A Note on Game Characterizations of Strong and Weak Bisimilarity

Jiří Srba

BRICS[*]
Department of Computer Science, Aalborg University,
Fredrik Bajersvej 7B, 9220 Aalborg East, Denmark
srba@cs.aau.dk

Abstract These notes are intended for use in Semantics and Verification course at AAU. They deal with strong and weak bisimilarity defined over labelled transition systems and focus on the game characterization of bisimilarity. The motivation and justification of the notions is provided in the other reading material of the course. Our aim is to explain thoroughly the definitions of strong and weak bisimilarity and provide a selection of examples which document their use.

1 Introduction

Perhaps the most abstract process behaviour can be described as follows: a process p performs an action and becomes a process p'. Processes are considered as agents that can execute actions in order to communicate with their environment. These actions can be observed by an external observer via communications and determine the visible behaviour of the process.

This simple idea is formally captured by the notion of a *labelled transition system*. Transition system with labels is perhaps the most basic model of a process behaviour. Most of the formalisms within concurrency theory are given their semantics by means of labelled transition systems via so called *structural operational semantics* (or simply *SOS*) invented by Plotkin [5].

In labelled transition systems, processes are understood as nodes of certain edge-labelled oriented graphs (labelled transition systems) and a change of a process state caused by performing an action is understood as moving along an edge labelled by the action name.

A labelled transition system consists therefore of a set of *states* (or *processes* or *configurations*), a set of *labels* (or *actions*), and a transition relation $\longrightarrow$ describing a change of a process state: if a process p can perform an action a and become a process p', we write $p \stackrel{a}{\longrightarrow} p'$. Sometimes a state is singled out as the *start state* in the labelled transition system under consideration.

Example 1. Let us start with the classical example of a tea/coffee vending machine. The very simplified behaviour of the process which determines the interaction of the machine with a costomer can be described as follows. From the initial state representing the situation "waiting for a request", (let us call the

[*] **B**asic **R**esearch in **C**omputer **S**cience,
Centre of the Danish National Research Foundation.

state p), two possible actions are enabled. Either the tea button or the coffee button can be pressed (the corresponding action '*tea*' or '*coffee*' is executed) and the internal state of the machine changes accordingly to p_1 or p_2. Formally, this can be described by the transitions

$$p \xrightarrow{tea} p_1 \quad \text{and} \quad p \xrightarrow{coffee} p_2.$$

Now the customer is asked to insert the corresponding amount of money, let us say one euro for a cup of tea and two euros for a cup of coffee. This is reflected in the control state of the vending machine with corresponding changes. It can be modelled by the transitions

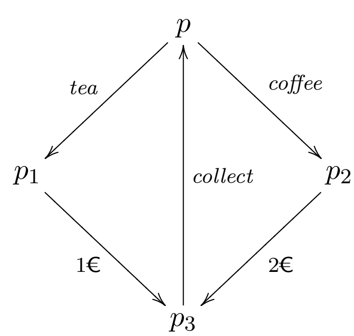
$$p_1 \xrightarrow{1€} p_3 \quad \text{and} \quad p_2 \xrightarrow{2€} p_3.$$

Finally, the drink is collected and the machine returns to its initial state p, ready to accept another customer. This corresponds to the transition

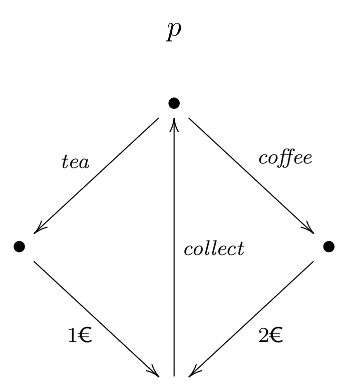
$$p_3 \xrightarrow{collect} p.$$

We shall often distinguish a so called *start state* (or *initial state*), which is one selected state in which the system initially starts (in our example the state p). □

It is convenient to use a graphical representation of labelled transition systems. The following picture represents the tea/coffee machine described above.



Sometimes, when referring only to the process p, we do not have to give names to the other process states (in our example p_1, p_2 and p_3) and it is sufficient to provide the following labelled transition system for the process p.



