Slicing Petri Nets*

Astrid Rakow

Department für Informatik, Univeristät Oldenburg
astrid.rakow@informatik.uni-oldenburg.de

Abstract. In this paper we introduce the notion of net-slice to describe a subnet of a marked Petri net Σ that approximates Σ's behaviour in respect to a set of places P. We show that a slice built for the set of atomic propositions of ϕ enables falsification of ϕ with ϕ being an LTL formula or verification of ϕ with ϕ being an LTL$_{-\mathrm{X}}$ formula, which is an LTL formula built without using the next-time operator. We first discuss the slicing approach on a basic Petri net slicing algorithm. This algorithm is refined to slice more aggressively. The refined algorithm generates slices that can be smaller than the original net Σ even if Σ is strongly connected.

1 Introduction

Slicing is a technique to syntactically reduce a model in such a way that at best the reduced model contains only those parts that may influence the property the model is analysed for. Slicing has originally been developed for software analysis [1] to minimise the program size by "slicing away" bits of the program that are not relevant for the current analysis. It has successfully been applied to support software developers in tasks like program understanding, debugging, differencing, integration, maintenance and testing [2]. Since Mark Weiser in his original publication [1] introduced the first program slicing algorithm, the concept of program slicing has been applied to formalisms other than programming languages such as attribute grammars [3], hierarchical state machines [4] and Z- and CSP-OZ-Specifications [5–7]. In [8] Chang and Wang present an algorithm on Petri nets, that slices out all sets of paths, called concurrency sets, such that all paths within the same set should be executed concurrently.

In this paper we define Petri net slices which enable verification of an LTL$_{-\mathrm{X}}$ formula or falsification of an LTL formula.

In Sect. 2 we introduce a basic slicing algorithm to discuss general aspects of our slicing approach. In Sect. 3 we refine the basic algorithm to construct the *r-slice (=refined-slice)* of a marked net Σ for a set of places P. An example illustrates the scope of our slicing technique in Sect. 4. In Sect. 5 we give a short overview of related work before drawing the conclusions in Sect. 6.

* This work is supported by the German Research Foundation (DFG), grant GRK 1076/1

2 Petri Net Slicing

In this paper we consider finite Place/Transition nets with arc weights. A marked net Σ is defined as $(N, M_0) = (S, T, W, M_0)$, i.e. Σ is a unmarked net N plus initial marking $M_0 \in \mathbb{N}^{|S|}$. N consists of a finite set of places S, a finite set of transitions T and a function $W : (S \times T) \cup (T \times S) \to \mathbb{N}$. $M[t\rangle M'$ denotes that firing transition t on marking M generates the marking M'. If we want to stress the net we also use the notation $M[t\rangle_\Sigma M'$. For $y \in S \cup T$ we denote the set $\{x \in S \cup T \mid W(x, y) > 0\}$ as ${}^\bullet y$ and analoguesly as $y^\bullet$ the set $\{x \in S \cup T \mid W(y, x) > 0\}$. A slice approximates the behaviour of its original net in respect to a set of places P, which is called slicing criterion.

2.1 Basic Slicing Algorithm

The following algorithm constructs the *b-slice (=basic slice)* of a marked Petri net $\Sigma = (S, T, W, M_0)$ for a set of places $P \subseteq S$. The key idea is to capture a possible token flow relevant for places in P. The token flow of a place s is determined by its pre- and post-transitions. Whether a transition can fire depends on the token count of its input places but not of its output places. So *b-slice*$(\Sigma, P) = (S', T', W', M_0')$ is defined as the subnet of Σ that includes all input places of all transitions connected to any place s in S', starting with $P \subseteq S'$.

Definition 1. *b-slice, slicing criterion*
Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and $P \subseteq S$ a non-empty set of places, which is called slicing criterion. The following algorithm constructs the net-slice b-slice(Σ, P).

```
generateBSlice(Σ, P){
  T' := ∅
  S' := P
  S_done := ∅
  forall s ∈ S' \ S_done {
    forall t ∈ ( •s ∪ s•) {
      T' := T' ∪ {t}
      S' := S' ∪ •t
    }
    S_done := S_done ∪ {s}
  }
  W' := W|_{T'∪S'}
  M_0' := M_0|_{S'}
  return (S', T', W', M_0')
}
```

Figure 1 illustrates the effect of the above algorithm. It shows the marked net Σ_1 and its *b-slice* for the slicing criterion $\{s4\}$.

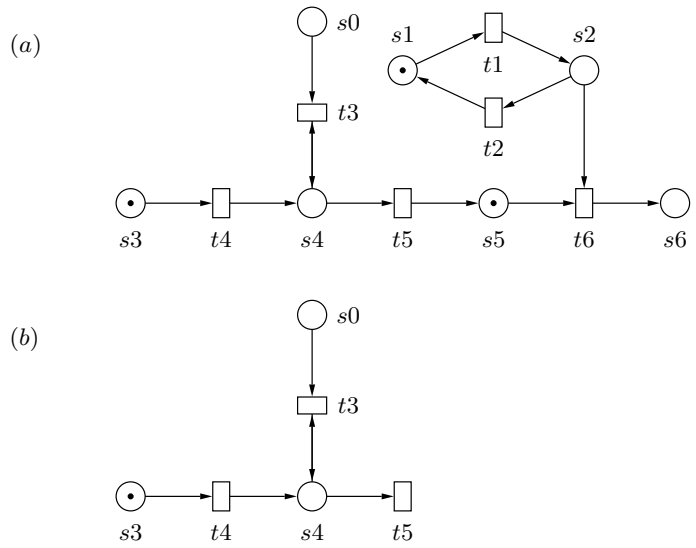

Fig. 1. (a) $\Sigma_1 = (N_1, M_1)$ (b) $\Sigma_1' = b\text{-}slice(\Sigma_1, \{s4\})$

2.2 Verification and Falsification

By analysing the slice we want to derive whether or not an LTL formula holds for Σ. If *"Σ models ϕ"* implies *"Σ's slice models ϕ"*, we can falsify that Σ models ϕ by analysing Σ's slice. Verification means that if the property holds for the slice, it also holds for the original system. So for verification by a net-slice Σ' we have to show that given all admissible firing sequences of Σ' satisfy a formula ϕ, it follows that all admissible firing sequences of Σ satisfy ϕ. Analoguesly, for falsification we have to show that given all admissible firing sequences of Σ satisfy ϕ, it follows that all admissible firing sequences of Σ' satisfy ϕ.

As we will show in the sequel, we have to make two restrictions to allow verification by a slice: one on the formulas and one on the set of admissible firing sequences in terms of fairness assumptions: Σ has more behaviour, which is intrinsic to slicing, as we intentionally do not to capture all behaviour. Fairness assumptions help to restrict the non-captured behaviour, so that verification of formulas without next-time operator becomes possible.

Slice-fairness Since a slice Σ' is defined to approximate a certain behaviour of Σ, we have to define what the behaviour of a net is. If we consider Σ under interleavings semantics, the set of admissible firing sequences consists of all maximal firing sequences of Σ. As maximal firing sequences we consider infinite and finite firing sequences:

Definition 2. *maximal firing sequence*
A firing sequence σ of a marked Petri net $\Sigma = (S, T, W, M_0)$ is maximal iff either σ is of infinite length or $\neg\exists t \in T : M_0[\sigma\, t\rangle$.

In the following, $|\sigma| \in (\mathbb{N} \cup \{\infty\})$ denotes the number of transitions of a firing sequence σ.

Definition 3. $\Sigma \models \phi$
Let Σ be a marked Petri net and ϕ an LTL formula.
$\Sigma \models \phi$ iff every maximal firing sequence of Σ models ϕ.

