Bisimulation up-to Equivalence

Tobias Kappé

November 25, 2021

Abstract

You've seen that bisimulation is a fundamental notion, which arises from the functor of the coalgebra under study. In many cases, one can come up with an algorithm that *decides* bisimilarity of two states.

In this lecture, we will start by developing an elegant algorithm that allows us to check bisimilarity for a specific functor. We then go on to consider an optimization of this concrete algorithm, and argue that it is sound. The second half of the lecture presents the coalgebraic view of this optimization, and proves that it is sound for a wide class of functors.

The material in these notes is largely adapted from [RBR13].

1 Bisimulation checking

Let's look at the deterministic automaton functor $FX = 2 \times X^A$. Recall that an F-bisimulation on $x : X \to FX$ is a relation R on X equipped with a coalgebra structure $\gamma : R \to FR$, such that the following diagram commutes:

$$\begin{array}{ccccc} X & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & X \\ {\scriptstyle x}\downarrow & & {\scriptstyle \gamma}\downarrow & & {\scriptstyle x}\downarrow \\ FX & \xleftarrow[F\pi_1]{} & FR & \xrightarrow[F\pi_2]{} & FX \end{array} \tag{1}$$

We write $x_1 \leftrightarroweq x_2$ when $x_1 \mathrel{R} x_2$ for some such F-bisimulation $\gamma : R \to FR$.

We recall Example 2.20 from the lecture notes.

Fact 1.1. *Let $\langle o, d \rangle : X \to FX$ be an F-coalgebra, and let R be a relation on X. Then R is a bisimulation if and only if the following hold for all $x_1 \mathrel{R} x_2$.*

$$o(x_1) = o(x_2) \qquad\qquad \forall a \in A.\ d(x_1, a) \mathrel{R} d(x_2, a)$$

Proof. For the forward implication, let R be a bisimulation with structure $\gamma : R \to FR$. Now, if if $x_1 \mathrel{R} x_2$, then commutativity of the bisimilarity diagram allows us to derive the following equalities for all $a \in A$:

$$o(x_1) = \pi_1(\gamma(b)) = o(x_2) \qquad\qquad d(x_1)(a) = \pi_2(\gamma(b))(a) = d(x_2)(a)$$

For the converse, let R be a relation satisfying those two constraints. Then we can put a coalgebra structure $\langle O, D \rangle : R \to FR$ on R, by setting

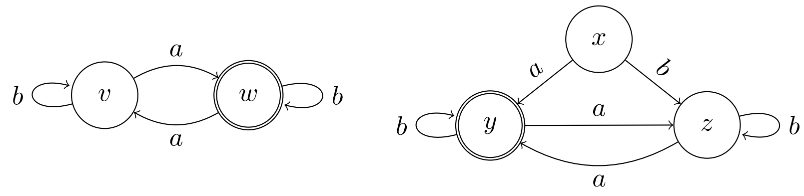
$$O(x_1, x_2) = o(x_1) = o(x_2) \qquad\qquad D(x_1, x_2)(a) = \langle d(x_1)(a), d(x_2)(a) \rangle$$

Both are well-defined because of the conditions on R. This makes the projections $\pi_1, \pi_2 : R \to X$ coalgebra homomorphisms, completing the proof. $\square$

1.1 Deriving an algorithm

The characterization derived above gives us a way to check F-bisimilarity "on the fly". To see how this works, let's look at some concrete examples.

Example 1.2. Suppose you have the following F-coalgebra $\langle o, d \rangle : X \to FX$:



States are indicated by circles. A double circle around x indicates that $o(x) = 1$; otherwise, $o(x) = 0$. Edges indicate transitions, e.g., $d(v)(a) = w$.

