On Combining Partial Order Reduction with Fairness Assumptions*

Luboš Brim, Ivana Černá, Pavel Moravec, and Jiří Šimša

Department of Computer Science, Faculty of Informatics
Masaryk University, Czech Republic
{brim,cerna,xmoravec,xsimsa}@fi.muni.cz

Abstract. We present a new approach to combine partial order reduction with fairness in the context of LTL model checking. For this purpose, we define several behaviour classes representing typical fairness assumptions and examine how various reduction techniques affect these classes. In particular, we consider both reductions preserving all behaviours and reductions preserving only some behaviours.

1 Introduction

Fairness and partial order reduction are often indispensable for the verification of liveness properties of concurrent systems. The former is mostly needed in order to eliminate some ”unrealistic” executions, while the latter is one of the most successful techniques for alleviating the state space explosion problem.

In model-based verification the adequacy of the model is important. As the model is a simplification of the system under consideration, some behaviours exhibited by the model may not be real ones. To tackle this problem the model can be refined or, alternatively, some assumptions that disqualify fictional behaviours in the model are used. For example, when modelling a multi-process concurrent system with a shared exclusive resource we may want to assume, for the sake of simplicity, that no process can starve though some behaviours of the model may not satisfy this assumption. This concept is commonly known as fairness assumptions or simply fairness.

The most common form of fairness [4,7] is *unconditional fairness* that considers only behaviours with some action occurring *infinitely* many times. Further it is reasonable to take into account *enabledness* of actions. This gives rise to even finer concepts. First, *strong fairness* that considers only behaviours where every action *enabled infinitely* many times is *taken infinitely* many times. Second, *weak fairness* that disqualifies behaviours with some action continuously enabled from a certain moment and subsequently never taken.

It might appear that the reason for using fairness is that it allows for simpler models. However, this simplicity is often outbalanced by the complexity of algorithms operating on a model with fairness. In fact, the main reason for using

* This work has been partially supported by the Grant Agency of Czech Republic grant No. GACR 201/06/1338. and by the Academy of Sciences of the Czech Republic grant. No. 1ET408050503.

L. Brim et al. (Eds.): FMICS and PDMC 2006, LNCS 4346, pp. 84–99, 2007.

fairness is that it simplifies the modelling process—work that has to be done by a human and not by a computer.

By contrast, partial order reduction allows for reduction of a state space of a modelled system [5,8,9,11]. A particular instance of the concept consists of a set of conditions that the reduction must satisfy. The idea behind partial order reduction is that it might not be necessary to consider *all* enabled transitions at a given state, but only a certain *ample* subset.

The justification for such reduction varies and depends on the nature of properties being examined. For example, we may select only one of mutually *independent* actions if we are interested in deadlocks. In general, a behavioural equivalence over a set of behaviours is defined and the reduction is required to contain a representative of each equivalence class.

In our previous work [2] we have proposed a combination of *distribution* and *partial order reduction*; concepts that push back the frontiers of practical verification, both fighting the state space explosion in its own way. In this paper we examine a combination of *partial order reduction* (and distribution) with *fairness*—a concept used for simplifying the process of modelling. We define four behaviour classes reflecting typical fairness assumptions and two partial order reduction techniques. For each behaviour class and reduction technique we prove or disprove that the reduction preserves behaviours of the class.

Closest to our work on combination of partial order reduction and fairness is that of Peled [8]. Peled uses equivalence robustness of properties to ensure that all fair runs in the original state space have at least one stuttering equivalent *fair* run in the reduced state space. Since fairness assumptions are not generally equivalently robust, one has to add more dependencies among transitions in order to achieve equivalence robustness. In author's later work [9] the discussion continues and on-the-fly state space generation is taken into account.

To contrast with our results, Peled considers only strong and weak fairness and aims at preservation of all behaviours. Whereas we examine more fairness assumptions and also study possibilities for better reduction.

This paper is organized as follows. Section 2 lays theoretical foundations for modelling of a system, reviews partial order reduction, and formulates two of its instances. The main theoretical contribution of the paper follows in Section 3 where we identify four behaviour classes and resolve whether they are preserved under the proposed reductions. After these results are established, we discuss their practical use in Section 4.

2 Partial Order Reduction

As a model we use labelled transition system. Let S be a set of *states* and *transition* be a partial function $\alpha : S \rightarrow S$, that is, a transition "can be taken" between different pairs of states. A *labelled transition system (LTS)* is then defined as a tuple $M = (S, s_0, \Delta, L)$, where $s_0 \in S$ is an *initial state*, Δ is a set of transitions over S, and $L : S \rightarrow 2^{AP}$ is a labelling function that assigns to each state a subset of some set AP of *atomic propositions*.

Furthermore, a set of transitions *enabled* at a state s, denoted $enabled(s)$, is a set of all $\alpha \in \Delta$ such that $\alpha(s)$ is defined. A *reduction* of M is then defined as a pair $(M, ample)$ where *ample* is a function assigning to each state s a subset of $enabled(s)$.

A *path* in M from a state s_1 is a finite or infinite sequence $\pi = s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{n-1}} s_n \xrightarrow{\alpha_n} \ldots$ of states interleaved with transitions—ending with a state in the finite case—such that $s_i \in S$, $\alpha_i \in \Delta$ and $\alpha_i(s_i, s_{i+1})$ for each index i.

Let $\eta = r_1 \xrightarrow{\alpha_1} r_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{m-1}} r_m$ be a finite path and $\sigma = s_1 \xrightarrow{\beta_1} s_2 \xrightarrow{\beta_2} \ldots \xrightarrow{\beta_{n-1}} s_n \xrightarrow{\beta_n} \ldots$ a finite or infinite path. Then $first(\sigma) = s_1$ denotes the *first state of* σ and $last(\eta) = r_m$ denotes *the last state of* η. If $last(\eta) = first(\sigma)$ then the path $\eta \circ \sigma = r_1 \xrightarrow{\alpha_1} r_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{m-1}} s_1 \xrightarrow{\beta_1} s_2 \xrightarrow{\beta_2} \ldots \xrightarrow{\beta_{n-1}} s_n \xrightarrow{\beta_n} \ldots$ is the *concatenation* of the paths η and σ.

