

Relating Apartness and Bisimulation

Herman Geuvers

Radboud Univ. Nijmegen and TU Eindhoven

December 17, 2020

jww Bart Jacobs

Overview

- ▶ Bisimulation and apartness
- ▶ LTSs and branching bisimulation
- ▶ Branching apartness
- ▶ Proving properties about branching apartness and using it

Deterministic Finite Automata

A DFA $M = (A, K, \delta, \downarrow)$ consists of an alphabet A , a set of states K and $\delta : K \times A \rightarrow K$, $\downarrow : K \rightarrow 2$. A DFA M gives rise to the notions of **bisimulation for M** and **apartness for M** .

- $R \subseteq K \times K$ is a **M -bisimulation** if it satisfies the following rule.

$$R(q_1, q_2)$$

$$q_1 \downarrow \Leftrightarrow q_2 \downarrow \quad \wedge \quad \forall a \in A \forall p_1, p_2 (q_1 \rightarrow_a p_1 \wedge q_2 \rightarrow_a p_2 \implies R(p_1, p_2))$$

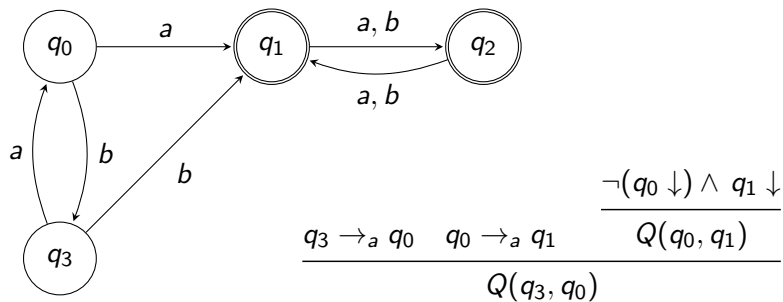
Two states $q_1, q_2 \in K$ are **M -bisimilar**, $q_1 \xleftrightarrow{M} q_2$, is defined by $q_1 \xleftrightarrow{M} q_2 := \exists R \subseteq K \times K$ (R is a M -bisimulation and $R(q_1, q_2)$).

- $Q \subseteq K \times K$ is a **M -apartness** if it satisfies the following rules.

$$\frac{q_1 \rightarrow_a p_1 \quad q_2 \rightarrow_a p_2 \quad Q(p_1, p_2)}{Q(q_1, q_2)} \qquad \frac{q_1 \downarrow \not\equiv q_2 \downarrow}{Q(q_1, q_2)}$$

Two states $q_1, q_2 \in K$ are **M -apart**, $q_1 \not\#^M q_2$, if $q_1 \not\#^M q_2 := \forall Q \subseteq K \times K$ (if Q is a M -apartness then $Q(q_1, q_2)$).

Example



A bisimulation is given by $q_1 \sim q_2$. It can be shown that $q_0 \not\sim^M q_3$ because for every apartness Q we have the derivation given on the right.

To be M -apart is the smallest relation satisfying specific closure properties, so it is an **inductive property**.

The closure properties yield a **derivation system** for proving that two elements are M -apart.

For the DFA case, these rules are:

$$\frac{q_1 \rightarrow_a p_1 \quad q_2 \rightarrow_a p_2 \quad p_1 \underline{\#}^M p_2}{q_1 \underline{\#}^M q_2}$$

$$\frac{q_1 \downarrow \not\equiv q_2 \downarrow}{q_1 \underline{\#}^M q_2}$$

Apartness in constructive analysis

In constructive real analysis (and similarly when talking about computable real numbers), one takes apartness as a primitive and defines equality as its negation:

$$x \underline{\#} y \simeq \text{we can find a proper distance } \delta \in \mathbb{Q} \text{ between } x \text{ and } y$$
$$x = y := \neg(x \underline{\#} y)$$

A relation is usually only called an apartness relation if it satisfies three properties.

DEFINITION. A relation $\underline{\#}$ is a **proper apartness** relation if it is

- ▶ **irreflexive**: $\forall x (\neg x \underline{\#} x)$,
- ▶ **symmetric**: $\forall x, y (x \underline{\#} y \implies y \underline{\#} x)$,
- ▶ **co-transitive**: $\forall x, y, z (x \underline{\#} y \implies x \underline{\#} z \vee z \underline{\#} y)$.

LEMMA. For R a relation, R is an equivalence relation if and only if $\neg R$ is a proper apartness relation.

