Bisimulation and Coinduction for Dummies

James Cheney

Programming Languages Interest Group

November 10, 2014

Motivation

- When we want to compare two systems, we often want to abstract over their **internal structure** and consider whether they provide the same **behavior**

- (e.g. observational equivalence for simple functional programs)

- The appropriate equivalence is sometimes not easy to define compositionally in terms of subcomponents.

Examples

- Infinite / lazy streams
- Functional programs with I/O behavior
- Concurrent processes (CCS, π-calculus)

Bisimulation and Coinduction

- **Bisimulation** is a way to define when two systems "behave the same", independently of their internal structure

- **Coinduction** is a basic mathematical tool to define bisimulation.

- Formally, coinduction is **dual** to induction, but typical uses of induction have stronger properties than (dualized) typical uses of coinduction

- So in practice, they have a very different "feel"

Review: Induction

- **Theorem:** All horses are of the same color.

- **Proof:**

  - Base case: trivial.

  - Inductive case: Suppose true for n horses. Consider a set of $n+1$ horses. Clearly, by induction, horses 1...n are of the same color. Likewise, by induction, horses 2...$n+1$ are of the same color. Obviously, the two sets overlap, so all $n+1$ horses are of the same color.

But seriously...

- Mathematical induction is a basic tool for computer science
  - particularly structural induction over syntax or rules
- Coinduction is also an important tool, but less well-known
  - (and in some sense less accessible)

Basic observations

- Let $(L, \leq)$ be a complete lattice (e.g. powerset lattice ordered by $\subseteq$)
  - i.e. $\leq$ is a reflexive, transitive and antisymmetric relation on L
  - such that all least upper bounds and greatest lower bounds exist
- A *fixed point* of $F : L \rightarrow L$ is an element x such that $F(X) = X$.
- We say $F : L \rightarrow L$ is *monotone* if $X \leq Y$ implies $F(X) \leq F(Y)$

Knaster-Tarski theorem

- Let $F : L \to L$ be monotone

- There exists a *least fixed-point*

$$lfp(F) = \bigwedge \{x \in L \mid F(x) \leq x\}$$

  aka the *least pre-fixed point*.

- Dually, there exists a *greatest fixed-point*

$$gfp(F) = \bigvee \{x \mid x \leq F(x)\}$$

  aka the *greatest post-fixed point*.

Induction

- When we define an object inductively, the object is the **least fixed-point** of an appropriate operator on an appropriate lattice (often left implicit)

- Example: $F(X) = \{[]\} \cup \{a :: y \mid a \in A, y \in X\}$ "defines" $List\ A$, finite lists of A's

- (Exercise: What is L?)

- The least fixed-point property justifies inductive proofs about such objects

Example

- Assume $[] \in P$ holds and for all a, y, we have $y \in P \Rightarrow a :: y \in P$

- Observe that

$$F(P) = \{[]\} \cup \{a :: y \mid a \in A, y \in P\} \subseteq P \cup P = P$$

  Hence, P is a pre-fixed point of F, so $List\ A \subseteq P$.

- (obviously by definition $P \subseteq List\ A$ so they are equal.)

Aside: Continuity

- Often, F has stronger property such as *continuity*
- so we also know that $lfp(F) = \bigvee_{i=0}^{\omega} F^n(\bot)$
- But this is not needed for fixed point theory generally:
- transfinite induction (over ordinals) can involve non-continuous operators
- Moreover, dual property (co-continuity) is rare for coinductive definitions

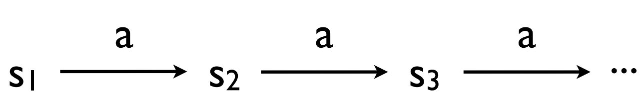
Labeled transition systems

- Consider labeled transition systems (LTSs)

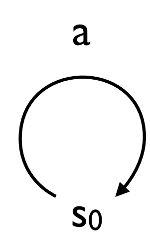
$$(S, A, (\to) \subseteq S \times A \times S)$$

  We write $s \xrightarrow{a} t$ to indicate that from state s there is a transition labeled a to state t.

- Examples:



LTS1



LTS2

Inductive equivalence

- Consider the following rule as an inductive definition of "equivalence" of states

$$\frac{\forall a, s'.s \xrightarrow{a} s' \Rightarrow \exists t'.t \xrightarrow{a} t' \wedge s' \equiv t' \qquad \forall a, t'.t \xrightarrow{a} t' \Rightarrow \exists s'.s \xrightarrow{a} s' \wedge s' \equiv t'}{s \equiv t}$$

- (Exercise: What is the base case?)