Finally, let $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ be a sequence of transitions from Δ. We say that γ is a *cycle* if for every state s, $\gamma_1 \in enabled(s), \gamma_2 \in enabled(\gamma_1(s)), \ldots, \gamma_n \in enabled(\gamma_{n-1}(\ldots(\gamma_1(s))\ldots))$ implies $\gamma_n(\ldots(\gamma_1(s))\ldots) = s$.

In order to simplify the presentation of the particular instance of the partial order reduction technique we are going to suggest, we define two relations which will help to formulate conditions constituting the instance.

Definition 1. *An* independence *relation* $\neg D \subseteq \Delta \times \Delta$ *is a symmetric, anti-reflexive relation, satisfying the following three conditions for each state* $s \in S$ *and for each* $(\alpha, \beta) \in \neg D$*:*

1. *Enabledness – If* $\alpha, \beta \in enabled(s)$*, then* $\alpha \in enabled(\beta(s))$*.*
2. *Commutativity – If* $\alpha, \beta \in enabled(s)$*, then* $\alpha(\beta(s)) = \beta(\alpha(s))$*.*
3. *Neutrality – If* $\alpha \in enabled(s)$ *and* $\beta \in enabled(\alpha(s))$*, then* $\beta \in enabled(s)$*.*

The dependency *relation* D *is the complement of* $\neg D$*.*

Note that our definition of $\neg D$ differs from the standard definition of $\neg D$ given in the literature. In particular, we add the *neutrality* condition and therefore our definition of $\neg D$ is more strict. We argue that, in practice, the relation $\neg D$ is approximated using rules conforming to our definition, which allows for more concise proofs.

Definition 2. *An* invisibility *relation* $\neg V \subseteq \Delta$ *is a unary relation with respect to a set of atomic propositions* AP*, where for each* $\alpha \in \neg V$ *and for each pair of states* $s, s' \in S$ *such that* $\alpha(s, s')$*,* $L(s) \cap AP = L(s') \cap AP$ *holds. The* visibility *relation* V *is the complement of* $\neg V$*.*

The *reduction* of a given state space is defined by providing a set of conditions the *ample* function has to fulfil to guarantee that behaviours with certain properties are preserved. In the case of properties expressed as formulas from a fragment of *Linear Temporal Logic without any next modalities* (LTL_{-X}) the following conditions are used [3].

C0. $ample(s) = \emptyset$ iff $enabled(s) = \emptyset$.

C1. Along every path in the model starting from s, the following condition holds: a transition that is dependent on a transition in $ample(s)$ cannot occur without a transition in $ample(s)$ occurring first.

C2. If $enabled(s) \neq ample(s)$, then every $\alpha \in ample(s)$ is invisible.

C3. A cycle is not allowed if it contains a state in which some transition α is enabled, but is never included in $ample(s)$ for any state s on the cycle.

Theorem 1 ([3]). *Let φ be a LTL$_{-X}$ formula, $M = (S, s_0, \Delta, L)$ be a LTS and $M' = (M, ample)$ a reduction of M satisfying conditions **C0** through **C3**. Then $M \models \varphi \Leftrightarrow M' \models \varphi$.*

We now formulate a new condition that is supposed to replace the condition **C3** and consequently allow for better reduction. Downside of the new condition is that reduction based on it may not preserve all behaviours.

C4. From every state s there is reachable a fully expanded state i.e. state such that $ample(s) = enabled(s)$.

In practice conditions **C3** and **C4** are ensured using provisos based on particular state space exploration algorithm. For example, when using depth first search the following provisos are used.

P3. If $ample(s) \neq enabled(s)$, then none of $ample(s)$ transitions points back to stack.

P4. If $ample(s) \neq enabled(s)$, then at least one of $ample(s)$ transitions does not point back to stack.

It can be shown by a simple argument that provisos **P3** and **P4** indeed imply conditions **C3** and **C4** respectively. Clearly, proviso **P4** is weaker than proviso **P3** and thus generally yields better reductions. Further advantage of condition **C4** over condition **C3** comes to light when combining partial order reduction with distribution; to ensure condition **C4** cycle detection is not necessary.

Based on the above conditions we can consider two reduction techniques. The first one uses the original set of conditions and the second one makes use of the new condition **C4**. In particular, when a reduction satisfies conditions **C0** through **C3** we say it is *safe* and when it satisfies conditions **C0** through **C2** and **C4** we say it is *aggressive*.

3 Behaviour Classes

In this section we identify several behaviour classes and investigate whether they are preserved by safe and/or aggressive reduction techniques. As we are interested in preservation of properties expressed in LTL$_{-X}$, we use stuttering equivalence as the behavioural equivalence.

Definition 3. *Two infinite paths $\eta = r_1 \xrightarrow{\alpha_1} r_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{n-1}} r_n \xrightarrow{\alpha_n} \ldots$ and $\sigma = s_1 \xrightarrow{\beta_1} s_2 \xrightarrow{\beta_2} \ldots \xrightarrow{\beta_{n-1}} s_n \xrightarrow{\beta_n} \ldots$ are* stuttering equivalent*, denoted $\sigma \sim_{st} \eta$,* *if there are two strictly increasing infinite sequences of integers* $(i_0, i_1, i_2, \ldots)$ *and* $(j_0, j_1, j_2, \ldots)$ *such that* $i_0 = j_0 = 0$ *and for every* $k \geq 0$*:*

$$L(s_{i_k}) = L(s_{i_k+1}) = \ldots L(s_{i_{k+1}-1}) = L(r_{j_k}) = L(r_{j_k+1}) = \ldots L(r_{j_{k+1}-1})$$

Definition 4. *Let* M *is an LTS. An* LTL_{-X} *formula* φ *is* invariant under stuttering *iff for each pair of paths* π *and* π' *such that* $\pi \sim_{st} \pi'$*,* $M, \pi \models \varphi$ *iff* $M, \pi' \models \varphi$*.*

Theorem 2 ([10]). *Any* LTL_{-X} *formula is invariant under stuttering.*

3.1 Paths with Infinitely Many Visible Transitions

Let $trans(\pi)$ denotes the *sequence of transitions* on a path π and $vis(\pi)$ denotes the *sequence of visible transitions* on a path π.

