CHALMERS
UNIVERSITY OF TECHNOLOGY

Unified and Simplified Bisimulation

Citation for the original published paper (version of record):

Lennartson, B. (2024). Unified and Simplified Bisimulation. IFAC Proceedings Volumes (IFAC-PapersOnline), 58(1): 222-227. http://dx.doi.org/10.1016/j.ifacol.2024.07.038

(article starts on next page)

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 58-1 (2024) 222–227

Unified and Simplified Bisimulation ⋆

Bengt Lennartson *

* *Division of Systems and Control, Chalmers University of Technology, SE-412 96 Göteborg, Sweden (e-mail: bengt.lennartson@chalmers.se).*

Abstract: Bisimulation is a powerful abstraction method, which can be used to perform model reduction, especially for modular transition systems. A unified formulation of strong, weak, stuttering, and branching bisimulation is presented. An ambiguity in branching bisimulation is also highlighted, and an equivalent reformulation is proposed where the ambiguity is avoided. A transitive and therefore an equivalence relation is also shown for the alternative formulation. A block transition based description that is more natural from a model reduction perspective is also shown to be equivalent to the original relation based bisimulations. All bisimulation formulations are based on general transition system models, which means that systems both including state and transition labels are handled in a unified way.

Keywords: Bisimulation, abstraction, transition systems, model reduction.

1. INTRODUCTION

The well known state space explosion in verification and synthesis of discrete event systems can be handled in different ways. One popular approach is to represent models by binary decision diagrams (BDDs) (Bryant, 1992). More recently SAT solvers have also shown to be very effective (Eén and Sörensson, 2004). An attractive alternative is to use abstractions such that reduced models, which preserve critical properties, can be used. One of the most well known abstractions is bisimulation, cf. Milner (1989); Park (1981), which is a general technique to determine if individual states of a transition system have the same future behavior. States with such common behavior are related and are said to be bisimilar. Bisimulation can therefore be used to reduce the number of states for transition systems.

Two bisimulation relations that have a strong coupling to temporal logic are 1) branching bisimulation for labeled transition systems and process algebra with event/transition labels (automata), see Van Glabbeek and Weijland (1996); Nicola and Vaandrager (1995), and 2) stuttering bisimulation for Kripke structures including state labels, see Baier and Katoen (2008). In Lennartson and Noori-Hosseini (2018), these two formulations are unified in a bisimulation, where both state and event labels are included. Two other similar formulations that unify state and transition labels are presented in Gerth et al. (1999) and Trčka (2007). They are, however, based on traditional relation based bisimulation formulations. Our alternative definition is directly formulated as an equivalence relation. All earlier bisimulation definitions are based on relations that are shown to be equivalence relations, sometimes including complex proofs, especially for branching bisimulation, cf. Van Glabbeek et al. (2009).

In this paper the four most common bisimulations, strong, weak, stuttering, and branching bisimulation are formulated in a unified manner, by introducing a generic transition operator followed by specific instances. Traditional relation formulations are given, as well as our block transition formulations, which by construction generate non-trivial equivalence relations. The branching bisimulation formulation is evaluated in more detail and an ambiguity in the original formulation is highlighted. Due to this ambiguity it was shown by a famous counter example in Basten (1996) that the original branching bisimulation is not transitive. The suggested solution was then to reformulate the relation to a similar one called semi-branching bisimulation.

Here we suggest instead a more natural solution (for those who prefer the original definition), namely to avoid the ambiguity by introducing an equivalent relation formulation, but also to introduce a formulation that more naturally unify branching and stuttering bisimulation. This bisimulation is proven to be an equivalence relation. The equivalence between the relation formulation and our block transition formulation is also shown.

The paper starts in Section 2 with a survey on the most important results for the basic bisimulation, also called strong bisimulation. In Section 3 a deeper analysis of branching and stuttering bisimulation is given, including the ambiguity mentioned above and an equivalence proof. In Section 4 weak bisimulation is presented in the unified framework, including both state and transition labels, followed by a unification in Section 5 of strong, weak, and branching bisimulation, all three also formulated in the alternative block transition formulation. In this section it is also proven that the traditional relation based bisimulation formulation and our block transition formulation are equivalent. Finally, some conclusions are given where interesting future investigations are added.

2. BISIMULATION

Bisimulation is a binary relation that determines which individual states in a transition system have the same future behavior. States with such common behavior are said to be *bisimilar*. The basic bisimulation relation, which is also called *strong bisimulation* (Milner, 1989; Park, 1981), is defined and illustrated in this section. It is also emphasized that strong bisimulation becomes an equivalence relation when the maximum number of pairs is included. The name of the relation is motivated by the fact that it is the strongest and most detailed bisimulation which is naturally formulated.

2.1 Strong bisimulation

Before the definition of strong bisimulation is given, a *transition system* G is defined as a six-tuple $G = \langle X, \Sigma, T, I, AP, \lambda \rangle$ where X is a set of states, Σ is a finite set of events, $T \subseteq X \times \Sigma \times X$ is a transition relation, where $t = (x, a, x') \in T$ includes

⋆ This work was supported by SyTec – Systematic Testing of Cyber-Physical Systems, a Swedish Science Foundation grant for strong research environment. The support is gratefully acknowledged.

10.1016/j.ifacol.2024.07.038

the source state x, the event label a, and the target state x' of the transition t, $I \subseteq X$ is a set of possible initial states, AP is a set of atomic propositions, and $\lambda : X \to 2^{AP}$ is a state labeling function. A transition (x, a, x') is also denoted $x \xrightarrow{a} x'$. A transition system without state labels is called an automaton or a labeled transition system (LTS), and a transition system without transition labels (events) is called a Kripke structure (Baier and Katoen, 2008).