Consider the Petri net Σ_1 and its *b-slice*$(\Sigma_1, \{s4\})$, Σ_1', in Fig. 1. The formula $\phi = \Diamond s4$ holds for the net-slice Σ_1', but does not hold for the original net Σ_1 due to the maximal firing sequence $t1\ t2\ t1\ t2 \ldots$. For verification we need the assertion that transitions of the slice may not be deferred indefinitely because transitions in $T \setminus T'$ preempt them, as $t1$ and $t2$ do in the example above. This leads to the notion of slice-fairness:

Definition 4. *permanently enabled*
Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let $\sigma = t_1 t_2 \ldots$ be an infinite firing sequence of Σ with $M_i[t_{i+1}\rangle M_{i+1}, \forall i,\ 0 \leq i$.
σ permanently enables $t \in T$ iff $\exists i,\ 0 \leq i : \forall j,\ i \leq j : M_j[t\rangle$.

Definition 5. *slice-fairness with respect to Σ'*
Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let $\Sigma' = (S', T', W', M_0')$ be its net-slice for a slicing criterion $P \subseteq S$. Let $\sigma = t_1 t_2 t_3 \ldots$ be a firing sequence of Σ and M_i the marking with $M_i[t_{i+1}\rangle M_{i+1}, \forall i,\ 0 \leq i < |\sigma|$.
σ is slice-fair w.r.t. Σ' iff

- *either σ is finite and $M_{|\sigma|}$ does not enable a transition $t \in T'$*
- *or σ is infinite, and, if it permanently enables some $t \in T'$, it then fires infinitely often some transition of T' (which may or may not be the same as t).*

Slice-fairness is a very weak fairness notion. Whereas weak fairness determines that every transition t of a system has to be fired infinitely often, if permanently enabled, slice-fairness concerns only the transitions of the slice, not of the entire net, and, if a transition t of the slice is permanently enabled, some transitions of the slice are required to fire infinitely often but not necessarily t. Slice-fairness with respect to a *b-slice* is weaker than weak fairness:

Proposition 6. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let $P \subseteq S$ be a slicing criterion. Let $\Sigma' = (S', T', W', M_0')$ be b-slice(Σ, P). Let σ be a weakly fair firing sequence of Σ.*
σ is slice-fair with respect to Σ'.

Proof. Let σ be a weakly fair firing sequence of Σ. Let us assume, σ is not slice-fair. In case σ is finite, this means that $M_{|\sigma|}[t\rangle$ for a transition $t \in T'$. In case σ is infinite, there is a permanently enabled transition $t \in T'$ but all transitions of T' are fired finitely often including t. So both cases contradict the assumption that σ is weakly fair.

Consider Σ_1's firing sequence $t4\,t5$, which is slice-fair. $t4\,t5$ is not weakly fair, since $t1$ is permanently enabled but never fired. So slice-fairness with respect to a *b-slice* is strictly weaker than weak fairness.

Definition 7. $\Sigma \models \phi$ *slice-fairly*
Let Σ be a marked Petri net and ϕ an LTL formula.

$\Sigma \models \phi$ slice-fairly iff every slice-fair (not necessarily maximal) firing sequence of Σ models ϕ.

As we will show in the sequel, verification is not possible under interleavings semantics, but if we assume slice-fairness.

LTL and LTL$_{-\mathrm{X}}$ Using the next-time operator it is possible to specify a condition to be true at a certain point in the net evolution as a position within a sequence of markings. Since slicing aims at building a reduced model, which should not reflect every state change of the original system, formulas specifying an exact point, counting the number of intermediate states, are not adequate. For an example consider the Petri net Σ_1 and its *b-slice* for $\{s4\}$ in Fig. 1. Firing transition $t1$ or $t6$ has no influence on the token count of $s4$ and thus the slice does not capture these state changes. By only examining the slice it is not possible to say at which position a transition of the slice may be fired within a firing sequence of the original net. A firing sequence of Σ_1, that fires $t4$, may fire $t1$ (or $t6$) before or after firing $t4$ or not at all. Hence for verification we consider LTL$_{-\mathrm{X}}$ formulas, which are formulas built over the set of atomic propositions by $\neg$, $\wedge$ and U. Lamport observed in [10] that every stutter-invariant property can be expressed as an LTL$_{-\mathrm{X}}$ formula, which was proven by Peled and Wilke in [11]. That means an LTL$_{-\mathrm{X}}$ formula cannot distinguish sequences of markings, which are the same up to replacing every occurrence of a finite sequence $M_iM_i...$ by a single occurrence of M_i.

While examining the net-slice does not allow to say when a transition of the original net may be fired, it is possible to say when a transition of a net-slice fires by studying the original net, as the following results show.

2.3 Results for *b-slice*

We start this section with simple observations on the token count and enabling relation for *b-slice*(Σ, P) and Σ.

As we have already seen in Sect. 2.2 the token count of places of the slice is determined by firings of transitions of the slice only. To formalise this observation we define the function slice:

Definition 8. $slice_{(N,N')}$
Let $N = (S, T, W)$ and $N' = (S', T', W')$ be two Petri nets with $T' \subseteq T$ and $S' \subseteq S$. We define the function

$slice_{(N,N')} \in [(T^* \cup T^\omega) \rightarrow (T'^* \cup T'^\omega)] \cup [\mathbb{N}^{|S|} \rightarrow \mathbb{N}^{|S'|}]$
such that a finite or infinite sequence of transitions σ is mapped onto the transition sequence σ' with σ' being derived from σ by omitting every transition $t \in T \setminus T'$. A marking M of N is mapped onto the marking M' of N' with $M' = M|_{S'}$.

In the following the function *slice* is used to project markings and firing sequences of a net Σ onto markings and firing sequences of its slices. We omit the indices referring to the nets and simply denote the function as *slice*, if this does not cause ambiguities.

The observation that transitions that are not part of the slice do not change the token count of places in the slice can now be formally expressed as follows:

Lemma 9. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and $P \subseteq S$ a slicing criterion. Let $\Sigma' = (S', T', W', M_0')$ be its b-slice(Σ, P). Let M_1 and M_2 be markings of Σ.*
$M_1[t\rangle M_2 \Rightarrow slice(M_1) = slice(M_2)$, $\forall t \in T \setminus T'$.

Proof. A transition t is included in the slice iff $t \in {}^\bullet s \cup s^\bullet$ for a place $s \in S'$. Thus a transition $t \in T \setminus T'$ does neither have input nor output places $s \in S'$ and hence cannot change the token count on any place $s \in S'$.

A transition of the slice is enabled in a marking M of Σ if and only if the token count on places in Σ' is sufficient:

Lemma 10. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and $P \subseteq S$ a slicing criterion. Let $\Sigma' = (S', T', W', M_0')$ be its b-slice(Σ, P). Let M be a marking of Σ and M' be a marking of Σ' with $slice(M) = M'$.*
$M[t\rangle_\Sigma \Leftrightarrow M'[t\rangle_{\Sigma'}$, $\forall t \in T'$.

Proof. Since a transition $t \in T'$ has the same input places in Σ and Σ' by Def. 1, $M' = M|_{S'}$ implies $M[t\rangle$ iff $M'[t\rangle$.

Firing Sequences Now we examine the relation of firing sequences of the *b-slice* and Σ: Does a firing sequence of Σ have a corresponding firing sequence of *b-slice* and vice versa?

The next proposition states that every firing sequence of the slice is also a firing sequence of the original net. The firing sequence generates on the original net and its slice the same token count on all places of the slice.

Proposition 11. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let $P \subseteq S$ be a slicing criterion. Let $\Sigma' = (S', T', W', M_0')$ be b-slice(Σ, P) and let σ' be a firing sequence of Σ' and M' be a marking of Σ'.*
$M_0'[\sigma'\rangle M' \Rightarrow \exists M \in \mathbb{N}^{|S|} : M_0[\sigma'\rangle M$ *and* $slice(M) = M'$.

Proof. The proof is by induction on the length l of σ'.

$l = 0$*:* The empty firing sequence generates the markings M'_0 and M_0 on Σ' and Σ, respectively. By Def. 1, $M'_0 = M_0|_{S'} = slice(M_0)$.

