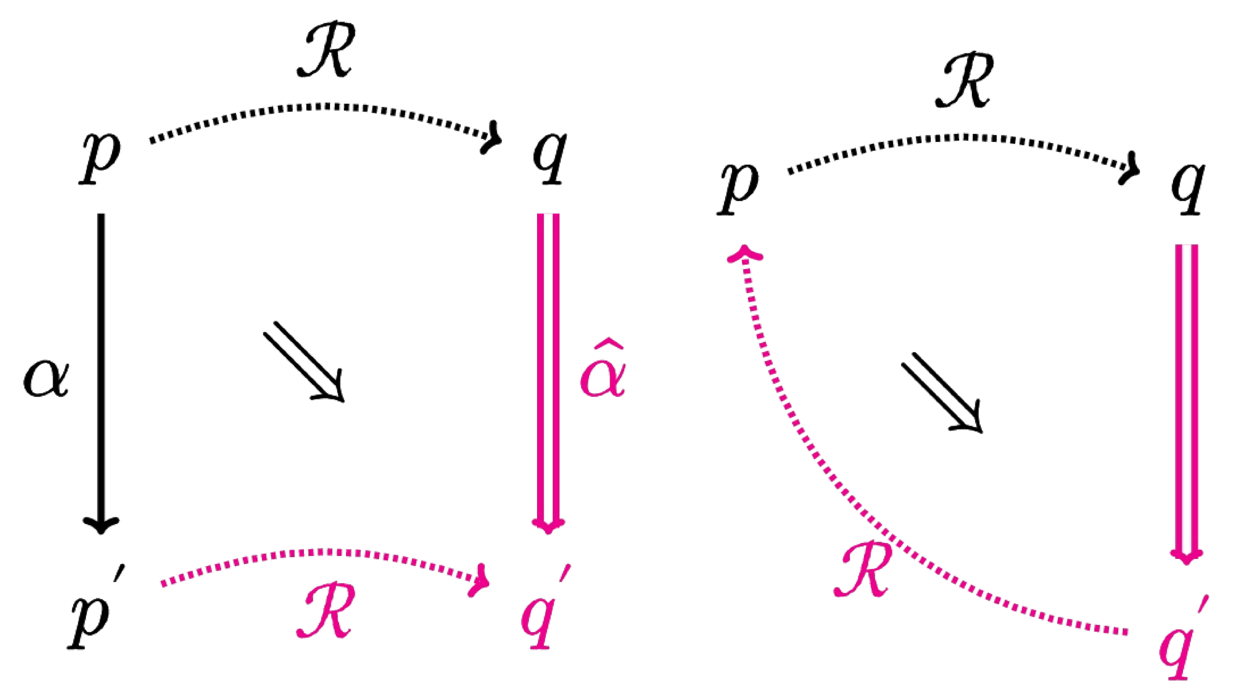
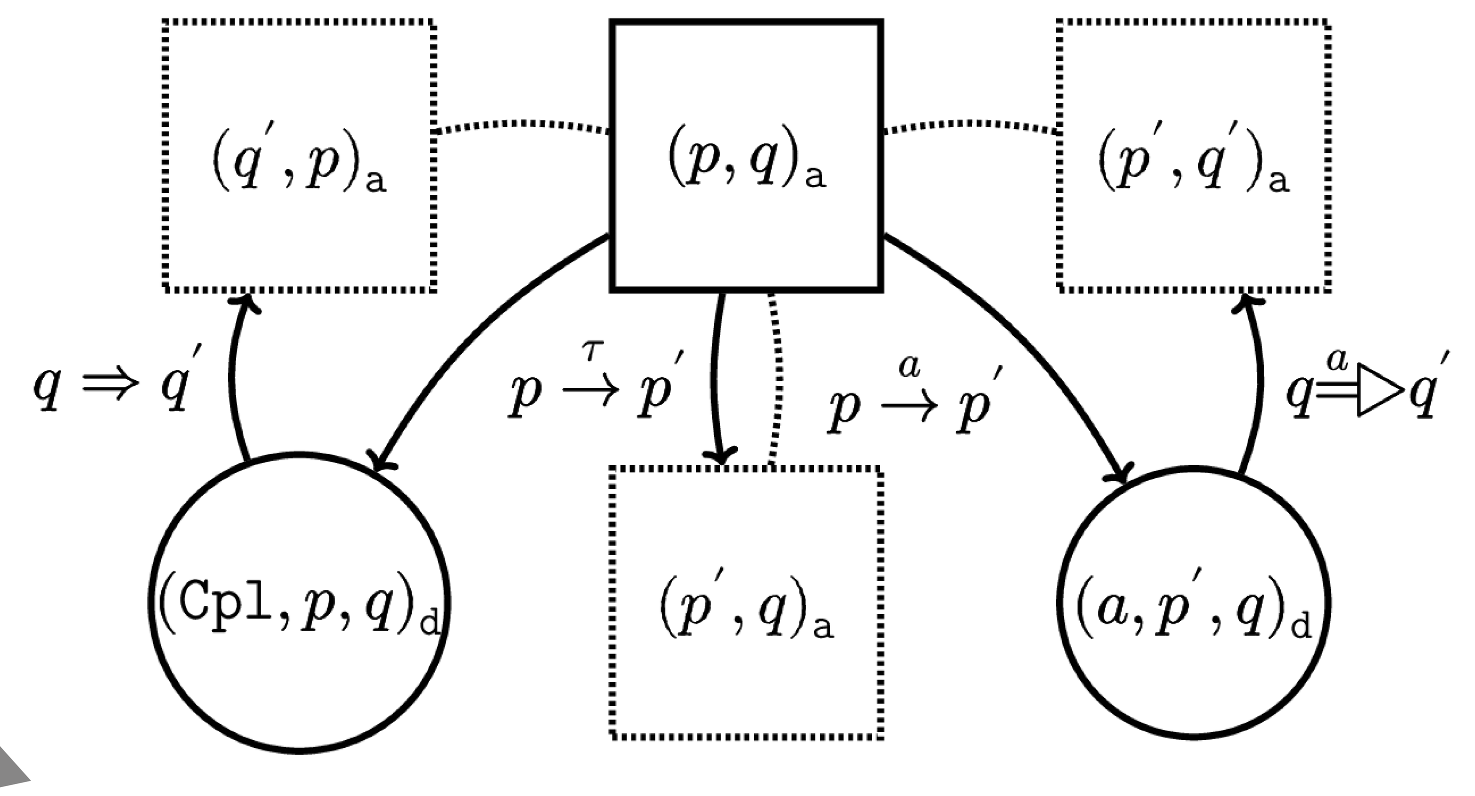


