RELATING APARTNESS AND BISIMULATION

HERMAN GEUVERS [a,b] AND BART JACOBS [a]

[a] ICIS, Radboud University Nijmegen

[b] Faculty of Mathematics and Computer Science, Technical University Eindhoven

e-mail address: herman@cs.ru.nl
e-mail address: bart@cs.ru.nl

ABSTRACT. A bisimulation for a coalgebra of a functor on the category of sets can be described via a coalgebra in the category of relations, of a lifted functor. A final coalgebra then gives rise to the coinduction principle, which states that two bisimilar elements are equal. For polynomial functors, this leads to well-known descriptions. In the present paper we look at the dual notion of "apartness". Intuitively, two elements are apart if there is a positive way to distinguish them. Phrased differently: two elements are apart if and only if they are not bisimilar. Since apartness is an inductive notion, described by a least fixed point, we can give a proof system, to derive that two elements are apart. This proof system has derivation rules and two elements are apart if and only if there is a finite derivation (using the rules) of this fact.

We study apartness versus bisimulation in two separate ways. First, for weak forms of bisimulation on labelled transition systems, where silent (τ) steps are included, we define an apartness notion that corresponds to weak bisimulation and another apartness that corresponds to branching bisimulation. The rules for apartness can be used to show that two states of a labelled transition system are not branching bismilar. To support the apartness view on labelled transition systems, we cast a number of well-known properties of branching bisimulation in terms of branching apartness and prove them. Next, we also study the more general categorical situation and show that indeed, apartness is the dual of bisimilarity in a precise categorical sense: apartness is an initial algebra and gives rise to an induction principle. In this analogy, we include the powerset functor, which gives a semantics to non-deterministic choice in process-theory.

1. INTRODUCTION

Bisimulation is a standard way of looking at indistinguishability of processes, labelled transitions, automata and streams, etc. These structures all have in common that they can be seen as coalgebraic: the elements are not built inductively, using constructors, but they are observed through "destructors" or "transition maps". The coinduction principle states that two elements that have the same observations are equal, when mapped to a "final" model. A bisimulation is a relation that is preserved along transitions: if two elements are bisimilar, and we perform a transition, then we either get two new bisimilar elements, or we get equal outputs (in case our observation is a basic value). Two elements are bisimilar if

and only if they are observationally indistinguishable, that is, if there is a bisimulation that relates them.

Coalgebraic structures have a natural notion of bisimulation, because the transfer principle can be defined directly from the type of the destructor, that is, from the functor involved. So bisimilarity, being the largest bisimulation is also defined directly from the destructor (transition operation), and it is well known that if one starts from a final coalgebra, then bisimilarity on the final coalgebra coincides with equality. This gives the coinduction principle: bisimilarity implies equality, see e.g. [JR11, Rut00, Jac16].

There is a dual way of looking at this, which has not been explored much[1]. Of course, the concept of observations is well-known, and there is work by Korver [Kor92], who presents an algorithm that, if two states are not branching bisimilar, produces a formula in Hennessy-Milner [HM85] logic with until operator that distinguishes the two states. Another work is Chow [Cho78] on testing equivalence of states in finite state machines and more recent work is by Smetsers et al. [SMJ16], where an efficient algorithm is presented for finding a minimal separating sequence for a pair of in-equivalent states in a finite state machine.

We take this further by developing the basic parts of a theory of "apartness". The idea is that two elements are apart if we can make an observation in finitely many steps that distinguishes these elements. This idea goes back to Brouwer, in his approach to real numbers, but here we introduce the notion of an "apartness relation" for a coalgebra, again directly from the definition of the type of the destructor, i.e. from the functor. Basically, a relation is an apartness relation if it satisfies the inverse of the transfer principle for bisimulations. We define two elements to be apart if they are in all apartness relations. It can be shown that a relation Q is an apartness relation if and only if its complement $\neg Q$ is a bisimulation relation. Thereby, two elements are apart if and only if they are not bisimilar, that is, distinguishable. Aside from providing a new view on bisimulation, apartness—being an inductive notion—also provides a proof system: two elements are apart if and only if there is a (finite, well-founded) derivation of that fact using the derivation rules. These derivation rules are the rules that define what an apartness is for that particular coalgebra, so they are directly derived from the type of the destructor. This again emphasizes that bisimilarity of two elements has to be proven coinductively, while "being apart" for two elements can be proven by giving a finite derivation (inductively).

This paper consists of two separate parts, one more concrete and one more abstract. The first, concrete part focuses on bisimulation and apartness for labelled transition systems (LTS). We apply these notions to the case of weak forms of bisimulation for labelled transition systems with "silent steps", usually referred to as τ-steps. Silent steps cannot be directly observed, but sometimes they do have some implicit side-effects as they may move the system from a state where a certain action is enabled to a state where this action is impossible. Therefore, several variations have been defined, like weak bisimulation and branching bisimulation. We study these from the point of view of apartness, and we define what it means to be a "weak apartness" relation and a "branching apartness" relation. Two states in a system are "weakly apart" if they are in the intersection of all weak apartness relations and are "branching apart" if they are in the intersection of all branching apartness relations. The main outcome of this first part is a derivation system for branching apartness.

---

[1]One of the authors (BJ) did write an article about bisimulation and apartness in 1995, entitled *Bisimulation and Apartness in Coalgebraic Specification*; it is available online at citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.4507 but was never published. Parts of that article are incorporated in the present text, esp. in Section 5.

This is a derivation system in the traditional (inductive) sense: a judgment holds if there is a finite derivation (so no infinite or circular derivations) that has that judgment as its conclusion. To show that the apartness view on LTSs is fruitful, we use the derivation system for branching apartness to show that the branching apartness relation is *co-transitive* and satisfies the *apartness stuttering property*. (These notions will be dealt with in Section 3.1.) These imply the stuttering property and the transitivity for branching bisimulation, properties that are known to be subtle to prove. (See [Bas96].) We also indicate how the derivation system can be used as an algorithm for proving branching apartness of two states in an LTS and we define and discuss the notion of *rooted branching apartness* which is the dual of rooted branching bisimulation.

The second part switches to a more abstract categorical level. It is restricted however to functors on the category of sets. First, the standard coalgebraic approach is recalled, in which a bisimulation is a coalgebra itself, for a lifting of the functor involved to the category of relations. This can be applied in particular to polynomial functors and yields familiar descriptions of bisimulation.

Next, apartness is described in an analogous manner. It does not use the category $\mathbf{Rel}$ of relations, nor its usual opposite $\mathbf{Rel}^{\mathrm{op}}$, but a special "fibred" opposite $\mathbf{Rel}^{\mathrm{fop}}$. A special lifting of a functor to $\mathbf{Rel}^{\mathrm{fop}}$ is described, via negation as a functor $\neg\colon \mathbf{Rel} \to \mathbf{Rel}^{\mathrm{fop}}$. An apartness relation is then defined as a coalgebra of the lifted functor (to $\mathbf{Rel}^{\mathrm{fop}}$). This set-up then guarantees that a relation R is a bisimulation iff $\neg R$ is an apartness relation. Moreover, there is an analogue of the coinduction principle, stating that two states of a coalgebraic system are apart iff they are non-equal when mapped to the final coalgebra.

A significant conclusion from this analysis is: bisimilarity is the *greatest* fixed point in a partial order of relations. But apartness is the *least* fixed point in that order. This means that apartness can be established in a finite number of steps. Hence it can be described via a system of proof rules. This, in the end, is the main reason why apartness can be more amenable than bisimulation.

We should emphasize that the two parts of this paper are really "apart" since there is no overlap. There is quite a bit of work on dealing with weak/branching bisimulation in a coalgebraic setting (see e.g. [SdVW09, BK17, Bre15, BMP15, GP14]), but there is no generic, broadly applicable approach. In this paper we are not solving this longstanding open problem. We have separate descriptions of weak/branching apartness (in the first part) and of categorical apartness (in the second part). The only hope that we can offer at this stage is that apartness might provide a fresh perspective on a common approach.

To clarify some terminology and relate the corresponding notions of the bisimulation view and the apartness view, we give the following table.

bisimulation relation	apartness relation
coinductive	inductive
bisimulation equivalence	proper apartness
congruence	strong extensionality

A bisimulation relation models an equality of processes or process terms, whereas an apartness relation models an inequality, so R will be a bisimulation (of some type) if and only if $\neg R$ is an apartness (of that same type). Bisimilarity is the largest bisimulation relation, which means that it is a coinductively defined concept. Apartness is the smallest apartness relation, which means that it is an inductively defined concept. A bisimulation should be (at

least) an equivalence relation, meaning that it satisfies reflexivity, symmetry and transitivity. The dual notions are irreflexivity, symmetry and co-transitivity, which together are usually called "apartness" in the literature. To avoid confusion, we have introduced the terminology "proper apartness" for a relation that satisfies irreflexivity, symmetry and co-transitivity. In process theory, bisimulation is not an equivalence relation by definition, so neither is an apartness a "proper apartness" by definition. There is really some work to do, so therefore it is important to single out these notions. A relation R is a congruence in case it is preserved by application of operators: if $R(x, y)$, then $R((f(x), f(y))$ for any operator f. The dual notion is strong extensionality, but in the "apartness view", this is not a property of the relation but of the operator. Operator f is strongly extensional (for apartness relation Q) if $Q(f(x), f(y))$ implies $Q(x, y)$. (Intuitively: if $f(x)$ and $f(y)$ are different, then x and y should be different.)

1.1. **Contents of the sections.** In Section 2, we introduce bisimulation and apartness for streams and for deterministic automata, as preparation for more general/complicated cases. In Section 3, we discuss weak and branching bisimulation and apartness and we indicate the potential use of reasoning with apartness instead of bisimulation. In Section 4 we recap the coalgebraic treatment of bisimulation for coalgebras in the category **Set** as a coalgebra in the category **Rel**. In Section 5 we introduce the dual case and give a coalgebraic treatment of apartness, as the opposite of bisimulation. For completeness, we give, in the Appendix, a syntactic treatment of the general picture of Section 2, where we have a general type of coalgebras for which we define bisimulation and apartness.

Special Thanks. We dedicate this article to Jos Baeten on the occasion of his retirement. Much of Jos' research has centered around process theory and process algebra, where various forms of bisimulation equivalence have always played a central role. As a math student, before going to the USA to do a PhD on a topic in the intersection of recursion theory and set theory, Jos was part of the Dutch "school" on constructive mathematics, and we think that it is nice to see that 'apartness', a notion which originates from constructive mathematics, also has a natural place in the study of process (non-)equivalence. The first author in particular would like to thank Jos for the years he has worked at the Technical University Eindhoven in the Formal Methods group, led by Jos, the many things he has learned during this period and the pleasant cooperation on topics of science, education and organisation. Thanks Jos!

2. Bisimulation and apartness for streams and deterministic automata

We start from the coalgebra of streams over an alphabet A and the coalgebra of DAs (Deterministic Automata) over A, for which we illustrate the notions of bisimulation and apartness. We work in the category **Set** of sets and functions. The coalgebra of streams over A is given by a function $c = \langle h, t \rangle \colon K \to A \times K$, where we associate every $s \in K$ with a stream by letting $h(s) \in A$ denote the head of s and $t(s) \in K$ the tail of s.

Definition 2.1. Let A be a fixed set/alphabet. A coalgebraic map $\langle h, t \rangle \colon K \to A \times K$ gives rise to the following notions of *bisimulation for c* and *apartness for c*.

(1) A relation $R \subseteq K \times K$ is a *c-bisimulation* if it satisfies the following rule

$$\frac{R(s_1, s_2)}{h(s_1) = h(s_2) \wedge R(t(s_1), t(s_2))}$$

Two states $s_1, s_2 \in K$ are *c-bisimilar*, notation $s_1 \leftrightarrow^c s_2$, is defined by

$$s_1 \leftrightarrow^c s_2 := \exists R \subseteq K \times K \, (R \text{ is a } c\text{-bisimulation and } R(s_1, s_2)).$$

(2) A relation $Q \subseteq K \times K$ is a *c-apartness* if it satisfies the following rules

$$\frac{Q(t(s_1), t(s_2))}{Q(s_1, s_2)} \qquad \frac{h(s_1) \neq h(s_2)}{Q(s_1, s_2)}$$

Two states $s_1, s_2 \in K$ are *c-apart*, notation $s_1 \mathrel{\#}^c s_2$, is defined by

$$s_1 \mathrel{\#}^c s_2 := \forall Q \subseteq K \times K \, (\text{if } Q \text{ is a } c\text{-apartness, then } Q(s_1, s_2)).$$

Before we prove some generalities about bisimulation and apartness, we now first treat the example of deterministic automata, DAs. A DA over A is given by a set of states, K, a transition function $\delta : K \times A \to K$ and a function $f : K \to \{0, 1\}$ denoting whether $q \in K$ is a final state or not. We write 2 for $\{0, 1\}$ and we view, as usual in coalgebra, a DA as a coalgebra $c : K \to K^A \times 2$, consisting of two maps $c = \langle \delta, f \rangle$ with $\delta : K \to K^A$ and $f : K \to 2$. We use the standard notation for automata and write $q \to_a q'$ if $\delta(q)(a) = q'$ and $q \downarrow$ if $f(q) = 0$.

We now introduce the notions of bisimulation and apartness for DAs. The first is well-known, the second less so. These notions can be defined in a canonical way for a large set of functors on **Set**. This we will describe categorically in Section 5. In the Appendix, we will give an outline in logical-syntactic terms.

Definition 2.2. Let A be an alphabet and let K be a set of states. A coalgebraic map $\langle c_1, c_2 \rangle : K \to K^A \times 2$ gives rise to the following notions of *bisimulation for c* and *apartness for c*.

(1) A relation $R \subseteq K \times K$ is a *c-bisimulation* if it satisfies the following rule.

$$\frac{R(q_1, q_2)}{\forall a \in A \, \forall p_1, p_2(q_1 \to_a p_1 \wedge q_2 \to_a p_2 \implies R(p_1, p_2)) \quad \wedge \quad q_1 \downarrow \Leftrightarrow q_2 \downarrow}$$

That two states $q_1, q_2 \in K$ are *c-bisimilar*, notation $q_1 \leftrightarrow^c q_2$, is defined by

$$q_1 \leftrightarrow^c q_2 \quad := \quad \exists R \subseteq K \times K \, (R \text{ is a } c\text{-bisimulation and } R(q_1, q_2)).$$

(2) A relation $Q \subseteq K \times K$ is a *c-apartness* if it satisfies the following rules.

$$\frac{q_1 \to_a p_1 \qquad q_2 \to_a p_2 \qquad Q(p_1, p_2)}{Q(q_1, q_2)} \qquad \frac{q_1 \downarrow \wedge \neg(q_2 \downarrow)}{Q(q_1, q_2)} \qquad \frac{\neg(q_1 \downarrow) \wedge q_2 \downarrow}{Q(q_1, q_2)}$$

As usual, rules are "schematic" in the free variables that occur in it, so the left rule represents a separate rule for each $a \in A$. That two states $q_1, q_2 \in K$ are *c-apart*, notation $q_1 \mathrel{\#}^c q_2$, is defined by

$$q_1 \mathrel{\#}^c q_2 \quad := \quad \forall Q \subseteq K \times K \, (\text{if } Q \text{ is a } c\text{-apartness, then } Q(q_1, q_2)).$$

In case the coalgebra c is clear from the context, we will ignore it. In DAs, two states are bisimilar if and only if they are not apart, which can easily be observed in the following example.