Remark 1. The definition of a labelled transitions systems allows situations like that in Figure 1 (where p is the initial state). This means that the state p_2 where

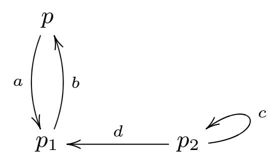


Figure 1. Labelled transition system with initial state p

the action c can be performed in a loop is irrelevant for the behaviour of the process p since p_2 can never be reached from p. This motivates us to introduce the notion of reachable states. We say that a state p' is *reachable* from p iff there exists an oriented path from p to p'. The set of all such states is called the *set of reachable states*. In our example this set contains exactly two states: p and p_1.

So far we were able to explicitly describe only finite-state processes (i.e. processes that have only finitely many states). We simply enumerate all the states and all the transitions. Transition systems can also be defined by using *process algebras*. The idea of process algebras is based on defining a set of *basic (atomic) processes* modelling very simple behaviours together with *composition operators* which enable us to define complex behaviours from the atomic processes by composing them in different ways. This main idea appears in many variations; let us mention e.g. the classical process algebras CCS [4], ACP [2] and CSP [3]. Process algebras enable us to define in a finite way labelled transition systems with infinitely many reachable states.

2 Labelled Transition Systems

We shall now formally define the notions discussed in the previous section.

Definition 1 (Labelled transition system).
A labelled transition system *is a triple* $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ *where*

- *Proc is a set of* states *(or* processes*),*
- *Act is a set of* labels *(or* actions*), and*
- *for every* $a \in Act$, $\xrightarrow{a} \subseteq Proc \times Proc$ *is a binary relation on states called the* transition relation*. We will use the infix notation* $s \xrightarrow{a} s'$ *meaning that* $(s, s') \in \xrightarrow{a}$.

We shall often distinguish a so called *start state* (or *initial state*), which is one selected state in which the system initially starts.

Sometimes the transition relation $\xrightarrow{a}$ is defined as a ternary relation $\longrightarrow \subseteq Proc \times Act \times Proc$ and we write $s \xrightarrow{a} s'$ whenever $(s, a, s') \in \longrightarrow$. This is an alternative way to define a labelled transition system and it gives exactly the same notion as Definition 1.

Remark 2. Let us now recall different notations that can be used in connection with labelled transitions systems.

– We can extend the transition relation to the elements of Act^* (all finite strings over Act including the empty string ϵ). The definition is as follows: $s \xrightarrow{\epsilon} s$ for every $s \in Proc$; and $s \xrightarrow{aw} s'$ iff there is $t \in Proc$ such that $s \xrightarrow{a} t$ and $t \xrightarrow{w} s'$ for every $s, s' \in Proc$, $a \in Act$ and $w \in Act^*$. In other words, if $w = a_1 a_2 \cdots a_n$ for $a_1, a_2 \ldots, a_n \in Act$ then we write $s \xrightarrow{w} s'$ whenever there exist states $s_1, \ldots, s_{n-1} \in Proc$ such that

$$s \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_3 \xrightarrow{a_4} \cdots \xrightarrow{a_{n-1}} s_{n-1} \xrightarrow{a_n} s'.$$

For the transition system in Figure 1 we have e.g. $p \xrightarrow{\epsilon} p$, $p \xrightarrow{ab} p$ and $p_1 \xrightarrow{bab} p$.

– We write $s \longrightarrow s'$ whenever there is an action $a \in Act$ such that $s \xrightarrow{a} s'$. For the transition system in Figure 1 we have e.g. $p \longrightarrow p_1$, $p_1 \longrightarrow p$, $p_2 \longrightarrow p_1$ and $p_2 \longrightarrow p_2$.

– We use the notation $s \xrightarrow{a}$ meaning that there is some $s' \in Proc$ such that $s \xrightarrow{a} s'$. We also write $s \stackrel{a}{\not\longrightarrow}$ whenever there is no $s' \in Proc$ such that $s \xrightarrow{a} s'$, and $s \not\longrightarrow$ whenever $s \stackrel{a}{\not\longrightarrow}$ for all $a \in Act$.
For the transition system in Figure 1 we have e.g. $p \xrightarrow{a}$, $p \stackrel{b}{\not\longrightarrow}$, $p_1 \xrightarrow{b}$, $p_1 \stackrel{c}{\not\longrightarrow}$.