PROOF. The only interesting property to check is that R is transitive iff $\neg R$ is co-transitive.

The general categorical picture

Bisimulation and apartness can be defined by induction over the structure of the polynomial functor $F : \mathbf{Set} \rightarrow \mathbf{Set}$ that we consider the coalgebra for.

- ▶ For DAs: $c : K \rightarrow F(K)$ with $F(X) = X^A \times 2$.
- ▶ For streams over A : $c : K \rightarrow F(K)$ with $F(X) = A \times X$.

We have the following result relating bisimulation and apartness.

LEMMA.

1. R is a bisimulation if and only if $\neg R$ is an apartness.
2. The relation $\underline{\leftrightarrow}$, the union of all bisimulations:
$$\underline{\leftrightarrow} = \bigcup \{R \mid R \text{ is a bisimulation}\},$$
 is itself a bisimulation.
3. The relation $\underline{\#}$, the intersection of all apartness relations:
$$\underline{\#} = \bigcap \{Q \mid Q \text{ is an apartness relation}\},$$
 is itself also an apartness relation.
4. $\underline{\leftrightarrow} = \neg \underline{\#}$.

LTSs and branching bisimulation

A **labelled transition systems**, LTS, is a tuple (X, A_τ, \rightarrow) , where

- ▶ X is a set of states,
- ▶ $A_\tau = A \cup \{\tau\}$ is a set of actions, with τ the special **silent action**,
- ▶ $\rightarrow \subseteq X \times A_\tau \times X$ is the **transition relation**.

We write $q_1 \rightarrow_u q_2$ for $(q_1, u, q_2) \in \rightarrow$. Furthermore, \rightarrow_τ denotes the reflexive transitive closure of \rightarrow_τ .

NB. We reserve $q_1 \rightarrow_a q_2$ to denote a transition with an a -step with $a \in A$ (so $a \neq \tau$).

The notion of bisimulation for LTSs we consider is **branching bisimulation**. Here, the categorical picture is not completely clear, so there is no “canonical” way for constructing the bisimulation and apartness from the functor and the co-algebra.

Branching bisimulation

We give the definition of branching bisimulation in **derivation rule** style: $R \subseteq X \times X$ is a **branching bisimulation relation** if the following derivation rules hold for R .

$$\frac{q \rightarrow_a q' \quad R(q, p)}{\exists p', p'' (p \twoheadrightarrow_\tau p' \rightarrow_a p'' \wedge R(q, p') \wedge R(q', p''))} \text{bis}_b$$

$$\frac{q \rightarrow_\tau q' \quad R(q, p)}{R(q', p) \vee \exists p', p'' (p \twoheadrightarrow_\tau p' \rightarrow_\tau p'' \wedge R(q, p') \wedge R(q', p''))} \text{bis}_{b\tau}$$

$$\frac{R(p, q)}{R(q, p)} \text{symm}$$

States q, p are **branching bisimilar**, $q \leftrightarrow_b p$ if there exists a branching bisimulation relation R such that $R(q, p)$.

Branching apartness

We define **branching apartness** by transporting the rules for branching bisimulation to derivation rules for $Q \subseteq X \times X$ where $\neg Q$ is a branching bisimulation.

DEFINITION. $Q \subseteq X \times X$ is a **branching apartness** in case the following derivation rules hold for Q .

$$\frac{q \rightarrow_a q' \quad \forall p', p'' (p \rightarrow_\tau p' \rightarrow_a p'' \implies Q(q, p') \vee Q(q', p''))}{Q(q, p)} \text{in}_b$$

$$\frac{q \rightarrow_\tau q' \quad Q(q', p) \quad \forall p', p'' (p \rightarrow_\tau p' \rightarrow_\tau p'' \implies Q(q, p') \vee Q(q', p''))}{Q(q, p)} \text{in}_{b\tau}$$

$$\frac{Q(p, q)}{Q(q, p)} \text{symm}$$

States q and p are **branching apart**, $q \not\#_b p$, if for all branching apartness relations Q , we have $Q(q, p)$.