Definition 1. (Strong bisimulation). Given a transition system $G = \langle X, \Sigma, T, I, AP, \lambda\rangle$, a binary relation $R \subseteq X \times X$ is a *strong bisimulation* (SB) if, for any states $x, y \in X$ and event $a \in \Sigma$, $xRy \Rightarrow \lambda(x) = \lambda(y) \wedge p_R^{SB}(x, y) \wedge p_R^{SB}(y, x)$, where the *transfer predicate*

$$p_R^{SB}(x, y) := \forall x', a : x \xrightarrow{a} x' \Rightarrow \exists y' : y \xrightarrow{a} y' \wedge x'Ry'.$$

Related states $(x, y) \in R$ are said to be *strongly bisimilar*, denoted $x \sim_s y$. Furthermore, if both x and y are terminal states and $\lambda(x) = \lambda(y)$, then $x \sim_s y$. □

The transfer predicate $p_R^{SB}(x, y)$ holds if, for the two source states x and y, every transition $x \xrightarrow{a} x'$, for all existing target states $x' \in X$ and events $a \in \Sigma$, is matched by at least one transition $y \xrightarrow{a} y'$, and the target states are also related, i.e. $x'Ry'$. Since the symmetric transfer predicate $p_R^{SB}(y, x)$ is also included, the bisimulation relation xRy only holds if furthermore every transition $y \xrightarrow{a} y'$ is matched by at least one transition $x \xrightarrow{a} x'$ and $y'Rx'$. This symmetric condition is further illustrated in Example 1.

Both state and transition labels Bisimulation is normally defined either for Kripke structures, only including state labels, or alternatively for labeled transition systems or process algebra (Milner, 1989), only including transition labels. In this paper both state and transition labels are accepted in bisimulation relations. Indeed, state labels are simply introduced by adding the equality condition $\lambda(x) = \lambda(y)$. Thus, a prerequisite for two states to be bisimilar is that they have the same state label.

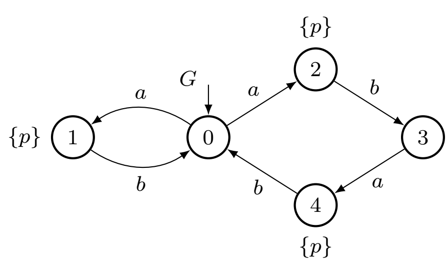

Figure 1. Transition system G with no state label in state 0 and 3, and state label $\{p\}$ in the remaining states.

Example 1. Consider the transition system G in Fig. 1. Based on Def. 1, the relation $R = \{(1, 4), (4, 1), (0, 0)\}$ is a strong bisimulation. The first pair $(1, 4) \in R$, since $\lambda(1) = \lambda(4) = \{p\}$, $1 \xrightarrow{b} 0$ is matched by $4 \xrightarrow{b} 0$ and $(0, 0) \in R$, and symmetrically $4 \xrightarrow{b} 0$ is matched by $1 \xrightarrow{b} 0$ and $(0, 0) \in R$. By the same argument also the second pair $(4, 1) \in R$, and $(0, 0) \in R$ since each state has the same future as itself. This is the most common and obvious strong bisimulation, where generally two source states with the same state and transition label, reaching the same target state, are bisimilar; they have the same future behavior. In Example 2 it will be shown that G includes a number of additional strongly related states. □

2.2 Equivalence relation and quotient transition system

Based on the strong bisimulation relation in Def. 1, an equivalence relation can be achieved. It is obtained by taking the union of all possible strong bisimulations, also called the *maximal strong bisimulation*. The following proposition states that this relation is an equivalence relation. A minor extension of the proof of Lemma 7.4 in Baier and Katoen (2008) proves this statement.

Proposition 1. (Strong bisimulation equivalence). The maximal strong bisimulation relation

$$R = \{(x, y) \mid \lambda(x) = \lambda(y) \wedge p_R^{SB}(x, y) \wedge p_R^{SB}(y, x)\},$$

where the transfer predicate $p_R^{SB}(x, y)$ is defined in Def. 1, is reflexive, symmetric, and transitive, and therefore an equivalence relation. □

Quotient transition system $G/\sim$ To obtain reduced transition systems, equivalent states for any equivalence relation are merged into equivalence classes $[x] = \{y \in X \mid x \sim y\}$, also called blocks. These blocks, which are non-overlapping subsets of X, divide the state space into the quotient set $X/\sim$, also called a partition Π of X. The block/equivalence class including state x is denoted $\Pi(x) = [x]$. A partition Π_1 that is finer than a partition Π_2 means that $\Pi_1(x) \subseteq \Pi_2(x)$ for all $x \in X$. It is denoted $\Pi_1 \preceq \Pi_2$.

Blocks are the states in reduced transition systems, and the notion partition Π is used in the computation of this model, while the resulting reduced model takes the equivalence perspective. For a transition system G, the reduced model is therefore called *quotient transition system*, denoted $G/\sim$, and for partition Π the state space of $G/\sim$ is the quotient set $X/\sim = \{[x] \mid [x] = \Pi(x)\}$.

Relation between G and $G/\sim_s$ By generating an extended model, including both G and the quotient transition system $G/\sim_s$ based on Def. 1, it can also be shown (Baier and Katoen, 2008) that every state $x \in X$ in G is strongly bisimilar to the corresponding block state $[x] \in X/\sim_s$ in $G/\sim_s$, that is $[x] \sim_s x$. The equivalence between every state x in G and corresponding block state $[x]$ in $G/\sim_s$ also means that the complete transition systems G and $G/\sim_d$ are said to be *strongly bisimulation equivalent*, denoted $G \sim_s G/\sim_s$.