Example 2.3. Consider the DA given to the left below

$$\frac{q_3 \rightarrow_a q_0 \quad q_0 \rightarrow_a q_1 \quad \dfrac{\neg(q_0 \downarrow) \wedge q_1 \downarrow}{Q(q_0, q_1)}}{Q(q_3, q_0)}$$

A bisimulation is given by $q_1 \sim q_2$. It can be shown that $q_0 \mathrel{\#} q_3$ because for every apartness Q we have the derivation given on the right.

We see that "being c-apart", being the smallest relation satisfying specific closure properties, is an inductive property. This implies that the closure properties yield a *derivation system* for proving that two elements are c-apart. This will be further explored in the next section. In the example, we are basically using this: we have proven $q_3 \mathrel{\#} q_0$ by giving a derivation.

A relation Q is usually (e.g. see [TvD88], Chapter 8) called an apartness relation if it is irreflexive, symmetric and co-transitive. As we have already used the terminology "apartness relation" for the dual of a bisimulation relation, we shall, for the present paper, refer to these as "proper apartness relations".

Definition 2.4. A relation Q is called a *proper apartness relation* if it is
- *irreflexive*: $\forall x \, \neg Q(x, x)$,
- *symmetric*: $\forall x, y \, (Q(x, y) \implies Q(y, x))$,
- *co-transitive*: $\forall x, y, z \, (Q(x, y) \implies Q(x, z) \vee Q(z, y))$.

It is easy to see that inequality on a set is a proper apartness relation. The following is a standard fact that relates equivalence relations and proper apartness relations.

Lemma 2.5. *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. If $\neg R(x, y)$ and $R(x, z)$, then $\neg R(z, y)$ by transitivity of R, so we have $\neg R(x, y) \implies \neg R(x, z) \vee \neg R(z, y))$. The other way around, suppose $R(x, y)$ and $R(y, z)$ and $\neg R(x, z)$. Then $\neg R(x, y) \vee \neg R(z, y)$ by co-transitivity of $\neg R$, contradiction, so $R(x, z)$. $\qquad \square$

Bisimulation and apartness for DAs and streams can be defined by induction over the structure of the functor $F : \mathbf{Set} \to \mathbf{Set}$ that we consider the coalgebra for. In the case of DAs, we have $c : K \to F(K)$ with $F(X) = X^A \times 2$ and for streams, we have $c : K \to F(K)$ with $F(X) = A \times X$. The general definition in category-theoretic terms can be found in Section 4. A purely logical-syntactic presentation can be found in the Appendix.

Lemma 2.6. *We have the following result relating bisimulation and apartness for the case of DAs and streams (but it also applies to the general case treated in the Appendix).*
(1) *R is a bisimulation if and only if $\neg R$ is an apartness.*
(2) *The relation $\underleftrightarrow{\hspace{0.3em}}$ is the union of all bisimulations, $\underleftrightarrow{\hspace{0.3em}} = \bigcup\{R \mid R \text{ is a bisimulation}\}$, and it is itself a bisimulation.*

(3) *The relation $\underline{\#}$ satisfies $\underline{\#} = \bigcap \{Q \mid Q$ is an apartness relation$\}$, and is thus the intersection of all apartness relations; it is itself also an apartness relation.*

(4) $\underline{\leftrightarrow} = \neg \underline{\#}$.

Proof. We show the first in some detail for the case of DAs (Definition 2.2). It rests on some simple logical equivalences. That R is a c-bisimulation is equivalent to:

$$\Leftrightarrow \forall q_1, q_2 (R(q_1, q_2) \implies \forall a \in A\, R(c_1(q_1)(a), c_1(q_2)(a)) \wedge c_2(q_1) = c_2(q_2))$$
$$\Leftrightarrow \forall q_1, q_2 (\neg \forall a \in A\, R(c_1(q_1)(a), c_1(q_2)(a)) \wedge c_2(q_1) = c_2(q_2)) \implies \neg R(q_1, q_2))$$
$$\Leftrightarrow \forall q_1, q_2 (\exists a \in A\, \neg R(c_1(q_1)(a), c_1(q_2)(a)) \vee c_2(q_1) \neq c_2(q_2)) \implies \neg R(q_1, q_2))$$
$$\Leftrightarrow \forall q_1, q_2 (\exists a \in A\, \neg R(c_1(q_1)(a), c_1(q_2)(a)) \implies \neg R(q_1, q_2) \wedge (c_2(q_1) \neq c_2(q_2) \implies \neg R(q_1, q_2))),$$

which states that $\neg R$ is a c-apartness.

The other items are easily verified: if R_1 and R_2 are bisimulations, then $R_1 \cup R_2$ is also a bisimulation, and if Q_1 and Q_2 are apartness relations, then $Q_1 \cap Q_2$ is also an apartness relation. $\qquad \square$

Remark 2.7. A relation R is a c-bisimulation in case it satisfies a specific *closure property* that is given in Definitions 2.1, 2.2 via a rule that R should satisfy. Similarly, there is a closure property that defines when Q is a c-apartness (also given via a rule that Q should satisfy).

To prove that s and t are c-bisimilar, we need to find an R that satisfies the rules for c-bisimulations such that $R(s, t)$ holds. Dually, to prove that s and t are c-apart, we need to show that $Q(s, t)$ holds for every Q that satisfies the rules for c-apartness. This means that we can use the rules for being a c-apartness as the *derivation rules* of the proof system for proving $s \,\underline{\#}^c\, t$: we have $s \,\underline{\#}^c\, t$ if and only if there is a finite derivation of $s \,\underline{\#}^c\, t$ using these rules.

So, for apartness, the rules that define "Q is a c-apartness" can be used as the derivation rules for proving $s \,\underline{\#}^c\, t$. This is obviously not the case for bisimilarity. There the rules just represent the closure properties that R should satisfy to be a c-bisimulation[2]

In Sections 4 and 5 we will give a more general categorical picture of bisimulation and apartness on coalgebras.

2.1. **Apartness in constructive mathematics.** The notion of apartness is standard in constructive real analysis and goes back to Brouwer, with Heyting giving the first axiomatic treatment in [Hey27]. (See also e.g. [TvD88] Chapter 8.) The observation is that, if one reasons in constructive logic, the primitive notion for real numbers is apartness: if two real numbers are apart, this can be positively decided in a finite number of steps, just by computing better and better approximations until one positively knows an ϵ-distance between them. Then equality on real numbers is defined as the negation of apartness: $x = y := \neg(x \# y)$.

As a matter of fact, one can start from apartness and define equality using its negation, and then build up the real numbers axiomatically from there. This is done in [GN00], where an axiomatic description of real numbers is given and it is shown how Cauchy sequences over the rationals form a model of that axiomatization, all in a constructive setting, i.e. without

_____

[2]One could think of using the rules for bisimulation as "infinitary proof rules", where one allows infinite derivations of some form, but we will not expand on that here.

using the excluded middle rule. If one assumes apartness $\#$ to be a proper apartness (as in our Definition 2.4), the defined equality is an equivalence relation.

In the setting of the present paper, these constructive issues do not play a role, because we reason classically. There is one point to make, which is the issue of congruence, which has been studied in depth in the context of process theory [BBR10, Fok00]. Then the question is if, in a theory of terms describing processes, with a notion of bisimilarity describing a semantic equivalence of the terms as labelled transition systems, bisimulation is preserved by the operators of the theory. Simply put: if $q_1 \leftrightarrow p_1$ and $q_2 \leftrightarrow p_2$, is it the case that $f(q_1, q_2) \leftrightarrow f(p_1, p_2)$? In constructive analysis, if one starts from apartness and defines equality as its negation, the corresponding notion is *strong extensionality*.

Definition 2.8. A function $f : K \times K \to K$ is *strongly extensional* if

$$\forall x_1, x_2, y_1, y_2 \in K (f(x_1, x_2) \# f(y_1, y_2) \implies x_1 \# y_1 \vee x_2 \# y_2).$$

A relation $R \subseteq K \times K$ is *strongly extensional* if

$$\forall x_1, x_2, y_1, y_2 \in K (R(x_1, x_2) \implies R(y_1, y_2) \vee x_1 \# y_1 \vee x_2 \# y_2).$$

It is easily checked that, if one defines an equivalence relation $\sim$ as the negation of $\#$, then strong extensionality implies congruence with respect to $\sim$. So, if we wish to deal with process theories in terms of apartness, we will have to require operations and relations to be strongly extensional. It turns out that weaker forms of bisimulation (e.g. branching bisimulation) are not congruences, and therefore one considers *rooted branching bisimulation*. In Section 3.2 we will briefly study its complement, *rooted branching apartness* and the connection between congruence and strong extensionality.

3. Weak and branching bisimulation

We now apply the techniques that we have seen before to weak and branching bisimulation. We do not give a categorical treatment, because the functors proposed for weak [SdVW09] and branching [BK17] bisimulation are not so easy to work with. Instead, we use the definition of "bisimulation" (for a specific type of system) to directly define the notion of "apartness" as its negation, and thereby we define a derivation system for apartness. Then, two states s and t are (weakly, branching) apart iff they are not (weakly, branching) bisimilar. We also apply our definitions in a simple example to show how apartness (and thereby the absence of a bisimulation) can be proved.

We also rephrase some known results about branching bisimulation in terms of apartness, notably we reprove the stuttering property for branching bisimulation and the fact that branching bisimulation is an equivalence relation by rephrasing these results in terms of branching apartness. In the known proofs of these results, the notion of *semi-branching bisimulation* is used. Here we use a notion of *semi-branching apartness* for similar purposes. Finally we look into applications of the derivation system for actually deriving that two states in an LTS are branching apart (and therefore not branching bisimilar) and we suggest some new rules, using both apartness and bisimulation, that may be useful for analyzing algorithms for branching bisimulation.

The systems we focus on are *labelled transition systems*, LTSs. An LTS is a tuple $(X, A_\tau, \to)$, where X is a set of states, $A_\tau = A \cup \{\tau\}$ is a set of actions (containing the special "silent action" τ), and $\to \subseteq X \times A_\tau \times X$ is the *transition relation*. We write $q_1 \to_u q_2$

for $(q_1, u, q_2) \in \rightarrow$ and we write $\twoheadrightarrow_\tau$ to denote the reflexive transitive closure of $\rightarrow_\tau$. So $q_1 \twoheadrightarrow_\tau q_2$ if $q_1 \rightarrow_\tau \ldots \rightarrow_\tau q_2$ in zero or more τ-steps.

Convention 3.1. We will reserve $q_1 \rightarrow_a q_2$ to denote a transition with an a-step with $a \in A$ (so $a \neq \tau$).

First we recapitulate the standard definitions of labelled transition system and weak and branching bisimulation. We do this in a "rule" style. The standard definition of $R \subseteq X \times X$ being a *weak* bisimulation relation is that we have, for all $q, p, q' \in X$ and all $a \in A$,

$$R(q,p) \wedge q \rightarrow_\tau q' \implies \exists p'(p \twoheadrightarrow_\tau p' \wedge R(q',p'))$$
$$R(q,p) \wedge q \rightarrow_a q' \implies \exists p', p'', p'''(p \twoheadrightarrow_\tau p' \rightarrow_a p'' \twoheadrightarrow_\tau p''' \wedge R(q', p''')),$$

and also the symmetric variants of these two properties:

$$R(p,q) \wedge q \rightarrow_\tau q' \implies \exists p'(p \twoheadrightarrow_\tau p' \wedge R(p',q'))$$
$$R(p,q) \wedge q \rightarrow_a q' \implies \exists p', p'', p'''(p \twoheadrightarrow_\tau p' \rightarrow_a p'' \twoheadrightarrow_\tau p''' \wedge R(p''', q')),$$

Many rules in the rest of this paper have symmetric variants, like branching bisimulation above. We will not give these explicitly, but just refer to them as the "symmetric variants" of the rules.

We will rephrase the properties of weak/branching bisimulation (equivalently) as rules. These look uncommon for bisimulation, but will turn out to be useful when we look at their inverse, apartness.

Definition 3.2. A relation $R \subseteq X \times X$ on a LTS $(X, A_\tau, \rightarrow)$ is a *weak bisimulation relation* if it the following two rules and their symmetric variants hold for R.

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

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

The states q, p are *weakly bismilar*, notation $q \underleftrightarrow{}_w p$ if and only if there exists a weak bisimulation relation R such that $R(q,p)$.

A relation $R \subseteq X \times X$ is a *branching bisimulation relation* if the following two rules and their symmetric variants hold for R.

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

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

The states q, p are *branching bisimilar*, notation $q \underleftrightarrow{}_b p$ if and only if there exists a branching bisimulation relation R such that $R(q,p)$.

It is well-known that weak bisimulation is really weaker than branching bisimulation (if $s \underleftrightarrow{}_b t$, then $s \underleftrightarrow{}_w t$, but in general not the other way around) and that various efficient algorithms for checking branching bisimulation exist ([GV90, JGKW20]). Here we wish to analyze these notions by looking at their opposite: *weak apartness* and *branching apartness*.

Definition 3.3. Given a labelled transition system $(X, A_\tau, \rightarrow)$, we say that $Q \subseteq X \times X$ is a *weak apartness relation* in case the following rules hold for Q.

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

$$\frac{q \rightarrow_\tau q' \qquad \forall p'(p \twoheadrightarrow_\tau p' \implies Q(q',p'))}{Q(q,p)} \text{ in}_{w\tau}$$

$$\frac{q \rightarrow_a q' \qquad \forall p',p'',p'''(p \twoheadrightarrow_\tau p' \rightarrow_a p'' \twoheadrightarrow_\tau p''' \implies Q(q',p'''))}{Q(q,p)} \text{ in}_w$$

The states q and p are *weakly apart*, notation $q \underline{\#}_w p$, if for all weak apartness relations Q, we have $Q(q,p)$.

The relation of "being weakly apart" is itself a weak apartness relation: it is the smallest weak apartness relation, so we have an inductive definition of "being weakly apart", using a derivation system. We express this explicitly in the following Corollary to the Definition.

Corollary 3.4. *Given a labelled transition system $(X, A_\tau, \rightarrow)$, and $q, p \in X$, we have $q \underline{\#}_w p$ if and only if this can be derived using the following derivation rules.*

$$\frac{p \underline{\#}_w q}{q \underline{\#}_w p} \text{ symm}$$

$$\frac{q \rightarrow_\tau q' \qquad \forall p'(p \twoheadrightarrow_\tau p' \implies q' \underline{\#}_w p')}{q \underline{\#}_w p} \text{ in}_{w\tau}$$

$$\frac{q \rightarrow_a q' \qquad \forall p',p'',p'''(p \twoheadrightarrow_\tau p' \rightarrow_a p'' \twoheadrightarrow_\tau p''' \implies q' \underline{\#}_w p''')}{q \underline{\#}_w p} \text{ in}_w$$

Remark 3.5 (Also see Remark 2.7)**.** The notions of weak bisimulation and weak apartness are defined using closure properties that a relation should satisfy. As weak apartness is an inductive notion, the rules that define the closure property for weak apartness can be used as the derivation rules of a proof system to derive $q \underline{\#}_w p$. More precisely: we have $q \underline{\#}_w p$ if and only if this can be derived using a finite derivation with the rules of Corollary 3.4. Again, this is not the case for weak bisimilarity.