We formalized CS-definitions and proved new characterizations using Isabelle/HOL!



Simulation + Coupling

We proved that delay simulation suffices.

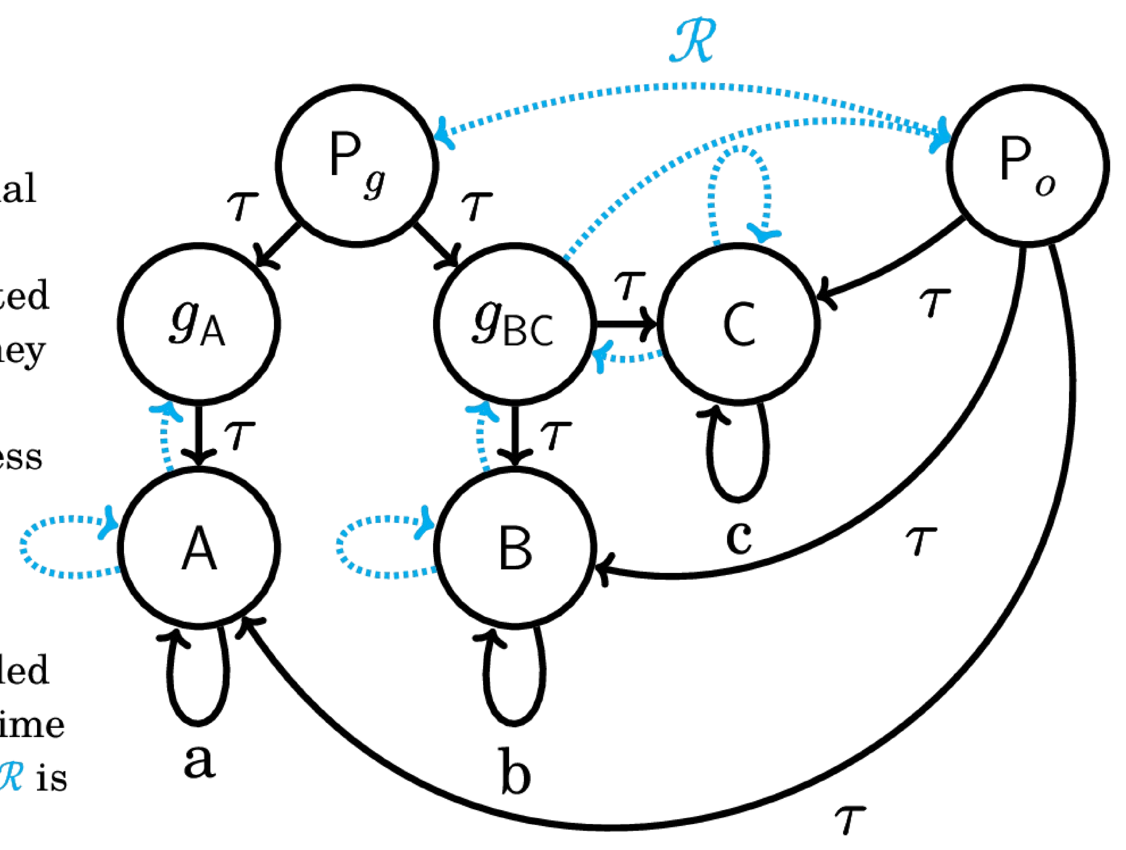
Coupled Simulation Game

The game characterization is the heart of our TACAS paper.

What
characterizes the coupled simulation preoder?

Why can't one just use weak bisimilarity?

Weak bisimilarity considers internal choices that can lead to the same results, like P_o and P_g in the depicted transition system, as different if they differ in the structure of partially committed states. This sensitiveness to gradual