Example 2. For the transition system G in Fig. 1 the maximal strong bisimulation relation is

$$R = \begin{matrix}\{(0,0),(0,3),\\ (3,0),(3,3)\}\end{matrix} \cup \begin{matrix}\{(1,1),(1,2),(1,4),\\ (2,1),(2,2),(2,4),\\ (4,1),(4,2),(4,4)\},\end{matrix}$$

which clearly shows the reflexive, symmetric, and transitive properties of this relation. The partition $\Pi = \{\{0, 3\}, \{1, 2, 4\}\}$ generates the reduced quotient transition system $G/\sim_s$ in Fig. 2. The repeated ab string means that $G/\sim_s$ only includes the block state $\{0, 3\}$ followed by the event a, and the block state $\{1, 2, 4\}$ followed by the event b. Note that the individual states in G are followed by the same event as the related block state in $G/\sim_s$. □

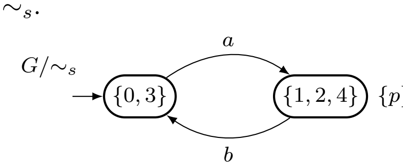

Figure 2. Quotient transition system $G/\sim_s$ of the transition system G in Fig. 1.

3. BRANCHING AND STUTTERING BISIMULATION

Events that are not synchronized with other subsystems are said to be local, and they can be *hidden* by replacing them with the transition label τ. Such τ events are also called *internal* or *silent* events. Some τ transitions $x \xrightarrow{\tau} x'$ can be removed by joining the source and target states into a block state $\{x, x'\}$, but not all of them.

Both in weak bisimulation (WB) and branching bisimulation (BB) some τ transitions are removed. WB is less restrictive than BB, which implies that the block states in a WB equivalence are often somewhat larger, resulting in less block states and a courser state partition. See further details in Section 4. On the other hand, more details are preserved in BB, where the name expresses that the branching structure is preserved. More specifically this implies that most temporal logic properties are preserved when abstraction is based on BB, while some relevant temporal properties are lost when WB is applied.

Normally, BB does not consider any state labels, but in this paper this is generalized to include both transition and state labels. BB for this type of general transition systems, without any label restrictions, is not common but has been formulated under the names visible bisimulation Gerth et al. (1999); Lennartson and Noori-Hosseini (2018) and silent bisimulation Trčka (2007). Since the only difference between these bisimulations and BB is that state labels are also included in a straightforward way, we keep the BB name used for labeled transition systems also for general transition systems, including both state and transition labels. This means that *stuttering bisimulation* is a special case, where no transition labels are involved. This can be interpreted as all transitions being labelled by τ, and the transition system is reduced to a Kripke structure.

3.1 Invisible transitions and stuttering paths

Before BB is presented, some basic results on invisible and visible transitions are presented. Some τ transitions are *visible* and some are *invisible*, while all transitions with label $a \neq \tau$ are visible.

Definition 2. (Invisible relation and invisible/visible transition). Given a transition system with state space X, a symmetric binary relation $R \subseteq X \times X$ is called an *invisible relation* if, for any states $x, y \in X$, $xRy \Rightarrow \lambda(x) = \lambda(y)$, and the transfer property

$$\forall x' : x \xrightarrow{\tau} x' \wedge xRy \Rightarrow x'Ry$$

holds. A τ transition that satisfies this transfer property is said to be an *invisible transition*, and consequently it is said to be a *visible transition* when it does not satisfy this transfer property. A transition $x \xrightarrow{a} x'$ where $a \neq \tau$ is always visible. □

A path only including τ-transitions is now defined, followed by a transfer property for this path.

Definition 3. (Stuttering path). Consider a transition system with a path $x = x_0 \xrightarrow{\tau} x_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} x_n = x'$, $n \geq 0$, also denoted $x \xrightarrow{\tau^*} x'$. When all the individual τ transitions in this path are invisible based on the invisible relation R in Def. 2, this path is called a *stuttering path*, denoted $x \xrightarrow[R]{\tau^*} x'$. □

Proposition 2. (Transfer property for stuttering path). Given a transition system with state space X, for any state $x \in X$ with a stuttering path $x \xrightarrow[R]{\tau^*} x'$ where $x = x_0 \xrightarrow{\tau} x_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} x_n = x'$, $n \geq 0$, all individual states are related to each other, i.e. $x_i R x_j$, $i, j \in \mathbb{N}_n$. Furthermore, for any state $y \in X$, the transfer property for this stuttering path is

$$\forall x' : x \xrightarrow[R]{\tau^*} x' \wedge xRy \Rightarrow x'Ry$$

Proof: Given an invisible relation R with transfer property according to Def. 2, this relation is transitive. It follows since $\forall x' : x \xrightarrow{\tau} x' \wedge xRy \wedge yRz \Rightarrow x'Ry \wedge yRz$ means that $\forall x' : x \xrightarrow{\tau} x' \wedge xR \circ Rz \Rightarrow x'R \circ Rz$. Hence, the relation $R \circ R$ is also an invisible relation.

Since every individual transition $x_j \xrightarrow{\tau} x_{j+1}$, $0 \leq j < n$, in the stuttering path is invisible, i.e. $\forall x_{j+1} : x_j \xrightarrow{\tau} x_{j+1} \wedge x_j Ry \Rightarrow x_{j+1}Ry$, the corresponding invisible transfer property for the specific choice $y = x_j$ implies that $x_{j+1}Rx_j$. Repeating the transitive property of this relation $i-j-1$ times $(i > j)$ implies due to the symmetry of R that $x_i R x_j$, $i, j \in \mathbb{N}_n$, and more specifically we find that $x'Rx$. Finally, the transfer property $\forall x' : x \xrightarrow[R]{\tau^*} x' \wedge xRy \Rightarrow x'Ry$ is trivially satisfied for $n = 0$, and for $n > 0$ it follows, since $x'Rx \wedge xRy$ implies that $x'Ry$. □

3.2 Branching Bisimulation

Branching bisimulation, as introduced in Van Glabbeek and Weijland (1996), is now defined for transition systems also including state labels.