Suppose you want to show that v is bisimilar to x. By Fact 1.1, it is (necessary and) sufficient to construct a relation R that relates v to x while satisfying those two conditions. Because $v \mathrel{R} x$, we can derive some facts. For instance,

$$d(v)(a) \mathrel{R} d(x)(a) \quad \text{or, in other words,} \quad w \mathrel{R} y$$

by the second condition. This, together with the second condition, implies that

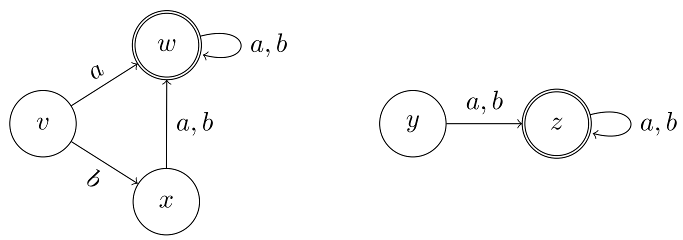
$$d(v)(b) \mathrel{R} d(x)(b) \quad \text{or, equivalently,} \quad v \mathrel{R} z$$

The latter could also lead us to conclude that $d(v)(a) \mathrel{R} d(z)(a)$. But this is precisely the same as $w \mathrel{R} y$ — something we already knew! The same holds for the other pairs we added: applying the second rule does not tell us anything new. In summary, we know that any relation we compute must satisfy

$$\{\langle v, x\rangle, \langle w, y\rangle, \langle v, z\rangle\} \subseteq R$$

But this set on the left already validates the conditions from Fact 1.1: it is closed under transitions (by construction), and all pairs are equally accepting Hence, we can conclude that v is bisimilar to z, as witnessed by this relation.

Example 1.3. Let's try a negative example. Consider the F-coalgebra drawn below, and suppose we were interested in whether v is bisimilar to y.



By the same technique as the previous example, we can find that any relation R with $v \mathrel{R} y$ that satisfies the second condition of Fact 1.1 should satisfy

$$\{\langle v, y\rangle, \langle w, z\rangle, \langle x, z\rangle\} \subseteq R$$

But because $o(x) \neq o(z)$ and $x \mathrel{R} z$, such an R can never satisfy the first condition! It follows that this R cannot exist, and hence v is *not* bisimilar to z.

We can further concretize this approach into the following algorithm, which simultaneously builds a list of pairs that should be in R by iterating the second condition, while also checking that the pairs we add satisfy the first condition. It works by keeping a "todo list" T of pairs to be checked Each checked pair puts further pairs on the "todo list", until we reach a fixpoint.

1. Initialize R to the empty set, and set $T = \{\langle x_1, x_2 \rangle\}$.
2. While T is not empty, do the following:
   (a) Remove $\langle x'_1, x'_2 \rangle$ from T.
   (b) If $\langle x'_1, x'_2 \rangle \in R$, then start the loop again.
   (c) If $o(x'_1) \neq o(x'_2)$, then decide negatively (no bisimulation possible).
   (d) Otherwise, add $\langle x'_1, x'_2 \rangle$ to R.
   (e) Furthermore, add $\{\langle d(x'_1, a), d(x'_2, a) \rangle : a \in A\}$ to T.
3. Decide positively (R is now a bisimulation).

Fact 1.4. *Suppose that the input coalgebra X is finite. Then the algorithm above is correct, in that it decides positively if and only if x_1 is bisimilar to x_2.*

Proof sketch. For termination, note that in each iteration either $X^2 \setminus R$ stays the same and T shrinks, or $X^2 \setminus R$ shrinks. Because X^2 is finite, this means that the loop (and hence the algorithm) eventually terminates.

For partial correctness, the main loop satisfies the following invariants:

1. If $R' \subseteq X \times X$ with $x_1 \mathrel{R} x_2$, and R' is a bisimulation, then $R \cup T \subseteq R'$.
2. If $\langle x'_1, x'_2 \rangle \in R$, then $o(x'_1) = o(x'_2)$ and $\langle d(x'_1)(a), d(x'_2)(a) \rangle \in R \cup T$.

If the algorithm decides negatively in the loop for the pair $\langle x'_1, x'_2 \rangle$, suppose towards a contradiction that x_1 is bisimilar to x_2. Then there exists a relation R' on X such that $x_1 \mathrel{R'} x_1$, and R' satisfies the conditions of Fact 1.1. But by the invariant, $x'_1 \mathrel{R'} x'_2$, despite $o(x'_1) \neq o(x'_2)$. This is a contradiction; we must therefore conclude that x_1 is indeed not bisimilar to x_2.