commitments, is too pedantic for branching-time semantics of programs with transparent distributability. Coupled similarity is the finest branching-time equivalence without the problem. $\mathcal{R}$ is a coupled simulation.



How
to compute it?

The game can be generated and solved in linear time of game size—and with massive parallelism! Exploiting the transitivity of the game, the computation of the τ-closure can be avoided, leading to space and time complexity of:

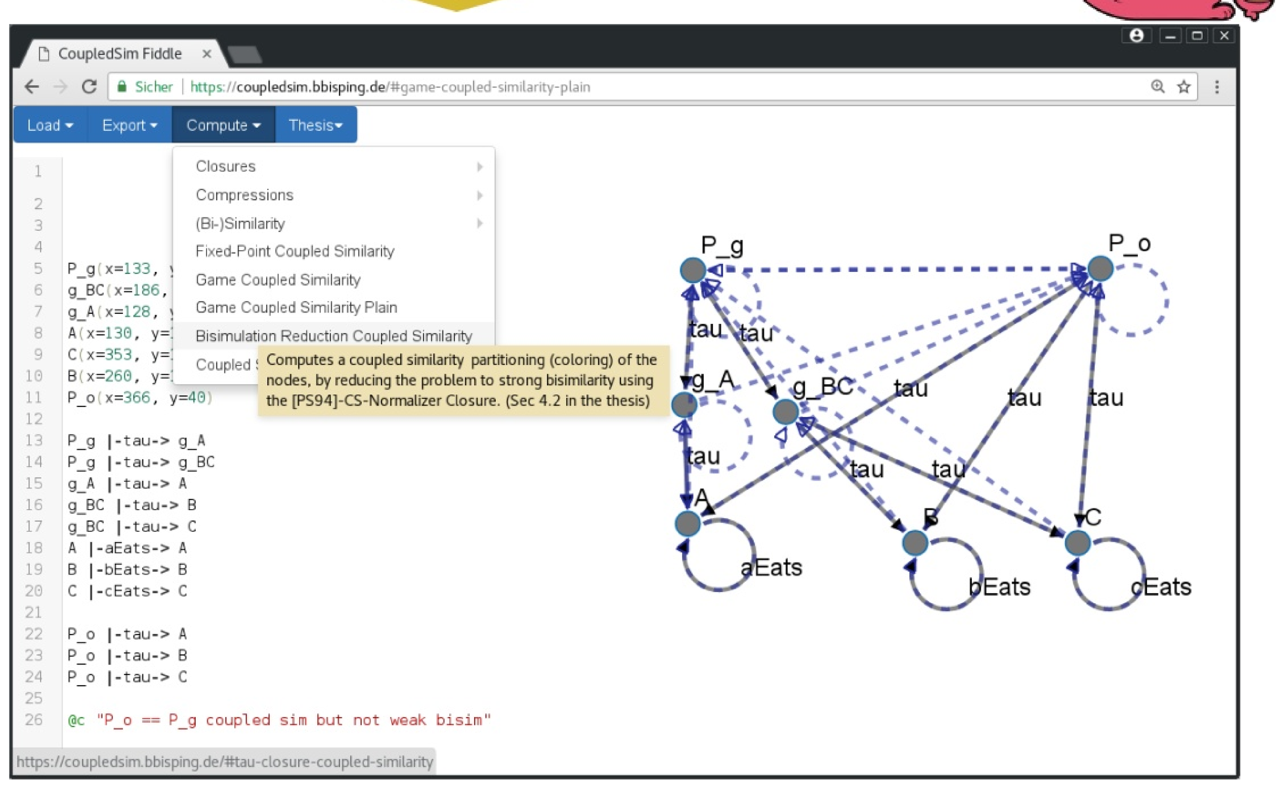
$$\mathcal{O}(\ |\dot{\rightarrow}|\,|S| + |\Sigma_\tau|\,|\overset{\tau}{\rightarrow}|\,|S|\ )$$