$l \to l+1$*:* Let σ' be a firing sequence of length l with $M'_0[\sigma'\rangle M'_l$. By the induction hypothesis, $M_0[\sigma'\rangle M_l$ with $slice(M_l) = M'_l$ holds. Let t_{l+1} be a transition in T', such that $\sigma' t_{l+1}$ is a firing sequence of Σ'. Let M'_{l+1} be the marking with $M'_l[t_{l+1}\rangle M'_{l+1}$. By Lemma 10, M_l enables t_{l+1}. Let M_{l+1} be the marking of Σ with $M_l[t_{l+1}\rangle M_{l+1}$. According to the state equation the markings M_{l+1} and M'_{l+1} are determined by $M_{l+1}(i) = M_l(i) + c_{i\,l+1}, \forall s_i \in S$, and $M'_{l+1}(i) = M'_l(i) + c'_{i\,l+1}, \forall s_i \in S'$, respectively. Since all transitions t with $t \in {}^\bullet s \cup s^\bullet$ are included in the slice and $W' = W|_{S' \cup T'}$, $\forall t_j \in T' : \forall s_i \in S' : c_{ij} = c'_{ij}$ holds. With $slice(M_l) = M_l|_{S'} = M'_l$, it follows that $slice(M_{l+1}) = M'_{l+1}$.

The projection of any of Σ's firing sequences onto T' is also a firing sequence of Σ'. The generated token count is same in both nets for all places $s \in S'$.

Proposition 12. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let σ be a firing sequence and M be a marking of Σ. Let $P \subseteq S$ be a slicing criterion and $\Sigma' = (S', T', W', M'_0)$ be its b-slice(Σ, P).*

$M_0[\sigma\rangle M \Rightarrow M'_0[slice(\sigma)\rangle slice(M)$.

Proof. The proof is by induction on the length l of σ.

$l = 0$*:* The empty firing sequence generates the markings M'_0 and M_0 on Σ' and Σ, respectively. By Def. 1, $M'_0 = M_0|_{S'} = slice(M_0)$.

$l \to l+1$*:* Let σ be a firing sequence of length l. Let M_l be a marking such that $M_0[\sigma\rangle M_l$. Let t_{l+1} be a transition in T and M_{l+1} be a marking of Σ such that $M_l[t_{l+1}\rangle M_{l+1}$. By the induction hypothesis $M'_0[slice(\sigma)\rangle M'_k$ and $slice(M_l) = M'_k$. Let us assume that t_{l+1} is in T'. The marking M_l enables t_{l+1}. Thus by Lemma 10, M'_k enables t_{l+1}. Let M'_{k+1} be the marking with $M'_k[t_{l+1}\rangle M'_{k+1}$. The markings M_{l+1} and M'_{k+1} are determined as $M_{l+1}(i) = M_l(i) + c_{i\,l+1}, \forall s_i \in S$ and $M'_{k+1}(i) = M'_k(i) + c_{i\,l+1}, \forall s_i \in S'$, respectively. With $M'_k = slice(M_l) = M_l|_{S'}$, it follows that $slice(M_{l+1}) = M'_{k+1}$. In case t_{l+1} is in $T \setminus T'$, $slice(\sigma) = slice(\sigma t_{l+1})$ and thus $M'_0[slice(\sigma t_{l+1})\rangle M'_k$. By Lemma 9, it follows that $slice(M_l) = slice(M_{l+1})$. □

Maximality and Slice-Fairness Whereas the previous section dealt with general firing sequences, in this section we are interested in slice-fair or maximal firing sequences.

Proposition 13. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and let $P \subseteq S$ be a slicing criterion. Let $\Sigma' = (S', T', W', M'_0)$ be b-slice(Σ, P). Let σ' be a maximal firing sequence of Σ'.*

σ' is a slice-fair firing sequence of Σ.

Proof. Let $\sigma' = t_1 t_2 \ldots$. Let M'_i be the marking of Σ', such that $M'_i[t_{i+1}\rangle M'_{i+1}$, $\forall i,\ 0 \leq i < |\sigma'|$. By Proposition 11, σ' is a firing sequence of Σ. Let M_i be the marking of Σ, such that $M_i[t_{i+1}\rangle M_{i+1}, \forall i,\ 0 \leq i < |\sigma'|$. In case σ' is finite, $M'_{|\sigma'|}$

does not enable any transitions $t' \in T'$. By Lemma 10, $M_{|\sigma'|}$ does not enable any transitions $t' \in T'$ either. If σ' is infinite it obviously fires infinitely often a transition $t' \in T'$ and thus is slice-fair. □

The projection of any slice-fair firing sequence of Σ onto the transitions of the slice is a maximal firing sequence of the slice.

Proposition 14. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let $\Sigma' = (S', T', W', M_0')$ be b-slice(Σ, P) with $P \subseteq S$. Let σ be a slice-fair firing sequence of Σ.*

slice(σ) is a maximal firing sequence of Σ'.

Proof. Let σ be equal $t_1 t_2 t_3 ...$ with $M_i[t_{i+1}\rangle M_{i+1}, \forall i, 0 \leq i < |\sigma|$. By Proposition 12, *slice*(σ) is a firing sequence of Σ. Let *slice*(σ) be $\sigma' = t_1' t_2' t_3' ...$ with $M_i'[t_{i+1}'\rangle M_{i+1}', \forall i, 0 \leq i < |\sigma'|$. Let us assume σ' is not maximal. Since we us assume that σ' is not maximal, σ' is finite. Let σ_1 be the smallest prefix of σ such that *slice*(σ_1) equals σ'. By Proposition 11, *slice*$(M_{|\sigma_1|}) = M'_{|\sigma'|}$. By Lemma 10, $M'_{|\sigma'|}[t'\rangle$ implies that $M_{|\sigma_1|}[t'\rangle$ and by Lemma 9, t' stays enabled for all markings M_j with $|\sigma_1| \leq j \leq |\sigma|$. But since no transitions of T' are fired, this is a contradiction to the assumption that σ is slice-fair.

Any maximal firing sequence of Σ' can be extended to a maximal firing sequence of Σ by firing transitions in $T \setminus T'$ only.

Proposition 15. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let $\Sigma' = (S', T', W', M_0')$ be b-slice(Σ, P) for a slicing criterion $P \subseteq S$. Let σ' be a maximal firing sequence of Σ'.*

There is a maximal firing sequence σ of Σ such that σ starts with σ' and slice$(\sigma) = \sigma'$

Proof. Let σ' be a maximal firing sequence of Σ'. By Proposition 11, σ' is a firing sequence of Σ. If σ' is of infinite length, it is also a maximal firing sequence of Σ. Assume that σ' is finite. Let σ'' be a transition sequence such that $\sigma = \sigma' \sigma''$ is a maximal firing sequence of Σ. By Proposition 12, *slice*$(\sigma) = \sigma'$*slice*(σ'') is a firing sequence of Σ'. As σ' is maximal, it follows that *slice*$(\sigma'') = \varepsilon$, i.e. $\sigma'' \in ((T \setminus T')^* \cup (T \setminus T')^\omega)$.

Verification and Falsification by *b-slice* For verification and falsification we build the slice for the set of atomic propositions of ϕ.

Definition 16. *scope*(ϕ)
Let A be the set of atomic propositions. Let ϕ, ϕ_1, ϕ_2 be LTL formulas. The function scope associates with every LTL formula ϕ the set of atomic propositions used in ϕ.

scope$(a) = \{a\}$ *for* $a \in A$
scope$(\otimes \phi) =$ *scope*(ϕ) *with* $\otimes \in \{\neg, X\}$
scope$(\phi_1 \otimes \phi_2) =$ *scope*$(\phi_1) \cup$ *scope*(ϕ_2) *with* $\otimes \in \{\wedge, U\}$

Firing the transitions of a firing sequence σ step by step generates a sequence of markings, which we denote as $\mathcal{M}(\sigma)$.

Definition 17. *marking sequence of σ, $\mathcal{M}(\sigma)$*
Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and $\sigma = t_1 t_2 t_3 ...$ be a firing sequence of Σ. Let M_i be the marking with $M_i[t_{i+1}\rangle M_{i+1}, \forall i, 0 \leq i < |\sigma|$.
$\mathcal{M}(\sigma) = M_0 M_1 M_2 ...$ is the marking sequence of σ.