Conversely, if the algorithm decides positively, then $T = \emptyset$. The second invariant then implies that R satisfies the conditions of Fact 1.1. Since $x_1 \mathrel{R} x_2$ by construction, we must conclude that x_1 is bisimilar to x_2. □

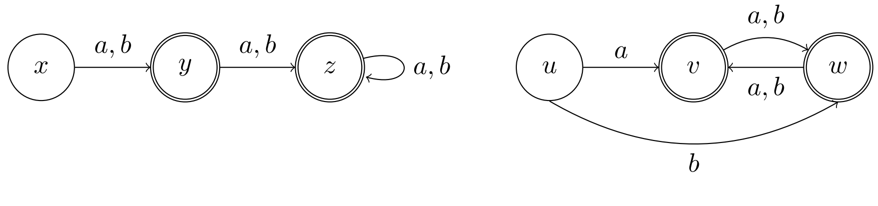
1.2 Optimization using equivalence

Sometimes, however, our algorithm can stop early, because we can sometimes conclude bisimilarity of pairs from other pairs, because of the following fact:

Fact 1.5. *The bisimilarity relation $\leftrightarroweq$ derived from F is an equivalence.*

The following example illustrates how we can exploit this.

Example 1.6. Consider the F-coalgebra depicted below [Rot15, Example 4.2.2]



If we run our algorithm to decide whether x is bisimilar to u, we end up constructing the following relation, with the pairs added in the order given:

$$R = \{\langle x,u\rangle, \langle y,v\rangle, \langle y,w\rangle, \langle z,w\rangle, \langle y,v\rangle, \langle z,v\rangle\}$$

But look at R when we added $\langle z,v\rangle$. At that point, the algorithm already decided that it needed to prove that $z \Leftrightarrow w$, $w \Leftrightarrow y$, and $y \Leftrightarrow v$. If all of these are truly bisimilar, then $z \Leftrightarrow v$ follows automatically by Fact 1.5. This begs the question: can we somehow avoid adding $\langle z,v\rangle$ to R based on this information, and by extension avoid adding all of its successors to T?

The next fact formally clarifies the intuition of the previous example.

Fact 1.7. *Let x_1 and x_2 be states of some F-coalgebra $\langle o,d\rangle$, with $FX = 2\times X^A$. Now $x_1 \Leftrightarrow x_2$ if and only if there exists a relation R such that the following hold:*

$$o(x_1) = o(x_2) \qquad\qquad \forall a \in A.\ d(x_1,a)\ \overline{R}\ d(x_2,a)$$

where $\overline{R}$ is the smallest equivalence relation containing R.

Proof. For the forward implication, note R satisfies the conditions of Fact 1.1. Since $R \subseteq \overline{R}$, the properties above hold immediately.

For the backward implication, a straightforward proof shows that if R is a relation satisfying the two conditions above, then $\overline{R}$ satisfies the conditions of Fact 1.1. Since $x_1\ \overline{R}\ x_2$, we then conclude that $x_1 \Leftrightarrow x_2$ by the same. □

We can adapt the algorithm from before, by adjusting step 2b to check whether $\langle x_1', x_2'\rangle \in \overline{R}$ instead. There exist efficient data structures to represent equivalence relations that allow you to do this. The algorithm that arises from this is due to Hopcroft and Karp [HK71].

Fact 1.8. *The adapted algorithm is also correct.*

Proof sketch. The structure of this proof is similar to the one for Fact 1.4, except that instead of appealing to Fact 1.1, we use Fact 1.7. The second loop invariant also uses $\overline{R} \cup T$ instead of $R \cup T$. □

2 Generalization

We considered a bisimulation checking algorithm derived for a specific functor F, and we showed that we could optimize it by working "up to equivalence". Similar algorithms to the first one can be derived for many different functors. The question then arises: can these algorithms likewise be optimized by the same principle? To answer this question, we must think categorically. First, let's formulate the coalgebraic equivalence of the relations from Fact 1.7.

