Preorder-Constrained Simulation for Nondeterministic Automata

Koko Muroya
RIMS, Kyoto University, Japan

Takahiro Sanada
RIMS, Kyoto University, Japan

Natsuki Urabe
National Institute of Informatics, Tokyo, Japan

Abstract
We describe our ongoing work on generalizing some quantitatively constrained notions of weak simulation up-to that are recently introduced for deterministic systems modeling program execution. We present and discuss a new notion dubbed preorder-constrained simulation that allows comparison between words using a preorder, instead of equality.

2012 ACM Subject Classification Theory of computation → Verification by model checking

Keywords and phrases simulation, weak simulation, up-to technique, language inclusion, preorder

1 Introduction: Simulation Notions with Bounded Number of Steps

In the literature of program semantics, coinductive techniques have often been used to establish equivalence between program behaviors. A recent approach utilizes weak simulations with quantitative constraints on the length of terminating runs. These constraints enable comparison of execution cost for programs, in terms of the number of execution steps it takes for a program to terminate.

One example is Accattoli et al.’s notion called improvement [1]. It was used to show that certain rewriting of a program before execution not only preserves the execution result, but also improves the execution cost by requiring less execution steps. Another example was used in the first author’s previous work [9]. It is dubbed \((Q,Q_1,Q_2)\)-simulation, parameterized by a triple \((Q,Q_1,Q_2)\) of preorders on natural numbers, i.e. on lengths of runs. The first preorder \(Q\) is used to compare lengths of accepted runs, and it generalizes the “greater-than-or-equal” preorder \(\geq\) used by improvements. The other two preorders \(Q_1,Q_2\) are for additionally incorporating the so-called up-to technique. Subtle conditions on these preorders are identified in loc. cit. to make the combination of weak simulations and the up-to technique work.

These two notions are both designed for unlabeled deterministic transition systems, which can model execution of deterministic programs only. We aim to pursue the idea of constraining terminating, or accepted, runs, in a more general setting. This abstract describes our ongoing work on generalizing \((Q,Q_1,Q_2)\)-simulations to nondeterministic automata. We present a novel notion of preorder-constrained simulation that is a weak simulation up-to constrained by preorders on words, not on natural numbers. It entails a generalized notion of language inclusion that compares words using a preorder instead of equality.
Main Contribution

Let $A_k = (X_k, \Sigma, \tau_k \subseteq X_k \times \Sigma \times X_k, F_k \subseteq X_k)$ ($k \in \{1, 2\}$) be nondeterministic automata, $x \in X_1$ and $y \in X_2$, and $L_{A_1}^x(x), L_{A_2}^y(y) \subseteq \Sigma^*$ be the set of words accepted from $x$ and $y$ respectively. The ordinary simulation notion [7] proves language inclusion $L_{A_1}^x(x) \subseteq L_{A_2}^y(y)$. Instead, our simulation notion proves $Q$-trace inclusion.

**Def. 1.** For a preorder $Q \subseteq \Sigma^* \times \Sigma^*$, we write $x \preceq_Q y$ and say $Q$-trace inclusion holds between $x$ and $y$ when $\forall w \in L_{A_1}^x(x), \exists w' \in L_{A_2}^y(y), wQw'$.

**Example 2.**
i) when $Q$ is the equality, $x \preceq_Q y$ iff $L_{A_1}^x(x) \subseteq L_{A_2}^y(y)$.
i) When $\tau$ contains a special letter $\tau$, and $wQw'$ means that $w$ and $w'$ are the same except for $\tau$, then $x \preceq_Q y$ iff weak language inclusion, i.e. language inclusion ignoring $\tau$, holds.
i) When $wQw'$ means that $w$ is a subword of $w'$, $x \preceq_Q y$ iff for each $w \in L_{A_1}^x(x)$ there exists $w' \in L_{A_2}^y(y)$ such that $w$ is a subword of $w'$.
i) When $\Sigma$ is the powerset $2^\Sigma$, some set $A$ and $a_1 \ldots a_k \subseteq^* Qa_1^* \ldots a_k^*$ means that $k = k'$ and $a_i \subseteq a_i'$ for each $i \in \{1, \ldots, k\}$, then $x \preceq_Q y$ iff for each $a_1 \ldots a_k \in L_{A_1}^x(x)$ there exists $a_1' \ldots a_k' \subseteq^* L_{A_2}^y(y)$ such that $a_i \subseteq a_i'$ for each $i \in \{1, \ldots, k\}$.

2.1 Preorder-Constrained Simulation without up-to

We hereby introduce a new simulation notion for witnessing $Q$-trace inclusion.