We now define the notion of *branching apartness*.

Definition 3.6. Given a labelled transition system $(X, A_\tau, \rightarrow)$, we say that $Q \subseteq X \times X$ is a *branching apartness* in case the following rules hold for Q.

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

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

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

The states q and p are *branching apart*, notation $q \mathrel{\underline{\#}_b} p$, if for all branching apartness relations Q, we have $Q(q, p)$.

Again, being branching apart is an inductive definition (it is the smallest branching apartness relation), so we have a derivation system. We express this explicitly in the following Corollary to the Definition, where again Remark 3.5 applies.

Corollary 3.7. *Given a labelled transition system* $(X, A_\tau, \rightarrow)$, *and* $q, p \in X$, *we have* $q \mathrel{\underline{\#}_b} p$ *if and only if this can be derived using the following derivation rules.*

$$\frac{p \mathrel{\underline{\#}_b} q}{q \mathrel{\underline{\#}_b} p} \text{ symm}$$

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

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

Remark 3.8 (A note on symmetry). In the rules, e.g. of Definition 3.6 and Corollary 3.7, there is a choice of adding symmetry as a rule, or adding symmetric variants of the rules. In our presentation, we choose to add symmetry as a rule. In the literature on bisimulation, it is standard to add symmetric variants of the rules, and then it can be shown that the relations themselves are symmetric. To be clear, the symmetric variants of the rules of Corollary 3.7 would be as follows.

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

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

and then one can prove that (without rule (symm)), the relation $\mathrel{\underline{\#}_b}$ is symmetric.

In the following, we will regularly prove properties about an apartness relation by *induction on the derivation* and then of course it matters which rules one has chosen. We found that having symmetry as a rule, and not a slightly informal "symmetric duplication" of all rules is a bit more clear and concise. In fact, for the proofs that are given below, it doesn't really matter what rules one has chosen: symmetry as a rule, or symmetry "built in" by adding the symmetric variants of the rules. The induction proofs that follow are mostly symmetric in either side of the apartness sign, with one notable exception, and that is the stuttering property, Lemma 3.21.

We now show how to use apartness on a few simple well-known examples. We show how we can *derive* that two states are branching apart (i.e. not branching bisimilar) by giving a derivation of this fact using the rules for $\mathrel{\underline{\#}_b}$.

Example 3.9. We describe two LTSs from [DV95] that serve as examples to show the difference between weak and branching bisimulation. We apply our apartness definitions to show the difference between $\mathrel{\underline{\#}_w}$ and $\mathrel{\underline{\#}_b}$. The LTS on the left consists of states $\{s, s_1, s_2, s_3, s_4, r, r_1, r_2, r_3\}$ and the point is that $s \mathrel{\underline{\#}_b} r$, while $s \mathrel{\underline{\leftrightarrow}_w} r$. The LTS on

the right consists of states $\{q, q_1, q_2, q_3, q_4, q_5, p, p_1, p_2, p_3, p_4\}$ and the point is that $q \mathrel{\#_b} p$, while $q \mathrel{\underleftrightarrow{}_w} p$.

In the LTS on the left, we have $s \mathrel{\#_b} r_1$, because s can do a d-step, while r_1 can not. Therefore, $s \mathrel{\#_b} r$, because $s \to_c s_2$ and the only possible c-step from r is $r \twoheadrightarrow_\tau r_1 \to_c r_3$, and $s \mathrel{\#_b} r_1$. Given that we now have a derivation system, we can also give a derivation of $s \mathrel{\#_b} r$:

$$\cfrac{s \to_c s_2 \qquad r \to_\tau r_1 \to_c r_3 \qquad \cfrac{\cfrac{s \to_d s_3}{s \mathrel{\#_b} r_1}}{s \mathrel{\#_b} r_1 \vee s_2 \mathrel{\#_b} r_3}}{s \mathrel{\#_b} r}$$

On the other hand we have $s \mathrel{\underleftrightarrow{}_w} r$. This can be seen by the weak bisimulation $\sim$ given by the following equivalence classes: $\{s, r\}, \{s_1, r_1\}, \{s_2, s_4, r_3\}, \{s_3, r_2\}$. This is indeed a weak bisimulation following Definition 3.2. A different way to prove $s \mathrel{\underleftrightarrow{}_w} r$ is by showing $\neg s \mathrel{\#_w} r$, which can be achieved by proving that there is no derivation of $s \mathrel{\#_w} r$. This is more involved, as we have to reason about all possible derivations of $s \mathrel{\#_w} r$. The only relevant candidate is below, which fails on finding a derivation of $s_2 \mathrel{\#_w} s_3$ (which does not exist).

$$\cfrac{s \to_c s_2 \qquad r \to_\tau r_1 \to_c r_3 \qquad \cfrac{??}{s_2 \mathrel{\#_w} r_3}}{s \mathrel{\#_w} r}$$

In the LTS on the right, we have $q_5 \mathrel{\#_b} p_1$, because q_5 cannot do an e-step. Therefore, $q \mathrel{\#_b} p$, because $q \to_c q_5$ and the only c-step from p leads to p_1 and $q_5 \mathrel{\#_b} p_1$. Also here, we can give a derivation:

$$\cfrac{q \to_c q_5 \qquad p \to_c p_1 \qquad \cfrac{\cfrac{p_1 \to_e p_3}{q_5 \mathrel{\#_b} p_1}}{q \mathrel{\#_b} p \vee q_5 \mathrel{\#_b} p_1}}{q \mathrel{\#_b} p}$$

The notions of weak, resp. branching, apartness and weak, resp. branching, bisimulation relate in the standard way we have seen before in Section 2: R is a weak (branching) apartness if and only if $\neg R$ is a weak (branching) bisimulation. This also implies that we can transfer properties from (weak/branching) bisimulation to (weak/branching) apartness and vice versa. In the next Section, we show how we can use apartness to proved results

about bisimulation. We now summarize the results that relate bisimulation and apartness in a couple of Lemmas.

Lemma 3.10. *A relation R over an LTS is a weak (resp. branching) bisimulation if and only if $\neg R$ is a weak (resp. branching) apartness.*

Proof. The proofs are by some standard logical manipulations, similar to the proof of Lemma 2.6. To simplify the work, it is easiest to first replace the rule (symm) by the "symmetric variants" of the other rules, as discussed in Remark 3.8. □

We have $\underline{\leftrightarrow}_w = \bigcup\{R \mid R$ is a weak bisimulation$\}$ and similarly for $\underline{\leftrightarrow}_b$ and it is straight-forward to verify that $\underline{\leftrightarrow}_w$ is itself a weak bisimulation (and similarly for $\underline{\leftrightarrow}_b$). For apartness we have the same result: $\underline{\#}_w = \bigcap\{Q \mid Q$ is a weak apartness$\}$, and similarly for $\underline{\#}_b$. The last part of the Lemma follows from

$$\neg(q \underline{\leftrightarrow}_w p) \Leftrightarrow \neg \exists R (R \text{ is a weak bisimulation} \wedge R(q,p))$$
$$\Leftrightarrow \forall R (R \text{ is a weak bisimulation} \implies \neg R(q,p))$$
$$\Leftrightarrow \forall Q (Q \text{ is a weak apartness} \implies Q(q,p))$$
$$\Leftrightarrow q \underline{\#}_w p.$$

This results in the following Lemma.

Lemma 3.11. (1) $\underline{\leftrightarrow}_w$ *(resp. $\underline{\leftrightarrow}_b$) is the largest weak (resp. branching) bisimulation.*
(2) $\underline{\#}_w$ *(resp. $\underline{\#}_b$) is the smallest weak (resp. branching) apartness.*
(3) $\underline{\#}_w = \neg \underline{\leftrightarrow}_w$ *and* $\underline{\#}_b = \neg \underline{\leftrightarrow}_b$.

3.1. **Using apartness to prove results about bisimulation.** The first result we prove is that weak apartness is included in branching apartness, which implies the well-known result that branching bisimulation is included in weak bisimulation. The interesting aspect is that we prove these results by induction (on the derivation). Then we will prove co-transitivity of branching apartness (which implies transitivity of branching bisimulation). We introduce semi-branching apartness as a means to prove a stuttering property and some other basic properties (for semi-branching apartness), from which we can conclude that semi-branching and branching apartness are the same, from which we derive co-transitivity.

Lemma 3.12. *If $s \underline{\#}_w t$, then $s \underline{\#}_b t$.*

Proof. By induction on the derivation of $s \underline{\#}_w t$, where we distinguish cases according to the last rule.

- Last rule is $(in_{w\tau})$

$$\frac{q \to_\tau q' \qquad \forall p'(p \twoheadrightarrow_\tau p' \implies q' \underline{\#}_w p')}{q \underline{\#}_w p} \, in_{w\tau}$$

By induction we have $\forall p'(p \twoheadrightarrow_\tau p' \implies q' \underline{\#}_b p')$, which implies $q' \underline{\#}_b p$ and $\forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies q \underline{\#}_b p' \vee q' \underline{\#}_b p'')$, which are the hypotheses for the rule $(in_{b\tau})$, so we conclude $q \underline{\#}_b p$ by the rule $(in_{b\tau})$.

- Last rule is (in_w)

$$\frac{q \to_a q' \qquad \forall p', p'', p'''(p \twoheadrightarrow_\tau p' \to_a p'' \twoheadrightarrow_\tau p''' \implies q' \mathbin{\underline{\#}}_w p''')}{q \mathbin{\underline{\#}}_w p} \text{in}_w$$

By induction we have $\forall p', p'', p'''(p \twoheadrightarrow_\tau p' \to_a p'' \twoheadrightarrow_\tau p''' \implies q' \mathbin{\underline{\#}}_b p''')$, which implies $\forall p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \implies q \mathbin{\underline{\#}}_b p' \vee q' \mathbin{\underline{\#}}_b p'')$, which is the hypothesis for the rule (in_b), so we conclude $q \mathbin{\underline{\#}}_b p$ by the rule (in_b).

□

It is well-known from the literature that the relations $\underline{\leftrightarrow}_w$ and $\underline{\leftrightarrow}_b$ are equivalence relations. For $\underline{\leftrightarrow}_w$, the proof is in [Mil89]. For $\underline{\leftrightarrow}_b$, the proof is remarkably subtle, as it is not the case in general that, if R_1 and R_2 are branching bisimulations, then $R_1 \circ R_2$ is a branching bisimulation. In [Bas96] the transitivity of $\underline{\leftrightarrow}_b$ is proven (and thereby that $\underline{\leftrightarrow}_b$ is an equivalence relation), using the notion of *semi-branching bisimulation*. In [GW96, Bas96], semi-branching bisimulation is also used to prove the so called *stuttering property*. The results from those papers can also be cast in terms of apartness, which we will do now. We prove that $\underline{\leftrightarrow}_b$ is an equivalence relation by proving that $\mathbin{\underline{\#}}_b$ is a proper apartness relation and using the fact that $\underline{\leftrightarrow}_b$ is the complement of $\mathbin{\underline{\#}}_b$. Similarly we prove an *apartness stuttering property* for $\mathbin{\underline{\#}}_b$ and conclude the stuttering property for $\underline{\leftrightarrow}_b$ from that. It turns out that, for proving co-transitivity of $\mathbin{\underline{\#}}_b$ (and also stuttering) we need a notion of *semi-branching apartness*, which is comparable to the complement of the notion of semi-branching bisimulation of [GW96, Bas96] (but slightly different). We introduce those notions first.

Definition 3.13. A relation $Q \subseteq X \times X$ is a *semi-branching apartness* in case the following derivation rules hold for Q. (So $\text{in}_{sb\tau}$ replaces the rule $\text{in}_{b\tau}$.)

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

$$\frac{q \to_\tau q' \qquad Q(q', p) \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies Q(q', p'') \vee (Q(q, p') \wedge Q(q, p'')))}{Q(q, p)} \text{in}_{sb\tau}$$

$$\frac{q \to_a q' \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \implies Q(q, p') \vee Q(q', p''))}{Q(q, p)} \text{in}_b$$

The states q and p are *semi-branching apart*, notation $q \mathbin{\underline{\#}}_{sb} p$, if for all semi-branching apartness relations Q, we have $Q(q, p)$.

So the rules symm and in_b are the same as for branching bisimulation of Definition 3.6, and only the rule for τ-steps has been modified. Note that in particular, to derive $Q(q, p)$ from $q \to_\tau q'$, we need to prove $Q(q', p)$ first.

Corollary 3.14. *The states q and p are semi-branching apart, $q \mathbin{\underline{\#}}_{sb} p$, if this can be derived from the following rules.*

$$\frac{p \mathbin{\underline{\#}}_{sb} q}{q \mathbin{\underline{\#}}_{sb} p} \text{symm}$$

$$\frac{q \to_\tau q' \qquad q' \mathbin{\underline{\#}}_{sb} p \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies q' \mathbin{\underline{\#}}_{sb} p'' \vee (q \mathbin{\underline{\#}}_{sb} p' \wedge q \mathbin{\underline{\#}}_{sb} p''))}{q \mathbin{\underline{\#}}_{sb} p} \text{in}_{sb\tau}$$

$$\frac{q \to_a q' \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \implies q \underline{\#}_{sb} p' \vee q' \underline{\#}_{sb} p'')}{q \underline{\#}_{sb} p} \; \text{in}_b$$

We also define the dual (complement) notion of a semi-branching bisimulation relation.

Definition 3.15. A relation $R \subseteq X \times X$ is a *semi-branching bisimulation relation* if the following two derivation rules and the symmetry rule hold for R.

$$\frac{q \to_\tau q' \qquad R(q, p)}{R(q', p) \vee \exists p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \wedge R(q', p'') \wedge (R(q, p') \vee R(q, p'')))} \; \text{bis}_{sb\tau}$$

$$\frac{q \to_a q' \qquad R(q, p)}{\exists p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \wedge R(q, p') \wedge R(q', p''))} \; \text{bis}_{sb}$$

The states q, p are *semi-branching bisimilar*, notation $q \underline{\leftrightarrow}_{sb} p$ if and only if there exists a semi-branching bisimulation relation R such that $R(q, p)$.

It can again be shown that Q is a semi-branching apartness if and only if $\neg Q$ is a semi-branching bisimulation. Using this and the fact that $\underline{\#}_{sb}$ is the smallest semi-branching apartness and $\underline{\leftrightarrow}_{sb}$ is the largest semi-branching bisimulation, we obtain that $q \underline{\#}_{sb} p \Leftrightarrow \neg(q \underline{\leftrightarrow}_{sb} p)$.

Our definition of semi-branching bisimulation is slightly different from the one in [Bas96] and [GW96], but it can be shown that they are equivalent.

The rest of this section will be devoted to proving the co-transitivity of $\underline{\#}_b$ (and thereby that $\underline{\leftrightarrow}_b$ is an equivalence relation) in the following steps.

(1) We prove that $q \underline{\#}_{sb} p \implies q \underline{\#}_b p$: Lemma 3.16.
(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 $\underline{\#}_b$: Lemma 3.17 and Corollary 3.19
(3) We prove the *apartness stuttering property* for $\underline{\#}_{sb}$: Lemma 3.21.
(4) We prove that $q \underline{\#}_b p \implies q \underline{\#}_{sb} p$, using the apartness stuttering property, and we conclude that $\underline{\#}_b = \underline{\#}_{sb}$: Lemma 3.22.
(5) We prove co-transitivity for $\underline{\#}_b$, using the basic lemmas mentioned above.

