Preorder-Constrained Simulation

Koko Muroya ✉
RIMS, Kyoto University, Japan

Takahiro Sanada ✉
RIMS, Kyoto University, Japan

Natsuki Urabe ✉
National Institute of Informatics, 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

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Category Early ideas abstract

Funding The first and second authors are supported by JST, ACT-X Grant No. JPMJAX190U, Japan. The third author is supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603).

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 [8]. It is dubbed (Q, Q_1, Q_2)-*simulation*, parameterized by a triple (Q, Q_1, Q_2) of preorders on natural numbers. This notion incorporates the so-called *up-to* technique, and the triple plays a crucial role to make the combination of weak simulations and the up-to technique work. The first preorder Q is used to compare lengths of accepted runs, generalizing the "greater-than-or-equal" preorder $\geq$ used by improvements.

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.

Preorder-Constrained Simulation Let $A_k = (X_k, \Sigma, \rightsquigarrow_k \subseteq X_k \times \Sigma \times X_k, F_k \subseteq X_k)$ $(k \in \{1, 2\})$ be nondeterministic automata, $x \in X_1$ and $y \in X_2$, and $L^*_{A_1}(x), L^*_{A_2}(y) \subseteq \Sigma^*$ be the set of words accepted from x and y respectively. The ordinary simulation notion [7] proves language inclusion $L^*_{A_1}(x) \subseteq L^*_{A_2}(y)$. Instead, for a preorder $\mathcal{Q} \subseteq \Sigma^* \times \Sigma^*$, we write $x \preceq_{\mathcal{Q}} y$ when $\forall w \in L^*_{A_1}(x).\ \exists w' \in L^*_{A_2}(y).\ w \mathcal{Q} w'$. Our simulation notion proves this.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:3
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Here are examples: when $\mathcal{Q}$ is the equality, $x \preceq_{\mathcal{Q}} y$ iff $L^*_{A_1}(x) \subseteq L^*_{A_2}(y)$. When Σ contains a special letter τ, and $w\mathcal{Q}w'$ means that w and w' are the same except for τ, then $x \preceq_{\mathcal{Q}} y$ iff language inclusion ignoring τ holds. When $w\mathcal{Q}w'$ means that w is a subword of w', $x \preceq_{\mathcal{Q}} y$ iff for each $w \in L^*_{A_1}(x)$ there exists $w' \in L^*_{A_2}(y)$ such that w is a subword of w'.

▶ **Def. 1.** Let $\mathcal{Q}, \mathcal{Q}_1, \mathcal{Q}_2 \subseteq \Sigma^* \times \Sigma^*$ be preorders. We call $R \subseteq X_1 \times X_2$ a $(\mathcal{Q}, \mathcal{Q}_1, \mathcal{Q}_2)$-*simulation* from A_1 to A_2 if, for any $(x, y) \in R$, the following holds.

$$\begin{tikzcd} x \arrow[r, rightsquigarrow, "w*"] \arrow[ddd, dash, dotted, "R"'] & x'' \arrow[d, dash, dotted, "\preceq_{\mathcal{Q}_1}"] \\ & \bullet \arrow[d, dash, dotted, "R"] \\ & \bullet \arrow[d, dash, dotted, "\preceq_{\mathcal{Q}_2}"] \\ y \arrow[r, rightsquigarrow, "w'*"] & y'' \end{tikzcd}$$

Final: $x \in F_1$ implies $y \overset{w}{\rightsquigarrow}{}^*_2\, y'$ for some $y' \in F_2$ and $w \in \Sigma^*$ such that $\varepsilon\mathcal{Q}w$.

Step: In the following game played on $X_1 \times \Sigma^*$ by Challenger and Simulator, Simulator is winning from a state (x, ε). In each round, a pebble on $(x', w) \in X_1 \times \Sigma^*$ is moved as follows.