Proposition 18. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and let ϕ be an LTL$_{-X}$ formula such that scope$(\phi) \subseteq S$. Let Σ' be either b-slice$(\Sigma,$ scope$(\phi))$. Let σ be a firing sequence of Σ. Then,*
$\mathcal{M}(\sigma) \models \phi \Leftrightarrow \mathcal{M}($slice$(\sigma)) \models \phi$.

Proof. We show Proposition 18 by induction on the structure of ϕ. Let $\Sigma' = (S', T', W', M_0')$. Let σ be equal $t_1 t_2 t_3 ...$ and slice(σ) be $\sigma' = t_1' t_2' t_3' ...$. Let $\mathcal{M}(\sigma) = M_0 M_1 M_2 ...$ and $\mathcal{M}(\sigma') = M_0' M_1' M_2' ...$.

$\phi =$ *true:* In this case nothing needs to be shown.

$\phi = \neg\psi$, $\phi = \psi_1 \wedge \psi_2$*:* Since the satisfiability of ϕ depends on the initial marking of scope(ϕ) only and scope$(\phi) \subseteq S' \subseteq S$, both directions hold.

$\phi = \psi_1 U \psi_2$*:* We assume $\mathcal{M}(\sigma) \models \psi_1 U \psi_2$. We can divide up σ such that $\sigma = \sigma_1 \sigma_2$ with $M_{|\sigma_1|} M_{|\sigma_1|+1} ... \models \psi_2$ and $\forall i, 0 \leq i < |\sigma_1| : M_i M_{i+1} ... \models \psi_1$. There are transition sequences σ_1' and σ_2' such that $\sigma' = \sigma_1' \sigma_2'$, slice$(\sigma_1) = \sigma_1'$, slice$(\sigma_2) = \sigma_2'$. By Proposition 11 it follows that $M'_{|\sigma_1'|} =$ slice$(M_{|\sigma_1|})$. Since $M_{|\sigma_1|} M_{|\sigma_1|+1} ... \models \psi_2$, $M'_{|\sigma_1|} M'_{|\sigma_1|+1} ... \models \psi_2$ by the induction hypothesis. Let ϱ' be a prefix of σ_1' such that $|\varrho| < |\sigma_1|$. Let ϱ be the prefix of σ_1 such that slice$(\varrho) = \varrho'$. Since ϱ' truncates at least one transition $t \in T'$, $|\varrho| < |\sigma_1|$. Since $M_{|\varrho|} M_{|\varrho|+1} ... \models \psi_1$, $M'_{|\varrho'|} M'_{|\varrho'|+1} ... \models \psi_1$ by the induction hypothesis.

We now assume $\mathcal{M}(\sigma') \models \psi_1 U \psi_2$. We can divide up σ' such that $\sigma' = \sigma_1' \sigma_2'$ with $M'_{|\sigma_1'|} M'_{|\sigma_1'|+1} ... \models \psi_2$ and $\forall i, 0 \leq i < |\sigma_1'| : M_i' M'_{i+1} ... \models \psi_1$. There are transition sequences σ_1 and σ_2 such that $\sigma = \sigma_1 \sigma_2$, slice$(\sigma_1) = \sigma_1'$, slice$(\sigma_2) = \sigma_2'$ and σ_1 does not end with a transition $t \in T \setminus T'$. By Proposition 11 it follows that $M'_{|\sigma_1'|} =$ slice$(M_{|\sigma_1|})$. Since $M'_{|\sigma_1'|} M'_{|\sigma_1'|+1} ... \models \psi_2$, $M_{|\sigma_1|} M_{|\sigma_1|+1} ... \models \psi_2$ by the induction hypothesis. Let ϱ be a prefix of σ_1 such that $|\varrho| < |\sigma_1|$. Let ϱ' be slice(ϱ). The firing sequence ϱ truncates at least one transition $t \in T'$, consequently $|\varrho'| < |\sigma_1'|$. Since $M'_{|\varrho'|} M'_{|\varrho'|+1} ... \models \psi_1$, $M_{|\varrho|} M_{|\varrho|+1} ... \models \psi_1$ by the induction hypothesis. □

The following theorem states that it is sufficient to examine whether Σ's b-slice$(\Sigma,$ scope$(\phi))$ models a formula ϕ by interleavings semantics, if we can assume slice-fairness for Σ.

Theorem 19. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let ϕ be an LTL formula such that scope$(\phi) \subseteq S$.*
$\Sigma \models \phi$ slice-fair $\Rightarrow$ b-slice$(\Sigma,$ scope$(\phi)) \models \phi$, for an LTL formula ϕ, and
$\Sigma \models \phi$ slice-fair $\Leftarrow$ b-slice$(\Sigma,$ scope$(\phi)) \models \phi$, for an LTL$_{-X}$ formula ϕ.

Proof. Let Σ be $(N, M_0) = (S, T, W, M_0)$. Let *b-slice*$(\Sigma, scope(\phi))$ be $\Sigma' = (S', T', W', M'_0)$.

First we show, "$\Sigma \models \phi$ slice-fair $\Rightarrow \Sigma' \models \phi$". Let us assume that Σ models ϕ slice-fair. Let σ' be a maximal firing sequence of Σ'. By Proposition 13, σ' is a slice-fair firing sequence of Σ. Since Σ models ϕ slice-fair, $\mathcal{M}_\Sigma(\sigma')$ satisfies ϕ. By Proposition 18, $\mathcal{M}_{\Sigma'}(\sigma') \models \phi$.

We show "$\Sigma \models \phi$ slice-fair $\Leftarrow \Sigma' \models \phi$". For $\phi = true, \phi = \neg\psi, \phi = \psi_1 \wedge \phi_2 \wedge \phi = \psi_1 U \psi_2$, let us assume Σ' models ϕ. Let σ be a slice-fair firing sequence of Σ. By Proposition 14, *slice*(σ) is a maximal firing sequence of Σ' and thus satisfies ϕ. By Proposition 18, it thus follows that σ satisfies ϕ.

Assume $\Sigma' \not\models X\psi$. Thus there is a maximal firing sequence $\sigma' = t'_1 t'_2 t'_3 ...$ of Σ' with $\mathcal{M}(\sigma) = M_0 M_1 M_2 ...$ and $M_1 M_2 ... \not\models \psi$. σ' is a firing sequence of Σ by Proposition 12. According to Proposition 13 there is a slice-fair firing sequence σ of (N, M_1) with *slice*$(\sigma) = t'_2 t'_3 ...$. By Proposition 18 follows, that $(N, M_1) \not\models \psi$, thus $\Sigma \not\models X\psi$. □

If we cannot assume slice-fairness for Σ, falsification by *b-slice*$(\Sigma, scope(\phi))$ is still possible.

Theorem 20. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let ϕ be an LTL formula such that scope$(\phi) \subseteq S$. Then it holds:*
$\Sigma \models \phi \Rightarrow$ *b-slice*$(\Sigma, scope(\phi)) \models \phi$.

Proof. Let Σ be $(N, M_0) = (S, T, W, M_0)$ and let *b-slice*$(\Sigma, scope(\phi))$ be $\Sigma' = (N', M'_0) = (S', T', W', M'_0)$. For $\phi = true, \phi = \neg\psi, \phi = \psi_1 \wedge \psi_2 \wedge \phi = \psi_1 U \psi_2$, let us assume, that $\Sigma \models \phi$. Let σ' be a maximal firing sequence of Σ' and $\mathcal{M}(\sigma') = M'_0 M'_1 M'_2 ...$. By Proposition 15, it follows that there is a maximal firing sequence σ with *slice*(σ) equals σ'. Since $\mathcal{M}(\sigma) \models \phi$, $\mathcal{M}(\sigma') \models \phi$ by Proposition 18.