Let's fix a functor F and a coalgebra $x : X \to FX$. A *bisimulation up-to equivalence* is a relation R on X, equipped with a coalgebra structure $\gamma : R \to FR$, such that the following diagram commutes:[1]

$$\begin{array}{ccccc} X & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & X \\ \downarrow{\scriptstyle x} & & \downarrow{\scriptstyle \gamma} & & \downarrow{\scriptstyle x} \\ FX & \xleftarrow[F\overline{\pi_1}]{} & F\overline{R} & \xrightarrow[F\overline{\pi_2}]{} & FX \end{array} \qquad (2)$$

[1]Henceforth, $\overline{\pi_1}$ and $\overline{\pi_2}$ denote the left- and right-projection from $\overline{R}$ to X.

When $x_1 \; R \; x_2$ for some bisimulation up-to equivalence R, we write $x_1 \underline{\leftrightarrow}_{\mathcal{E}} x_2$.

Note how this is subtly different from Diagram 1: here, we have $F\overline{R}$ instead of FR. When F is the functor $FX = 2 \times X^A$, we can relate bisimulations up-to equivalence to regular bisimulations, using our earlier results specific to F.

Fact 2.1. *Suppose F is the finite automata functor, and let $x : X \to FX$ be an F-coalgebra. For all states $x_1, x_2 \in X$, we have $x_1 \underline{\leftrightarrow} x_2$ if and only if $x_1 \underline{\leftrightarrow}_{\mathcal{E}} x_2$.*

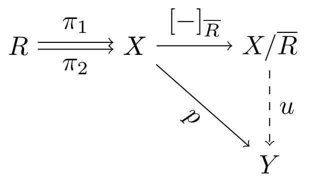
Proof. We can show that $x_1 \underline{\leftrightarrow}_{\mathcal{E}} x_2$ if and only if there exists an R such that $x_1 \; R \; x_2$, which moreover satisfies the conditions of Fact 1.7; the proof is similar to that of Fact 1.1. The claim then follows by the statement of Fact 1.7. □

So, for that functor, we have an equivalence between bisimulation and bisimulation up-to equivalence. However, the proof does not give a lot of immediate hope of a generalization to *all* functors, because the argument there is specific to the particular functor. What's more, the proofs rely on the idea of an equivalence relation and equivalence closure; how do those work in a diagram?

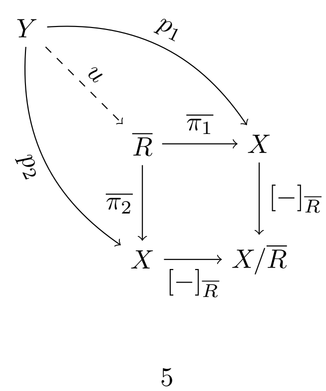
2.1 Detour: equivalence closure in a diagram

To get a better idea of the categorical picture, let's first characterize the equivalence closure of a relation in terms of diagrams. Just to be on the same page, we should fix some notation first. Let X be a relation, and let R be a set. When $x \in X$, we write $[x]_{\overline{R}}$ for the equivalence class of x under $\overline{R}$, i.e., the set $\{x' \in X : x \; \overline{R} \; x'\}$. Furthermore, we write $X/\overline{R}$ for the *quotient* of X by $\overline{R}$, i.e., the set $\{[x]_{\overline{R}} : x \in X\}$. Note that this makes $[-]_{\overline{R}}$ a map from X to $X/\overline{R}$.

Fact 2.2. *Let R be a relation on some set X. Then $[-]_{\overline{R}} : X \to X/\overline{R}$ is the* coequalizer *of $\pi_1, \pi_2 : R \to X$. Specifically, this means that $\pi_1 \circ [-]_{\overline{R}} = \pi_2 \circ [-]_{\overline{R}}$, and for every function $p : X \to Y$ such that $p \circ \pi_1 = p \circ \pi_2$, there exists a (unique) function $u : X/\overline{R} \to Y$ that makes the following diagram commute:*



