *Up set/Enabling Driven* part the construction of the stubborn set is initiated from the transitions in the up set but it does not favour stubborn sets containing transitions in the up set. The *DFG* columns represent a depth-first generation of the state space with *early termination*, i.e., as soon as a marking has been found where the state property holds the generation stops. The *CG* columns represent a complete generation, i.e., the generation continues even though a marking has been found where the state property holds. This gives information about how large a state space the corresponding algorithm considers in the worst-case. For those properties where no marking is reachable where the property holds, depth-first and complete generation coincide, and only the numbers for the complete generation is given. The entries in the *Min Length* columns are of the form x/y, where x gives the number of nodes in a shortest path leading to a marking where the property holds for the depth-first generation (if such one exists), and y gives the corresponding number for the complete generation. This gives information about how good the algorithm is at providing short witness paths.

Table 2 gives information about the performance of the *Attractor Set* algorithm and the full state space. The table contains two main parts. The *Full State Space* part lists the size of the full state space. In the *Attractor Set Method* part, the *DFG* and *CG* columns represent depth-first generation with early termination, and complete generation, respectively. The *BFG* column represents a breadth-first generation with early termination. The entries in the *Min Length* are of the form x/y, where x gives the number of nodes in a shortest path leading to a marking where the property holds for the depth-first generation (if such one exists), and y gives the corresponding number for the *BFG* generation. It was proved in [6] that the latter equals the number of nodes in one of the shortest paths of the full state space. All state spaces reported on in this section were generated in less than 2 minutes on a 166 Mhz PII PC.

If we first compare the numbers for the complete generation (*CG*) in Tables 1 and 2 then in all cases the *Up set/Enabling Driven* implementation gives much better reduction than the *Attractor Set Method*. The *Up set Driven* implementation gives approximately the same reduction as the *Attractor Set Method*. As a consequence of this the *Up set/Enabling Driven* implementation outperforms the *Attractor Set Method* in the cases where the state property does not hold. If we consider the set of state properties which holds then for the first three examples the *Attractor Set Method* seems slightly better than the cycle detection algorithm in terms of yielding small state spaces and generating short witness paths. However, when we turn to the larger *Master/Slave* example, then the *Up set/Enabling Driven* implementation again outperforms the *Attractor Set Method* in terms of reduction and it is still able to generate a short witness path. The intuitive reason for the *Up set/Enabling Driven* implementation to be better in these cases is that if the state property is located “far” from the initial

Table 1. Experimental results – Cycle detection algorithm.

Model/ Property	Up set Driven					Up set/Enabling Driven				
	DFG		CG		Min	DFG		CG		Min
	Nodes	Arcs	Nodes	Arcs	Length	Nodes	Arcs	Nodes	Arcs	Length
Peterson										
Reach. of CS	9	8	36	59	9/6	6	5	34	51	6/6
Mutual Excl.	-	-	48	84	-	-	-	47	79	-
Hyman										
Reach. of CS	5	4	60	95	5/5	8	7	38	46	8/8
Mutual Excl.	19	19	64	106	17/12	18	20	45	57	12/12
Reader/Writer										
Reach. of Write	3	2	77	221	3/3	7	6	14	17	7/7
Reach. of Read	3	2	85	197	3/3	14	17	14	17	7/7
Excl. Write	-	-	46	111	-	-	-	24	37	-
Master/Slave										
DoneIdle-3	60	67	229	548	60/15	30	30	130	152	30/15
DoneIdle-5	230	277	7,837	32,412	230/23	691	790	1,654	2,172	483/23
DoneIdle-6	516	622	46,781	233,276	513/27	1,744	2,084	5,600	7,658	1053/27
DoneWIdle-3	-	-	231	562	-	-	-	185	272	-
DoneWIdle-5	-	-	7,839	32,494	-	-	-	3,745	6,592	-
DoneWIdle-6	-	-	46,783	233,470	-	-	-	16,769	31,168	-

marking (as is the case for the *Master/Slave* example), then the *Attractor Set Method* and to some extent also the *Up Set Driven* implementation have a high risk of investigating wrong branches of the state space first. For the cases where the state property holds the *Up set/Enabling Driven* implementation therefore seem to represent a good solution to the trade-off between generating short witness paths and considering large state spaces.