Definition 4. (Branching bisimulation). Given a transition system $G = \langle X, \Sigma, T, I, AP, \lambda \rangle$, a binary relation $R \subseteq X \times X$ is a *branching bisimulation* (BB) if, for any states $x, y \in X$ and event $a \in \Sigma$, $xRy \Rightarrow \lambda(x) = \lambda(y) \wedge p_R^{BB}(x, y) \wedge p_R^{BB}(y, x)$. The *transfer predicate*

$$p_R^{BB}(x, y) := \forall x', a : x \xrightarrow{a} x' \Rightarrow p_R^{iv}(x', y) \vee p_R^a(x, x', y),$$

where

$$p_R^{iv}(x', y) := a = \tau \wedge x'Ry \tag{1}$$

$$p_R^a(x, x', y) := \exists y', y'' : y \xrightarrow{\tau^*} y'' \xrightarrow{a} y' \wedge xRy'' \wedge x'Ry'. \tag{2}$$

see Fig. 3. If both x and y' are terminal states, $\lambda(x) = \lambda(y)$, and $y \xrightarrow[R]{\tau^*} y'$, then xRy'. Related states $(x, y) \in R$ are said to be *branching bisimilar*, denoted $x \sim_b y$. □

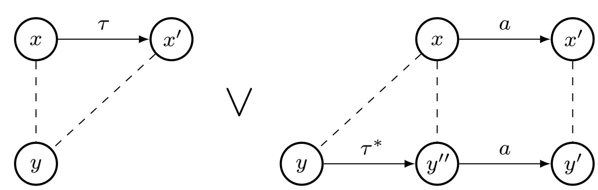

Figure 3. Transfer diagrams for the two disjunctive relations in branching bisimulation.

Example 3. Consider the transition system G in Fig. 4. Based on Def. 4 the relation $R = \{(0, 4), (4, 0), (1, 3), (3, 1), (1, 4), (4, 1), (2, 3), (3, 2), (2, 4), (4, 2), (5, 5)\}$ is a (non-maximal) BB. The pair $(1, 3) \in R$, since according to predicate p_R^{iv} in (1), $1 \xrightarrow{\tau} 2$ and $(2, 3) \in R$, and due to symmetry $3 \xrightarrow{\tau} 4$ and $(4, 1) \in R$. However, according to predicate p_R^a in (2), $(1, 3) \in R$ also because

$$1 \xrightarrow{\tau} 2 \text{ is matched by } 3 \xrightarrow{\tau} 4 \text{ and } (2, 4) \in R,$$
$$3 \xrightarrow{\tau} 4 \text{ is matched by } 1 \xrightarrow{\tau} 2 \text{ and } (4, 2) \in R.$$

□

This example illustrates that the predicate p_R^{iv} (1) (corresponding to Def. 2), but sometimes also the predicate p_R^a (2), hold for branching bisimilar invisible transitions. On the other hand, p_R^{iv} never holds for any visible transitions, only for branching bisimilar invisible transitions. Indeed, this ambiguity creates a problem, showing that BB is a transitive relation and therefore an equivalence relation.

By a famous counter example in Basten (1996) it was shown that the BB relation R in Def. 4 is not transitive. The suggested

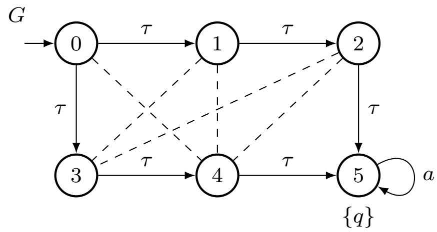

Figure 4. Transition system G, where related branching bisimilar states are connected by dashed lines.

solution was then to reformulate BB to a similar relation called semi-BB. Here we suggest instead a more natural solution (for those who prefer the original BB definition), namely to exclude the second predicate p_R^a to be involved in any invisible transitions.

3.3 Visible invisible separated branching bisimulation

Since $p \vee q \equiv p \vee (\neg p \wedge q)$, the transfer predicate p_R^{BB} in Def. 4 can equivalently be formulated as

$$p_R^{BB}(x,y) := \ \forall x', a : x \xrightarrow{a} x' \Rightarrow$$
$$(a = \tau \wedge x'Ry) \vee \big(\neg(a = \tau \wedge x'Ry) \wedge p_R^a(x,x',y)\big). \quad (3)$$

The fact that this predicate is equivalent to p_R^{BB} in Def. 4 does not mean that the individual expressions in these two predicates have the same meaning. The second part in the two disjunctions is obviously different. This implies that in the end of Example 3, the matching transitions, due to the predicate p_R^a, are removed in the determination of the BB relation pair $(1,3)$.

This reformulation of p_R^{BB} also means that the predicate p_R^a can be sharpened to include a stuttering path $y \overset{\tau^*}{\underset{R}{\to}} y''$, according to Def. 3. The result is presented in the following proposition.