Fact 2.3. *Let R be a relation on some set X. Then $\langle \overline{R}, \overline{\pi_1}, \overline{\pi_2} \rangle$ is the pullback of $[-]_{\overline{R}} : X \to X/\overline{R}$ along itself. Specifically, this means that for every pair of functions $p_1, p_2 : Y \to X$ such that $[-]_{\overline{R}} \circ p_1 = [-]_{\overline{R}} \circ p_2$, there exists a unique function $u : Y \to \overline{R}$ such that the following diagram commutes:*

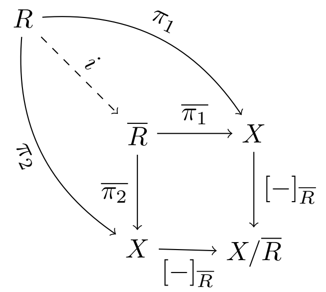


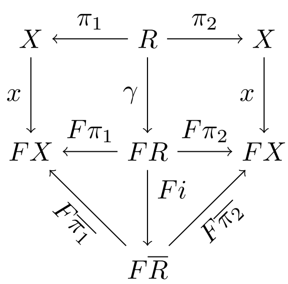
2.2 Completeness of bisimilarity up-to equivalence

The completeness proof for bisimilarity up-to equivalence for the deterministic automata (the forward direction in Fact 1.7) was fairly easy. It turns out we can lift this to the categorical setting without much ado.

Fact 2.4. *Let $x : X \to FX$ be an F-coalgebra. Then bisimilarity is contained in bisimilarity up-to equivalence: for all $x_1, x_2 \in X$, if $x_1 \leftrightarrows x_2$, then $x_1 \leftrightarrows_{\mathcal{E}} x_2$.*

Proof. Let R be some bisimulation witnessing that $x_1 \leftrightarrows x_2$. Classically, R is contained in its equivalence closure by definition. Categorically, this containment is witnessed by the function $i : R \to \overline{R}$ that makes the diagram below on the left commute, which exists uniquely by Fact 2.2.





We then find that the diagram above on the right commutes: the two squares on top commute because R is a bisimulation, and the two triangles below commute because they are the image of the commuting triangles above on the left under F. This makes R a bisimulation up-to equivalence witnessing $x_1 \leftrightarrows_{\mathcal{E}} x_2$. $\square$

2.3 Soundness of bisimilarity up-to equivalence

The other direction, i.e., that bisimilarity up-to equivalence implies bisimilarity, is a bit harder to show. Indeed, it turns out that the equivalence does *not* hold in general, as witnessed by the following example.

Example 2.5. We provide the counterexample hinted at in [RBR13, Example 6]. Let F be the 2–3 functor from the lecture notes, i.e.,

$$FX = \{(x_1, x_2, x_2) \in X^3 : |\{x_1, x_2, x_3\}| \leq 2\}$$

$$Ff(x_1, x_2, x_3) = (f(x_1), f(x_2), f(x_3))$$

We now choose $X = \{0, 1, 2, \tilde{0}, \tilde{1}\}$, and set $x : X \to FX$ as follows:

$$x(0) = (0,0,1) \qquad x(\tilde{0}) = (0,0,0) \qquad x(2) = (2,2,2)$$
$$x(1) = (1,0,0) \qquad x(\tilde{1}) = (1,1,1)$$

Suppose towards a contradiction that $\tilde{0} \leftrightarrows \tilde{1}$. Then there exists a bisimulation $\gamma : R \to FR$ with $\tilde{0} \; R \; \tilde{1}$. Let $\gamma(\langle \tilde{0}, \tilde{1} \rangle) = \langle \langle p_{00}, p_{01} \rangle, \langle p_{10}, p_{11} \rangle, \langle p_{20}, p_{21} \rangle \rangle$. Now,

$$\langle 0,0,0 \rangle = x(\tilde{0}) = F\pi_1(\gamma(\langle \tilde{0}, \tilde{1} \rangle)) = \langle p_{00}, p_{10}, p_{20} \rangle$$

$$\langle 1,1,1 \rangle = x(\tilde{1}) = F\pi_2(\gamma(\langle \tilde{0}, \tilde{1} \rangle)) = \langle p_{01}, p_{11}, p_{21} \rangle$$

Thus, we have $0 \; R \; 1$. But then, R witnesses that $0 \leftrightarroweq 1$, which is false — see Example 3.2 on page 3–1 in the course lecture notes. Hence, $\tilde{0} \not\leftrightarroweq \tilde{1}$.