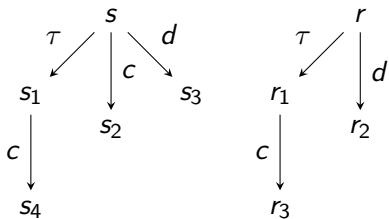
Branching bisimulation and branching apartness

- ▶ By definition: Q is a branching apartness iff $\neg Q$ is a branching bisimulation, so
- ▶ $q \#_b p$ if and only if $\neg(q \leftrightarrow_b p)$.
- ▶ As $q \#_b p$ is an inductive notion, we have that $q \#_b p$ if and only if it is derivable using the (symm) rule and the following two:

$$\frac{q \rightarrow_a q' \quad \forall p', p'' (p \twoheadrightarrow_\tau p' \rightarrow_a p'' \implies q \#_b p' \vee q' \#_b p'')}{q \#_b p} \text{in}_b$$

$$\frac{q \rightarrow_\tau q' \quad q' \#_b p \quad \forall p', p'' (p \twoheadrightarrow_\tau p' \rightarrow_\tau p'' \implies q \#_b p' \vee q' \#_b p'')}{q \#_b p} \text{in}_{b\tau}$$

Example



We give a derivation of $s \#_b r$:

$$\frac{
 \frac{
 \frac{
 s \rightarrow_d s_3 \quad \checkmark
 }{
 s \#_b r_1
 }
 }{
 s \#_b r_1 \vee s_2 \#_b r_3
 }
 \quad
 \frac{
 s \rightarrow_c s_2 \quad [r \rightarrow_\tau r_1 \rightarrow_c r_3]
 }{
 }
 }{
 s \#_b r
 }$$

NB: Remember the derivation rule:

$$\frac{
 q \rightarrow_a q' \quad \forall p', p'' (p \rightarrow_\tau p' \rightarrow_a p'' \implies q \#_b p' \vee q' \#_b p'')
 }{
 q \#_b p
 }_{in_b}$$

Proving that \leftrightarrow_b is an equivalence relation

This is remarkably tricky, because if R_1, R_2 are branching bisimulation relations, then $R_1 \circ R_2$ need not be a branching bisimulation relation. (So the “obvious” proof of transitivity fails.)

Twan Basten used the notion of **semi-branching bisimulation relation** and proved that (1) “being semi-branching bisimilar”, \leftrightarrow_{sb} , is an equivalence relation and (2) \leftrightarrow_{sb} coincides with \leftrightarrow_b .

We similarly introduce **semi-branching apartness relation**, $\#_{sb}$, by replacing rule ($\text{in}_{b\tau}$) by

$$\frac{q \rightarrow_{\tau} q' \quad q' \#_{sb} p \quad \forall p', p'' (p \twoheadrightarrow_{\tau} p' \rightarrow_{\tau} p'' \implies q' \#_{sb} p'' \vee (q \#_{sb} p' \wedge q \#_{sb} p''))}{q \#_{sb} p} \text{in}_{sb\tau}$$

So, $q \#_{sb} p$ in case this is derivable by these adapted set of rules.

Proving the co-transitivity of branching apartness

The proof of co-transitivity of $\underline{\#}_b$ (and thus that \leftrightarrow_b is an equivalence relation) proceeds in the following steps.

1. We prove $q \underline{\#}_{sb} p \implies q \underline{\#}_b p$ (by induction on $q \underline{\#}_{sb} p$).
2. We prove a number of basic lemmas for $\underline{\#}_{sb}$.
(Typically useful results we would also like to have for $\underline{\#}_b$, but we can't obtain directly for $\underline{\#}_b$.)
3. We prove the **apartness stuttering property** for $\underline{\#}_{sb}$.
4. We prove that $q \underline{\#}_b p \implies q \underline{\#}_{sb} p$ (by induction on $q \underline{\#}_{sb} p$, using the apartness stuttering property) and we conclude that $\underline{\#}_b = \underline{\#}_{sb}$.
5. We prove co-transitivity for $\underline{\#}_b$ using the lemmas under (2).

For one of the basic lemmas under (2) we move over to the “bisimulation view”, as the result seems easier to obtain there.

Stuttering and apartness stuttering

The **stuttering property** states that the following holds (for \leftrightarrow_b)

$$\frac{r \rightarrow_{\tau} r_1 \rightarrow_{\tau} \dots \rightarrow_{\tau} r_n \rightarrow t \quad (n \geq 0) \quad r \leftrightarrow_b p \quad t \leftrightarrow_b p}{\forall i (1 \leq i \leq n) r_i \leftrightarrow_b p}$$

If we cast this as a property about apartness we obtain the following **apartness stuttering property**

$$\frac{r \twoheadrightarrow_{\tau} q \twoheadrightarrow_{\tau} t \quad q \underline{\#} p}{r \underline{\#} p \vee p \underline{\#} t} \text{stut}$$

LEMMA. The apartness stuttering property holds for $\underline{\#}_{sb}$

PROOF. By induction on $q \underline{\#}_{sb} p$ (using various auxiliary properties).