Assume $\Sigma' \not\models X\psi$. Thus there is a maximal firing sequence $\sigma' = t'_1 t'_2 t'_3 ...$ of Σ' with $\mathcal{M}(\sigma') = M'_0 M'_1 M'_2 ..$ and $M'_1 M'_2 ... \not\models \psi$. According to Proposition 15 there is a firing sequence σ of (N, M_1) that starts with $t'_2 t'_3 ... t'_{|\sigma'|}$ and *slice*$(\sigma) = t'_2 t'_3 ... t_{|\sigma'|}$. By Proposition 18, it follows that $\mathcal{M}(\sigma) \not\models X\psi$. □

The *b-slice*(Σ, P) will only be smaller than its original net Σ, if there is a bottom subnet $N'' = (S'', T'', W'')$ $(\forall s \in S'' : (s^\bullet)^\bullet \subseteq S'')$ of Σ that does not contain a place $p \in P$. *b-slice*(Σ, P) will hence be equal Σ for a strongly connected Petri net Σ. The slicing algorithm presented in this section is very basic. It constructs the *b-slice* by examining the graph structure for whether a token on a relevant place may be generated. But the algorithm does not take into account, for instance, whether transitions may ever be fired or whether the transitions may actually generate new tokens. So the transition $t3$ of Σ_1 in Fig. 1, which has no influence on the token count on $s4$, is included in the slice. In the next section a refined slicing algorithm will be introduced, which uses transitions like $t3$ to slice more aggressively and hence may reduce strongly connected nets.

3 Refined Slicing Algorithm

Key idea for the construction of the refined algorithm is to distinguish between reading and non-reading transitions. A reading transition with respect to R cannot change the token count of any place in R:

Definition 21. *reading transition, non-reading transition*
Let $N = (S, T, W)$ be a Petri net. Let R be a subset of S. Let t be a transition of N.

t is a reading transition of R iff $\forall s \in R : W(s,t) = W(t,s) \wedge \exists s \in R : W(s,t) = W(t,s)$. t is a non-reading transition iff $\exists s \in R : W(s,t) \neq W(t,s)$.

In the sequel we are only interested in reading and non-reading transitions of the set of places of a given slice.

Again the slice $\Sigma^r = (S', T', W', M_0')$ of Σ for a slicing criterion P is constructed by taking into the slice transitions connected to a place $s \in S'$ and their input places, starting with $S' = P$. But for the construction of Σ^r reading transitions and their input places are not included in the slice.

Definition 22. *r-slice*
Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and $P \subseteq S$ a slicing criterion. The following algorithm constructs the net-slice r-slice(Σ, P).

```
generateRSlice(Σ, P, mode){
  T' := ∅
  S' := P
  S_done := ∅
  forall s ∈ S' \ S_done {
    forall t ∈ (•s ∪ s•) {
      if(W(s,t) ≠ W(t,s)) {
        S' := S' ∪ •t
        T' := T' ∪ {t}
      }
    }
    S_done := S_done ∪ {s}
  }
  W' := W|_{T'∪S'}
  M_0' := M_0|_{S'}
  return (S', T', W', M_0')
}
```

Figure 2 illustrates the construction of the *r-slice*.

3.1 Firing Sequences of *r-slice*

In this section we concentrate on a single firing sequence and its corresponding transition sequence σ', i.e. the firing sequence derived from σ by omitting all

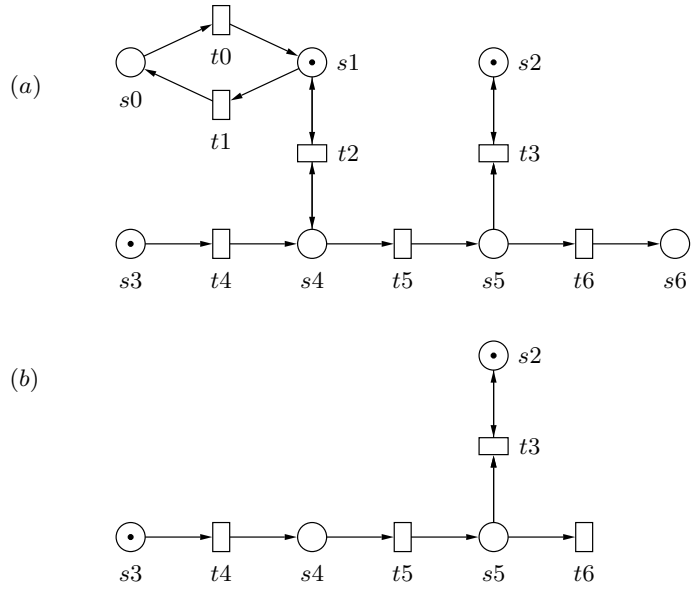

Fig. 2. (a) $\Sigma_2 = (N_2, M_2)$ (b) $r\text{-}slice(\Sigma_2, \{s5\})$

transitions that are not in Σ^r. We examine the three aspects: given a firing sequence $\sigma \in T^* \cup T^\omega$ on Σ, is σ' a firing sequence on Σ^r; given a firing sequence $\sigma' \in T'^* \cup T'^\omega$ on Σ^r, does a corresponding firing sequence on Σ exist, and does the LTL satisfiability change if we consider σ' instead of σ or vice versa?

Lemma 23. *Let $\Sigma = (S, T, W, M_0)$ be a marked net and let Σ^r be $r\text{-}slice(\Sigma, P)$ for $P \subseteq S$.*

The coefficients c_{ij} of the incidence matrix equal zero for all places $s_i \in S'$ and transitions $t_j \in T \setminus T'$.

Proof. Let Σ^r be $r\text{-}slice(\Sigma, P)$. A transition $t \in T$ is element of $T' \subseteq T$, if it may change the token count of a place $s \in S'$. Thus a transition $t \in T \setminus T'$ either is not connected to a place $s \in S'$ or cannot change its token count. □

Lemma 24. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let $P \subseteq S$ be a slicing criterion. Let Σ^r be its $r\text{-}slice(\Sigma, P)$. Let M be a marking of Σ and M' be a marking of Σ^r with $M' = slice(M)$.*

$M[t\rangle \Leftrightarrow M'[t\rangle$, $\forall t \in T'$.

Proof. Let $\Sigma^r = (S', T', W', M'_0)$. Since a transition $t \in T'$ has the same input places in Σ and Σ^r by Def. 22, $M' = M|_{S'}$ implies $M[t\rangle$ iff $M'[t\rangle$.

Every firing sequence σ of Σ projected onto the transitions of T' is also a firing sequence of the net-slice Σ^r. The resulting markings M and M' assign the same number of tokens to places in $S' \subseteq S$.

Proposition 25. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net, let $P \subseteq S$ be the slicing criterion and let $\Sigma^r = (S', T', W', M_0')$ be r-slice(Σ, P). Let σ be a firing sequence of Σ and let M be a marking of Σ.*

$M_0 \,[\sigma\rangle\, M \Rightarrow M_0' \,[slice(\sigma)\rangle\, slice(M)$

Proof. We show Proposition 25 by induction over the length l of σ. Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net, σ a firing sequence of Σ and $P \subseteq S$ a slicing criterion. Let $\Sigma^r = (S', T', W', M_0')$ be r-slice(Σ, P).

$l = 0$*:* In this case *slice*(σ) equals ε. Thus the initial marking of Σ and Σ^r is generated by firing ε. By Def. 22 and Def. 8, $M_0' = M_0|_{S'} = slice(M_0)$.