Theorem 3. *Let* M *be an LTS and* $M' = (M, ample)$ *be a safe reduction. Then for each path* σ *in* M *such that* $|vis(\sigma)| = \infty$ *there is a path* η *in* M' *such that* $\sigma \sim_{st} \eta$ *with* $|vis(\eta)| = \infty$*.*

For the proof we refer to the construction of infinite sequence of infinite paths π_0,π_1,π_2,... from the proof of Theorem 1 (see [3], Section 10.6). For the hint on the construction see appendix A.

Theorem 4. *Let* M *be an LTS and* $M' = (M, ample)$ *an aggressive reduction. Then for each path* σ *in* M *such that* $|vis(\sigma)| = \infty$ *there is a path* η *in* M' *such that* $\sigma \sim_{st} \eta$*.*

There are two key steps to prove Theorem 4. The first step is an observation that it is sufficient to consider only paths without *scattered cycles*. The next step is a construction of stuttering equivalent path for a path without scattered cycles.

Definition 5. *A path* σ *contains a* scattered cycle $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ *iff:*

- γ *is a cycle*
- *every transition from* γ *is invisible*
- *there are paths* $\theta_1, \ldots, \theta_{n+1}$ *such that all transitions in* $\theta_1, \theta_2, \ldots \theta_i$ *are independent on the transition* γ_i *and* $\sigma = \theta_1 \circ (\mathit{last}(\theta_1) \xrightarrow{\gamma_1} \mathit{first}(\theta_2)) \circ \theta_2 \circ \ldots \circ \theta_n \circ (\mathit{last}(\theta_n) \xrightarrow{\gamma_n} \mathit{first}(\theta_{n+1})) \circ \theta_{n+1}$*.*

Lemma 1. *For each path* σ *in* M *with* $|vis(\sigma)| = \infty$ *there is an infinite path* σ' *in* M *such that* $\sigma \sim_{st} \sigma'$*,* $\mathit{first}(\sigma) = \mathit{first}(\sigma')$ *and* σ' *does not contain any scattered cycle.*

Proof: Let us suppose that σ contains a scattered cycle $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ and $\sigma = \theta_1 \circ (\mathit{last}(\theta_1) \xrightarrow{\gamma_1} \mathit{first}(\theta_2)) \circ \theta_2 \circ \ldots \circ \theta_n \circ (\mathit{last}(\theta_n) \xrightarrow{\gamma_n} \mathit{first}(\theta_{n+1})) \circ \theta_{n+1}$.

According to the definition of the scattered cycle, the transition γ_2 is enabled in the state $\mathit{first}(\theta_2)$ and is independent on all transitions in θ_2. Therefore there is a path in M containing the scattered cycle γ and such that the transition γ_2 precedes all transitions from θ_2. Using the same argument repeatedly we show that there is a path $\theta_1 \circ (\mathit{last}(\theta_1) \xrightarrow{\gamma_1} \ldots \xrightarrow{\gamma_n} \mathit{last}(\theta_1)) \circ \theta'_2 \circ \ldots \theta'_n \circ \theta_{n+1}$ in M where $\mathit{trans}(\theta_i) = \mathit{trans}(\theta'_i)$ for all $i = 2, \ldots, n$.

As γ is a cycle, $\theta_1 \circ \theta'_2 \circ \ldots \theta'_n \circ \theta_{n+1}$ is a path in M stuttering equivalent to σ. It seems that in this manner we could iteratively remove all scattered cycles appearing in σ. However, by removing a scattered cycle from a path we could create a new one. Therefore to prove existence of stuttering equivalent path without scattered cycles we consider all existing as well as potential scattered cycles on the path σ simultaneously.

Let $\delta = (\delta_1, \delta_2, \ldots)$ be a subsequence of $\mathit{trans}(\sigma)$ such that either δ_i is a transition of a scattered cycle in σ or there is a finite number of scattered cycles that can be removed from σ—through the construction above—with δ_i becoming a transition of a scattered cycle afterwards.

Let $(\alpha_1, \alpha_2, \ldots)$ be a sequence of transitions which remain in $\mathit{trans}(\sigma)$ after removing the subsequence δ. We prove that there is an infinite path σ' in M such that $\mathit{first}(\sigma) = \mathit{first}(\sigma')$ and $\mathit{trans}(\sigma') = (\alpha_1, \alpha_2, \ldots)$ as these together guarantee $\sigma \sim_{st} \sigma'$.

We show that for all i, $\alpha_i \in \mathit{enabled}(\alpha_{i-1}(\ldots(\alpha_1(\mathit{first}(\sigma)))\ldots))$. Let δ_j occurs in σ before α_i. Then δ_j can be removed from the path together—with the cycle it belongs to—and α_i still remains enabled thanks to the arguments mentioned above. Consequently, σ' is a path in M and as $\mathit{vis}(\sigma) = \mathit{vis}(\sigma')$ and $|\mathit{vis}(\sigma)| = \infty$, it is infinite. □

It can be shown that any aggressive reduction contains a path stuttering equivalent to a given path in M without any scattered cycle. The construction of the stuttering equivalent path is suspended until Appendix.

3.2 Process Fair Paths

In this subsection we assume, that LTS M is modelling a multi-process system and $\mathcal{P}$ denotes the set of its processes. Further, let $\pi_{\geq i}$ denotes the *suffix* of a path π that is a subsequence of π starting at i-th state.

Definition 6. *Let σ be an infinite path and M an LTS. Then for $\mathcal{X} \subseteq \mathcal{P}$, $\mathit{trans}(\mathcal{X}, \sigma)$ denotes the* set of all transitions on σ of a process from $\mathcal{X}$.