**Def. 3.** We call $R \subseteq X_1 \times X_2$ a $Q$-constrained simulation from $A_1$ to $A_2$ if, for any $(x, y) \in R$, the following holds (see also Fig. 1).

**Final:** If $x \in F_1$ then there exist $w' \in \Sigma^*$ and $y' \in F_2$ such that $\psi Qw'$ and $y' \psi w'$. 

**Step:** For each $a_1 \ldots a_n \in \Sigma^*$ and $x_1 \ldots x_n \in X_1^+$ such that $x \psi a_1 \psi x_2 \psi \ldots \psi a_n \psi x_n$, and $x_n \in F_1$, there exist $k \in \{1, \ldots, n\}$, $w' \in \Sigma^*$ and $y' \in X_2$ such that $a_1 \ldots a_k \psi w'$, $w' \psi a_{k+1} \psi \ldots \psi a_n \psi y'$, and i) $x_k R y'$ or ii) $k = n$ and $y' \in F_2$.

**Example 4.** We continue Ex. 2(iv). Let $A_P = \mathbb{R}^2$. For a formula $\varphi(a, b)$ with free variables $a$ and $b$, let $\langle \varphi \rangle := \{(a, b) \in \mathbb{R}^2 \mid \varphi(a, b)\}$. For nondeterministic automata illustrated in Fig. 2, as $\langle a^2 + b^2 \leq 1 \rangle \subseteq \langle |a| + |b| \leq 1 \rangle$, $R := \{(x, y), (x_1, y_1), (x_2, y_2)\}$ is a $Q$-constrained simulation. Note that $x_1$, $y_11$ and $y_12$ are not involved by $R$.

**Prop. 5 (soundness).** If $Q$ is closed under concatenation (i.e. $w_1 Qw'_1$ and $w_2 Qw'_2$ imply $w_1 w_2 Q w'_1 w'_2$), $xRy$ implies $x \preceq_Q y$.

Unfortunately, it seems that $Q$-constrained simulation is not practicable as it is hard to check if given $R$ satisfies Step in Def. 3 for all $a_1 \ldots a_n \in \Sigma^*$ and $x_1 \ldots x_n \in X_1^+$. In fact, by letting $R := \{(x, y) \mid x \preceq_Q y\}$, we can easily see that $\preceq_Q$ is a $Q$-constrained simulation.

**Prop. 6 (completeness).** If $Q$ is closed under concatenation, there exists a $Q$-constrained simulation $R$ such that $x \preceq_Q y$ implies $xRy$.\(\blacksquare\)
This means that existence of a \( Q \)-constrained simulation relating two states is very difficult to determine in many cases. For example, ordinary language inclusion between nondeterministic automata (i.e. \( Q \)-trace inclusion when \( Q \) is the equality) is known to be PSPACE-complete [8]. We therefore consider approximating \( Q \)-constrained simulation. We consider fixing \( M \in \mathbb{N} \) and replacing Step with the following (here \( A^{[m,M]} := \bigcup_{m \leq i \leq M} A^i \)):

**Step\(^{\leq M} \):** For each \( a_1 \ldots a_n \in \Sigma^{[1,M]} \) and \( x_1 \ldots x_n \in \Sigma^{[1,M]} \) such that \( x \xrightarrow{a_1} x_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} x_n \) and either \( x_n \in F_1 \) or \( n = M \), there exist \( k \in \{1, \ldots, n\} \), \( w' \in \Sigma^* \) and \( y' \in X_2 \) such that \( a_1, \ldots, a_k \in Q w', y w'^2 y' \) and i) \( x_k R y' \) or ii) \( k = n < M \) and \( y' \in F_2 \).

**Prop. 7.** Given \( x \in X_1 \) and \( y \in X_2 \), existence of \( R \) satisfying Final, \( \text{Step}^{\leq M} \) and \( x R y \) can be checked in polynomial time to \(|\Sigma|, |X_1|, |X_2|\) and \( T_M(|\Sigma|, |X_2|) \). Here \( T_M(p, q) \) is the computation time for the following problem: given \( w \in \Sigma^{[1,M]} \) and a nondeterministic automaton whose alphabet and state space have sizes of \( p \) and \( q \), check if the set \( \{w' | w Q w'\} \) intersects with the language of the automaton.

As \( \text{Step}^{\leq M} \) implies Step, Prop. 5 still holds after the modification. Moreover, by Prop. 7, if \( M \) is fixed and \( T_M(|\Sigma|, |X_2|) \) is polynomial to \(|X_2|\) and \(|\Sigma|\) (it holds for all \( Q \) illustrated above), then existence of a \( Q \)-constrained simulation relating two states can be checked in polynomial time.