Many of the proofs will proceed by induction on the derivation, where we use the apartness as an inductively defined relation (defined via derivation rules). For one of the basic Lemmas under (2) we will move over to the "bisimulation view", as the result seems easier to obtain there.

Lemma 3.16. *For all states q, p, $q \underline{\#}_{sb} p \implies q \underline{\#}_b p$.*

Proof. By induction on the derivation of $q \underline{\#}_{sb} p$. The only interesting case is when the last rule applied is $(\text{in}_{sb\tau})$.

$$\frac{q \to_\tau q' \qquad q' \underline{\#}_{sb} p \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies q' \underline{\#}_{sb} p'' \vee (q \underline{\#}_{sb} p' \wedge q \underline{\#}_{sb} p''))}{q \underline{\#}_{sb} p} \; \text{in}_{sb\tau}$$

We have $q \to_\tau q'$ and by induction hypothesis $q' \underline{\#}_b p$ and $\forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies q' \underline{\#}_b p'' \vee (q \underline{\#}_b p' \wedge q \underline{\#}_b p''))$. To apply rule $(\text{in}_{b\tau})$ and conclude $q \underline{\#}_b p$ we need to prove $\forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies q \underline{\#}_b p' \vee q' \underline{\#}_b p'')$. Let p', p'' be such that $p \twoheadrightarrow_\tau p' \to_\tau p''$. Form the induction hypothesis we have two cases.

- Case $q' \underline{\#}_b p''$; then $q \underline{\#}_b p' \vee q' \underline{\#}_b p''$, so done.
- Case $q \underline{\#}_b p' \wedge q \underline{\#}_b p''$; then $q \underline{\#}_b p'$ and so $q \underline{\#}_b p' \vee q' \underline{\#}_b p''$ and done.

$\square$

We first state two simple derivable rules, that are nevertheless convenient to make explicit for use in further proofs.

Lemma 3.17. *The following two derived rules holds for* $\underline{\#}_{sb}$.

(1)

$$\frac{p \twoheadrightarrow_\tau t \quad q \rightarrow_\tau q' \quad q' \underline{\#}_{sb} p \quad \forall p', p''(p \twoheadrightarrow_\tau p' \rightarrow_\tau p'' \implies q' \underline{\#}_{sb} p'' \vee (q \underline{\#}_{sb} p' \wedge q \underline{\#}_{sb} p''))}{q \underline{\#}_{sb} t}$$

(2)

$$\frac{p \twoheadrightarrow_\tau t \quad q \rightarrow_a q' \quad \forall p', p''(p \twoheadrightarrow_\tau p' \rightarrow_a p'' \implies q \underline{\#}_{sb} p' \vee q' \underline{\#}_{sb} p'')}{q \underline{\#}_{sb} t}$$

Proof. For the proof of (1), assume (a) $p \twoheadrightarrow_\tau t$, (b) $q \rightarrow_\tau q'$, (c) $q' \underline{\#}_{sb} p$ and (d) $\forall p', p''(p \twoheadrightarrow_\tau p' \rightarrow_\tau p'' \implies q' \underline{\#}_{sb} p'' \vee (q \underline{\#}_{sb} p' \wedge q \underline{\#}_{sb} p''))$. Then by rule $(\text{in}_{sb\tau})$, we find $q \underline{\#}_{sb} p$, so we may assume that (a) $p \twoheadrightarrow_\tau t$ is non-empty and we have (e) $p \twoheadrightarrow_\tau p' \rightarrow_\tau t$.

We use (e) in (d), taking t for p'' and find that $q' \underline{\#}_{sb} t$ or $q \underline{\#}_{sb} p' \wedge q \underline{\#}_{sb} t$. In the latter case we have $q \underline{\#}_{sb} t$ and we are done. In case $q' \underline{\#}_{sb} t$, to prove $q \underline{\#}_{sb} t$, we apply rule $(\text{in}_{sb\tau})$. We need to show that $\forall t', t''(t \twoheadrightarrow_\tau t' \rightarrow_\tau t'' \implies q' \underline{\#}_{sb} t'' \vee (q \underline{\#}_{sb} t' \wedge q \underline{\#}_{sb} t''))$, which follows from $p \twoheadrightarrow_\tau p' \rightarrow_\tau t$ and (d).

The proof of (2) is similar, but slightly simpler.

$\square$

Lemma 3.18. *The following two derived rules holds for* $\underline{\leftrightarrow}_{sb}$ *(and as a matter of fact they hold for any semi-branching bisimulation relation).*

(1)

$$\frac{q_0 \twoheadrightarrow_\tau q \rightarrow_\tau q' \quad q_0 \underline{\leftrightarrow}_{sb} p}{q' \underline{\leftrightarrow}_{sb} p \vee \exists p', p''(p \twoheadrightarrow_\tau p' \rightarrow_\tau p'' \wedge q' \underline{\leftrightarrow}_{sb} p'' \wedge (q \underline{\leftrightarrow}_{sb} p' \vee q \underline{\leftrightarrow}_{sb} p''))}$$

(2)

$$\frac{q_0 \twoheadrightarrow_\tau q \rightarrow_a q' \quad q_0 \underline{\leftrightarrow}_{sb} p}{\exists p', p''(p \twoheadrightarrow_\tau p' \rightarrow_a p'' \wedge q \underline{\leftrightarrow}_{sb} p' \wedge q' \underline{\leftrightarrow}_{sb} p'')}$$

Proof. Assuming $q_0 \twoheadrightarrow_\tau q'$ has the shape $q_0 \rightarrow_\tau q_1 \ldots \rightarrow_\tau q_n = q$, the proof proceeds by induction on n. We only treat (1), because (2) is similar (but slightly simpler).

- ($n = 0$) We need to show that the following holds

$$\frac{q_0 \rightarrow_\tau q' \quad q_0 \underline{\leftrightarrow}_{sb} p}{q' \underline{\leftrightarrow}_{sb} p \vee \exists p', p''(p \twoheadrightarrow_\tau p' \rightarrow_\tau p'' \wedge q' \underline{\leftrightarrow}_{sb} p'' \wedge (q_0 \underline{\leftrightarrow}_{sb} p' \vee q_0 \underline{\leftrightarrow}_{sb} p''))}$$

  which is immediate by $(\text{bis}_{sb\tau})$

- ($n > 0$) We need to show that the following holds

$$\frac{q_0 \twoheadrightarrow_\tau q_1 \rightarrow_\tau q \rightarrow_\tau q' \quad q_0 \underline{\leftrightarrow}_{sb} p}{q' \underline{\leftrightarrow}_{sb} p \vee \exists p', p''(p \twoheadrightarrow_\tau p' \rightarrow_\tau p'' \wedge q' \underline{\leftrightarrow}_{sb} p'' \wedge (q \underline{\leftrightarrow}_{sb} p' \vee q \underline{\leftrightarrow}_{sb} p''))}$$

By the induction hypothesis, we find

$$q \mathrel{\underline{\leftrightarrow}}_{sb} p \vee \exists p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \wedge q \mathrel{\underline{\leftrightarrow}}_{sb} p'' \wedge (q_1 \mathrel{\underline{\leftrightarrow}}_{sb} p' \vee q_1 \mathrel{\underline{\leftrightarrow}}_{sb} p'')).$$

- In case $q \mathrel{\underline{\leftrightarrow}}_{sb} p$, we apply rule $(\text{bis}_{sb\tau})$ and conclude $q' \mathrel{\underline{\leftrightarrow}}_{sb} p \vee \exists p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \wedge q' \mathrel{\underline{\leftrightarrow}}_{sb} p'' \wedge (q \mathrel{\underline{\leftrightarrow}}_{sb} p' \vee q \mathrel{\underline{\leftrightarrow}}_{sb} p''))$ and we are done.
- In the other case, consider the p', p'' for which $p \twoheadrightarrow_\tau p' \to_\tau p''$, $q \mathrel{\underline{\leftrightarrow}}_{sb} p''$ and $q_1 \mathrel{\underline{\leftrightarrow}}_{sb} p' \vee q_1 \mathrel{\underline{\leftrightarrow}}_{sb} p''$. We have $q \to_\tau q'$ and $q \mathrel{\underline{\leftrightarrow}}_{sb} p''$, so by rule $(\text{bis}_{sb\tau})$ we derive $q' \mathrel{\underline{\leftrightarrow}}_{sb} p'' \vee \exists p_1, p_2(p'' \twoheadrightarrow_\tau p_1 \to_\tau p_2 \wedge q' \mathrel{\underline{\leftrightarrow}}_{sb} p_2 \wedge (q \mathrel{\underline{\leftrightarrow}}_{sb} p_1 \vee q \mathrel{\underline{\leftrightarrow}}_{sb} p_2))$.
  * In case $q' \mathrel{\underline{\leftrightarrow}}_{sb} p''$, we have $q' \mathrel{\underline{\leftrightarrow}}_{sb} p'' \wedge (q \mathrel{\underline{\leftrightarrow}}_{sb} p' \vee q \mathrel{\underline{\leftrightarrow}}_{sb} p'')$ and we are done.
  * In case $\exists p_1, p_2(p'' \twoheadrightarrow_\tau p_1 \to_\tau p_2 \wedge q' \mathrel{\underline{\leftrightarrow}}_{sb} p_2 \wedge (q \mathrel{\underline{\leftrightarrow}}_{sb} p_1 \vee q \mathrel{\underline{\leftrightarrow}}_{sb} p_2))$ we also have $\exists p_1, p_2(p \twoheadrightarrow_\tau p_1 \to_\tau p_2 \wedge q' \mathrel{\underline{\leftrightarrow}}_{sb} p_2 \wedge (q \mathrel{\underline{\leftrightarrow}}_{sb} p_1 \vee q \mathrel{\underline{\leftrightarrow}}_{sb} p_2))$ and we are done.

$\square$

Corollary 3.19. *The following two derived rules holds for $\mathrel{\underline{\#}}_{sb}$.*

(1)

$$\frac{q_0 \twoheadrightarrow_\tau q \to_\tau q' \qquad q' \mathrel{\underline{\#}}_{sb} p \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies q' \mathrel{\underline{\#}}_{sb} p'' \vee (q \mathrel{\underline{\#}}_{sb} p' \wedge q \mathrel{\underline{\#}}_{sb} p''))}{q_0 \mathrel{\underline{\#}}_{sb} p}$$

(2)

$$\frac{q_0 \twoheadrightarrow_\tau q \to_a q' \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \implies q \mathrel{\underline{\#}}_{sb} p' \vee q' \mathrel{\underline{\#}}_{sb} p'')}{q_0 \mathrel{\underline{\#}}_{sb} p}$$

Proof. Immediately from Lemma 3.18 by taking the complement. $\square$

In the literature on branching bisimulation, the "stuttering property" refers to the following property for a relation R, that we depict as a rule here.

$$\frac{r \to_\tau r_1 \to_\tau \ldots \to_\tau r_n \to t \quad (n \geq 0) \qquad R(r, p) \qquad R(t, p)}{\forall i (1 \leq i \leq n) R(r_i, p)} \tag{3.1}$$

So, if in a τ-path, the first and the last state are bisimilar with p, then all states in between are bisimilar with p. In [GW96] (and also in other papers), the stuttering property is proved for $\mathrel{\underline{\leftrightarrow}}_b$. We cast this property in terms of apartness.

Definition 3.20. A relation Q satisfies the *apartness stuttering property* if the following rule holds for Q.

$$\frac{r \twoheadrightarrow_\tau q \twoheadrightarrow_\tau t \qquad Q(q, p)}{Q(r, p) \vee Q(t, p)} \text{ stut}$$

The equivalence between Q being an apartness stuttering property and $\neg Q$ satisfying the stuttering property of 3.1 should be clear. Another way of phrasing the stuttering property for bisimulations, e.g. in [DV95], is as follows.

$$\frac{r \to_\tau r_1 \to_\tau \ldots \to_\tau r_n \to t \quad (n \geq 0) \qquad R(r, t)}{\forall i (1 \leq i \leq n) R(r_0, r_i)} \tag{3.2}$$

The apartness variation of this property is

$$\frac{r \twoheadrightarrow_\tau s \twoheadrightarrow_\tau t \qquad Q(r, s)}{Q(r, t)} \tag{3.3}$$

Property 3.3 follows easily from the apartness stuttering property of Definition 3.20, using irreflexivity of Q.

Lemma 3.21. *The relation $\underline{\#}_{sb}$ (semi-branching apartness) satisfies the apartness stuttering property:*

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

Proof. By induction on the proof of $q \underline{\#}_{sb} p$. There are four cases to consider: either $q \underline{\#}_{sb} p$ was derived by rule $(\text{in}_{sb\tau})$ or (in_b), or $p \underline{\#}_{sb} q$ was derived by rule $(\text{in}_{sb\tau})$ or (in_b), and then $q \underline{\#}_{sb} p$ was derived by symmetry (symm).

- Case $q \underline{\#}_{sb} p$ was derived using rule $(\text{in}_{sb\tau})$. So we have

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

  Then we conclude $r \underline{\#}_{sb} p$ using Corollary 3.19 (1), and so $r \underline{\#}_{sb} p \vee t \underline{\#}_{sb} p$.

- Case $q \underline{\#}_{sb} p$ was derived using rule (in_b). So we have

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

  Then we conclude $r \underline{\#}_{sb} p$ using Corollary 3.19 (2), and so $r \underline{\#}_{sb} p \vee t \underline{\#}_{sb} p$.

- Case $p \underline{\#}_{sb} q$ was derived using rule $(\text{in}_{sb\tau})$. So we have

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

  Then we conclude $p \underline{\#}_{sb} t$ using Lemma 3.17 (1), and so $r \underline{\#}_{sb} p \vee t \underline{\#}_{sb} p$.

- Case $p \underline{\#}_{sb} q$ was derived using rule (in_b). So we have

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

  Then we conclude $p \underline{\#}_{sb} t$ using Lemma 3.17 (2), and so $r \underline{\#}_{sb} p \vee t \underline{\#}_{sb} p$.

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \square$

Lemma 3.22. *Branching apartness is included in semi-branching apartness and thereby the two notions coincide: $\underline{\#}_b = \underline{\#}_{sb}$.*

Proof. We prove $q \underline{\#}_b p \implies q \underline{\#}_{sb} p$ by induction on the derivation of $q \underline{\#}_b p$, using the apartness stuttering property. We conclude $\underline{\#}_b = \underline{\#}_{sb}$ using Lemma 3.16.

For the induction we only treat the case where $q \underline{\#}_b p$ has been derived using the rule $(\text{in}_{b\tau})$, as the other cases are immediate. So assume we have the following.

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

So we have $q \to_\tau q'$ and by induction hypothesis we have (a) $q' \mathrel{\#_{sb}} p$ and (b) $\forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies q \mathrel{\#_{sb}} p' \vee q' \mathrel{\#_{sb}} p'')$. To be able to apply the rule $(\mathrm{in}_{sb\tau})$ to conclude $q \mathrel{\#_{sb}} p$, we need to prove

$$\forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies q' \mathrel{\#_{sb}} p'' \vee (q \mathrel{\#_{sb}} p' \wedge q \mathrel{\#_{sb}} p'')).$$

Let p', p'' be such that $p \twoheadrightarrow_\tau p' \to_\tau p''$. Using (b) we have two cases.