– We write $s \longrightarrow^* s'$ iff $s \xrightarrow{w} s'$ for some $w \in Act^*$. In other words, $\longrightarrow^*$ is the reflexive and transitive closure of the relation $\longrightarrow$.
For the transition system in Figure 1 we have e.g. $p \longrightarrow^* p$, $p \longrightarrow^* p_1$, and $p_2 \longrightarrow^* p$.

Definition 2 (Reachable states).
Let $T = (Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ be a labelled transition system and $s \in Proc$ its initial state. We say that $s' \in Proc$ is reachable *in the transition system T iff $s \longrightarrow^* s'$. The set of* reachable states *contains all states reachable in T.*

In the transition system from Figure 1 where p is the initial state, the set of all reachable states is equal to $\{p, p_1\}$.

3 Strong Bisimilarity

In this section we shall define strong bisimilarity and introduce its game characterization.

Definition 3 (Strong bisimulation).
Let $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ be a labelled transition system. A binary relation $R \subseteq Proc \times Proc$ is a strong bisimulation *iff whenever $(s, t) \in R$ then for each $a \in Act$:*

– *if $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ for some $t' \in Proc$ such that $(s', t') \in R$, and*
– *if $t \xrightarrow{a} t'$ then $s \xrightarrow{a} s'$ for some $s' \in Proc$ such that $(s', t') \in R$.*

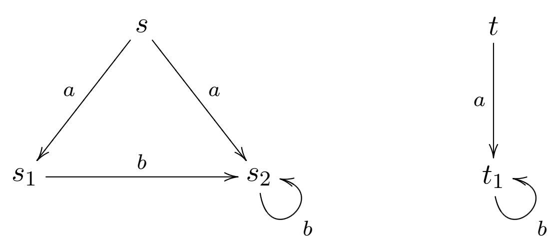
We say that two states $p_1, p_2 \in Proc$ are *strongly bisimilar*, and write $p_1 \sim p_2$, if and only if there exists a strong bisimulation R such that $(p_1, p_2) \in R$.

Several properties of the relation $\sim$ are mentioned in [1], in particular $\sim$ is an equivalence relation and it is the largest strong bisimulation. We call it *strong bisimilarity* or *strong bisimulation equivalence*.

Example 2. Consider a labelled transition system $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ where

- $Proc = \{s, s_1, s_2, t, t_1\}$
- $Act = \{a, b\}$
- $\xrightarrow{a} = \{(s, s_1), (s, s_2), (t, t_1)\}$ and $\xrightarrow{b} = \{(s_1, s_2), (s_2, s_2), (t_1, t_1)\}$.

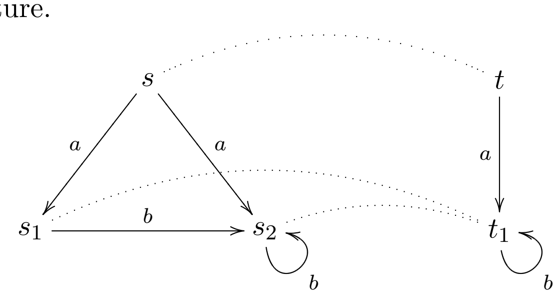
Here is a graphical representation of the transition system.



We will show that $s \sim t$. In order to do that, we have to define a strong bisimulation R such that $(s, t) \in R$. Let us define it as

$$R = \{(s, t), (s_1, t_1), (s_2, t_1)\}.$$

The binary relation R can be graphically depicted by dotted lines like in the following picture.



Obviously, $(s, t) \in R$. We have to show that R is a strong bisimulation, i.e., that it satisfies Definition 3. For every pair of states from R, we have to investigate all the possible transitions from both states and see whether they can be matched by corresponding transitions from the other state. Note that a transition under some label a can be matched only by a transition under the same label a. We will now do the complete analysis of all steps needed to show that R is a strong bisimulation, even though they are very simple and tedious.