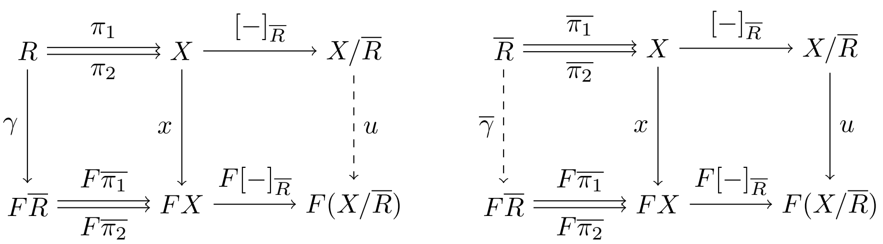
However, $\tilde{0} \leftrightarroweq_{\mathcal{E}} \tilde{1}$ *does* hold. To see this, choose $R = \{\langle \tilde{0}, \tilde{1} \rangle, \langle 0, 2 \rangle, \langle 2, 1 \rangle\}$, and $\gamma : R \to F\overline{R}$, defined as follows:

$$\begin{aligned} \gamma(\langle \tilde{0}, \tilde{1} \rangle) &= \langle \langle 0, 1 \rangle, \langle 0, 1 \rangle, \langle 0, 1 \rangle \rangle \\ \gamma(\langle 0, 2 \rangle) &= \langle \langle 0, 2 \rangle, \langle 0, 2 \rangle, \langle 1, 2 \rangle \rangle \\ \gamma(\langle 2, 1 \rangle) &= \langle \langle 2, 0 \rangle, \langle 2, 1 \rangle, \langle 2, 0 \rangle \rangle \end{aligned}$$

Note in particular that $0 \; \overline{R} \; 1$, so γ is well-defined. This makes R a bisimulation up-to equivalence witnessing that $\tilde{0} \leftrightarroweq_{\mathcal{E}} \tilde{1}$.

Fact 2.6. *If F preserves weak pullbacks, then bisimilarity up-to equivalence implies bisimilarity: for all $x_1, x_2 \in X$, if $x_1 \leftrightarroweq_{\mathcal{E}} x_2$, then $x_1 \leftrightarroweq x_2$.*

Proof. Let R be some bisimulation up-to equivalence witnessing that $x_1 \leftrightarroweq_{\mathcal{E}} x_2$. Because $[-]_{\overline{R}}$ is the coequalizer of π_1 and π_2, and $F[-]_{\overline{R}} \circ x \circ \pi_1 = F[-]_{\overline{R}} \circ x \circ \pi_2$, we obtain $u : X/\overline{R} \to F(X/\overline{R})$ making the diagram below on the left commute.



Furthermore, because F preserves weak pullbacks, and $\overline{R}$ is the pullback of $[-]_{\overline{R}}$ along itself, it follows that $F\overline{R}$ is the weak pullback of $F[-]_{\overline{R}}$ along itself. By the weak pullback property and the fact that $F[-]_{\overline{R}} \circ x = u \circ [-]_{\overline{R}}$, we then obtain $\overline{\gamma} : \overline{R} \to F\overline{R}$ making the diagram above on the right commute. This makes $\overline{R}$ a bisimulation. Since $x_1 \; \overline{R} \; x_2$, it follows that $x_1 \leftrightarroweq x_2$. □

References

[HK71] John E. Hopcroft and Richard M. Karp. A linear algorithm for testing equivalence of finite automata. Technical Report TR71-114, Cornell University, December 1971.

[RBR13] Jurriaan Rot, Marcello M. Bonsangue, and Jan J. M. M. Rutten. Coalgebraic bisimulation-up-to. In *Proc. Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 369–381, 2013.

[Rot15] Jurriaan Rot. *Enhanced Coinduction.* PhD thesis, Leiden University, 2015.