- Case $q \mathrel{\#_{sb}} p'$. Then, by $p \twoheadrightarrow_\tau p' \to_\tau p''$ and the stuttering property (Lemma 3.21), we have $q \mathrel{\#_{sb}} p \vee q \mathrel{\#_{sb}} p''$. In case $q \mathrel{\#_{sb}} p$, we are done, because that's exactly what we had to prove in the end; in case $q \mathrel{\#_{sb}} p''$ we have $q' \mathrel{\#_{sb}} p'' \vee (q \mathrel{\#_{sb}} p' \wedge q \mathrel{\#_{sb}} p'')$ and we are done.
- Case $q' \mathrel{\#_{sb}} p''$. Then $q' \mathrel{\#_{sb}} p'' \vee (q \mathrel{\#_{sb}} p' \wedge q \mathrel{\#_{sb}} p'')$ and we are done.

$\square$

As a consequence of this Lemma, Corollary 3.19 and Lemma 3.17 also apply to branching apartness, $\mathrel{\#_b}$.

Lemma 3.23. *The relation $\mathrel{\#_b}$ is co-transitive: for all q, p, r: if $q \mathrel{\#_b} p$, then $q \mathrel{\#_b} r \vee r \mathrel{\#_b} p$.*

Proof. We prove $q \mathrel{\#_b} p \implies \forall r(q \mathrel{\#_b} r \vee r \mathrel{\#_b} p)$ by induction on the derivation of $q \mathrel{\#_b} p$, using the properties we have proved before about $\mathrel{\#_b}$ and $\mathrel{\#_{sb}}$.

- Case $q \mathrel{\#_b} p$ was derived using $(\mathrm{in}_{b\tau})$.

$$\frac{q \to_\tau q' \qquad q' \mathrel{\#_b} p \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \implies q \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p'')}{q \mathrel{\#_b} p} \, \mathrm{in}_{b\tau}$$

Let r be a state. If (a) $q' \mathrel{\#_b} r$ and (b) $\forall r', r''(r \twoheadrightarrow_\tau r' \to_\tau r'' \implies q \mathrel{\#_b} r' \vee q' \mathrel{\#_b} r'')$, then $q \mathrel{\#_b} r$ and we are done. Otherwise, $\neg(q' \mathrel{\#_b} r)$ or

$$\exists r', r''(r \twoheadrightarrow_\tau r' \to_\tau r'' \wedge \neg(q \mathrel{\#_b} r') \wedge \neg(q' \mathrel{\#_b} r'')).$$

If $\neg(q' \mathrel{\#_b} r)$, we apply induction on $q' \mathrel{\#_b} p$ to derive $q' \mathrel{\#_b} r \vee r \mathrel{\#_b} p$, from which we conclude $r \mathrel{\#_b} p$ and we are done.

In the other case we consider r', r'' with (d) $\neg(q \mathrel{\#_b} r')$ and (e) $\neg(q' \mathrel{\#_b} r'')$. We will prove that $r \mathrel{\#_b} p$. Let p', p'' be such that $p \twoheadrightarrow_\tau p' \to_\tau p''$. (If there are no such p', p'', then $r \mathrel{\#_b} p$ due to Corollary 3.19 (1) and the fact that $r'' \mathrel{\#_b} p$, which follows from induction on $q' \mathrel{\#_b} p$, which yields $q' \mathrel{\#_b} r'' \vee r'' \mathrel{\#_b} p$, but we know $\neg(q' \mathrel{\#_b} r'')$ from (e).) Then $q \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p''$.
  – Case $q \mathrel{\#_b} p'$. Then by induction $q \mathrel{\#_b} r' \vee p' \mathrel{\#_b} r'$, so $p' \mathrel{\#_b} r'$ by (d) and so $p' \mathrel{\#_b} r' \vee p'' \mathrel{\#_b} r''$.
  – Case $q' \mathrel{\#_b} p''$. Then by induction $q' \mathrel{\#_b} r'' \vee p'' \mathrel{\#_b} r''$, so $p'' \mathrel{\#_b} r''$ by (e) and so $p' \mathrel{\#_b} r' \vee p'' \mathrel{\#_b} r''$.
  So $r' \mathrel{\#_b} p$ and we apply Corollary 3.19 (1), to conclude $r \mathrel{\#_b} p$.
- Case $q \mathrel{\#_b} p$ was derived using (in_b).

$$\frac{q \to_a q' \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \implies q \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p'')}{q \mathrel{\#_b} p} \, \mathrm{in}_b$$

Let r be a state. If $\forall r', r''(r \twoheadrightarrow_\tau r' \to_a r'' \implies q \mathrel{\underline{\#}_b} r' \vee q' \mathrel{\underline{\#}_b} r'')$, then $q \mathrel{\underline{\#}_b} r$ and we are done. Otherwise

$$\exists r', r''(r \twoheadrightarrow_\tau r' \to_a r'' \wedge \neg(q \mathrel{\underline{\#}_b} r') \wedge \neg(q' \mathrel{\underline{\#}_b} r'')).$$

Consider r', r'' with (d) $\neg(q \mathrel{\underline{\#}_b} r')$ and (e) $\neg(q' \mathrel{\underline{\#}_b} r'')$). We will prove that $r \mathrel{\underline{\#}_b} p$. Let p', p'' be such that $p \twoheadrightarrow_\tau p' \to_a p''$. (If there are no such p', p'', then $r \mathrel{\underline{\#}_b} p$ due to Corollary 3.19 (2).) Then $q \mathrel{\underline{\#}_b} p' \vee q' \mathrel{\underline{\#}_b} p''$.
– Case $q \mathrel{\underline{\#}_b} p'$. Then by induction $q \mathrel{\underline{\#}_b} r' \vee p' \mathrel{\underline{\#}_b} r'$, so $p' \mathrel{\underline{\#}_b} r'$ by (d) and so $p' \mathrel{\underline{\#}_b} r' \vee p'' \mathrel{\underline{\#}_b} r''$.
– Case $q' \mathrel{\underline{\#}_b} p''$. Then by induction $q' \mathrel{\underline{\#}_b} r'' \vee p'' \mathrel{\underline{\#}_b} r''$, so $p'' \mathrel{\underline{\#}_b} r''$ by (e) and so $p' \mathrel{\underline{\#}_b} r' \vee p'' \mathrel{\underline{\#}_b} r''$.
So $r' \mathrel{\underline{\#}_b} p$ and we apply Corollary 3.19 (2), to conclude $r \mathrel{\underline{\#}_b} p$.

$\square$

The co-transitivity is the crucial property for showing that $\mathrel{\underline{\#}_b}$ is a proper apartness relation.

Theorem 3.24. *The relation $\mathrel{\underline{\#}_b}$ is a proper apartness relation (in the sense of Definition 2.4).*

Proof. We need to verify irreflexivity, symmetry and co-transitivity. Symmetry is built in and co-transitivity has been proved in Lemma 3.23. For irreflexivity, consider the shortest derivation of $q \mathrel{\underline{\#}_b} q$ (for some q). If this is derived using rule in_b, we have $q \to_a q'$ and $q \mathrel{\underline{\#}_b} q \vee q' \mathrel{\underline{\#}_b} q'$, which means that there is a shorter derivation of a reflexivity, contradiction. If this is derived using rule $in_{b\tau}$, we have $q \to_\tau q'$ and $q \mathrel{\underline{\#}_b} q \vee q' \mathrel{\underline{\#}_b} q'$, which again means that there is a shorter derivation of a reflexivity, contradiction. So there is no derivation of $q \mathrel{\underline{\#}_b} q$ for any q.

$\square$

Corollary 3.25. *The relation $\mathrel{\underline{\leftrightarrow}_b}$ is an equivalence relation.*

Proof. Immediately from Theorem 3.24 using the fact that $\mathrel{\underline{\leftrightarrow}_b}$ is the complement of $\mathrel{\underline{\#}_b}$. $\square$

3.2. **Using branching apartness.** Further research has to establish whether the notion of apartness is really useful in the study and analysis of labelled transition systems. In the previous section we have shown how to use apartness in the meta-theory of branching bisimulation to give some new proofs for known properties. In follow up research we would like to analyze well-known algorithms for checking branching bisimulation, as in [JGKW20], and possibly develop variations on those algorithms. One way to decide branching bisimilarity of states in a finite LTS is by deciding branching apartness. In the present section, we give some ideas of what an algorithm for deciding branching apartness could look like and we also give some variations of the rules for branching apartness, also combined with branching bisimulation that might provide useful. In the end of this section, we briefly mention *rooted branching apartness* as the complement of *rooted branching bisimulation*. Rooted branching bisimulation is a congruence [GW96, Fok00], while branching bisimulation is not. For apartness this means that operations are strongly extensional with respect to rooted branching apartness, while they are not with respect to branching apartness.

An obvious algorithm to decide $q \mathrel{\underline{\#}_b} p$ is by trying to find a derivation of $q \mathrel{\underline{\#}_b} p$ in a structured way and concluding that $q \mathrel{\underline{\leftrightarrow}_b} p$ holds in case such a derivation cannot be found.

It may look as if, for LTSs with loops, this could lead to an infinite search process. But this can be avoided if we look for a *shortest derivation* and keep track of goals that we have already encountered. If we encounter the goal again, we can conclude it is not provable. Also, some of the goals will be disjunctions of apartness assertions, like $q \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p''$. In that case we will search for a proof of $q \mathrel{\#_b} p'$ and for a proof of $q' \mathrel{\#_b} p''$, in parallel, and we conclude as soon as we have found a proof of one of them. To clarify this point a bit better we show two pairs of LTSs with loops and how a proof of branching apartness is found for the first pair, and a proof of branching bisimilarity for the second pair.

Example 3.26. We give 4 LTSs with loops.

In the first two LTSs, we have $q_0 \mathrel{\#_b} p_0$, which is established by the derivation below.

Observe that an algorithm would have to go through all possible d-steps from q_0 and "replay" them from p_0. We have chosen the "successful" d-step that leads to a derivation of $q_0 \mathrel{\#_b} p_0$. Similarly in proving $q_0 \mathrel{\#_b} p_2$, we have chosen the successful d-step, $q_0 \rightarrow_d q_1$. When proving a disjunction, an algorithm would have to try to prove both parts of the disjunction in parallel. We have only shown the successful one.

For the third and fourth LTS, we have $q \mathrel{\underleftrightarrow{}_b} p$, so we want to show that $\neg(q \mathrel{\#_b} p)$. This is achieved by trying to find the shortest derivation and observing there is none. This search leads to the following derivation.

Note that this is the complete search tree for a derivation of $q \mathrel{\#_b} p$, where we have stopped at a branch as soon as we find a goal that we have already encountered. Therefore we fail at

the goal $q' \mathrel{\#_b} p \vee q \mathrel{\#_b} p$, because both $q' \mathrel{\#_b} p$ and $q \mathrel{\#_b} p$ have already been encountered. Conclusion: $\neg(q \mathrel{\#_b} p)$, that is $q \mathrel{\leftrightarrow_b} p$.

We now look into some variations on the rules for branching apartness.

Lemma 3.27. *The following alternative* in_b-*rule*[3] *is sound for* $\mathrel{\#_b}$.

$$\frac{q \to_a q' \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \implies p \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p'')}{q \mathrel{\#_b} p} \; \mathrm{in}_b^A$$

Proof. Assume we have $q \to_a q'$ and $\forall p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \implies p \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p'')$. We need to prove $q \mathrel{\#_b} p$. Suppose $\neg q \mathrel{\#_b} p$. We want to apply the original in_b-rule, so we need to prove the hypothesis to that rule, which is $\forall p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \implies q \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p'')$. Let p', p'' be such that $p \twoheadrightarrow_\tau p' \to_a p''$. Then $p \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p''$.

- Case $p \mathrel{\#_b} p'$. Then $q \mathrel{\#_b} p' \vee q \mathrel{\#_b} p$ by co-transitivity. We know from our assumption that $\neg q \mathrel{\#_b} p$, so $q \mathrel{\#_b} p'$ and so $q \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p''$ and done.
- Case $q' \mathrel{\#_b} p''$. Then $q \mathrel{\#_b} p' \vee q' \mathrel{\#_b} p''$ and done. So we can apply the original in_b-rule and conclude $q \mathrel{\#_b} p$. This contradicts our assumption $\neg q \mathrel{\#_b} p$, so we conclude $q \mathrel{\#_b} p$. □

We conjecture that the rule (in_b^A) is also complete for proving $\mathrel{\#_b}$, that is: if we replace rule (in_b) with rule (in_b^A) we can derive the same apartness judgments. If we write $\mathrel{\#_b^A}$ for the system with rule (in_b) replaced by rule (in_b^A), Lemma 3.27 states that $q \mathrel{\#_b^A} p \implies q \mathrel{\#_b} p$. For the proof of completeness, $q \mathrel{\#_b} p \implies q \mathrel{\#_b^A} p$, it seems we need to prove co-transitivity of $\mathrel{\#_b^A}$ first.

Using the notion of apartness, we can also add some rules that combine apartness and bisimulation and that may be useful in analyzing or developing new algorithms for checking branching bisimulation, as in [JGKW20].

Lemma 3.28. *The following two rules are sound for proving branching apartness* $\mathrel{\#_b}$.

$$\frac{q \to_\tau q' \qquad q' \mathrel{\#_b} p \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \wedge q' \mathrel{\leftrightarrow_b} p'' \implies q \mathrel{\#_b} p')}{q \mathrel{\#_b} p} \; \mathrm{in}_{b\tau}^{\leftrightarrow_b}$$

$$\frac{q \to_a q' \qquad \forall p', p''(p \twoheadrightarrow_\tau p' \to_a p'' \wedge q' \mathrel{\leftrightarrow_b} p'' \implies q \mathrel{\#_b} p')}{q \mathrel{\#_b} p} \; \mathrm{in}_b^{\leftrightarrow_b}$$

Proof. The proof is immediate from the fact that $\mathrel{\#_b} = \neg \mathrel{\leftrightarrow_b}$ and Corollary 3.7. □

In the literature, the rules concerning bisimulation are often depicted in a diagram for better memorization. The two rules above can be depicted as follows.

---

[3]Thanks to David N. Jansen for suggesting this rule

On the left: Suppose that $q' \underline{\#}_b\, p$ and for all p', p'' with $p \twoheadrightarrow_\tau p' \to_\tau p''$, if $q' \underline{\leftrightarrow}_b p''$, then $q \underline{\#}_b p'$. Then (in dashes): $q \underline{\#}_b p$.

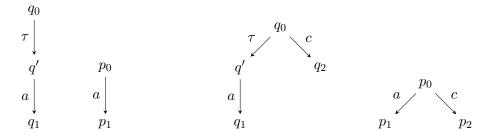Similarly, we introduce an adapted rule for branching bisimulation.