11 Conclusions

We have presented two new stubborn set methods for reasoning about state properties. The method for determining whether a reachable marking exists in which a given state property holds was based on ideas first presented in [6]. The main difference between our new method and [6] is in how progress towards the state property is ensured. We have replaced the *always progress* condition of [6] with the weaker *eventual progress* condition, which have the potential of leading to better reduction results, and which contains the always progress condition as a special case. We have demonstrated the potential on some practical case studies by means of an implementation of the new method. The case studies showed that the new stubborn set method is significantly better when the state property does not holds in any reachable marking. When a reachable marking exists in which the state property does hold, then it represents good solution to the trade-off between short witness paths and large state spaces. From an implementation point of view the more powerful implementations which we have suggested for the eventual progress condition seems to require *strong stubborn sets*, whereas for the always progress implementation it suffices to use *weak stubborn sets*.

We have extended the first stubborn set method to obtain a second stubborn set method representing a novel technique for determining, e.g., whether a marking is a home marking, and for checking liveness of only a single transition. Like

Table 2. Experimental results – Full state space and attractor set algorithm.

Model/	Full		Attractor Set Method						
Property	State Space		DFG		BFG		CG		Min.
	Nodes	Arcs	Nodes	Arcs	Nodes	Arcs	Nodes	Arcs	Length
Peterson	58	116							
Reach. of CS			9	8	10	11	39	67	9/5
Mutual Excl.			-	-	-	-	50	90	-
Hyman	80	160							
Reach. of CS			7	6	14	17	60	95	7/5
Mutual Excl.			36	42	49	76	64	106	30/12
Reader/Writer	136	532							
Reach. of Write			3	2	3	2	77	221	3/3
Reach. of Read			3	2	3	2	85	197	3/3
Excl. Write			-	-	-	-	118	419	-
Master/Slave									
DoneIdle-3	232	588	61	79	212	520	229	548	45/15
DoneIdle-5	7,840	32,656	1,623	4,144	7,700	31,966	7,837	32,412	575/23
DoneIdle-6	46,784	233,856	9,819	37,155	33,092	156,317	46,701	233,276	1566/27
DoneWIdle-3	232	588	-	-	-	-	231	562	-
DoneWIdle-5	7,840	32,656	-	-	-	-	7,839	32,494	-
DoneWIdle-6	46,784	233,856	-	-	-	-	46,783	233,470	-

existing methods for checking liveness of transitions it relies on strong stubborn sets, but it does not require *ignoring* to be eliminated.

As an application to boundedness properties we have illustrated the use of the results presented in this paper as a tool for developing stubborn set methods for state properties beyond those considered in the paper. In fact, it can be observed that we only directly referred to PT-nets in the implementation of up and down sets, and hence the suggested methods can be transferred to other modelling formalisms – provided that they allow for the definition of sets of transitions satisfying the properties of up and down sets.

References

1. S. Christensen, J. B. Jørgensen, and L. M. Kristensen. Design/CPN - A Computer Tool for Coloured Petri Nets. In E. Brinksma, editor, *Proceedings of TACAS'97*, volume 1217 of *Lecture Notes in Computer Science*, pages 209–223. Springer-Verlag, 1997.
2. A. Gibbons. *Algorithmic Graph Theory.* Cambridge University Press, 1985.
3. P. Godefroid. Using Partial Orders to Improve Automatic Verification Methods. In *Proceedings of Computer-Aided Verification '90*, volume 531 of *Lecture Notes in Computer Science*, pages 175–186. Springer-Verlag, 1990.
4. P. Godefroid. *Partial-Order Methods for the Verification of Concurrent Systems, An Approach to the State-Explosion Problem*, volume 1032 of *Lecture Notes in Computer Science.* Springer-Verlag, 1996.
5. D. Peled. All from One, One for All: On Model Checking Using Representatives. In *Proceedings of Computer-Aided Verification '93*, volume 697 of *Lecture Notes in Computer Science*, pages 409–423. Springer-Verlag, 1993.
6. K. Schmidt. Stubborn Sets for Standard Properties. In S. Donatelli and J. Kleijn, editors, *Proceedings of ICATPN'99*, volume 1639 of *Lecture Notes in Computer Science*, pages 46–65. Springer-Verlag, 1999.

7. A. Valmari. Error Detection by Reduced Reachability Graph Generation. In *Proceedings of the 9th European Workshop on Application and Theory of Petri Nets*, pages 95–112, 1988.
8. A. Valmari. A Stubborn Attack on State Explosion. In *Proceedings of Computer-Aided Verification '90*, volume 531 of *Lecture Notes in Computer Scienc*, pages 156–165. Springer-Verlag, 1990.
9. A. Valmari. Stubborn Sets for Reduced State Space Generation. In G. Rozenberg, editor, *Advances in Petri Nets '90*, volume 483 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.
10. A. Valmari. The State Explosion Problem. In W. Reisig and G. Rozenberg, editors, *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*, pages 429–528. Springer-Verlag, 1998.