$l \to l + 1$*:* Let σ be a firing sequence of length l and M_l be a marking of Σ with $M_0[\sigma\rangle M_l$. Let t_{l+1} be a transition in T and M_{l+1} a marking of Σ such that $M_l[t_{l+1}\rangle M_{l+1}$. By the induction hypothesis, $M_0'[slice(\sigma)\rangle M_k'$ and $slice(M_l) = M_k'$. If t_{l+1} is an element of T', it follows by Lemma 24, that M_k' enables t_{l+1}, since M_l enables t_{l+1}. By the state equation, the resulting marking M_{k+1}' is determined by $M_{k+1}'(i) = M_k'(i) + c_{i\,l+1}, \forall s_i \in S'$ and M_{l+1} is determined by $M_{l+1}(i) = M_l(i) + c_{i\,l+1}, \forall s_i \in S$. Since $slice(M_l) = M_l|_{S'} = M_k'$, it thus follows that $slice(M_{l+1}) = M_{k+1}'$. If t_{l+1} is an element of $T \setminus T'$, $slice(\sigma) = slice(\sigma t_{l+1})$ and thus $M_0'[slice(\sigma t_{l+1})\rangle M_k'$. A transition $t \in T \setminus T'$ cannot change the token count on any place $s \in S'$. By Lemma 23 and the state equation, $slice(M_{l+1}) = slice(M_l)$. □

A firing sequence σ' of the net r-slice(Σ, P) is also a firing sequence of Σ. The markings resulting from firing σ' on Σ and Σ^r, respectively, assign the same token count to places $s \in S'$.

Proposition 26. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and let $P \subseteq S$ be a slicing criterion. Let $\Sigma^r = (S', T', W', M_0')$ be r-slice(Σ, P). Let σ' be a firing sequence of Σ^r and M' a marking of Σ^r.*

$M_0'[\sigma'\rangle M' \;\Rightarrow\; \exists M \in \mathbb{N}^{|S|} : M' = slice(M) \wedge M_0[\sigma'\rangle M$

Proof. The proof is by induction over the length l of σ'.

$l = 0$*:* The empty firing sequence generates the marking M_0 on Σ and the marking M_0', which is defined as $M_0|_{S'}$, on Σ^r. By Def. 8 $slice(M_0) = M_0|_{S'}$.

$l \to l + 1$*:* Let $\sigma' = t_1 ... t_{l+1}$ be a firing sequence of Σ^r with length $l + 1$. Let M_l' and M_{l+1}' be markings of Σ^r such that $M_0'[t_1 ... t_l\rangle M_l'[t_{l+1}\rangle M_{l+1}'$. Let M_l be the marking of Σ with $M_0[t_1 ... t_l\rangle M_l$ and $slice(M_l) = M_l'$, which exists according to the induction hypothesis. By Lemma 24, M_l enables t_{l+1}.

The marking M_{l+1} satisfies $\forall s_i \in S : M_{l+1}(i) = M_l(i) + c_{i\,l+1}$ and M_{l+1}' satisfies $\forall s_i \in S' : M_{l+1}'(i) = M_l'(i) + c_{i\,l+1}$. With $M_l|_{S'} = slice(M_l) = M_l'$, it follows that $slice(M_{l+1})$ is equal to M_{l+1}'. □

3.2 LTL and Firing Sequences of *r-slice*

In the sequel, we will show that the sequence of markings generated by a firing sequence σ on Σ and the sequence of markings generated by $slice(\sigma)$ satisfy the same LTL$_{-\mathrm{X}}$ formulas ϕ.

Proposition 27. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and let ϕ be an LTL$_{-X}$ formula such that scope$(\phi) \subseteq S$. Let Σ^r be r-slice$(\Sigma,$ scope$(\phi))$. Let σ be a firing sequence of Σ. Then,*

$\mathcal{M}(\sigma) \models \phi \Leftrightarrow \mathcal{M}(slice(\sigma)) \models \phi$.

Proof. We show Proposition 27 by induction on the structure of ϕ. Let $\Sigma^r = (S', T', W', M_0')$. Let σ be equal $t_1 t_2 t_3 ...$ and $slice(\sigma)$ be $\sigma' = t_1' t_2' t_3' ...$. Let $\mathcal{M}(\sigma) = M_0 M_1 M_2 ...$ and $\mathcal{M}(\sigma') = M_0' M_1' M_2' ...$.

$\phi = true$: In this case nothing needs to be shown.

$\phi = \neg\psi$, $\phi = \psi_1 \wedge \psi_2$: Since the satisfiability of ϕ depends on the initial marking of $scope(\phi)$ only and $scope(\phi) \subseteq S' \subseteq S$, both directions hold.

$\phi = \psi_1 U \psi_2$: We assume $\mathcal{M}(\sigma') \models \psi_1 U \psi_2$. We can divide up σ' such that $\sigma' = \sigma_1' \sigma_2'$ with $M'_{|\sigma_1'|} M'_{|\sigma_1'|+1} ... \models \psi_2$ and $\forall i,\ 0 \leq i < |\sigma_1'| : M_i' M_{i+1}' ... \models \psi_1$. There are transition sequences σ_1 and σ_2 such that $\sigma = \sigma_1 \sigma_2$, $slice(\sigma_1) = \sigma_1'$, $slice(\sigma_2) = \sigma_2'$ and σ_1 does not end with a transition $t \in T \setminus T'$. By Proposition 25 it follows that $M'_{|\sigma_1'|} = slice(M_{|\sigma_1|})$. Since $M'_{|\sigma_1'|} M'_{|\sigma_1'|+1} ... \models \psi_2$, $M_{|\sigma_1|} M_{|\sigma_1|+1} ... \models \psi_2$ by the induction hypothesis. Let ϱ be a prefix of σ_1 such that $|\varrho| < |\sigma_1|$. Let ϱ' be $slice(\varrho)$. The firing sequence ϱ truncates at least one transition $t \in T'$, consequently $|\varrho'| < |\sigma_1'|$. Since $M'_{|\varrho'|} M'_{|\varrho'|+1} ... \models \psi_1$, $M_{|\varrho|} M_{|\varrho|+1} ... \models \psi_1$ by the induction hypothesis.

Analogously, it can be shown that $\mathcal{M}(\sigma) \models \psi_1 U \psi_2$ implies $\mathcal{M}(\sigma') \models \psi_1 U \psi_2$. □

This result cannot be generalised to LTL formulas. Consider the Petri net Σ_2 of Fig. 2. For the firing sequence $\sigma_1 = t1\, t4\, t5\, t3$ of Σ_2 the formula $\phi_1 = XXXs5$ holds, but does not hold for the firing sequence $slice(\sigma_1) = t4\, t5\, t3$. The firing sequence $\sigma_2' = t4\, t5$ satisfies $\phi_2 = XXs5$, but the firing sequence $\sigma_2 = t1\, t0\, t4\, t5$ with $slice(\sigma_2) = \sigma_2'$ does not satisfy ϕ_2.

3.3 Verification and Falsification

In this section we present the main results for *r-slice*. Although r-slice(Σ, P) may be substantially smaller than b-slice(Σ, P), as we will see in Sect. 4, we get the same theorems as in Sect. 2.3.

Proposition 28. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and Σ^r the r-slice(Σ, P) for slicing criterion $P \subseteq S$. Let σ' be a maximal firing sequence of Σ^r.*

σ' is a slice-fair firing sequence of Σ.

Proof. Let $\sigma' = t_1 t_2 ...$. Let M_i' be the marking of Σ', such that $M_i'[t_{i+1}\rangle M_{i+1}'$, $\forall i,\ 0 \leq i < |\sigma'|$. By Proposition 26, σ' is a firing sequence of Σ. Let M_i be the marking of Σ, such that $M_i[t_{i+1}\rangle M_{i+1}, \forall i,\ 0 \leq i < |\sigma'|$. In case σ' is finite, $M'_{|\sigma'|}$ does not enable any transitions $t' \in T'$. By Lemma 24, $M_{|\sigma'|}$ does not enable any transitions $t' \in T'$ either. If σ' is infinite it obviously fires infinitely often a transition $t' \in T'$ and thus is slice-fair.

Proposition 29. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let $\Sigma^r = (S', T', W', M_0')$ be r-slice(Σ, P) for slicing criterion $P \subseteq S$. Let σ be a slice-fair firing sequence of Σ.*

slice(σ) is a maximal firing sequence of Σ^r.