Lemma 3.29. *The following rule is sound for branching bisimulations. If R is a branching bisimulation, then the following rule (and its symmetric variant) holds.*

$$\frac{q \to_\tau q' \qquad R(q,p) \qquad q' \underline{\#}_b\, p}{\exists p', p''(p \twoheadrightarrow_\tau p' \to_\tau p'' \wedge R(q,p') \wedge R(q',p''))}$$

Proof. The proof is immediate from the definition of branching bisimulation (Definition 3.2) and the fact that, if R is a branching bisimulation, then $q' \underline{\#}_b\, p \implies \neg R(q',p)$.   □

The notion of rooted branching bisimulation has been introduced to recover the failure of the congruence property for branching bisimulation [GW96, BBR10, Fok00]. If two labelled transition systems are branching bisimilar and one composes them both with a third one, one expects the newly obtained LTSs to be branching bisimilar again. But this is not the case. Of course, this also depends on the notion of composition and therefore this problem is usually cast in terms of process terms and a process operator f, where congruence means that if $q_1 \underline{\leftrightarrow}_b p_1$ and $q_2 \underline{\leftrightarrow}_b p_2$, then $f(q_1, q_2) \underline{\leftrightarrow}_b f(p_1, p_2)$. Here, the bisimulation equivalence should be understood as being between the interpretation of the process terms as LTSs. A main example of non-congruence arises from the non-deterministic choice operator $+$. A process $q + p$ can non-deterministically choose for q and do a step in q or for p and do a step in p. The LTS for $q + p$ arises from joining the LTS for q with the one for p at the root node. This leads to the non-congruence exemplified below.

Example 3.30. For the two LTSs on the left we have $q_0 \underline{\leftrightarrow}_b p_0$. If we compose them via non-deterministic choice with an LTS that does just a c-step, we get the two LTSs on the right, for which we have $\neg(q_0 \underline{\leftrightarrow}_b p_0)$.

For the LTSs on the right, a derivation of $q_0 \underline{\#}_b p_0$ is as follows.

$$\dfrac{q_0 \to_\tau q' \qquad \dfrac{\dfrac{\dfrac{p_0 \to_c p_2}{p_0 \underline{\#}_b q'}}{q' \underline{\#}_b p_0} \qquad \dfrac{\text{trivial}}{\forall p', p''(p_0 \twoheadrightarrow_\tau p' \to_\tau p'' \implies q_0 \underline{\#}_b p' \vee q' \underline{\#}_b p'')}}{q_o \underline{\#}_b p_0}}{}$$

To remedy the non-congruence, the notion of *rooted branching bisimulation* has been introduced.

Definition 3.31 [GW96]. A relation R on an LTS is a *rooted branching bisimulation* if the following properties hold.

- R is a branching bisimulation.
- R satisfies the rule (bis_{rb}), where x ranges over A_τ,

$$\dfrac{q \to_x q' \qquad R(q,p)}{\exists p'(p \to_x p' \wedge q' \underline{\leftrightarrow}_b p')} \; \text{bis}_{rb}$$

The states q and p are *rooted branching bisimilar*, notation $q \underline{\leftrightarrow}_{rb} p$ if there is a rooted branching bisimulation R such that $R(q,p)$.

As a complement we define that the relation Q on an LTS is a *rooted branching apartness* if the following properties hold.

- Q is a branching apartness.
- Q satisfies the rule (in_{rb}), where x ranges over A_τ,

$$\dfrac{q \to_x q' \qquad \forall p'(p \to_x p' \implies q' \underline{\#}_b p')}{Q(q,p)} \; \text{in}_{rb}$$

The states q and p are *rooted branching apart*, notation $q \underline{\#}_{rb} p$ if for all rooted branching apartness relations Q, we have $Q(q,p)$.

It is easy to see that $q \underline{\leftrightarrow}_{rb} p$ if and only if $\neg(q \underline{\#}_b p)$ and that $q \underline{\#}_b p$ if and only if this is derivable using the rules $(\underline{\#}_b)$, $(\text{in}_{b\tau})$ and (in_{rb}).

Example 3.32. As a continuation of Example 3.30, we can now show that both for the two LTSs on the right and also for the two LTSs on the left, we have $q_0 \underline{\#}_{rb} p_0$. For the pair on the left, the derivation is simply as follows.

$$\dfrac{q_0 \to_\tau q' \qquad \dfrac{\text{trivial}}{\forall p'(p_0 \to_\tau p' \implies q' \underline{\#}_b p')}}{q_0 \underline{\#}_{rb} p_0}$$

Thereby, congruence has been regained, by refining branching bisimulation (or strengthening branching apartness).

To understand how congruence is regained in general and relate that to strong extensionality, assume we have an operation $+$ on LTSs that represents non-deterministic choice, which means to join the root-nodes of two LTSs. So process terms of the form $q + p$ have the following behavior as LTS.

$$\frac{p_1 \to_x p_1'}{p_1 + p_2 \to_x p_1'} \qquad \frac{p_2 \to_x p_2'}{p_1 + p_2 \to_x p_2'}$$

The interpretation of $q + p$ as an LTS should be understood as the union of the LTSs arising from q and p that are joined at their root node. That rooted branching bisimulation is a congruence for the $+$ operator means that, if $q_1 \underline{\leftrightarrow}_{rb} p_1$ and $q_2 \underline{\leftrightarrow}_{rb} p_2$, then $(q_1 + q_2) \underline{\leftrightarrow}_{rb} (p_1 + p_2)$. We will look into this property from the "apartness side", where we want $+$ to be strongly extensional for $\underline{\#}_{rb}$: if $(q_1 + q_2) \underline{\#}_{rb} (p_1 + p_2)$, then $q_1 \underline{\#}_{rb} p_1 \vee q_2 \underline{\#}_{rb} p_2$. We can prove strong extensionality of $+$ as follows. Assume that $(q_1 + q_2) \underline{\#}_{rb} (p_1 + p_2)$. This means that there is a state r and $x \in A_\tau$ with $(q_1 + q_2) \to_x r$ and $\forall r'((p_1 + p_2) \to_x r' \implies r \underline{\#}_b r')$. The transition $(q_1 + q_2) \to_x r$ either arises from $q_1 \to_x r$ or from $q_2 \to_x r$. In case $q_1 \to_x r$ we have the derivation below. In case $q_2 \to_x r$ we have a symmetric situation that we don't depict here.

$$\frac{\dfrac{q_1 \to_x r}{(q_1 + q_2) \to_x r} \qquad \forall r'((p_1 + p_2) \to_x r' \implies r \underline{\#}_b r')}{(q_1 + q_2) \underline{\#}_{rb} (p_1 + p_2)}$$

So we have $\forall r'((p_1 + p_2) \to_x r' \implies r \underline{\#}_b r')$, which implies $\forall r'(p_1 \to_x r' \implies r \underline{\#}_b r')$, which, together with $q_1 \to_x r$, using rule (in_{rb}) gives $q_1 \underline{\#}_{rb} p_1$. In case $q_2 \to_x r$ we have a symmetric situation, so we can conclude $q_1 \underline{\#}_{rb} p_1 \vee q_2 \underline{\#}_{rb} p_2$.
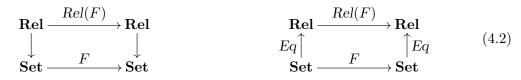
4. Coalgebras and lifting

This section recalls some basic facts about the description of bisimulations on coalgebras in terms of lifting of the functor from sets to relations.

We write **Rel** for the category of binary relations $R \subseteq X \times X$. A morphism $f \colon (R \subseteq X \times X) \to (S \subseteq Y \times Y)$ in **Rel** is a function $f \colon X \to Y$ between the underlying sets satisfying $(x_1, x_2) \in R \implies (f(x_1), f(x_2)) \in S$. This can equivalently be expressed via the existence of a function f' in:

$$
\begin{array}{ccc}
R & \xdashrightarrow{\;\;f'\;\;} & S \\
{\scriptstyle \langle r_1, r_2 \rangle} \downarrow & & \downarrow {\scriptstyle \langle s_1, s_2 \rangle} \\
X \times X & \xrightarrow{\;f \times f\;} & Y \times Y
\end{array}
\tag{4.1}
$$

We can also describe this situation via an inclusion $R \subseteq (f \times f)^{-1}(S)$, where $(f \times f)^{-1}(S) = \{(x_1, x_2) \mid (f(x_1), f(x_2)) \in S\}$. There is an obvious functor **Rel** $\to$ **Set** which sends a relation $R \subseteq X \times X$ to its underlying set X. The poset $\mathcal{P}(X \times X)$ of relations on a set X is often called the "fibre over X", since it mapped by this functor to X.

In this setting we restrict ourselves to (endo)functors $F \colon$ **Set** $\to$ **Set**, with associated category $CoAlg(F)$ of coalgebras. There is a standard way to "lift" such a functor F from **Set** to **Rel** in a commuting diagram, as on the left below.

$$
\begin{array}{ccc}
\mathbf{Rel} & \xrightarrow{\;Rel(F)\;} & \mathbf{Rel} \\
\downarrow & & \downarrow \\
\mathbf{Set} & \xrightarrow{\;F\;} & \mathbf{Set}
\end{array}
\qquad\qquad
\begin{array}{ccc}
\mathbf{Rel} & \xrightarrow{\;Rel(F)\;} & \mathbf{Rel} \\
{\scriptstyle Eq}\uparrow & & \uparrow{\scriptstyle Eq} \\
\mathbf{Set} & \xrightarrow{\;F\;} & \mathbf{Set}
\end{array}
\tag{4.2}
$$

For $R \subseteq X \times X$ one obtains $Rel(F)(R) \subseteq F(X) \times F(X)$ via:

$$Rel(F)(R) := \{(u_1, u_2) \mid \exists w \in F(R). \, F(r_1)(w) = u_1 \text{ and } F(r_2)(w) = u_2\}. \qquad (4.3)$$
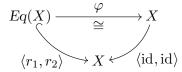
Here we write the inclusion map $R \hookrightarrow X \times X$ as a pair $\langle r_1, r_2 \rangle \colon R \to X \times X$.

It is not hard to see that $Rel(F)$ is functorial: for a morphism $f \colon (R \subseteq X \times X) \to (S \subseteq Y \times Y)$ in **Rel** there is a (unique) map $f' \colon R \to S$ as in (4.1). We show that $F(f)$ is a morphism $Rel(f)(R) \to Rel(F)(S)$. So let $(u_1, u_2) \in Rel(F)(R)$, say via $w \in F(R)$ with $F(r_i)(w) = u_i$. Then $w' := F(f')(w) \in F(S)$ satisfies $F(s_i)(w') = F(s_i \circ f')(w) = F(f \circ r_i)(w) = F(f)(u_i)$. This shows that the pair $(F(f)(u_1), F(f)(u_2))$ is in $Rel(F)(S)$.

A *bisimulation* for a coalgebra $c \colon X \to F(X)$ is a relation $R \subseteq X \times X$ for which c is a map in the category **Rel** of the form $c \colon R \to Rel(F)(R)$. Thus we may consider the category of coalgebras $CoAlg\big(Rel(F)\big)$ as the category of bisimulations — for F-coalgebras.

Lemma 4.1. *Relation lifting commutes with equality, as in the diagram on the right in (4.2), where* $Eq(X) = \{(x,x) \mid x \in X\} \subseteq X \times X$.

Proof. There is an obvious isomorphism φ in:

$$
\begin{array}{ccc}
Eq(X) & \xrightarrow{\;\;\varphi\;\;} & X \\
 & \cong & \\
\langle r_1, r_2 \rangle \searrow \;\; X \;\; \swarrow \langle \mathrm{id}, \mathrm{id} \rangle & &
\end{array}
$$

In particular, $r_1 = r_2$.

Hence if $(u_1, u_2) \in Rel(F)(Eq(X))$, say via $w \in F(Eq(X))$ with $F(r_i) = u_i$, then $u_1 = u_2$ since $r_1 = r_2$. In the other direction, for $u \in F(X)$ we have $(u, u) \in Rel(F)(Eq(X))$ via $w = F(\varphi^{-1})(u) \in F(Eq(X))$ that satisfies $F(r_i)(w) = F(\mathrm{id})(u) = u$. $\qquad \square$

For (Kripke) polynomial functors the generic form of relation lifting (4.3) specializes to well-known formulas, see [Jac16] for details.

Lemma 4.2. *Relation lifting satisfies:*

(1) $Rel(\mathrm{id}) = \mathrm{id}$;
(2) $Rel(A) = Eq(A)$, *where A on the left is the constant-A functor;*
(3) $Rel(F_1 \times F_2)(R) = \{(u, v) \mid (\pi_1 u, \pi_1 v) \in Rel(F_1)(R) \text{ and } (\pi_2 u, \pi_2 v) \in Rel(F_1)(R)\}$;
(4) $Rel(F_1 + F_2)(R) = \{(\kappa_1 u, \kappa_1 v) \mid (u, v) \in Rel(F_1)(R)\} \cup \{(\kappa_2 u, \kappa_2 v) \mid (u, v) \in Rel(F_2)(R)\}$;
(5) $Rel(F^A)(R) = \{(f, g) \mid \forall a \in A. \, (f(a), g(a)) \in Rel(F)(R)\}$;
(6) $Rel(\mathcal{P})(R) = \{(U, V) \mid \forall x \in U. \exists y \in V. \, (x, y) \in R \text{ and } \forall y \in V. \exists x \in U. \, (x, y) \in R\}$. $\quad \square$

Example 4.3. We shall elaborate what it means that a coalgebra $c \colon Y \to F(Y)$ also forms a coalgebra $c \colon R \to Rel(F)(R)$, for a relation $R \subseteq Y \times Y$. We shall do so for three different instantiations of the functor F. This gives uniform description of the same examples used in Section 2.

(1) Let F be the functor $F(X) = A \times X$ for streams, as in Definition 2.1. We can then describe the coalgebra $c \colon Y \to A \times Y$ as a pair $c = \langle h, t \rangle$ for a "head" function $h \colon Y \to A$ and a "tail" function $t \colon Y \to Y$. The lifting is:

$$
\begin{aligned}
Rel(F)(R) &= \{((a_1, y_1), (a_2, y_2)) \mid a_1 = a_2 \text{ and } R(y_1, y_2)\} \\
&= \{((a, y_1), (a, y_2)) \mid R(y_1, y_2)\}.
\end{aligned}
$$

Thus, $\langle h, t \rangle$ being a map $R \to Rel(F)(R)$ in **Rel** corresponds to the usual definition of bisimulation:

$$R(x_1, x_2) \implies h(x_1) = h(x_2) \text{ and } R\big(t(x_1), t(x_2)\big).$$

(2) For deterministic automata one uses the functor $F(X) = X^A \times 2$, where $2 = \{0, 1\}$, as in Definition 2.2. A coalgebra is again a tuple $c = \langle c_1, c_2 \rangle \colon Y \to Y^A \times 2$, with the following standard notation:

$$y \to_a y' \Leftrightarrow c_1(y)(a) = y' \qquad \text{and} \qquad y \downarrow \Leftrightarrow c_2(y) = 0.$$

The notation $y \downarrow$ denotes that y is a final state. Having a coalgebra $\langle c_1, c_2 \rangle \colon R \to Rel(F)(R)$ now means:

$$R(x_1, x_2) \implies \begin{cases} x_1 \downarrow \text{ iff } x_2 \downarrow \\ \quad \text{and} \\ \forall a_1, a_2. \forall y_1, y_2. x_1 \to_{a_1} y_1 \;\&\; x_2 \to_{a_2} y_2 \Rightarrow a_1 = a_2 \;\&\; R(y_1, y_2). \end{cases}$$

(3) We now use a functor $F(X) = \mathcal{P}\big(X + X \times A \times X\big)$ and investigate the associated form of bisimulation. We show that it resembles the formulation that we have seen earlier for weaker forms of bisimulation.