For every $\mathcal{X} \subseteq \mathcal{P}$ such that all $\alpha \in \mathit{trans}(\mathcal{X}, \sigma)$ are independent on all $\beta \in \mathit{trans}(\mathcal{P} \setminus \mathcal{X}, \sigma)$ that is $(\alpha, \beta) \in \neg D$, we define a path $\mathit{proj}(\mathcal{X}, \sigma)$ as a path resulting from σ after removing all transitions of processes from $\mathcal{P} \setminus \mathcal{X}$.

Definition 7. *An infinite path σ is* process fair *if for every $P \in \mathcal{P}$ the number of P's transition on σ is infinite.*

Theorem 5. *Let M be an LTS and $M' = (M, \mathit{ample})$ a safe reduction. Then for each process fair path σ in M there is a process fair path η in M' such that $\pi \sim_{st} \eta$.*

Again, for the proof we refer to the construction of infinite sequence of infinite paths π_0,π_1,π_2,..., from the proof of Theorem 1 and we omit the rest.

Theorem 6. *Let M be an LTS and $M' = (M, ample)$ an aggressive reduction. Then for each process fair path σ in M there is a path η in M' such that $\pi \sim_{st} \eta$.*

Similarly to the proof of Theorem 4, there are two key steps to prove Theorem 6. The first step is an observation that it is sufficient to consider only *non-reducible paths*. The next step is the construction of stuttering equivalent path path for a non-reducible path.

Definition 8. *Let $\sigma = s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{n-1}} s_n \xrightarrow{\alpha_n} \ldots$ be a path in M.*
If exists $k \in \mathbb{N}$ and a non-empty set of processes $\mathcal{X} \neq \mathcal{P}$ such that

- *all $\alpha \in trans(\mathcal{X}, \sigma_{\geq k})$ are independent on all $\beta \in trans(\mathcal{P} \setminus \mathcal{X}, \sigma_{\geq k})$,*
- *all transitions from $trans(\mathcal{P} \setminus \mathcal{X}, \sigma_{\geq k})$ are invisible,*
- *both $proj(\mathcal{X}, \sigma_{\geq k})$ and $proj(\mathcal{P} \setminus \mathcal{X}, \sigma_{\geq k})$ are infinite,*

*then σ is k-*reducible *and the path $\sigma' = s_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{k-1}} s_k \circ proj(\mathcal{X}, \sigma_{\geq k})$ is a* k-reduction of *σ. If no such k and $\mathcal{X}$ exists then σ is called* non-reducible*.*

Lemma 2. *Let σ be a path in M and σ' be a k-reduction of σ. Then σ' is a path in M and $vis(\sigma) = vis(\sigma')$.*

Proof: By a simple argument from definition of $\neg D$ and k-reducibility. □

Lemma 3. *Let σ be a process fair path in M. Then there is an infinite path σ' in M such that $\sigma \sim_{st} \sigma'$ and σ' is* non-reducible*.*

Proof: We inductively construct a finite sequence of paths $\sigma_0, \sigma_1, \ldots, \sigma_n$ such that $\sigma_0 = \sigma$ and $\sigma_n = \sigma'$ and show that σ_i is a k-reduction of σ_{i-1} for every $i = 1, \ldots, n$.

We start with $\sigma_0 = \sigma$. If σ_i is k-reducible for some k and $\mathcal{X}$ we take the smallest k and subsequently smallest possible $\mathcal{X}$ and we let σ_{i+1} to be the respective k-reduction of σ_i. Otherwise, the construction is finished.

Note that the construction is deterministic – as we choose the smallest possible k and $\mathcal{X}$ – and finite since the sequence is strictly decreasing in the number of processes which take a transition infinitely many times. □

Let σ be a non-reducible path in M resulting from the process fair path transformation outlined above. The construction of a path stutter equivalent to σ in an aggressive reduction $(M, ample)$ can be found in Appendix.

3.3 Weakly Fair Paths

Definition 9. *Let $\sigma = s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{n-1}} s_n \xrightarrow{\alpha_n} \ldots$ be a path. If there do not exist i and β such that for all $j \geq i$, $\beta \in enabled(s_j)$ and $\beta \neq \alpha_j$, then σ is* weakly fair*.*

It can be shown by a simple argument using induction, that every weakly fair path in a model has stuttering equivalent weakly fair path in any safe reduction

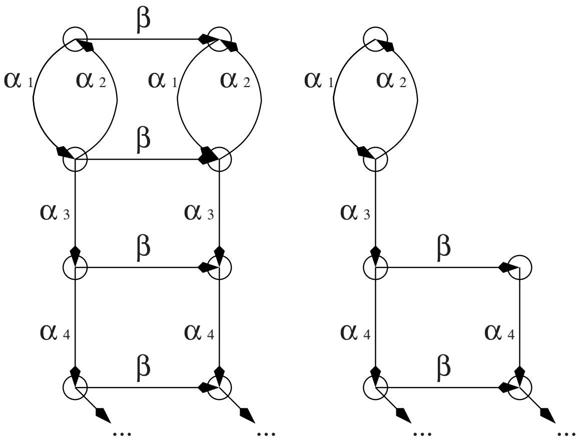

Fig. 1. Model and its reduction

of the model. For the idea of the proof we refer once again to the construction of infinite sequence of infinite paths π_0,π_1,π_2,..., from the proof of Theorem 1.

As Figure 1 demonstrates, weakly fair behaviour does not have to be preserved in aggressive reductions. On the left there is a part of the model state space and on the right there is a part of its reduction state space. Let α transitions be mutually dependent and transitions β and α_4 be visible. Then weakly fair path $\beta \cdot (\alpha_1 \cdot \alpha_2)^\omega$ in the model has no stuttering equivalent path in the reduction.

3.4 Strongly Fair Paths

Definition 10. *Let $\sigma = s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{n-1}} s_n \xrightarrow{\alpha_n} \ldots$ be a path. If for every β enabled in infinitely many states on σ there exists infinitely many j's such that $\beta = \alpha_j$ then σ is* strongly fair*.*

It can be shown by a simple argument using induction, that every strongly fair path in a model has a stuttering equivalent path in a safe reduction. However, this path may not be strongly fair as Figure 2 demonstrates.