Proposition 3. (Branching bisimulation). Given a transition system $G = \langle X, \Sigma, T, I, AP, \lambda\rangle$, a binary relation $R \subseteq X \times X$ is a *branching bisimulation* (BB) if, for any states $x, y \in X$ and event $a \in \Sigma$,

$$xRy \ \Rightarrow\ \lambda(x) = \lambda(y) \wedge p_R^{BB}(x,y) \wedge p_R^{BB}(y,x).$$

The *transfer predicate*

$$p_R^{BB}(x,y) := \forall x', a : x \xrightarrow{a} x' \Rightarrow p_R^{iv}(x',y) \vee p_R^{v}(x,y),$$

where the predicate for *invisible transitions* $p_R^{iv}(x',y)$ is given in Def. 4, while the predicate for *visible transitions*

$$p_R^{v}(x,y) := \neg p_R^{iv}(x',y) \wedge p_R^a(x,x',y),$$

and, compared to (2), p_R^a is reformulated as

$$p_R^a(x,x',y) := \exists y', y'' : y \overset{\tau^*}{\underset{R}{\to}} y'' \xrightarrow{a} y' \wedge xRy'' \wedge x'Ry'.$$

Proof: Repeat the transfer property for invisible relations in Def. 2. For $i \in \mathbb{N}_n = \{0, \ldots, n\}$ this implies that $y_i \xrightarrow{\tau} y_{i+1} \wedge y_iRx \Rightarrow y_{i+1}Rx$. Thus, y_iRx for $i \in \mathbb{N}_n$, and according to Def. 3, $y \xrightarrow{\tau^*} y''$, where $y = y_0$ and $y'' = y_n$, is a stuttering path denoted $y \overset{\tau^*}{\underset{R}{\to}} y''$. □

The BB formulated in this proposition clearly shows the separation between invisible and visible transitions, handled by the corresponding predicates p_R^{iv} and p_R^{v}, respectively. This separation simplifies the following relatively short lemma, which shows that the BB formulated in Prop. 3 is transitive.

Lemma 4. (Branching bisimulation is transitive). Based on the BB relation R in Prop. 3, it follows that

$$xR\circ Rz \Rightarrow p_{R\circ R}^{BB}(x,z).$$

Proof: According to (1) and Prop. 3,

$$\begin{aligned}
&xR\circ Rz \wedge \forall x', a : x \xrightarrow{a} x' \equiv\\
&\quad \exists y : xRy \wedge yRz \wedge \forall x', a : x \xrightarrow{a} x' \Rightarrow\\
&\quad \exists y\, \forall x', a : x \xrightarrow{a} x' \wedge yRz \Rightarrow\\
&\quad (a = \tau \wedge x'Ry \wedge yRz) \vee\\
&\quad \big(\neg(a = \tau \wedge x'Ry) \wedge p_R^a(x,x',y) \wedge yRz\big)
\end{aligned}$$

Since $p_R^{iv}(x',y) \wedge yRz \equiv (a = \tau \wedge x'Ry \wedge yRz) \equiv p_{R\circ R}^{iv}(x',z)$ and $\neg p_R^{iv}(x',y) \wedge yRz \equiv \neg(p_R^{iv}(x',y) \wedge yRz) \wedge yRz \equiv \neg p_{R\circ R}^{iv}(x',z) \wedge yRz$, the expression $xR\circ Rz \wedge \forall x', a : x \xrightarrow{a} x'$ implies that

$$\forall x', a : p_{R\circ R}^{iv}(x',z) \vee \big(\neg p_{R\circ R}^{iv}(x',z) \wedge \exists y : p_R^a(x,x',y) \wedge yRz\big) \quad (4)$$

According to Prop. 2, $y \overset{\tau^*}{\underset{R}{\to}} y'' \wedge yRz \Rightarrow y''Rz$. Furthermore, $p_R^{iv}(y',z) \wedge x'Ry' \equiv p_{R\circ R}^{iv}(x',z)$, and $\neg p_R^{iv}(y',z) \wedge x'Ry' \equiv \neg p_{R\circ R}^{iv}(x',z) \wedge x'Ry'$. Therefore,

$$\begin{aligned}
&\exists y : p_R^a(x,x',y) \wedge yRz\\
&\quad \Rightarrow \exists y, y', y'' : y \overset{\tau^*}{\underset{R}{\to}} y'' \wedge yRz \wedge y'' \xrightarrow{a} y' \wedge xRy'' \wedge x'Ry'\\
&\quad \Rightarrow \exists y', y'' : y''Rz \wedge y'' \xrightarrow{a} y' \wedge xRy'' \wedge x'Ry'\\
&\quad \Rightarrow \exists y', y'' : (p_{R\circ R}^{iv}(x',z) \wedge xRy'') \vee \big(\neg p_{R\circ R}^{iv}(x',z) \wedge x'Ry'\\
&\qquad \wedge xRy'' \wedge \exists z', z'' : z \overset{\tau^*}{\underset{R}{\to}} z'' \xrightarrow{a} z' \wedge y''Rz'' \wedge y'Rz'\big)
\end{aligned}$$

Inserting this result in (4) gives

$$\begin{aligned}
&xR\circ Rz \Rightarrow \forall x', a : x \xrightarrow{a} x' \Rightarrow p_{R\circ R}^{iv}(x',z) \vee \big(\neg p_{R\circ R}^{iv}(x',z) \wedge\\
&\exists z', z'' : z \overset{\tau^*}{\underset{R}{\to}} z'' \xrightarrow{a} z' \wedge xR\circ Rz'' \wedge x'R\circ Rz'\big)
\end{aligned}$$

which finally can be expressed as

$$xR\circ Rz \Rightarrow p_{R\circ R}^{BB}(x,z)$$

□

Theorem 5. (Branching bisimulation equivalence). Given a transition system $G = \langle X, \Sigma, T, I, AP, \lambda\rangle$, the BB relation R formulated in Prop. 3 is an equivalence relation.