Variations on the rules

We can show that other rules are sound for proving apartness, for example (thanks to David Jansen):

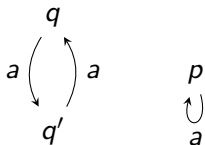
$$\frac{q \rightarrow_a q' \quad \forall p', p'' (p \rightarrow_\tau p' \rightarrow_a p'' \implies p \#_b p' \vee q' \#_b p'')}{q \#_b p} \text{in}_b^A$$

Or, combining bisimulation and apartness, the following rule is sound:

$$\frac{q \rightarrow_a q' \quad \forall p', p'' (p \rightarrow_\tau p' \rightarrow_a p'' \wedge q' \leftrightarrow_b p'' \implies q \#_b p')}{q \#_b p} \text{in}_b^{\leftrightarrow}$$

Using $\underline{\#}_b$ to prove $q \Leftrightarrow_b p$

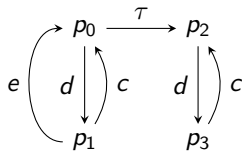
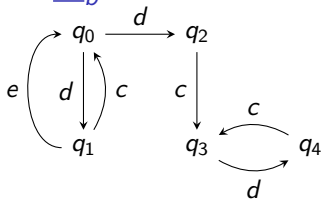
EXAMPLE.



We search for the **shortest derivation** of $q \underline{\#}_b p$ and notice it doesn't exist, and therefore we can conclude that $\neg q \underline{\#}_b p$ and so $q \Leftrightarrow_b p$. In our search for a deduction we keep track of goals that we have already encountered.

$$\frac{
 \frac{
 \frac{
 q' \rightarrow_a q \quad \frac{\text{fail}}{q' \underline{\#}_b p \vee q \underline{\#}_b p}
 }{q' \underline{\#}_b p}
 }{q \rightarrow_a q' \quad q \underline{\#}_b p \vee q' \underline{\#}_b p}
 }{q \underline{\#}_b p}$$

From $q \not\equiv_b p$ to a distinguishing formula (example)



$$q_1 \rightarrow_e q_0$$

$$q_1 \not\equiv_b p_3$$

$$q_0 \not\equiv_b p_2 \vee q_1 \not\equiv_b p_3$$

$$p_1 \rightarrow_e p_0$$

$$p_1 \not\equiv_b q_2$$

$$q_2 \not\equiv_b p_1$$

$$q_0 \not\equiv_b p_0 \vee q_2 \not\equiv_b p_1$$

$$q_0 \rightarrow_d q_1 \quad \forall p', p'' (p_2 \rightarrow_\tau p' \rightarrow_d p'' \implies q_0 \not\equiv_b p' \vee q_1 \not\equiv_b p'')$$

$$q_0 \not\equiv_b p_2$$

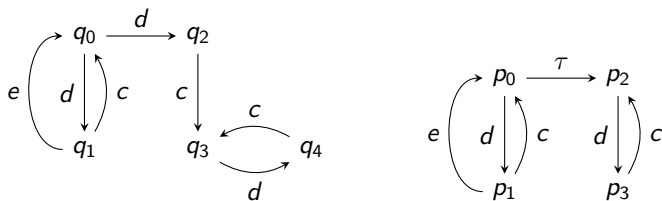
$$q_0 \not\equiv_b p_2 \vee q_2 \not\equiv_b p_3$$

$$q_0 \rightarrow_d q_2$$

$$\forall p', p'' (p_0 \rightarrow_\tau p' \rightarrow_d p'' \implies q_0 \not\equiv_b p' \vee q_2 \not\equiv_b p'')$$

$$q_0 \not\equiv_b p_0$$

From $q \not\equiv_b p$ to a distinguishing formula (example)



- ▶ Henri Korver has an algorithm that generates an HMLU (Hennessy-Milner Logic with Until) formula Φ that distinguishes two states s and t in case $\neg(s \leftrightarrow_b t)$.
- ▶ We can extract such a formula from a derivation of $s \not\equiv_b t$.

For the example, the formula derived from the derivation of $q_0 \not\equiv_b p_0$ is

$$\Phi := (tt \langle d \rangle (tt \langle e \rangle tt)) \langle d \rangle \neg(tt \langle e \rangle tt)$$

We have $q_0 \models \Phi$ and $p_0 \not\models \Phi$.

Questions?