- This correctly relates states that have the same **finite** observations

- But what about infinite / cyclic behavior (LTS_1 vs. LTS_2)?

$$s_1 \not\equiv s_0$$

Coinduction

- When we define an object **coinductively**, the object is the **greatest fixed-point** of an appropriate operator on an appropriate lattice (often left implicit)

- Example: $F(X) = \{[]\} \cup \{a :: y \mid a \in A, x \in X\}$ defines the set of **finite or infinite streams of A's**, or $Stream\ A$.

- (Exercise: What is L?)

- The greatest fixed point property justifies **coinductive** reasoning principles for such objects

Example

- Let's prove that 010101... is an infinite stream.

- First attempt: Let $P = \{010101...\}$. Try to show $P \subseteq F(P)$. Not true; after removing initial 0, we have 101010... which is not in P.

- Second attempt: Let $P = \{010101..., 101010...\}$. Then we can show that $P \subseteq F(P)$:

$$\begin{array}{rcl} F(P) & = & \{[]\} \cup \{a :: y \mid a \in \{0,1\}, y \in P\} \\ & = & \{[]\} \cup \{1010101..., 0010101..., 1101010..., 0101010...\} \\ & \supseteq & \{010101..., 101010...\} \end{array}$$

Example

- Consider the following rule as a **coinductive** definition of "equivalence" of states

$$\frac{\forall a, s'. s \xrightarrow{a} s' \Rightarrow \exists t'. t \xrightarrow{a} t' \wedge s' \sim t' \qquad \forall a, t'. t \xrightarrow{a} t' \Rightarrow \exists s'. s \xrightarrow{a} s' \wedge s' \sim t'}{s \sim t}$$

- This correctly relates states that have the same observations and step to "equivalent" states

- This correctly handles cyclic/infinite behavior (e.g. LTS_1 vs. LTS_2)

$$s_1 \sim s_0$$

More formally

- For any LTS $(S, A, \rightarrow)$, we can define a *bisimulation* to be any relation R such that for all $(s, t) \in R$:
  - for all $a \in A, s' \in S$ such that $s \xrightarrow{a} s'$, there exists $t' \in S$ such that $t \xrightarrow{a} t'$ and $(s', t') \in R$
  - and dually: for all $a \in A, t' \in S$ such that $t \xrightarrow{a} t'$, there exists $s' \in S$ such that $s \xrightarrow{a} s'$ and $(s', t') \in R$

- *Bisimilarity* $(\sim)$ is the union of all bisimulations:

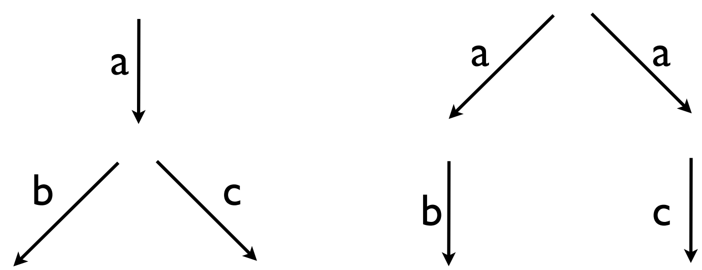
$$(\sim) = \bigcup \{R \mid R \text{ is a bisimulation}\}$$

Trace equivalence

- Another natural-seeming equivalence on LTSs:
- Let $traces(s)$ be the set of all possible (finite or infinite) transition sequences startng at s.
- Example: $traces(s_i) = \{a^\omega\} = traces(s_0)$ in LTS_1, LTS_2
- Define $s =_{tr} t$ to mean $traces(s) = traces(t)$
- Example: $s_0 =_{tr} s_1 = \cdots =_{tr} s_i$

Bisimulation vs. trace equivalence

- Trace equivalence is a bisimulation
- but different from bisimilarity (in the presence of nondeterminsm):



- Top states have the same traces $\{ab, ac\}$ but are not bisimilar

Bisimilarity and fixed points

- There is an associated monotone closure operator on $P(S \times S)$:

$$\begin{array}{rcl} F(X) & = & \{(s,t) \mid \forall s', a. s \xrightarrow{a} s' \Rightarrow \exists t'. t \xrightarrow{a} t' \wedge (s',t') \in X\} \\ & \cup & \{(s,t) \mid \forall t', a. t \xrightarrow{a} t' \Rightarrow \exists s'. s \xrightarrow{a} s' \wedge (s',t') \in X\} \end{array}$$

- and $\sim$ is its greatest fixed point.

- Key point: **bisimilarity is a bisimulation**.

- Hence, the greatest fixed point property justifies *proof by coinduction* for bisimilarity.

Proof by coinduction

- Suppose we want to show $s_0 \sim t_0$.