Computing Coupled Similarity

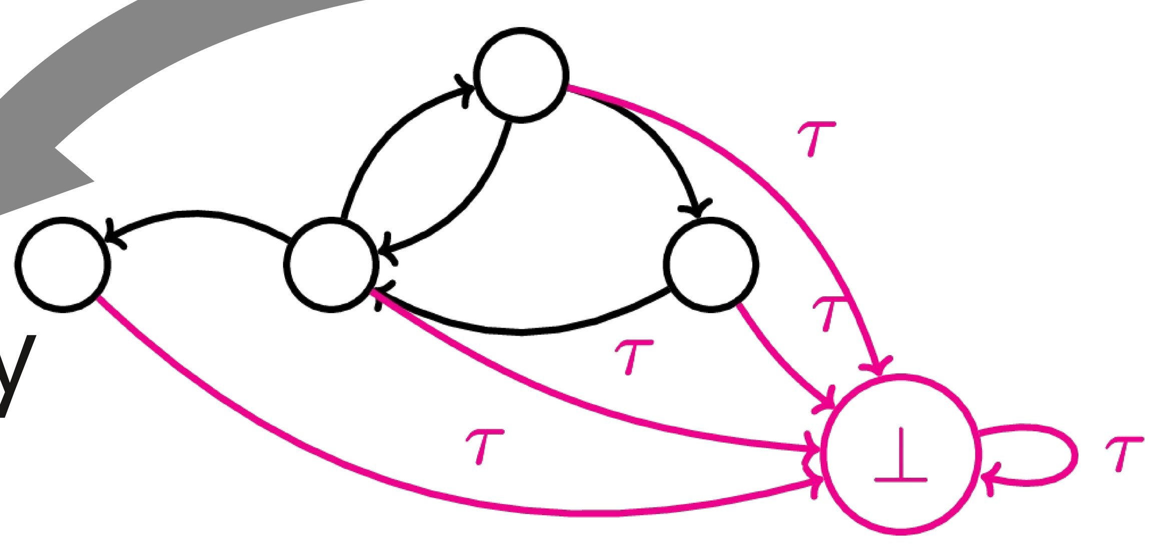
More material at:
coupledsim.bbisping.de

And in our paper "Computing Coupled Similarity" published in the TACAS 2019 proceedings.

We implemented a prototype using the scalable data-flow engine Apache Flink and provide a Scala.js webtool.



Is cubic complexity okay?



For any transition system $\mathcal{S} = (S, \Sigma_\tau, \rightarrow)$, coupled similarity and weak similarity coincide on its extension by a τ-sink:

$$\mathcal{S}^{\perp} := (S \cup \{\perp\},\quad \Sigma_\tau,\quad \rightarrow \cup \{(p, \tau, \perp) \mid p \in S \cup \{\perp\}\})$$

So, deciding weak similarity can be reduced to deciding coupled similarity. The best known similarity algorithms are cubic in the size of the transition system state space.

Because of the reduction, cubic time complexity probably is as good as it can get.

Technische Universität Berlin

Benjamin Bisping
benjamin.bisping@tu-berlin.de | @benkeks

Uwe Nestmann
uwe.nestmann@tu-berlin.de | @defconq