On the left is a part of model state space and on the right is a part of its reduction state space. Let α transitions be mutually dependent as well as β transitions and γ be dependent on all α and β transitions. Further let α_1, α_2 and γ be visible transitions. For the strongly fair path $(\alpha_1 \cdot \alpha_2 \cdot \beta_1 \cdot \beta_2)^\omega$ in the model state space, there is no stuttering equivalent strongly fair path in the reduction state space.

Furthermore, Figure 3 demonstrates that a strongly fair behaviour does not have to be preserved in aggressive reductions. On the left is a part of the model state space and on the right is a part of its reduction state space. Let α transitions be mutually dependent and transition β and α_3 be visible. Then a strongly fair path $\beta \cdot (\alpha_1 \cdot \alpha_2)^\omega$ in the model has no stuttering equivalent path in the reduction.

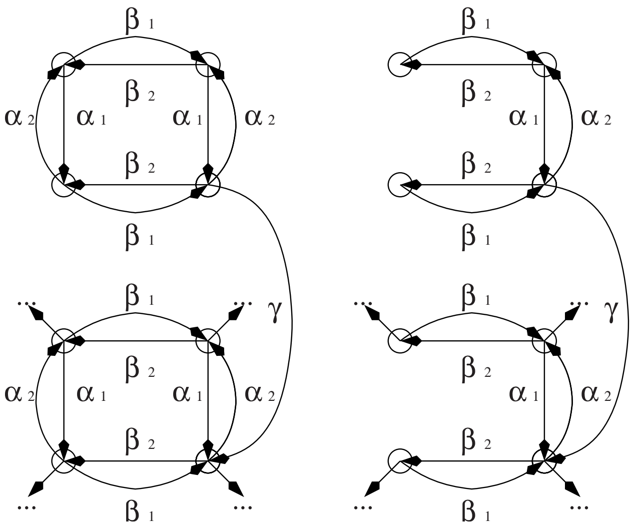

Fig. 2. Model and its reduction

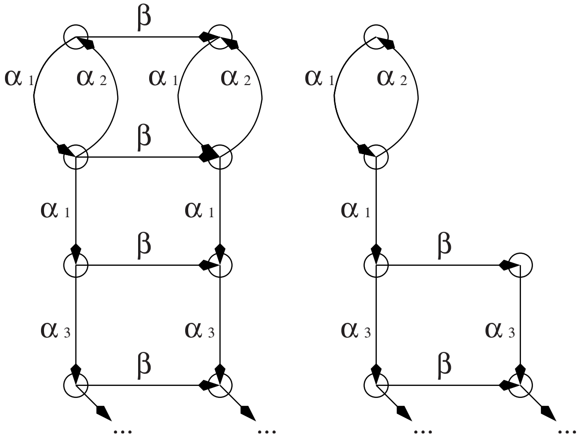

Fig. 3. Model and its reduction

4 Applications

In this section, we identify typical fairness assumptions and relate them, one by one, to results established in the previous section. We try to point out situations where either aggressive or safe reduction may be of use.

Another issue to be discussed is related to usage of fairness model checking algorithms. Although a reduction may preserve all fair behaviours, it may not preserve them "fairly". Therefore, a fairness model checking algorithm may return different results when applied on the model and on the reduction.

Situation 1. Certain subset of actions of modelled system is considered and each of them is taken infinitely many times.

If at least one of the relevant actions is visible, we can apply an aggressive reduction as every behaviour of our interest is preserved in such a reduction. Moreover, the same fairness model checking algorithm can be applied on the reduction.

Otherwise, aggressive reduction does not guarantee that the desired behaviours are preserved in the reduction. On the contrary, safe reductions preserve all behaviours. Furthermore, as the respective construction of a stuttering equivalent path for this set of conditions does not remove any transition from the original path, the same fairness model checking algorithm can be applied.

Situation 2. Certain subset of processes of multi-process system is considered and each of them performs some action infinitely many times.

First, if the subset is equal to the set of all processes, the result for *process fair* paths can be applied. Unfortunately, a *non-reducible* representative of a process fair path might not be fair. Consider the example on Figure 4. On the left there is a part of model state space and on the right there is a part of its reduction state space. Let α transitions be mutually dependent as well as β transitions. Further let β_1, β_2 and α_4 be visible transitions. Finally, let $\{\alpha_1, \beta_1\}$ be the fairness assumption.

The path $(\beta_1 \cdot \beta_2 \cdot \alpha_1 \cdot \alpha_2)^\omega$ in the model state space satisfies the assumption. However, there is no stuttering equivalent path in the reduction state space, which would satisfy the assumption. Thus, one cannot use the same fairness

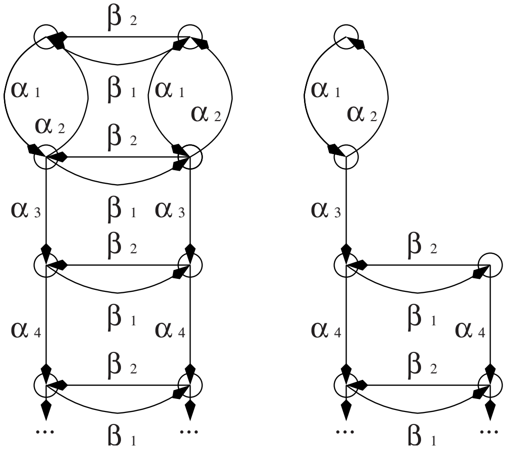

Fig. 4. Model and its reduction

model checking algorithm both for a model M and its aggressive reduction M_R and the equivalence $M \models_F \varphi \Leftrightarrow M_R \models_F \varphi$ does not hold in general.

Nevertheless, this approach can be used for checking validity of φ as $M_R \models \varphi \Rightarrow M \models_F \varphi$. Actually, we find this result to be quite interesting, as checking validity is generally more “space and time demanding” task than checking invalidity—which can be partially dealt with using approximation and stochastic techniques. Again, a safe reduction preserves all behaviours. Moreover, the respective construction of a stutter equivalent path in a safe reduction does not remove any transition from the original path and thus the original fairness model checking algorithm can be used as well.