So lets start with a coalgebra $c \colon Y \to \mathcal{P}\big(Y + Y \times A \times Y\big)$ and write:

$$x \to_1 y \Leftrightarrow \kappa_1 y \in c(x) \qquad \text{and} \qquad x \to_2 y \to_a z \Leftrightarrow \kappa_2(y, a, z) \in c(x).$$

Here we regard $\to_1$ and $\to_2$ as two different forms of silent steps. There is a coalgebra $c \colon R \to Rel(F)(R)$ when:

$$R(x_1, x_2) \implies \begin{cases} x_1 \to_1 y_1 \implies \exists y_2. x_2 \to_1 y_2, \quad \text{and} \\ x_1 \to_2 y_1 \to_a z_1 \implies \exists y_2, z_2. x_2 \to_2 y_2 \to_a z_2 \\ \quad \text{and} \\ x_2 \to_1 y_2 \implies \exists y_1. x_1 \to_1 y_1, \quad \text{and} \\ x_2 \to_2 y_2 \to_a z_2 \implies \exists y_1, z_1. x_1 \to_2 y_1 \to_a z_1. \end{cases}$$

These clauses show that with this categorical way of capturing silent steps (via $\to_1$ and $\to_2$) means that the numbers of silent steps in both coordinates must be equal. This differs in an important way from weak/branching bisimulation where a single silent step on one coordinate can be mimicked by multiple (zero or more) silent steps in the other coordinate. We refer to the literature for different ways of bringing the two approaches closer togeter [SdVW09, BK17, Bre15, BMP15, GP14].

The next result gives a concrete description of what is captured abstractly in [HJ98].

Proposition 4.4. *The equality functor* $Eq \colon \mathbf{Set} \to \mathbf{Rel}$ *restricts to* $Eq \colon CoAlg(F) \to CoAlg(Rel(F))$, *for each functor* F, *and preserves final coalgebras. This implies that bisimilar elements become equal when mapped to the final coalgebra.*

Proof. Let $c \colon X \to F(X)$ be an arbitrary coalgebra. Applying the equality functor to it yields a $Rel(F)$-coalgebra:

$$Eq(X) \xrightarrow{\quad c \quad} Eq(F(X)) = Rel(F)(Eq(X)).$$

Now let $\zeta\colon Z \xrightarrow{\cong} F(Z)$ be the final F-coalgebra. Let $c\colon R \to Rel(F)(R)$ be a coalgebra, so that R is a bisimulation on $c\colon X \to F(X)$. We write X/R for the quotient of X under (the least equivalence relation containing) R, with quotient map $q_R\colon X \to X/R$. Then $q_R\colon R \hookrightarrow Eq(X/R)$ in **Rel**, giving:

$$R \xrightarrow{\ c\ } Rel(F)(R) \xrightarrow{\ F(q_R)\ } Rel(F)\big(Eq(X/R)\big) = Eq(F(X/R)).$$

This means that there is a unique coalgebra $c/R\colon X/R \to F(X/R)$ with $c/R \circ q_R = F(q_R) \circ c$. As a result, the unique coalgebra homomorphism g from c to the final coalgebra ζ factors as:

$$
\begin{array}{ccccc}
F(X) & \xrightarrow{\ F(q_R)\ } & F(X/R) & \dashrightarrow^{F(f)} & F(Z) \\
{\scriptstyle c}\big\uparrow & & {\scriptstyle c/R}\big\uparrow & & {\scriptstyle \zeta}\big\uparrow{\scriptstyle \cong} \\
X & \xrightarrow[\ q_R\ ]{} & X/R & \dashrightarrow_{f} & Z \\
& \multicolumn{3}{c}{\xrightarrow[\qquad\qquad g \qquad\qquad]{}} &
\end{array}
$$

Then, for $(x, x') \in R$, $q_R(x) = q_R(x')$, and therefore $g(x) = g(x')$. This means that g is a $Rel(F)$-coalgebra homomorphism $R \to Eq(Z)$. By finality of $\zeta\colon Z \xrightarrow{\cong} F(Z)$ there can be at most one such homomorphism. $\qquad\square$

We conclude this section with the following observation. As we have seen, R is a bisimulation for a coalgebra c when:

$$R \subseteq (c \times c)^{-1}\big(Rel(F)(R)\big).$$

Thus, the bisimilarity $\underline{\leftrightarrow}^c$ — that is, the greatest bisimulation on c — can be obtained as the greatest post-fixed point (final coalgebra) of the monotone operator $R \longmapsto (c \times c)^{-1}\big(Rel(F)(R)\big)$.

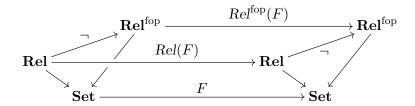## 5. Apartness

The opposite **Rel**$^{\mathrm{op}}$ of the category of relations contains relations as objects with reversed arrows. We are going to use a different category **Rel**$^{\mathrm{fop}}$ which is the "fibred opposite", where the order relations in the fibres are reversed. This is an instance of a more general construction [Jac99, Defn. 1.10.11].

Definition 5.1. The category **Rel**$^{\mathrm{fop}}$ has binary relations as objects. A morphism $f\colon (R \subseteq X \times X) \to (S \subseteq Y \times Y)$ is a function $f\colon X \to Y$ satisfying $R \supseteq (f \times f)^{-1}(S)$. This means that $(f(x_1), f(x_2)) \in S$ implies $(x_1, x_2) \in R$. There is an obvious forgetful functor **Rel**$^{\mathrm{fop}} \to$ **Set**, given by $(R \subseteq X \times X) \mapsto X$.

(1) We shall write $\neg\colon$ **Rel** $\to$ **Rel**$^{\mathrm{fop}}$ for the negation functor, where $\neg R = \{(x, x') \mid (x, x') \notin R\}$. On morphisms we have $\neg(f) = f$.
(2) We write $nEq := \neg \circ Eq\colon$ **Set** $\to$ **Rel**$^{\mathrm{fop}}$ for the inequality functor, sending a set X to $nEq(X) = \{(x, x') \mid x \neq x'\} \subseteq X \times X$.

(3) For a functor $F\colon \mathbf{Set} \to \mathbf{Set}$ we define 'opposite relation lifting' $Rel^{\mathrm{fop}}(F) \coloneqq \neg \circ Rel(F) \circ \neg\colon \mathbf{Rel}^{\mathrm{fop}} \to \mathbf{Rel}^{\mathrm{fop}}$. Then we have a situation:

It is easy to see that $\left(\mathbf{Rel}^{\mathrm{fop}}\right)^{\mathrm{fop}} = \mathbf{Rel}$ and that $\neg$ is a functor $\mathbf{Rel} \to \mathbf{Rel}^{\mathrm{fop}}$: if $R \subseteq (f \times f)^{-1}(S)$ then:

$$\neg R \supseteq \neg(f \times f)^{-1}(S) = (f \times f)^{-1}(\neg S).$$

Moreover, inequality is a functor since for $f\colon X \to Y$ one has $f(x) \neq f(x') \Longrightarrow x \neq x'$, that is, $nEq(X) \supseteq (f \times f)^{-1}(nEq(Y))$. Similarly, $\neg$ can be described as a functor $\neg\colon \mathbf{Rel}^{\mathrm{fop}} \to \mathbf{Rel}$. Clearly, the composite $\neg \circ \neg\colon \mathbf{Rel} \to \mathbf{Rel}^{\mathrm{fop}} \to \mathbf{Rel}$ is the identity functor, and similarly for $\neg \circ \neg\colon \mathbf{Rel}^{\mathrm{fop}} \to \mathbf{Rel} \to \mathbf{Rel}^{\mathrm{fop}}$.

Since $Rel(F)$ commutes with equality, see Lemma 4.1, the opposite relation lifting $Rel^{\mathrm{fop}}(F)$ commutes with inequality:

$$
\begin{aligned}
Rel^{\mathrm{fop}}(F)\big(nEq(X)\big) \;=\; \neg Rel(F)\big(\neg\neg Eq(X)\big) &\;=\; \neg Rel(F)\big(Eq(X)\big) \\
&\;=\; \neg Eq(F(X)) \;=\; nEq(F(X)).
\end{aligned}
$$

We have the following analogue of Lemma 4.2 for opposite relation lifting.

Lemma 5.2. *Relation lifting satisfies:*

(1) $Rel^{\mathrm{fop}}(\mathrm{id}) = \mathrm{id}$;

(2) $Rel^{\mathrm{fop}}(A) = nEq(A)$;

(3) $Rel^{\mathrm{fop}}(F_1 \times F_2)(R) = \{(u,v) \mid (\pi_1 u, \pi_1 v) \in Rel^{\mathrm{fop}}(F_1)(R) \ or \ (\pi_2 u, \pi_2 v) \in Rel^{\mathrm{fop}}(F_1)(R)\}$;

(4) $Rel^{\mathrm{fop}}(F_1 + F_2)(R) = \{(\kappa_1 u, \kappa_2 v) \mid u \in F_1(X), v \in F_2(X)\} \ \cup$

$$
\begin{aligned}
&\{(\kappa_1 u, \kappa_2 v) \mid u \in F_1(X), v \in F_2(X)\} \ \cup \\
&\{(\kappa_1 u, \kappa_2 v) \mid (u,v) \in Rel^{\mathrm{fop}}(F_1)(R)\} \ \cup \\
&\{(\kappa_1 u, \kappa_2 v) \mid (u,v) \in Rel^{\mathrm{fop}}(F_2)(R)\} \\
=\; &\{(a,b) \mid \forall u,v.\ a = \kappa_1 u, b = \kappa_1 v \Rightarrow (u,v) \in Rel^{\mathrm{fop}}(F_1)(R) \ and \\
&\qquad\qquad\qquad a = \kappa_2 u, b = \kappa_2 v \Rightarrow (u,v) \in Rel^{\mathrm{fop}}(F_2)(R)\}
\end{aligned}
$$

(5) $Rel^{\mathrm{fop}}(F^A)(R) = \{(f,g) \mid \exists a \in A.\ (f(a), g(a)) \in Rel^{\mathrm{fop}}(F)(R)\}$;

(6) $Rel^{\mathrm{fop}}(\mathcal{P})(R) = \{(U,V) \mid \exists x \in U.\forall y \in V.\ (x,y) \in R \ and \ \exists y \in V.\forall x \in U.\ (x,y) \in R\}$. $\square$

Example 5.3. We look at the analogue of Example 4.3, using apartness instead of bisimulation.

(1) A relation $R \subseteq Y \times Y$ is an apartness relation for the functor $F(X) = A \times X$ when there is a coalgebra $\langle h, t \rangle\colon R \to Rel^{\mathrm{fop}}(F)(R)$ in $\mathbf{Rel}^{\mathrm{fop}}$. This amounts to:

$$h(x_1) \neq h(x_2) \Longrightarrow R\big(x_1, x_2\big) \qquad \text{and} \qquad R\big(t(x_1), t(x_2)\big) \Longrightarrow R\big(x_1, x_2\big).$$

Apartness $\#$ is the least relation R for which these two implications hold. In such a situation one commonly writes these implications as rules:

$$\frac{h(x_1) \neq h(x_2)}{x_1 \mathrel{\underline{\#}} x_2} \qquad\qquad \frac{t(x_1) \mathrel{\underline{\#}} t(x_2)}{x_1 \mathrel{\underline{\#}} x_2}$$

Alternatively, $x_1 \mathrel{\underline{\#}} x_2$ iff $h\big(t^n(x_1)\big) \neq h\big(t^n(x_2)\big)$ for some $n \in \mathbb{N}$.

(2) For a deterministic automaton, R is an apartness relation when:

$$\left.\begin{array}{c} x_1 \downarrow \ \& \ \neg\big(x_2 \downarrow\big) \\[4pt] \text{or} \\[4pt] x_2 \downarrow \ \& \ \neg\big(x_1 \downarrow\big) \\[4pt] \text{or} \\[4pt] \text{for some } a, \ x_1 \to_a y_1 \ \& \ x_2 \to_a y_2 \ \& \ R\big(y_1, y_2\big) \end{array}\right\} \implies R\big(x_1, x_2\big).$$

(3) For an apartness relation R for the functor $F(X) = \mathcal{P}\big(X + X \times A \times X\big)$ there are many cases to distinghuish: one has $R(x_1, x_2)$ if either:

- $x_1 \to_1 y_1$, but there is no y_2 with $x_2 \to_1 y_2$;
- $x_1 \to_1 y_1$ and $x_2 \to_1 y_2$ with $R(y_1, y_2)$;
- $x_1 \to_2 y_1 \to_{a_1} z_1$, but there are no y_2, a_2, z_2 with $x_2 \to_2 y_2 \to_{a_2} z_2$;
- $x_1 \to_2 y_1 \to_a z_1$, but there are no y_2, z_2 with $x_2 \to_2 y_2 \to_a z_2$;
- $x_1 \to_2 y_1 \to_a z_1$ and $x_2 \to_2 y_2 \to_a z_2$ with $R(y_1, y_2)$;
- $x_1 \to_2 y_1 \to_a z_1$ and $x_2 \to_2 y_2 \to_a z_2$ with $R(z_1, z_2)$,

and similarly for the six symmetric cases, starting with x_2.

Definition 5.4. Let $c\colon X \to F(X)$ be an arbitrary coalgebra.

(1) An *apartness relation* for c is a relation R on X for which c is a coalgebra $c\colon R \to Rel^{\mathrm{fop}}(F)(R)$. This means that:

$$R \supseteq (c \times c)^{-1}\big(Rel^{\mathrm{fop}}(F)(R)\big) \quad \text{or, equivalently} \quad \neg R \subseteq (c \times c)^{-1}\big(Rel(F)(\neg R)\big).$$

Thus, R is an apartness relation iff $\neg R$ is a bisimulation relation.

(2) *Apartness* $\#^c$ is the greatest apartness relation, w.r.t. the order $\supseteq$.

The category $CoAlg(Rel^{\mathrm{fop}}(F))$ thus has apartness relations as objects. In order to find the apartness relation on a coalgebra $c\colon X \to F(X)$ we need to find the *greatest post-fixed point* (final coalgebra) of the mapping $R \longmapsto (c \times c)^{-1}\big(Rel(F)(\neg R)\big)$ in the poset $(\mathcal{P}(X \times X), \supseteq)$ with opposite order. A crucial observation is that this is the *least pre-fixed point* (initial algebra) in $\mathcal{P}(X \times X)$ with usual inclusion order $\subseteq$. Elements of such an initial algebra can typically be constructed in a finite number of steps. This corresponds to the idea that finding a difference in behaviour of coalgebra can be done in finitely many steps — although you may not know how many steps — whereas showing indistinguishability of behaviour involves all steps.

Proposition 5.5. *The inequality functor* $nEq\colon \mathbf{Set} \to \mathbf{Rel}^{\mathrm{fop}}$ *restricts to* $nEq\colon CoAlg(F) \to CoAlg(Rel^{\mathrm{fop}}(F))$, *for each functor* F, *and preserves final coalgebras. This implies that elements that are apart become non-equal (different) when mapped to the final coalgebra.*

Proof. We assume a final F-coalgebra $\zeta\colon Z \to F(Z)$. Let $c\colon R \to Rel^{\mathrm{fop}}(F)(R)$ describe an apartness relation $R \subseteq X \times X$, for a coalgebra $c\colon X \to F(X)$. The latter has a unique

coalgebra homomorphism $f \colon X \to Z$. Since $\neg R$ is a bisimulation, Proposition 4.4 says that we have a map $f \colon \neg R \to Eq(Z)$ in **Rel**. By functoriality of $\neg$ we get $f \colon R \to \neg Eq(Z) = nEq(Z)$ in $\mathbf{Rel}^{\mathrm{fop}}$, as required. $\square$

6. Conclusion and Further directions

In this paper we have explored the notion of "apartness" from a coalgebraic perspective, as the negation of bisimulation. We have shown what this means concretely in the simple cases of streams and deterministic automata and in the cases of weak and branching bisimulation for labelled transition systems. We have also given a general categorical treatment of apartness as the negation of bisimulation. An important contribution of this view is that it yields a logic for proving that two states are apart, proving that they are not bisimilar. This applies to the general situation for coalgebras of a polynomial functor, but also to the specific situation of weak and branching bisimulation.