- The pair (s, t):
  - transitions from s:
    - $s \xrightarrow{a} s_1$ can be matched by $t \xrightarrow{a} t_1$ and $(s_1, t_1) \in R$
    - $s \xrightarrow{a} s_2$ can be matched by $t \xrightarrow{a} t_1$ and $(s_2, t_1) \in R$

    * these are all the transitions from s
  - transitions from t:
    * $t \xrightarrow{a} t_1$ can be matched e.g. by $s \xrightarrow{a} s_2$ and $(s_2, t_1) \in R$ (another possibility would be to match it by $s \xrightarrow{a} s_1$ but finding one possibility is enough)
    * these are all the transitions from t
- The pair (s_1, t_1):
  - transitions from s_1:
    * $s_1 \xrightarrow{b} s_2$ can be matched by $t_1 \xrightarrow{b} t_1$ and $(s_2, t_1) \in R$
    * these are all the transitions from s_1
  - transitions from t_1:
    * $t_1 \xrightarrow{b} t_1$ can be matched by $s_1 \xrightarrow{b} s_2$ and $(s_2, t_1) \in R$
    * these are all the transitions from t_1
- The pair (s_2, t_1):
  - transitions from s_2:
    * $s_2 \xrightarrow{b} s_2$ can be matched by $t_1 \xrightarrow{b} t_1$ and $(s_2, t_1) \in R$
    * these are all the transitions from s_2
  - transitions from t_1:
    * $t_1 \xrightarrow{b} t_1$ can be matched by $s_2 \xrightarrow{b} s_2$ and $(s_2, t_1) \in R$
    * these are all the transitions from t_1

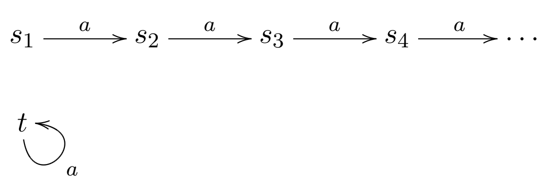
This completes the proof that R is a strong bisimulation and because $(s, t) \in R$ we get that $s \sim t$.

In order to prove that e.g. $s_1 \sim s_2$ we can use the following relation $R = \{(s_1, s_2), (s_2, s_2)\}$. The reader is invited to verify that R is a strong bisimulation. □

Example 3. In this example we shall demonstrate that it is possible that the initial state of a labelled transition system with infinitely many reachable states can be strongly bisimilar to a state from which only finitely many states are reachable. Consider $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ where

- $Proc = \{s_i \mid i \in \mathbb{N}\} \cup \{t\}$ where $\mathbb{N} = \{1, 2, 3, \ldots\}$ is the set of natural numbers
- $Act = \{a\}$
- $\xrightarrow{a} = \{(s_i, s_{i+1}) \mid i \in \mathbb{N}\} \cup \{(t, t)\}$.

Here is a graphical representation of the transition system.



We can now observe that $s_1 \sim t$ because

$$R = \{(s_i, t) \mid i \in \mathbb{N}\}$$

is a strong bisimulation and it contains the pair (s_1, t). The reader is invited to verify this simple fact. □

We can naturally ask the following question:

> *What techniques do we have to show that two states are* not *bisimilar?*

In order to prove that for two given states s and t it is the case that $s \not\sim t$, we should by Definition 3 enumerate all binary relations over the set of states and for each of them show that if it contains the pair (s, t) then it is not a strong bisimulation. For the transition system from Example 2 this translates to investigating 2^{25} different candidates and in general for a transition system with n states one would have to go through 2^{n^2} different binary relations. In what follows, we will introduce a game characterization of strong bisimilarity, which will enable us to determine much more effectively that two states are not strongly bisimilar.

The idea is that there are two players in the bisimulation game, called *'attacker'* and *'defender'*. The attacker is trying to show that two given states are not bisimilar while the defender aims to show the opposite. The formal definition follows.