Proof: The BB formulation in Prop. 3 is by definition symmetric, and Lemma 4 shows that if xRy and yRz are BB relations, then $xR\circ Rz$ is also a BB relation. Finally, by replacing y with x in Prop. 3, xRx is also a BB relation. Thus, the BB formulation in Prop. 3 is an equivalence relation. □

Proposition 6. (Extended branching bisimulation). By introducing the notation

$$x \overset{a}{\underset{R}{\twoheadrightarrow}} x' := \exists x'' : x \overset{\tau^*}{\underset{R}{\to}} x'' \xrightarrow{a} x'$$

where it is also assumed that $a \neq \tau \vee \neg x'Ry$, the transfer predicate for the BB relation in Prop. 3 can equivalently be expressed as

$$p_R^{BB}(x,y) := \forall x', a : x \overset{a}{\underset{R}{\twoheadrightarrow}} x' \Rightarrow \exists y' : y \overset{a}{\underset{R}{\twoheadrightarrow}} y' \wedge x'Ry'.$$

Proof: The condition $a \neq \tau \vee \neg x'Ry$ comes from the visible predicate in Prop. 3. The rest follows by combining the transfer property for the stuttering path in Prop. 2, where x' is replaced by x'', with the transfer property for the BB in Prop. 3, with x replaced by x''. □

This transfer predicate has the same form as the transfer predicate for the SB relation in Def. 1. In the next section a corresponding formulation will be given for WS.

4. WEAK BISIMULATION

As already mentioned, WB is less restrictive than BB, which results in less block states and a courser state partition. Before this is illustrated by a small example, a definition of WB on the same form as SB is introduced (Milner, 1989).

Definition 5. (Weak bisimulation). Given a transition system $G = \langle X, \Sigma, T, I, AP, \lambda \rangle$, a binary relation $R \subseteq X \times X$ is a *weak bisimulation* (WB) if, for any states $x, y \in X$ and event $a \in \Sigma$, $xRy \Rightarrow \lambda(x) = \lambda(y) \wedge p_R^{WB}(x,y) \wedge p_R^{WB}(y,x)$, and using the notation

$$x \overset{a}{\twoheadrightarrow} x' := \exists x'', x''' : x \overset{\tau^*}{\rightarrow} x'' \overset{a}{\rightarrow} x''' \overset{\tau^*}{\rightarrow} x'$$

the *transfer predicate*

$$p_R^{WB}(x,y) := \forall x', a : x \overset{a}{\twoheadrightarrow} x' \Rightarrow \exists y' : y \overset{a}{\twoheadrightarrow} y' \wedge x'Ry'.$$

If both x and y are terminal states and $\lambda(x) = \lambda(y)$, $y \overset{\tau^*}{\rightarrow} y'$, then xRy'. Related states $(x,y) \in R$ are said to be *weakly bisimilar*, denoted $x \sim_w y$. □

Example 4. For the transition system G in Fig. 5, branching bisimulation does not give any reduction, while the first two states are joined in the weak bisimular quotient transition system $G/{\sim_w}$. The reason is that WB does not differentiate between the direct path to the final state $0 \overset{a}{\rightarrow} 3$ and the path via state 1 which also has an alternative b transition to the final state. This reduction is not accepted by BB that preserves the branching information. □

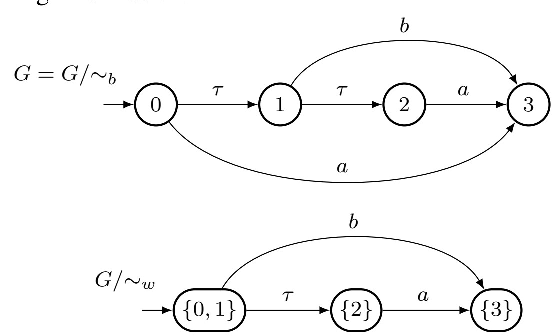

Figure 5. Transition system G that is equal to the branching bisimalar quotient transition system $G/{\sim_b}$ and the coarser weak bisimalar quotient transition system $G/{\sim_w}$.

5. UNIFIED BISIMULATION RELATION AND EQUIVALENCE

The three bisimulations we have presented in this paper, where stuttering bisimulation has been incorporated in BB, can all three be formulated in the same way by introducing, for a transition label a, a generic transition operator $\overset{a}{\hookrightarrow}$. This results in the following generic bisimulation relation.

Definition 6. (Generic bisimulation relation). Given a transition system $G = \langle X, \Sigma, T, I, AP, \lambda \rangle$, a binary relation $R \subseteq X \times X$ is a *generic bisimulation* if, for any states $x, y \in X$ and event $a \in \Sigma$, the implication

$$xRy \Rightarrow \lambda(x) = \lambda(y) \wedge p_R(x,y) \wedge p_R(y,x)$$

holds, where the *transfer predicate*

$$p_R(x,y) := \forall x', a : x \overset{a}{\hookrightarrow} x' \Rightarrow \exists y' : y \overset{a}{\hookrightarrow} y' \wedge x'Ry'$$

Related states $(x,y) \in R$ are said to be *bisimilar*, denoted $x \sim y$. □

In the same way as in Section 2, a maximal bisimulation relation can be formulated based on the generic bisimulation in Def. 6, resulting in an equivalence relation, see Prop. 1. For BB, a detailed equivalence proof was presented in Section 3.

An alternative formulation of BB was presented in Lennartson and Noori-Hosseini (2018), directly based on a state partition, and therefore directly generating an equivalence formulation. This formulation is here called bisimulation equivalence, and will in this section be proved to be equivalent to the maximal generic bisimulation relation, but also coupled to the specific bisimulation relations presented in this paper.