Proof. Let σ be equal $t_1 t_2 t_3 ...$ with $M_i[t_{i+1}\rangle M_{i+1}, \forall i,\ 0 \leq i < |\sigma|$. By Proposition 25, *slice*(σ) is a firing sequence of Σ^r. Let *slice*(σ) be $\sigma' = t_1' t_2' t_3' ...$ with $M_i'[t_{i+1}'\rangle M_{i+1}', \forall i,\ 0 \leq i < |\sigma'|$. Let us assume σ' is not a maximal firing sequence of Σ^r. Thus σ' is finite and there is a transition t' in T' with $M_{|\sigma'|}'[t'\rangle$. Let σ_1 be the smallest prefix of σ such that *slice*(σ_1) equals σ'. By Proposition 25, *slice*$(M_{|\sigma_1|}) = M_{|\sigma'|}'$. By Lemma 24, $M_{|\sigma'|}'[t'\rangle$ implies that $M_{|\sigma_1|}[t'\rangle$. After firing σ_1, σ does not fire any transitions of T'. By Lemma 23 and the state equation it follows, that *slice*$(M_{|\sigma_1|}) =$ *slice*$(M_{|\sigma_1|+1}) = ...$. So t' stays enabled for all markings M_j with $|\sigma_1| \leq j \leq |\sigma|$ but is fired finitely many times only. This is a contradiction to the assumption that σ is slice-fair.

It is sufficient to examine whether r-*slice*$(\Sigma,$ *scope*$(\phi))$ models a formula ϕ by interleavings semantics to derive whether a slice-fair Σ satisfies ϕ:

Theorem 30. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let Σ^r be the r-slice$(\Sigma,$ scope$(\phi))$. Let ϕ be an LTL formula with scope$(\phi) \subseteq S$.*

$\Sigma \models \phi$ slice-fairly $\Rightarrow \Sigma^r \models \phi$, for an LTL formula ϕ.

$\Sigma \models \phi$ slice-fairly $\Leftarrow \Sigma^r \models \phi$, for an LTL$_{-X}$ formula ϕ.

Proof. We first show "$\Sigma \models \phi$ slice-fairly $\Rightarrow \Sigma^r \models \phi$". Let us assume that $\Sigma \models \phi$ slice-fairly holds. Let σ' be a maximal firing sequence of Σ^r. Since σ' is a slice-fair firing sequence of Σ by Proposition 28, $\mathcal{M}(\sigma') \models \phi$.

Let us now assume $\Sigma^r \models \phi$. Let σ be a slice-fair firing sequence of Σ. By Proposition 29, *slice*(σ) is a maximal firing sequence of Σ^r and thus satisfies ϕ. By Proposition 27, it follows that σ satisfies ϕ. □

Proposition 31. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net and Σ^r the r-slice(Σ, P) for slicing criterion $P \subseteq S$. Let σ_m' be a maximal firing sequence of Σ^r.*

There is a maximal firing sequence σ_m of Σ that starts with σ_m' and for which slice$(\sigma_m) = \sigma_m'$ holds.

Proof. Let $\Sigma^r = (S', T', W', M_0')$. By Proposition 26, σ_m' is also a firing sequence of Σ. First, let us assume, that σ_m' is finite. Let σ_m be a maximal firing sequence of Σ with $\sigma_m = \sigma_m' \sigma$ where $\sigma \in T^* \cup T^\omega$. Let σ' be the transition sequence with $\sigma' =$ *slice*(σ_m) or equivalently $\sigma' = \sigma_m'$*slice*(σ). By Proposition 25, σ' is a firing sequence of Σ^r. Since σ_m' is maximal, it follows that *slice*$(\sigma) = \varepsilon$, i.e. $\sigma \in (T \setminus T')^* \cup (T \setminus T')^\omega$. In case σ' is infinite, σ' is infinite firing sequence of Σ by Proposition 26. □

If we cannot assume slice-fairness for Σ, falsification by r-*slice*$(\Sigma,$ *scope*$(\phi))$ is still possible.

Theorem 32. *Let $\Sigma = (S, T, W, M_0)$ be a marked Petri net. Let Σ^r be the r-slice$(\Sigma, scope(\phi))$. Let ϕ be an LTL formula with $scope(\phi) \subseteq S$.*

$\Sigma \models \phi \Rightarrow \Sigma^r \models \phi$.

Proof. The proof is by induction on the structure of ϕ. Let $\Sigma = (N, M_0) = (S, T, W, M_0)$ and $\Sigma^r = (N', M_0') = (S', T', W', M_0')$.

For $\phi = true, \phi = \neg\psi, \phi = \psi_1 \wedge \psi_2$ and $\phi = \psi_1 U \psi_2$, let us assume, $\Sigma \models \phi$ holds. Let $\sigma_m' = t_1' t_2'...$ be a maximal firing sequence of Σ^r and $\mathcal{M}(\sigma_m') = M_0' M_1' M_2'...$. Let $\sigma_m = t_1 t_2...$ be a maximal firing sequence of Σ such that $slice(\sigma_m) = \sigma_m'$ and $\mathcal{M}(\sigma) = M_0 M_1 M_2...$. Such a σ_m exists according to Proposition 31. By Proposition 27, $M_0' M_1'... \models \phi$ follows.

To show $\Sigma \models X\psi \Rightarrow \Sigma^r \models X\psi$, let us assume $\Sigma^r \not\models X\psi$. Hence there is a maximal firing sequence $\sigma' = t_1' t_2'...$ with $\mathcal{M}(\sigma') = M_0' M_1'...$ such that $M_1' M_2'... \not\models \psi$. By Proposition 31, there is a maximal firing sequence σ of Σ that starts with σ' and $slice(\sigma) = \sigma'$. Hence there is a marking M_1 of Σ such that $M_0[t_1'\rangle M_1$ and $slice(M_1) = M_1'$ by Proposition 26. By the induction hypothesis, it follows from $(N', M_1') \not\models \psi$ that $(N, M_1) \not\models \psi$, hence $\Sigma \not\models X\psi$. □

For an example showing that "$\Sigma^r \models \phi \Rightarrow \Sigma \models \phi$" does not hold, consider Fig. 2. Σ_2 does not satisfy $\Diamond s5$, because of firing sequences like $t1\,t0\,t1\,t0...$. Its r-$slice(\Sigma_2, \{s5\})$ does satisfy $\Diamond s5$.

4 Example

As an example application of our refined slicing algorithm we consider a Petri net modelling the daily routine of two employees and their boss. They all have four basic states: They are at home, in their respective offices, on break or at a meeting. The boss starts his working day in his office. Before he goes on a break, he may schedule a meeting. After the break he either goes home directly, if no meeting is scheduled, or attends the meeting and then goes home. The net modelling his daily routine is presented in Fig. 3. Place $B1$ denotes "boss is at home", $B2$ "boss is in his office", $B3$ "boss is on break" and $B4$ "boss is in a meeting". A token on the place M means, that the boss scheduled a meeting, a token on NM, that he did not schedule a meeting.

The employees have to wait at home, until the boss is in his office to let them in. They take a break before they either go to a meeting, given a meeting is scheduled, or go home directly. They always go home after their boss does. Figure 4 shows the full net modelling the daily routine of the boss and his two employees. The places of the two employee-subnets are named in analogy to the net in Fig. 3.