It would be interesting to see if apartness can be applied in the analysis or description of algorithms for computing bisimulation, in particular branching bisimulation [JGKW20, GV90]. In existing algorithms, apartness clearly plays a role, as they are described in terms of a collection of "blocks" that is refined. States in different blocks are apart, so these algorithms refine an apartness until in the limit, the finest possible apartness is reached. Our approach is not intended to replace bisimulation with apartness. In fact, we view a combined approach as most promising, e.g. as we have in Lemma 3.29. In the period of submission and revision of the present paper, the concept of apartness has already been picked up and applied to develop a new approach to active automata learning [VGRW21], which underlines the potential fruitfulness of the concept.

Acknowledgments

References

[Bas96]   Twan Basten. Branching bisimilarity is an equivalence indeed. *Inform. Process. Lett*, pages 141–147, 1996.

[BBR10]   J. Baeten, T. Basten, and M. Reniers. *Process Algebra: Equational Theories of Communicating Processes*. Cambridge University Press, 2010.

[BK17]   Harsh Beohar and Sebastian Küpper. On path-based coalgebras and weak notions of bisimulation. In Filippo Bonchi and Barbara König, editors, *7th Conference on Algebra and Coalgebra in Computer Science, CALCO 2017, June 12-16, 2017, Ljubljana, Slovenia*, volume 72 of *LIPIcs*, pages 6:1–6:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[BMP15]   T. Brengos, M. Miculan, and M. Peressotti. Behavioural equivalences for coalgebras with unobservable moves. *Journ. of Logical and Algebraic Methods in Programming*, 84(6):826–852, 2015.

[Bre15]   T. Brengos. Weak bisimulation for coalgebras over order enriched monads. *Logical Methods in Comp. Sci.*, 11(2), 2015.

[Cho78]   T. S. Chow. Testing software design modeled by finite-state machines. *IEEE Trans. Softw. Eng.*, 4(3):178–187, May 1978.

[DV95]   Rocco De Nicola and Frits W. Vaandrager. Three logics for branching bisimulation. *J. ACM*, 42(2):458–487, 1995.

[Fok00]   Wan Fokkink. Rooted branching bisimulation as a congruence. *J. Comput. Syst. Sci.*, 60(1):13–37, 2000.

[GN00]   Herman Geuvers and Milad Niqui. Constructive reals in Coq: Axioms and categoricity. In Paul Callaghan, Zhaohui Luo, James McKinna, and Robert Pollack, editors, *Types for Proofs and Programs, International Workshop, TYPES 2000, Durham, UK, December 8-12, 2000, Selected Papers*, volume 2277 of *Lecture Notes in Computer Science*, pages 79–95. Springer, 2000.

[GP14]   S. Goncharov and D. Pattinson. Categorical views on computations on trees. In J. Esparza, P. Fraigniaud, T. Husfeldt, and E. Koutsoupias, editors, *Int. Coll. on Automata, Languages and Programming*, number 8573 in LNCS, pages 196–207. Springer, Berlin, 2014.

[GV90]   Jan Friso Groote and Frits W. Vaandrager. An efficient algorithm for branching bisimulation and stuttering equivalence. In Mike Paterson, editor, *Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, UK, July 16-20, 1990, Proceedings*, volume 443 of *Lecture Notes in Computer Science*, pages 626–638. Springer, 1990.

[GW96]   Rob Van Glabbeek and Peter Weijland. Branching time and abstraction in bisimulation semantics. *Jornal of the ACM*, 43:613–618, 1996.

[Hey27]   A. Heyting. Zur intuitionistischen Axiomatik der projektiven Geometrie. *Mathematische Annalen*, 98:491–538, 1927.

[HJ98]   C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Information & Computation*, 145:107–152, 1998.

[HM85]   M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161,, 1985.

[Jac99]   B. Jacobs. *Categorical Logic and Type Theory*. North Holland, Amsterdam, 1999.

[Jac16]   B. Jacobs. *Introduction to Coalgebra. Towards Mathematics of States and Observations*. Number 59 in Tracts in Theor. Comp. Sci. Cambridge Univ. Press, 2016.

[JGKW20]   David N. Jansen, Jan Friso Groote, Jeroen J. A. Keiren, and Anton Wijs. An O(m log n) algorithm for branching bisimilarity on labelled transition systems. In Armin Biere and David Parker, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 26th International Conference, TACAS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings, Part II*, volume 12079 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2020.

[JR11]   B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. In D. Sangiorgi and J. Rutten, editors, *Advanced topics in bisimulation and coinduction*, number 52 in Tracts in Theor. Comp. Sci., pages 38–99. Cambridge Univ. Press, 2011.

[Kor92]   Henri Korver. Computing distinguishing formulas for branching bisimulation. In Kim G. Larsen and Arne Skou, editors, *Computer Aided Verification (CAV)*, volume 575 of *Lecture Notes in Computer Science*, pages 13–23. Springer, 1992.

[Mil89]   Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[Rut00]   J. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 249:3–80, 2000.

[SdVW09]   Ana Sokolova, Erik P. de Vink, and Harald Woracek. Coalgebraic weak bisimulation for action-type systems. *Sci. Ann. Comp. Sci.*, 19:93–144, 2009.

[SMJ16]   Rick Smetsers, Joshua Moerman, and David N. Jansen. Minimal separating sequences for all pairs of states. In Adrian-Horia Dediu, Jan Janoušek, Carlos Martín-Vide, and Bianca Truthe, editors, *Language and Automata Theory and Applications (LATA)*, volume 9618 of *Lecture Notes in Computer Science*, pages 181–193. Springer, 2016.

[TvD88]   A.S. Troelstra and D. van Dalen. *Constructivism in Mathematics, volume II*. Number volume 123 in Studies in Logic and the Foundations of Mathematics. Elsevier Science, 1988.

[VGRW21]   Frits W. Vaandrager, Bharat Garhewal, Jurriaan Rot, and Thorsten Wißmann. A new approach for active automata learning based on apartness. *CoRR*, abs/2107.05419, 2021.

Appendix A. Syntax and logic for bisimulation and apartness

As we have seen, the treatment of bisimulation and apartness in Section 2 can be generalized using category theory. We could also have chosen for a purely logical-syntactic approach, which we briefly summarize here. As a matter of fact, the treatment in Sections 4 and 5 can be seen as a semantics of the syntax we introduce here.

We start with a description of the systems we want to deal with, which are basically a slight generalization of polynomial functors. A coalgebra has a carrier and a destructor operation. We describe the "types" of the possible range of a destructor.

Definition A.1. Let a set At of symbols fot atomic types be given and let X be a special symbol (a type variable) not in At.

$$\text{Ty} ::= \text{At} \mid X \mid 1 \mid \text{Ty} + \text{Ty} \mid \text{Ty} \times \text{Ty} \mid \text{Ty}^{\text{At}} \mid 2^{\text{Ty}}.$$

A *destructor signature* is a pair (d, σ) with

$$d : X \to \sigma.$$

A *coalgebra for the destructor signature* (d, σ) is a $K \in \mathbf{Set}$ and $h : K \to F(K)$ (the *destructor*), where F is the functor on $\mathbf{Set}$ defined from $X \mapsto \sigma(X)$ in the obvious way.

We have seen examples for this definition in 2.2, 2.1, 4.3 and 5.3.

Definition A.2. Let K be a coalgebra for the signature of Definition A.1 with destructor $h : K \to F(K)$.

(1) A relation $R \subseteq K \times K$ is called a *bisimulation* (or a *coinduction assumption*) on K if the following holds for all $x, y \in K$.

$$R(x, y) \Rightarrow h(x) \overset{F(K)}{\underset{R}{\leftrightarrow}} h(y).$$

Here, the relation $\overset{B}{\underset{R}{\leftrightarrow}}$ is defined by induction on the structure of B as follows.
- $B = D \in \text{At}$, then $x \overset{B}{\underset{R}{\leftrightarrow}} y := Eq_D(x, y)$, equality on D.
- $B = X$, then $x \overset{B}{\underset{R}{\leftrightarrow}} y := R(x, y)$.
- $B = 1$, then $x \overset{B}{\underset{R}{\leftrightarrow}} y$ is true.
- $B = B_1 + B_2$, then $x \overset{B}{\underset{R}{\leftrightarrow}} y :=$
  $\exists u, v(x = \text{inl}(u) \land y = \text{inl}(v) \land u \overset{B_1}{\underset{R}{\leftrightarrow}} v) \lor (x = \text{inr}(u) \land y = \text{inr}(v) \land u \overset{B_2}{\underset{R}{\leftrightarrow}} v)$.
- $B = B_1 \times B_2$, then $x \overset{B}{\underset{R}{\leftrightarrow}} y :=$
  $\exists u_1, u_2, v_1, v_2(x = \langle u_1, u_2 \rangle \land y = \langle v_1, v_2 \rangle \land u_1 \overset{B_1}{\underset{R}{\leftrightarrow}} v_1 \land u_2 \overset{B_2}{\underset{R}{\leftrightarrow}} v_2)$.
- $B = B_1^D$ with $D \in \text{At}$, then $x \overset{B}{\underset{R}{\leftrightarrow}} y := \forall d \in D(x(d) \overset{B_1}{\underset{R}{\leftrightarrow}} y(d))$.
- $B = 2^{B_1}$, then $x \overset{B}{\underset{R}{\leftrightarrow}} y := (\forall u(x(u) = 1 \implies \exists v(y(v) = 1 \land u \overset{B_1}{\underset{R}{\leftrightarrow}} v)) \land (\forall v(y(v) = 1 \implies \exists u(x(u) = 1 \land u \overset{B_1}{\underset{R}{\leftrightarrow}} v))$.

(2) Two elements $x, y \in K$ are called *bisimilar*, notation $x \underline{\leftrightarrow} y$, if there exists a bisimulation R for which $R(x, y)$ holds. Thus:

$$\underline{\leftrightarrow} = \bigcup \{ R \subseteq K \times K \mid R \text{ is a bisimulation} \}.$$

The condition that $R \subseteq K \times K$ should satisfy in order to be a bisimulation can be viewed as a *derivation rule*. This gives an alternative way to say that R is a bisimulation.

Remark A.3. The relation $R \subseteq K \times K$ is a bisimulation if the following derivation rules hold for R.

$$\frac{R(x,y)}{h(x) \overset{F(K)}{\underset{R}{\leftrightarrow}} h(y)}$$

Lemma A.4. *The relation $\leftrightarrow$ is itself a bisimulation relation, and therefore $\leftrightarrow$ is the largest bisimulation relation.*

Proof. This is a standard fact about bisimulations that is easily verified for this more general setting. The crucial propertyis that if $x \overset{B}{\leftrightarrow}_R y$ for some bisimulation R, then $x \overset{B}{\leftrightarrow}_{\leftrightarrow} y$. $\qquad\square$

Similar to the definition of bisimulation (Definition A.2), we can give a general definition of apartness for a coalgebra of a destructor signature.

Definition A.5. Let again K be a coalgebra for the signature of Definition A.1 with destructor $h : K \to F(K)$.

(1) A relation $Q \subseteq K \times K$ is called an *apartness* on K if the following holds for all $x, y \in K$.

$$h(x) \overset{F(K)}{\#}_Q h(y) \Rightarrow Q(x,y).$$

Here, the relation $\overset{B}{\#}_Q$ is defined by induction on the structure of B as follows.

- $B = D \in \mathrm{At}$, then $x \overset{B}{\#}_Q y := \neg Eq_D(x,y)$, in-equality on D.
- $B = X$, then $x \overset{B}{\#}_Q y := Q(x,y)$.
- $B = 1$, then $x \overset{B}{\#}_Q y$ is false.
- $B = B_1 + B_2$, then $x \overset{B}{\#}_Q y :=$

  $\forall u, v((x = \mathrm{inl}(u) \wedge y = \mathrm{inl}(v)) \Rightarrow u \overset{B_1}{\#}_Q v) \wedge ((x = \mathrm{inr}(u) \wedge y = \mathrm{inr}(v)) \Rightarrow u \overset{B_2}{\#}_Q v).$
- $B = B_1 \times B_2$, then $x \overset{B}{\#}_R y :=$

  $\forall u_1, u_2, v_1, v_2(x = \langle u_1, u_2 \rangle \wedge y = \langle v_1, v_2 \rangle) \Rightarrow u_1 \overset{B_1}{\#}_Q v_1 \vee u_2 \overset{B_2}{\#}_Q v_2).$
- $B = B_1^D$ with $D \in \mathrm{At}$, then $x \overset{B}{\#}_R y := \exists d \in D(x(d) \overset{B_1}{\#}_R y(d)).$
- $B = 2^{B_1}$, then $x \overset{B}{\#}_R y := (\exists u(x(u) = 1 \wedge \forall v(y(v) = 1 \implies u \overset{B_1}{\#}_R v)) \vee$

  $(\exists v(y(v) = 1 \wedge \forall u(x(u) = 1 \implies u \overset{B_1}{\#}_R v)).$
(2) Two elements $x, y \in K$ are called *apart*, notation $x \mathrel{\#} y$, if for all apartness relations Q we have $Q(x, y)$. Thus:

$$\mathrel{\#} = \bigcap \{Q \subseteq K \times K \mid Q \text{ is an apartness}\}.$$

Just as for bisimulation, we can phrase the property that a relation should satisfy in order to be an apartness in the form of aderivation rule.

Remark A.6. A relation $Q \subseteq K \times K$ is an apartness if the following derivation rule holds for Q.

$$\frac{h(x) \overset{F(K)}{\#}_Q h(y)}{Q(x,y)}$$

Lemma A.7. *The relation $\#$ is itself an apartness relation, and therefore $\#$ is the smallest apartness relation.*

Proof. That the rule of Remark A.6 is sound for $\#$ is easily verified by observing that, if $x \overset{B}{\#}_{\#} y$, then $x \overset{B}{\#}_{Q} y$ for any apartness Q. $\qquad\square$

As $x \# y$ is the smallest apartness relation, the rule of Remark A.6 gives a complete derivation system for proving $x \# y$ for $x, y \in K$.

We can summarize the relations between apartness and bisimulation as follows. This is the general statement of Lemma 2.6. The proof is simply checking all the properties.

Lemma A.8. (1) *R is a bisimulation if and only if $\neg R$ is an apartness.*
(2) *The relation $\underline{\leftrightarrow}$ is the union of all bisimulations, $\underline{\leftrightarrow} = \bigcup\{R \mid R \text{ is a bisimulation}\}$, and it is itself a bisimulation.*
(3) *The relation $\#$ is the intersection of all apartness relations, $\# = \bigcap\{Q \mid Q \text{ is an apartness}\}$, and it is itself an apartness.*
(4) *$\underline{\leftrightarrow} = \neg \#$.*