Definition 7. (Bisimulation equivalences). Given a transition system $G = \langle X, \Sigma, T, I, AP, \lambda \rangle$, a partition Π, which for any state $x \in X$ satisfies the greatest fixpoint

$$\Pi(x) = \{ y \in X \mid \lambda(y) = \lambda(x) \wedge T_\Pi(x) = T_\Pi(y) \}, \quad (5)$$

is a *bisimulation equivalence* where the set of block transitions

$$T_\Pi(x) = \{ \Pi(x) \overset{a}{\rightarrow} \Pi(x') \mid \exists x', a : x \overset{a}{\hookrightarrow} x' \}. \quad (6)$$

(i) It is a *strong bisimulation equivalence* when the generic transition operator $\overset{a}{\hookrightarrow}$ is replaced by $\overset{a}{\rightarrow}$, and states $x, y \in \Pi(x)$ are strongly bisimilar, denoted $x \sim_s y$.

(ii) It is a *weak bisimulation equivalence* when the generic transition operator $\overset{a}{\hookrightarrow}$ is replaced by $\overset{a}{\twoheadrightarrow}$, and states $x, y \in \Pi(x)$ are weakly bisimilar, denoted $x \sim_w y$.

(iii) It is a *branching bisimulation equivalence* when the generic transition operator $\overset{a}{\hookrightarrow}$ is replaced by $\underset{\Pi}{\overset{a}{\twoheadrightarrow}}$, and states $x, y \in \Pi(x)$ are branching bisimilar, denoted $x \sim_b y$. □

The equivalence between these bisimulation equivalences and corresponding maximal relations will now proved.

Theorem 7. (Bisimulation equivalences and maximal relations). The bisimulation equivalence given by the fixpoint (5) in Def. 7 is equivalent to the maximal relation

$$R = \{ (x,y) \mid \lambda(x) = \lambda(y) \wedge p_R(x,y) \wedge p_R(y,x) \} \quad (7)$$

where the transfer predicate $p_R(x,y)$ is defined in Def. 6. Strong, weak, and branching bisimulation are obtained by replacing the generic transition operator $\overset{a}{\hookrightarrow}$ with $\overset{a}{\rightarrow}$, $\overset{a}{\twoheadrightarrow}$, and $\underset{R}{\overset{a}{\twoheadrightarrow}}$, respectively.

Proof: For an equivalence relation R, the relation predicate $xRy \equiv y \in \Pi(x)$. The equality $T_\Pi(x) = T_\Pi(y)$ in (5) means that $\Pi(x) = \Pi(y)$ and $\Pi(x') = \Pi(y')$, and therefore $y \in \Pi(x)$ and $y' \in \Pi(x')$. Furthermore, $T_\Pi(x) = T_\Pi(y)$ can equivalently be formulated as $\big(T_\Pi(x) \subseteq T_\Pi(y)\big) \wedge \big(T_\Pi(y) \subseteq T_\Pi(x)\big)$. The definition of subsets applied to $T_\Pi(x) \subseteq T_\Pi(y)$ then gives the equivalent formulation

$$\forall x', a : \Pi(x) \overset{a}{\rightarrow} \Pi(x') \in T_\Pi(x) \Rightarrow \Pi(x) \overset{a}{\rightarrow} \Pi(x') \in T_\Pi(y)$$

Furthermore, the fact that $y \in \Pi(x) = \Pi(y)$ and $y' \in \Pi(x') = \Pi(y')$ results in one more equivalent expression

$$\forall x', a : \Pi(x) \overset{a}{\rightarrow} \Pi(x') \in T_\Pi(x) \Rightarrow$$
$$\exists y' : \Pi(y) \overset{a}{\rightarrow} \Pi(y') \in T_\Pi(y) \wedge y' \in \Pi(x')$$

Finally, the definition of $T_\Pi(x)$ (6) in Def. 7 gives the equivalent implication

$$\forall x', a : x \overset{a}{\hookrightarrow} x' \Rightarrow \exists y' : y \overset{a}{\hookrightarrow} y' \wedge y' \in \Pi(x')$$

This corresponds to the predicate $p_R(x,y)$ in Def. 6, with $y' \in \Pi(x')$ replaced by the equivalent predicate $x'Ry'$ and more specifically for BB $\underset{\Pi}{\overset{a}{\twoheadrightarrow}}$ replaced by the equivalent transition operator $\underset{R}{\overset{a}{\twoheadrightarrow}}$. Hence, the fixpoint for $\Pi(x)$ can be expressed as

$$\Pi(x) = \{ y \in X \mid \lambda(x) = \lambda(y) \wedge p_R(x,y) \wedge p_R(y,x) \},$$

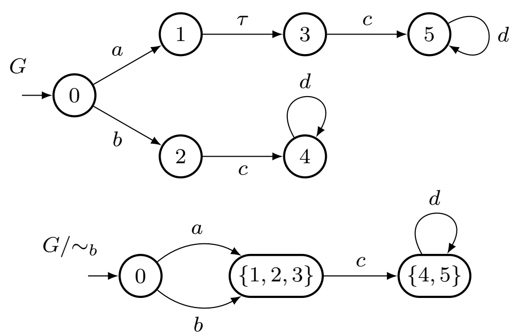

Figure 6. Transition system G and its branching bisimilar quotient transition system $G/{\sim_b}$.

which corresponds to the maximal relation (7). The results for the strong and weak bisimulations follows in the same way, where the only difference is the transition operator that is changed to the actual bisimulation. □

What we have achieved is an alternative bisimulation formulation, which more directly focuses on the model reduction application, where the partition and the block states in the reduced model are directly expressed in the bisimulation. Furthermore, this formulation directly results in a fixed point algorithm that can be used both for hand calculations and as an efficient computational algorithm, which is easily formulated for parallel computations. This result is not surprising, since the idea behind this alternative formulation is the signature algorithm, proposed in Blom and Orzan (2003), where our block states are closely related to their signatures. Our formulation includes state labels, and can therefore also be applied to stuttering bisimulation. The proofs in this paper are also more simple compared to the signature formulation. The fixed point iteration is presented in the following proposition, and is then illustrated by an example.