We want to verify that every time the boss does not schedule a meeting, he will be at home eventually. To put it formally, we are interested in whether $\Sigma_3 \models \Box(NM \Rightarrow \Diamond B1)$. The *b-slice* of Σ_3 for any slicing-criterion would be Σ_3, since it is strongly connected. We build the *r-slice* of Σ_3 for the slicing criterion $\{NM, B1\}$. The algorithm of Def. 22 generates the slice shown in Fig. 3. Σ_3 is approximated by a net leaving out all of the employees' nets. Studying the real

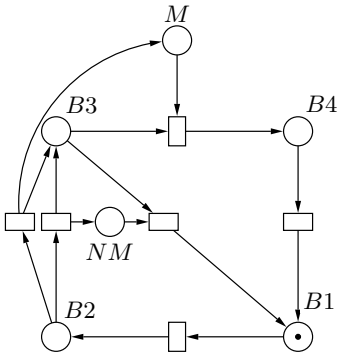

Fig. 3. Net modelling the boss's daily routine

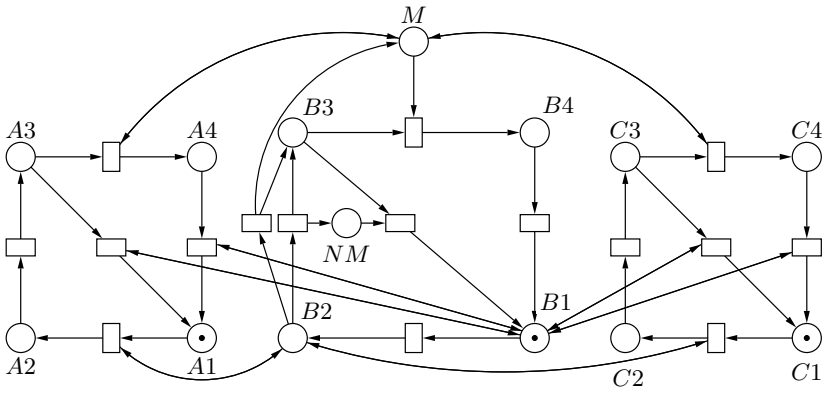

Fig. 4. Σ_3, the full net modelling two employees and their boss

system, i.e. the boss and his employees, we conclude that we can assume slice-fairness for Σ with respect to $r\text{-}slice(\Sigma_3, \{NM, B1\})$. From the initial marking of Σ_3 80 states (=markings) via 168 state transitions are reachable, whereas $r\text{-}slice(\Sigma_3, \{NM, B1\})$ has 5 reachable markings via 6 state transitions. This holds despite the fact that Σ_3 is strongly connected.

As an example of falsification by a slice, we want to falsify that, every time the boss does not schedule a meeting, employee A will be at home eventually, i.e. $\phi_3 = \Box(NM \Rightarrow \Diamond A1)$. Property ϕ_3 is violated when employee A is at work and the boss does not schedule a meeting and then everytime the boss goes home, he goes back to work before employee A went home. Figure 5 shows the $r\text{-}slice(\Sigma_3, \{NM, A1\})$. The reachability graph consists of 20 states and 33 state transitions.

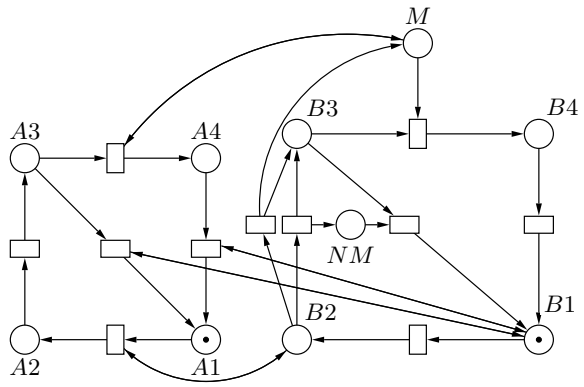

Fig. 5. $r\text{-}slice(\Sigma_3, \{NM, A1\})$

Whether or not the algorithm is able to reduce the net size depends on the structure of the net and the position of the slicing criterion within the net. The slice will be smaller, if the original net Σ is composable into two subnets Σ' and Σ'' such that the slicing criterion is in a subnet Σ', which is not written by Σ'', i.e. Σ'' has not any input places of a non-reading transition t, whose output places are in Σ'.

5 Related Work

There are several groups working on slicing and on slicing as a method to tackle the state-explosion problem for model-checking in particular. For program slicing Dwyer and Hatcliff show in [12] that a program P and its program slice P' either both satisfy ϕ or both do not satisfy ϕ given the set of atomic propositions of an $\text{LTL}_{-\text{X}}$ formula ϕ is the slicing criterion.

Cone-of-influence reduction [13], a similar technique used in hardware verification, constructs a reduced model P' for the set of atomic propositions of a formula ϕ and guarantees that the reduced model P' satisfies ϕ if and only if the original model P satisfies ϕ.

In [6] Brückner develops a technique for slicing CSP-OZ specifications. Brückner and Wehrheim show in [7] the correctness of their approach to slicing Object-Z specifications.

6 Conclusions and Future Work

This paper introduces Petri net slicing to reduce the size of a net in order to alleviate the state explosion problem for model checking Petri nets. As a first approach we define a basic slicing algorithm based on the observations that the token count of a place p is determined by the firings of transitions connected with p and these are determined by the token count of their input places. This leads to the definition of $b\text{-}slice(\Sigma, P)$, which allows falsification and verification given Σ is slice-fair. The refined algorithm is based on the observation that the token count of p is not influenced by reading transitions. The *r-slice* also allows falsification and verification if Σ is slice-fair, but is additionally able to reduce strongly connected nets. Our approach is quite general by imposing no restrictions on the LTL or $\text{LTL}_{-\text{X}}$ formula and little restrictions on the net in terms of fairness assumptions. Nevertheless, our results show that slicing is a technique that can help to alleviate the state explosion problem for model checking Petri nets.

The implementation of the algorithms presented here is planned. We are investigating two ideas for the development of algorithms that allow for more aggressive slicing. One approach could be the generalisation of reading transitions under stronger fairness assumptions to reading subnets that may temporarily remove tokens. An other approach could be to find restrictions on the net that allow to identify subnets that play the role of data and control flow, so that the concept of relevant variables, as defined in [1], is applicable.

7 Acknowledgement

I would like to thank Eike Best, Hans Fleischhack and Harro Wimmel for plenty of feedback. I am grateful to Javier Esparza and Maciej Koutny for reading and commenting.

References

1. Weiser, M.: Program slicing. In: Proceedings of the 5th international conference on Software engineering, IEEE Press Piscataway, NJ, USA (1981) 439–449
2. Tip, F.: A survey of program slicing techniques. In: Journal of programming languages **3** (1995) 121–189

3. Sloane, A.M., Holdsworth, J.: Beyond traditional program slicing. In: International Symposium on Software Testing and Analysis, San Diego, CA, ACM Press (1996) 180–186
4. Heimdahl, M.P.E., Whalen, M.W.: Reduction and slicing of hierarchical state machines. In Jazayeri, M., Schauer, H., eds.: Proceedings of the Sixth European Software Engineering Conference (ESEC/FSE 97), Springer–Verlag (1997) 450–467
5. Chang, J., Richardson, D.J.: Static and dynamic specification slicing. In: Proceedings of the Fourth Irvine Software Symposium. (1994)
6. Brückner, I.: Slicing CSP-OZ specifications. In: Nordic Workshop on Programming Theory. (2004)
7. Brückner, I., Wehrheim, H.: Slicing Object-Z specifications for verification. In: Treharne, H., King, S., Henson, M., and Schneider, S., eds.: Formal Specification and Development in Z and B, LNCS 3455 (2005) 414–433.
8. Chang, C.K., Wang, H.: A slicing algorithm of concurrency modeling based on Petri nets. In Hwang, K., Jacobs, S.M., Swartzlander, E.E., eds.: Proc. of the 1986 Int. Conf. on Parallel Processing, Washington, IEEE Computer Society Press (1987) 789–792.
9. Rakow, A.: Slicing petri nets. Technical Report, parsys.informatik.uni-oldenburg.de/pub/index.html#SlPN_tr.pdf, 2007.
10. Lamport, L.: What good is temporal logic? In Manson, R.E.A. ed., Information Processing 1983: Proc. of the IFIO 9th World Computer Congress, Paris, France (Nort-Holland, Amsterdam, 1983), 657-668.
11. Peled, D., Wilke, T.: Stutter-invariant temporal properties are expressible without the next-time operator. In: Information Processing Letters. Elsevier Science B.V. (1997) 243–246
12. Hatcliff, J., Dwyer, M.B., Zheng, H.: Slicing software for model construction. In: Higher-Order and Symbolic Computation. (2000) 315–353
13. Berezin, S., Campos, S., Clarke, E.M.: Compositional reasoning in model checking. In: Proc. COMPOS. COMPOS (1997)