- Since bisimilarity is the union of all bisimulations, suffices to:

  1. define a **single** relation R such that $(s_0, t_0) \in R$
  2. prove $(s, t) \in R$ and $s \xrightarrow{a} s'$ implies $\exists t'.t \xrightarrow{a} t' \wedge (s', t') \in R$
  3. and dually $(s, t) \in R$ and $t \xrightarrow{a} t'$ implies $\exists s'.s \xrightarrow{a} s' \wedge (s', t') \in R$

- Since R is a bisimulation, we conclude $(s_0, t_0) \in R \subseteq (\sim)$, i.e. $s_0 \sim t_0$

Example, continued

- Proof by coinduction that $s_1 \sim s_0$:

- Let $R = \{(s_i, s) \mid i \in \mathbb{N}\}$

- Show that whenever $(s, t) \in R$, we have:
  - $\forall a, s'. s \xrightarrow{a} s' \Rightarrow \exists t'. t \xrightarrow{a} t' \wedge (s', t') \in R$
  - and dually $\forall a, t'. t \xrightarrow{a} t' \Rightarrow \exists s'. s \xrightarrow{a} s' \wedge (s', t') \in R$

- Often (but not always) one part is “obvious by construction” and the other nontrivial

Example, continued

- Suppose $(s, t) \in R$ and let a, s' be given with $s \xrightarrow{a} s'$.

- Then clearly $s = s_i$ and $s' = s_{i+1}$ for some i.

- Likewise, clearly $t = s_0$, and observe that $s_0 \xrightarrow{a} s_0$.

- Observe that $(s_{i+1}, s_0) \in R$. QED for the first part.

Example, continued

- Suppose $(s, t) \in R$ and let a, t' be given with $t \stackrel{a}{\to} t'$.

- Then clearly $t = s_0 = t'$.

- Likewise, clearly $s = s_i$ for some i, and recall that $s_i \stackrel{a}{\to} s_{i+1}$ for each i.

- Observe that $(s_{i+1}, s_0) \in R$. QED for the second part.

Similarities and differences

- Induction and coinduction: both involve "local" checks

- Induction involves showing that property/set is closed under rules "forward", hence it contains inductively defined set

- Coinduction involves guessing a property/set and showing that it is closed under rules "backwards", hence it is contained in coinductively defined set

- Induction (continuous): Each state has a finite "rank"

- Coinduction: There is usually no inherent notion of "rank"

A little history

He [i.e., David Park] came down during breakfast one morning carrying my CCS book and said [“]there’s something wrong!”. So I prepared to defend myself. He pointed out the non coinductive way that I had set up observation equivalence, as the limit of a decreasing ω-chain of relations, which didn’t quite reach the maximal fixed point.

After about 10 minutes I reali[z]ed he was right, and through that day I got excited about the coinductive proof technique.

That was what David meant by [“]something’s wrong”. Not only had I missed the (fixed!) point—which I had reali[z]ed—but also my proof technique (involving induction on the iteration of the functions) for establishing instances of the equivalences was clumsy. I immediately saw that he had liberated me from a misconception, and that the whole theory was going to look very much better by using maximal fixed points and (what I now recogni[z]e as) coinduction. [...]

That same day we went for a walk in the hills around Edinburgh, and the express purpose was to agree what the pre-fixed points and the maximal fixed point should be called. We thought of a lot of words; David at one point liked [“]mimicry”, which I vetoed. I think [“]bisimulation” was my suggestion; in any case, we both liked it, partly because we could use that word for the pre-fixed points and [“]bisimilarity” for the maximal fixed point itself. I think David demurred because there are five syllables; but we then thought that they were a lot easier to pronounce than the three syllables of [“]mimicry”!

— Robin Milner (in Sangiorgi [2009])

But that's not all!

- There are many different variations on this theme:
  - e.g. "weak bisimulation" (allows ignoring "silent" transitions)
  - early/late bisimulation in π-calculus
  - barbed equivalences, testing equivalences
  - many more!

- Beyond scope of this talk

Bisimulation and coinduction in other contexts

- Modal logic/games: existence of bisimulation = existence of winning strategy

- Databases: graph bisimulation can be a useful substitute for subgraph isomorphism (and easier to check)

- Bisimulation also appears in e.g. equivalence of symmetric/edit lenses

- Algebras/coalgebras further generalize inductive/coinductive ideas (as I understand it)

Conclusion

- Goal of the talk: just give a taste of the main ideas of bisimulation and coinduction

- Fully exploring these, e.g. in context of π-calculus or CCS, could be a whole course of its own

- Hopefully, however, this gave you some pointers to where to look if bisimulation/coinduction appear relevant to your work

Sources/further reading

- Davide Sangiorgi. 2009. On the origins of bisimulation and coinduction. ACM Trans. Program. Lang. Syst. 31, 4, Article 15 (May 2009), 41 pages.

- Introduction to Bisimulation and Coinduction, Davide Sangiorgi, Cambridge University Press, 2012