We conclude this section by giving a game theoretic characterization for \( Q \)-constrained simulations, namely the safety game in Fig. 4 played by Challenger and Simulator.

**Prop. 8.** Simulator is winning in the two-player game in Fig. 4 from a state \((\varepsilon, x, y)\) if and only if there exists a \( Q \)-constrained simulation \( R \) such that \( x R y \).

Intuitively, \( w \in \Sigma^* \) in Fig. 4 is understood as a queue that saves labels executed on \( A_1 \) by Challenger. Basically, Simulator can skip the turn until she can construct a path labeled by a word \( w' \in \Sigma^* \) such that \( w Q w' \). However, when an accepting state is reached on \( A_1 \), Challenger can also declare the last turn and force Simulator to construct a path immediately, although if Simulator succeeded in constructing a path then Challenger loses.

The modification of \( Q \)-constrained simulation stated above, i.e. replacing Step with \( \text{Step}^{\leq M} \), corresponds to replacing \( \Sigma^* \) and \( \Sigma^+ \) with \( \Sigma^{[0,M]} \) and \( \Sigma^{[1,M]} \) respectively, and prohibiting Simulator from skipping the turn when \(|w| = M \) in Fig. 4.

Figure 4 Two-player game characterizing \( Q \)-constrained simulation. Simulator wins if \( \text{sim-win} \) is reached, Challenger gets stuck, or a play continues infinitely.
2.2 Preorder-Constrained Simulation with up-to

We can think of an up-to variant of \( Q \)-constrained simulations from \( A_1 \) to \( A_2 \).

**Prop. 10.** Assume \( Q \) is closed under concatenation. If \( R_1 \subseteq \preceq Q_1 \) and \( R_2 \subseteq \preceq Q_2 \) for preorders \( Q_1, Q_2 \subseteq \Sigma^* \times \Sigma^* \) satisfying the following conditions, then \( xRy \) implies \( x \preceq Q y \): i) \( Q_1 \subseteq Q_2 \); and ii) \( wQ_1 w' \implies |w| \geq |w'| \).

Cond. (i) ensures compatibility of \( \preceq Q_1, \preceq Q_2 \) with \( \preceq Q \), and Cond. (ii) ensures safe integration of the up-to technique. They are strongly inspired by \((Q, Q_1, Q_2)\)-simulation for unlabeled and deterministic automata [9].

3 Related Work

The above notion is similar to buffered simulation [3], which was developed to enable more relations to witness language inclusion. Buffered simulations allow Simulator to skip his turn, to buffer Challenger’s moves and to simulate them later together, which has a similar flavor to our simulation notion (cf. Ex. 4). Hence our simulation notion can be also thought of as a generalization of buffered simulation.

Preorder-constrained simulations allow a quantitative reasoning such as comparing lengths of accepted runs. There exist quantitative simulation notions for comparing costs of weighted automata. Many of them are for probabilistic systems [6, 5, 4]. One simulation notion for automata weighted with costs was introduced as a matrix over real numbers [12]. A methodology for comparing infinite runs of weighted automata is also known [2]. In contrast to weighted automata, which are labeled with both letters and weights, our target is automata labeled with letters only. Quantities appear in the set of words, in our approach.

4 Research Directions

Our simulation notion focuses on finite languages. As is the case for the ordinary simulation notion, our notion may fail to prove inclusion of finite languages when there is no inclusion of infinite languages. We are looking into possible solutions.

We suspect that Cond. (ii) of Prop. 5, whose analogues are also in existing notions of weak simulation up-to, is too strong. We think \( Q_1 \) violating Cond. (ii) can be allowed finitely many times. However, at the same time, we should note that the relaxation makes the definition of simulations a global one, which can result in a more complicated algorithm for finding it. We should make sure that it does not ruin efficiency gained by up-to techniques.

Ex. 4 suggests that our simulation notion works well with systems whose alphabet \( \Sigma \) carries an order. Such a system also arises in the study of linear temporal logic (LTL). An LTL formula induces a Büchi automaton labeled with the powerset \( 2^{\text{AP}} \) of atomic propositions [13]. The alphabet \( 2^{\text{AP}} \) is ordered by the inclusion, which induces a preorder on \((2^{\text{AP}})^*\).
We are also interested in a categorical study of our simulation notion. One possible strategy would be to use the category $\text{PreOrd}$ of preordered sets as the base category. The nondeterministic branching would be then captured by the powerset functor (or possibly a monad) $\mathcal{P}$ lifted to $\text{PreOrd}$. The categorical generalization might allow us to transfer our simulation notion to systems with other branching types, e.g. probabilistic one.

References