Proposition 8. (Bisimulation greatest fixpoint). The greatest fixpoint of (5) in Def. 7 is obtained by iterating

$$\Pi_{k+1}(x) = \{y \in X \,|\, T_{\Pi_k}(x) = T_{\Pi_k}(y)\} \tag{8}$$

until $\Pi_{k+1}(x) = \Pi_k(x)$, with the initial partition $\Pi_0(x) = \{y \,|\, \lambda(x) = \lambda(y)\}$.

Proof: The block $\Pi_{k+1}(x) = \{y \in X \,|\, T_{\Pi_k}(x) = T_{\Pi_k}(y)\}$, and therefore $\Pi_{k+1}(x) \preceq \Pi_k(x)$ for all states $x \in X$ and $k \geq 0$. This holds since

$$\begin{aligned}\Pi_{k+1}(x) = \{y \in X \,|\, \exists x', y', a :\\ (\Pi_k(x), a, \Pi_k(x')) = (\Pi_k(y), a, \Pi_k(y'))\} \preceq \Pi_k(x)\end{aligned}$$

Obviously, $\Pi_{k+1}(x)$ is further restricted compared to $\Pi_k(x)$ by also requiring the next block states to be equal, i.e. $\Pi_k(x') = \Pi_k(y')$. Thus, the iteration in (8) generates a monotonically decreasing partition Π_k as k increases, until the greatest fixpoint $\Pi_{k+1} = \Pi_k$ is reached, according to Knaster–Tarski's famous fixed point theorem (Tarski, 1955). □

Example 5. For the transition system G in Fig. 6, the first fixed point iteration of (8) gives for $\Pi_0 = X = \{0, 1, 2, 3, 4, 5\}$

$$\begin{aligned}&T_{\Pi_0}(0) = \{X \xrightarrow{a} X, X \xrightarrow{b} X\},\\ &T_{\Pi_0}(1) = T_{\Pi_0}(2) = T_{\Pi_0}(3) = \{X \xrightarrow{c} X\},\\ &T_{\Pi_0}(4) = T_{\Pi_0}(5) = \{X \xrightarrow{d} X\},\end{aligned}$$

resulting in $\Pi_1 = \{\{0\}, B_1, B_2\}$, where $B_1 = \{1, 2, 3\}$ and $B_2 = \{4, 5\}$. The second iteration gives

$$\begin{aligned}&T_{\Pi_1}(0) = \{\{0\} \xrightarrow{a} B_1, \{0\} \xrightarrow{b} B_1\},\\ &T_{\Pi_1}(1) = T_{\Pi_1}(2) = T_{\Pi_1}(3) = \{B_1 \xrightarrow{c} B_2\},\\ &T_{\Pi_1}(4) = T_{\Pi_1}(5) = \{B_2 \xrightarrow{d} B_2\},\end{aligned}$$

which results in the fixed point $\Pi_2 = \Pi_1$. The obtained branching bisimilar quotient transition system $G/{\sim_b}$ is also shown in Fig. 6. □

6. CONCLUSIONS

A unified formulation of strong, weak, stuttering, and branching bisimulation has been presented in this paper. An alternative block transition based formulation that is more natural from a model reduction and computational perspective is also shown to be equivalent to the original relation based bisimulation formulations. Future steps is to generalize the unified formulation to also include transitions with weights, typically representing time or energy cost.

ACKNOWLEDGEMENTS

Many thanks to Professor Rob van Glabbeek, the founder of BB, who kindly evaluated our proposed alternative bisimulation formulation and suggested to prove the equivalence between his original formulation and the one presented in Section 5.

REFERENCES

Baier, C. and Katoen, J.P. (2008). *Principles of Model Checking*. The MIT Press, Cambridge, MA.

Basten, T. (1996). Branching bisimilarity is an equivalence indeed! *Information Processing Letters*, 58(3), 141–147.

Blom, S. and Orzan, S. (2003). Distributed branching bisimulation reduction of state spaces. *Electronic Notes in Theoretical Computer Science*, 89, 99–113.

Bryant, R.E. (1992). Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, 24(3), 293–318.

Eén, N. and Sörensson, N. (2004). An extensible SAT-solver. In E. Giunchiglia and A. Tacchella (eds.), *Lecture Notes in Computer Science*, volume 2919, 502–518. Springer.

Gerth, R., Kuiper, R., Peled, D., and Penczek, W. (1999). A partial order approach to branching time logic model checking. *Journal of the ACM*, 150, 132–152.

Lennartson, B. and Noori-Hosseini, M. (2018). Visible bisimulation equivalence— a unified abstraction for temporal logic verification. In *Proc. 14th International Workshop on Discrete Event Systems, WODES'18*.

Milner, R. (1989). *Communication and Concurrency*. Series in Computer Science. Prentice-Hall.

Nicola, R.D. and Vaandrager, F. (1995). Three logics for branching bisimulation. *Journal of the ACM*, 42, 458–487.

Park, D.M.R. (1981). Concurrency and automata on infinite sequences. In P. Deussen (ed.), *5th GI-Conference on Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, 167–183. Springer, Karlsruhe, Germany.

Tarski, A. (1955). A Lattice-Theoretical Fixpoint Theorem and Its Applications. *Pacific Journal of Mathematics*, 5, 285–309.

Trčka, N. (2007). Silent Steps in Transition Systems and Markov Chains. IPA Dissertation Series 2007-08, Technische Universiteit Eindhoven.

Van Glabbeek, R.J., Luttik, B., and Trčka, N. (2009). Branching bisimilarity with explicit divergence. *Fundamenta Informaticae*, 93, 371–392.

Van Glabbeek, R.J. and Weijland, W.P. (1996). Branching time and abstraction in bisimulation semantics. *Journal of the ACM*, 43, 555–600.