Alternatively, if we somehow guarantee that every time a process performs infinitely many actions, it performs infinitely many visible actions as well, then all desired behaviours are preserved even by an aggressive reduction and the same fairness model checking algorithm can be applied. However, the more visible actions there are, the smaller the reduction generally is.

Situation 3. Only weakly fair behaviours are considered.

As a weakly fair path might not have a stuttering equivalent path in an aggressive reduction, we discuss this assumption in the context of safe reductions.

These reductions preserve all behaviours and as the respective construction of a stuttering equivalent paths does not remove any transition from the original path, the original fairness model checking algorithm can be applied.

Situation 4. Only strongly fair behaviours are considered.

In general aggressive reductions do not preserve strongly fair behaviours. On the contrary, safe reductions preserve all strongly fair behaviours, but the resulting stuttering equivalent paths do not have to be strongly fair. Therefore the same fairness model checking algorithm cannot be applied.

In order to use the same fairness model checking algorithm the dependency relation can be modified as described in [8]. Alternatively, any model checking algorithm can be used for checking the validity of φ as $M_R \models \varphi \Rightarrow M \models_F \varphi$.

Finally, if the model represents a multi-process system where every process has always enabled at least one action, strong fairness implies process fairness and aggressive reductions can be used for checking validity.

5 Conclusions

The paper explores a combination of two concepts: partial order reduction and fairness, both used in the context of LTL model checking. While the first one is essential in alleviating the state space explosion, the second one simplifies the modelling process.

For the partial order reduction we consider a well-known safe variant together with a new variant represented by condition **C4** which is supposed to replace condition **C3**. It allows for better reduction in general and yet ensures that certain subset of behaviours is preserved. We have defined safe reduction as any reduction satisfying conditions **C0** through **C3** and we have used the new condition to define aggressive reduction. Then we have defined four behavioural classes motivated by typical fairness assumptions. The paper gives a detailed analysis of fairness concepts and demonstrates how they are affected by safe and aggressive reductions.

For several reductions we have encountered the following problem. Even though fair behaviour is preserved by the reduction it does not have an equivalent *fair* behaviour representative in the reduced model. This disables the possibility to use the same fairness model checking algorithm. On the contrary, as all fair behaviours in a model M have a stuttering equivalent behaviour in a reduction M_R and $M_R \models \varphi \implies M \models_F \varphi$, we can actually check formula validity under fairness assumptions. Whether our results can be extended to checking invalidity is left as an open problem.

References

1. D. Bosnacki. Partial order reduction in presence of rendez-vous communications with unless constructs and weak fairness. In *Theoretical and Practical Aspects of SPIN Model Checking (SPIN 1999)*, volume 1680 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 1999.
2. L. Brim, I. Černá, P. Moravec, and J. Šimša. Distributed Partial Order Reduction. *Electronic Notes in Theoretical Computer Science*, 128:63–74, April 2005.
3. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* The MIT Press, Cambridge, Massachusetts, 1999.
4. N. Francez. *Fairness.* Texts and Monographs in Computer Science. Springer, 1986.
5. P. Godefroid and D. Pirottin. Refining dependencies improves partial-order verification methods. In *Proc. of the 5th Conference on Computer-Aided Verification*, volume 697 of *LNCS*, pages 438–449. Springer, 1992.
6. G.J. Holzmann, P. Godefroid, and D. Pirottin. Coverage preserving reduction strategies for reachability analysis. In *Proc. 12th Int. Conf on Protocol Specification, Testing, and Verification, INWG/IFIP*, Orlando, Fl., June 1992.
7. T. Latvala and K. Heljanko. Coping with strong fairness. *Fundamenta Informaticae*, pages 175–193, 2000.
8. D. Peled. All from one, one from all: on model checking using representatives. In *Proceedings of the 5th International Conference on Computer Aided Verification, Greece*, number 697 in Lecture Notes in Computer Science, pages 409–423, Berlin-Heidelberg-New York, 1993. Springer.
9. D. Peled. Combining partial order reductions with on-the-fly model-checking. In *Proceedings of CAV'94*, pages 377–390. Springer Verlag, LNCS 818, 1994.
10. D. Peled and T. Wilke. Stutter-invariant temporal properties are expressible without the nexttime operator. *Information Processing Letters*, 1997.
11. A. Valmari. A stubborn attack on state explosion. In *Proc. of the 2nd Workshop on Computer-Aided Verification*, volume 531 of *LNCS*, pages 156–165. Springer, 1991.

A Appendix

Proof of Theorems 4 and 6 follow the same direction. Thus, we present it just once and we distinguish between different context only when necessary.

Our goal is the following. Given a path σ in M and a reduction $M' = (M, \mathit{ample})$ satisfying conditions **C0** through **C2** and **C4**, show that there is a path in M' stuttering equivalent to σ.

First, we inductively describe an infinite sequence of paths π_0, π_1, π_2, ..., where $\sigma = \pi_0$ and for every i, $\pi_i = \eta_i \circ \theta_i$ is a path in M, η_i is a path in M', and $\mid \eta_i \mid = i$.

Basic step. Let $\eta_0 = \varepsilon$, $\theta_0 = \sigma$.
Inductive step. Let $s_0 = \mathit{last}(\eta_i) = \mathit{first}(\theta_i)$, $\theta_i = s_0 \overset{\alpha_1}{\to} s_1 \overset{\alpha_2}{\to} s_2 \ldots$

There are two possibilities:

A If $\alpha_1 \in \mathit{ample}(s_0)$ then $\eta_{i+1} = \eta_i \circ (s_0 \overset{\alpha_1}{\to} s_1)$, $\theta_{i+1} = s_1 \overset{\alpha_2}{\to} s_2 \ldots$

B The case $\alpha_1 \notin \mathit{ample}(s_0)$ divides into two subcases:

B1 There is k such that $\alpha_k \in \mathit{ample}(s_0)$ and (α_j, α_k) are independent for all $1 \leq j < k$. We choose the *smallest* possible k. Then $\eta_{i+1} = \eta_i \circ (s_0 \overset{\alpha_k}{\to} \alpha_k(s_0))$. As transitions α_j are independent, $\alpha_k(s_0) \overset{\alpha_1}{\to} \alpha_k(s_1) \overset{\alpha_2}{\to} \alpha_k(s_2) \ldots$ is a path in M. Let $\theta_{i+1} = s_0 \overset{\alpha_k}{\to} \alpha_k(s_0) \overset{\alpha_1}{\to} \alpha_k(s_1) \overset{\alpha_2}{\to} \ldots \overset{\alpha_{k-1}}{\to} \alpha_k(s_k) \overset{\alpha_{k+1}}{\to} s_{k+2} \overset{\alpha_{k+2}}{\to} \ldots$

B2 $\alpha_k \notin \mathit{ample}(s_0)$ for any k. Then from the condition **C1** all transitions in $\mathit{ample}(s_0)$ are independent on all transitions in θ_i. Let ξ be the shortest path in M' from s_0 to a fully expanded state (the existence of such a path is guaranteed by **C4**) and let β be the first transition of ξ. Then $\eta_{i+1} = \eta_i \circ (s_0 \overset{\beta}{\to} \beta(s_0))$, $\theta_{i+1} = \beta(s_0) \overset{\alpha_1}{\to} \beta(s_1) \overset{\alpha_2}{\to} \beta(s_2) \ldots$

Cases **B1** and **B2** cover all possibilities conforming to **C1**.

Notice that a simple argument based on the definition of $\neg D$ yields that the rule **B1** cannot be applied after the rule **B2** without the rule **A** being applied in the meantime. This fact is implicitly employed in proof of Lemma 7.

Next, we characterise properties of the path η and then we prove the stuttering equivalence between η and σ.

Properties of η

Lemma 4. *For every i, $\pi_i = \eta_i \circ \theta_i$ is a path in M, η_i is a path in M', and $\mid \eta_i \mid = i$.*

Proof: By induction. Induction basis for $i = 0$ holds trivially. In induction step, we first prove that π_i is a path in M. It obviously holds for the case **A**. In the case **B1**, (α_j, α_k) are independent, for all $j < k$. Hence there is a path $\xi = s_0 \overset{\alpha_k}{\to} \alpha_k(s_0) \overset{\alpha_1}{\to} \alpha_k(s_1) \overset{\alpha_2}{\to} \ldots \overset{\alpha_{k-1}}{\to} \alpha_k(s_k) \overset{\alpha_{k+1}}{\to} s_{k+2} \overset{\alpha_{k+2}}{\to} \ldots$ in M, where α_k is moved before $\alpha_1\alpha_2\alpha_3 \ldots \alpha_{k-1}$. Note that $\alpha_k(s_k) = s_{k+1}$. Therefore, $\alpha_k(s_k) \overset{\alpha_{k+1}}{\to} s_{k+2}$ is the same as $s_{k+1} \overset{\alpha_{k+1}}{\to} s_{k+2}$. In the case **B2** we execute a transition which is independent on all transitions in θ_{i-1}, hence θ_i is obviously a path in M. Certainly η_i is a path in M' and $\mid \eta_i \mid = i$ in all cases, as we append to η_{i-1} exactly one transition from $\mathit{ample}(\mathit{last}(\eta_{i-1}))$. □

Lemma 5. *Let $\eta = \lim_{i\to\infty} \eta_i$. Then η is a path in M'.*

Proof: By induction to i. □

Stuttering equivalence

Lemma 6. *The following holds for all i, j such that $j \geq i \geq 0$.*

1. *$\pi_i \sim_{st} \pi_j$.*
2. *$vis(\pi_i) = vis(\pi_j)$.*
3. *Let ξ_i be a prefix of π_i and ξ_j be a prefix of π_j such that $vis(\xi_i) = vis(\xi_j)$. Then $L(last(\xi_i)) = L(last(\xi_j))$.*

Proof: It is sufficient to consider the case where $j = i + 1$. Consider three ways of constructing π_{i+1} from π_i. In case **A**, $\pi_{i+1} = \pi_i$ and the statement holds trivially.

In case **B1**, π_{i+1} is obtained from π_i by executing a invisible transition α_k in π_{i+1} earlier than it is executed in π_i. In this case, we replace the sequence $s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{k-1}} s_{k-1} \xrightarrow{\alpha_k} s_k$ by $s_0 \xrightarrow{\alpha_k} \alpha_k(s_0) \xrightarrow{\alpha_1} \alpha_k(s_1) \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{k-1}} \alpha_k(s_{k-1})$. Because α_k is invisible, corresponding states have the same label, that is, for each $0 < l \leq k$, $L(s_l) = L(\alpha_k(s_l))$. Also, the order of the visible transitions remains unchanged. Parts 1, 2, and 3 follow immediately.

Finally, consider case **B2**, where the difference between π_i and π_{i+1} is that π_{i+1} includes an additional invisible transition β. Thus, we replace some suffix $s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots$ by $s_0 \xrightarrow{\beta} \beta(s_0)) \xrightarrow{\alpha_1} \beta(s_1) \xrightarrow{\alpha_2} \ldots$. So, $L(s_l) = L(\beta(s_l))$ for $l \geq 0$. Again, the order of visible transitions remains unchanged and parts 1, 2, and 3 follow immediately. □

In the following lemma we have to differentiate between individual cases.

Lemma 7. *During the construction of η, the case* **A** *is chosen infinitely often.*

Proof: for paths with infinitely many visible transitions

First, we prove that for every i, θ_i does not contain any scattered cycle.

By induction to i. For $\theta_0 = \sigma$ the statement holds trivially. If θ_i is constructed applying **A** or **B2** it does not contain any cycle as θ_{i-1} does not contain any. In case of **B1**, a presence of a scattered cycle in θ_i would imply a presence of a scattered cycle in θ_{i-1}

Now, let us assume that there is an index j such that during the construction of $\pi_j, \pi_{j+1}, \ldots$ only the rule **B** is applied. Then either **B1** or **B2** is applied infinitely many times.