1. Challenger chooses $a \in \Sigma$ and $x'' \in X_1$ such that $x' \overset{a}{\rightsquigarrow}_1 x''$, and let $w := wa$.
2. Simulator chooses either of the following: i) choose $y'' \in X_2$ and $w' \in \Sigma^*$ such that $y \overset{w'}{\rightsquigarrow}{}^*_2\, y''$, $w\mathcal{Q}w'$ and $x'' \preceq_{\mathcal{Q}_1} R \preceq_{\mathcal{Q}_2} y''$, and end the game; or ii) skip his turn.

Simulator wins the game if (i) is chosen on his turn.

▶ **Prop. 2.** *If the following conditions are satisfied, xRy implies $x \preceq_{\mathcal{Q}} y$: i) $w_1\mathcal{Q}w'_1$ and $w'_2\mathcal{Q}w'_2$ imply $w_1w_2\mathcal{Q}w'_1w'_2$; ii) $\mathcal{Q}_1\mathcal{Q}\mathcal{Q}_2 \subseteq \mathcal{Q}$; and iii) $w\mathcal{Q}_1w'$ implies $|w| \geq |w'|$.* ◀

It is known that a naïve combination of weak simulations and up-to techniques leads to unsoundness, and require special cares [9, 10]. In Prop. 2, it is dealt with by Cond.(iii).

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. 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 [11]. 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.

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. (iii) of Prop. 2, whose analogues are also in existing notions of weak simulation up-to, is too strong. We think $\mathcal{Q}_1$ violating Cond. (iii) 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.

Our simulation notion works well with systems whose alphabet Σ carries an order. Such a system arises in the study of linear temporal logic (LTL). An LTL formula induces a Büchi automaton labeled with the powerset 2^{AP} of atomic propositions [12]. The alphabet 2^{AP} is ordered by the inclusion, which induces a preorder on $(2^{\mathsf{AP}})^*$.

We are also interested in a categorical study of our simulation notion. One possible strategy would be to use the category **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 **PreOrd**. The categorical generalization might allow us to transfer our simulation notion to systems with other branching types, e.g. probabilistic one.

References

1. Beniamino Accattoli, Ugo Dal Lago, and Gabriele Vanoni. The machinery of interaction. In *PPDP 2020*, pages 4:1–4:15. ACM, 2020.
2. Suguman Bansal, Swarat Chaudhuri, and Moshe Y. Vardi. Comparator automata in quantitative verification. In *FoSSaCS 2018*, volume 10803 of *Lecture Notes in Computer Science*, pages 420–437. Springer, 2018.
3. Milka Hutagalung, Martin Lange, and Étienne Lozes. Buffered simulation games for büchi automata. In Zoltán Ésik and Zoltán Fülöp, editors, *AFL 2014*, volume 151 of *EPTCS*, pages 286–300, 2014.
4. Bart Jacobs and Jesse Hughes. Simulations in coalgebra. *Electronic Notes in Theoretical Computer Science*, 82(1):128–149, 2003.
5. Bengt Jonsson and Kim Guldstrand Larsen. Specification and refinement of probabilistic processes. In *LICS 1991*, pages 266–277. IEEE Computer Society, 1991.
6. Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.
7. Nancy A. Lynch and Frits W. Vaandrager. Forward and backward simulations – Part I: Untimed systems. Technical report, NLD, 1993.
8. Koko Muroya. *Hypernet Semantics of Programming Languages*. PhD thesis, University of Birmingham, 2020.
9. Damien Pous. Up-to techniques for weak bisimulation. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 730–741. Springer, 2005.
10. Damien Pous. New up-to techniques for weak bisimulation. *Theoretical Computer Science*, 380(1):164–180, 2007. Automata, Languages and Programming.
11. Natsuki Urabe and Ichiro Hasuo. Generic forward and backward simulations III: quantitative simulations by matrices. In *CONCUR 2014*, volume 8704 of *Lecture Notes in Computer Science*, pages 451–466. Springer, 2014.
12. Moshe Y. Vardi and Pierre Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.