In case rule **B1** is applied infinitely many times there is an infinite sequence of transitions which are added to the prefix η_{j-1}. These transitions are invisible and independent on all other transitions in θ_j. From finiteness of the set of states we have that some of the considered transitions form a cycle, which is moreover a scattered cycle in θ_j. Hence a contradiction.

This gives us an existence of an index $k \geq j$ such that for the construction of $\pi_k, \pi_{k+1}, \ldots$ only the rule **B2** is applied. But this is a contradiction to the fact that in **B2** we always choose a transition from the shortest path to a fully expanded state. □

Proof: for process fair paths

First, we prove that for every i, θ_i is non-reducible.

By induction to i. For $\theta_0 = \sigma$ the statement holds trivially. If θ_i is constructed applying **A**, **B1** or **B2** it is non-reducible as θ_{i-1} is.

Now, let us assume that there is an index j such that during the construction of $\pi_j, \pi_{j+1}, \ldots$ only the rule **B** is applied. Then either **B1** or **B2** is applied infinitely many times.

In case rule **B1** is applied infinitely many times there is an infinite sequence of transitions which are added to the prefix η_{j-1}. These transitions are invisible and independent on all other transitions in θ_j. Let P be the process taking α transition on θ_j. If $\mid proj(\{P\}, \theta_j) \mid = \infty$, then θ_j is 0-reducible and we get a contradiction. Therefore $\mid proj(\{P\}, \theta_j) \mid$ must be finite and θ_j is not reducible for any k. Moreover, σ is a result of process fair path transformation described in Lemma 3. The original process fair path contained infinitely many P's transitions. Thus, during the construction of σ, P's transitions were removed because of some k-reduction. But in that particular moment of the construction a j-reduction removing transitions selected by **B1** rule would be possible too. Finally, as j is strictly smaller than k, this is a contradiction as well.

This gives us an existence of an index $k \geq j$ such that for the construction of $\pi_k, \pi_{k+1}, \ldots$ only the rule **B2** is applied. But this is a contradiction to the fact that in **B2** we always choose a transition from the shortest path to a fully expanded state. □

Lemma 8. *Let α be the first transition of θ_i. Then there exists $j > i$: α is the last transition of η_j and $\forall k : i \leq k < j$: α is the first transition of θ_k.*

Proof: The rules **B1** and **B2** leave the first transition α of θ_i unchanged, the rule **A** shifts the transition α to η_i. Thus it is sufficient to prove that during the construction of η, the rule **A** is applied infinitely often. This follows from Lemma 7. □

Lemma 9. *Let δ be the first visible transition on θ_i, $prefix_\delta(\theta_i)$ be the maximal prefix of $trans(\theta_i)$ that does not contain δ. Then **either** δ is the first transition of θ_i and the last transition of η_{i+1} **or** δ is the first visible transition of θ_{i+1}, the last transition of η_{i+1} is invisible and $prefix_\delta(\theta_{i+1})$ is a subsequence of $prefix_\delta(\theta_i)$.*

Proof:

- If θ_{i+1} is constructed according to **A**, then δ is the last transition of η_{i+1}.
- If **B1** is applied then an invisible transition α_k from θ_i is appended to η_i to form η_{i+1} and δ is still the first visible transition of θ_{i+1}. The prefix $prefix_\delta(\theta_i)$ is either unchanged or shortened by the transition α_k.
- Otherwise an invisible transition β is appended to η_i to form η_{i+1} and $prefix_\delta(\theta_{i+1}) = prefix_\delta(\theta_i)$. □

Lemma 10. *Let v be a prefix of $vis(\sigma)$. Then there exists a path η_i such that $v = vis(\eta_i)$.*

Proof: By induction to the length of v. For the basic step $\mid v \mid= 0$ the statement holds trivially. For the induction step we must prove that if $v \cdot \delta$ is a prefix of $vis(\sigma)$ and there is a path η_i such that $vis(\eta_i) = v$, then there is a path η_j with $j > i$ such that $vis(\eta_{i+1}) = v \cdot \delta$. Thus, we need to show that δ will be eventually added to η_j for some $j > i$, and that no other visible transition will be added to η_k for $i < k < j$. According to the case **A** in the construction, we may add a visible transition to the end of η_k to form η_{k+1} only if it appears as the first transition of θ_k. Lemma 9 shows that δ remains the first visible transition in successive paths θ_k after θ_i unless it is being added to some η_j. Moreover, the sequence of transitions before δ can only shrink. Lemma 8 shows that the first transition in each θ_k is eventually removed and added to the end of some η_l for $l > k$. Thus, δ as well is eventually added to some sequence η_j. □

Proof: of Theorems 4 and 6

We will show that the described path $\eta = \lim_{i\to\infty} \eta_i$ is stutter equivalent to the original path σ.

First note that $vis(\sigma) = vis(\eta)$. It follows from Lemma 10 that for every prefix of σ there is a prefix of η with the same sequence of visible transitions. The opposite follows from Lemma 6.

Next we construct two infinite sequences of indexes $0 = i_0 < i_1 < \ldots$ and $0 = j_0 < j_1 < \ldots$ that define corresponding stuttering blocks of σ and η, as required in Definition 3. For every natural n, let i_n be the length of the smallest prefix ξ_{i_n} of σ that contains exactly n visible transitions. Let j_n be the length of the smallest prefix η_{j_n} of η that contains the same sequence of visible transitions as ξ_{i_n}. Recall that η_{j_n} is a prefix of π_{j_n}. Then by Lemma 6, $L(s_{i_n}) = L(r_{j_n})$. By the definition of visible transitions we also know that if $n > 0$, for $i_{n-1} \leq k < i_n - 1$, $L(s_k) = L(s_{i_{n-1}})$. This is because i_{n-1} is the length of the smallest prefix $\xi_{i_{n-1}}$ of σ that contains exactly $n-1$ visible transitions. Thus, there is no visible transition between i_{n-1} and $i_n - 1$. Similarly, for $j_{n-1} \leq l < j_n - 1$, $L(r_l) = L(r_{j_{n-1}})$. □