Definition 4 (Strong Bisimulation Game).
Let $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ *be a labelled transition system. A* strong bisimulation game *starting from the pair of states* $(s_1, t_1) \in Proc \times Proc$ *is a two-player game of an* 'attacker' *and a* 'defender'*.*

The game is played in rounds *and configurations of the game are pairs of states from* $Proc \times Proc$*. In every round exactly one configuration is called* current*; initially the configuration* (s_1, t_1) *is the current one.*

In each round the players change the current configuration (s, t) *according to the following rules.*

1. *The attacker chooses either a left or right side of the current configuration* (s, t) *and an action* a *from* Act*.*
   - *If the attacker chose left then he has to perform a transition* $s \xrightarrow{a} s'$ *for some state* $s' \in Proc$*.*
   - *If the attacker chose right then he has to perform a transition* $t \xrightarrow{a} t'$ *for some state* $t' \in Proc$*.*
2. *In this step the defender must provide an answer to the attack made in the previous step.*
   - *If the attacker chose left then the defender plays on the right side and has to respond by making a transitions* $t \xrightarrow{a} t'$ *for some* $t' \in Proc$*.*
   - *If the attacker chose right then the defender plays on the left side and has to respond by making a transitions* $s \xrightarrow{a} s'$ *for some* $s' \in Proc$*.*
3. *The configuration* (s', t') *becomes the current configuration and the game continues by another round according to the rules described above.*

A *play* of the game is a maximal sequence of configurations formed by the players according to the rules described above, and starting from the initial configuration (s_1, t_1). Note that a bisimulation game can have many different plays according to the choices made by the attacker and the defender. The attacker can choose a side, an action and a transition. The defender's only choice is in selecting one of the available transitions that are labelled with the same action picked by the attacker.

We shall now define when a play is winning for the attacker and when for the defender.

A finite play is lost by the player who is stuck and cannot make a move from the current configuration (s, t) according to the rules of the game. Note that attacker loses only if both $s \not\longrightarrow$ and $t \not\longrightarrow$, i.e., there is no transition from both the left and the right side of the configuration. The defender loses if he has (on his side of the configuration) no available transition under the action selected by the attacker.

It can also be the case that none of the players is stuck in any configuration and the play is infinite. In this situation the defender is the winner of the play.

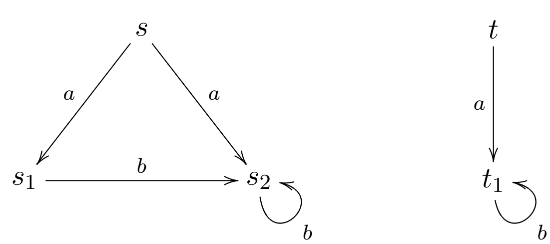
A given play is always winning either for the attacker or the defender and it cannot be winning for both at the same time.

The following proposition relates strong bisimilarity with the corresponding game characterization (see e.g. [6,7]).

Proposition 1. *States s_1 and t_1 of a labelled transition system are strongly bisimilar if and only if the defender has a universal winning strategy in the strong bisimulation game starting from the configuration (s_1, t_1). The states s_1 and t_1 are not strongly bisimilar if and only if the attacker has a universal winning strategy.*

By universal winning strategy we mean that the player can always win the game, irrelevant of how the other player is selecting his moves. In case that the opponent has more than one choice how to continue from the current configuration, all these possibilities have to be considered.

Example 4. Let us recall the transition system from Example 2.



We will show that the defender has a universal winning strategy from the configuration (s, t) and hence show that $s \sim t$. In order to do that, we have to consider all possible attacker's moves from this configuration and define defender's response to each of them. The attacker can make three different moves from (s, t):

1. attacker selects right side, action a and makes the move $t \xrightarrow{a} t_1$
2. attacker selects left side, action a and makes the move $s \xrightarrow{a} s_2$
3. attacker selects left side, action a and makes the move $s \xrightarrow{a} s_1$

– Defender's answer on attack 1. is by playing $s \xrightarrow{a} s_2$.
  (Even though there are more possibilities it is sufficient to provide only one.)
  The current configuration becomes (s_2, t_1).

– Defender's answer on attack 2. is by playing $t \xrightarrow{a} t_1$.
  The current configuration becomes again (s_2, t_1).
– Defender's answer on attack 3. is by playing $t \xrightarrow{a} t_1$.
  The current configuration becomes again (s_1, t_1).

Now it remains to show that the defender has a universal winning strategy from the configurations (s_2, t_1) and (s_1, t_1).

From (s_2, t_1) is easy to see that any continuation of the game will always go through the same current configuration (s_2, t_1) and hence the game will be necessarily infinite. According to the definition, the defender is the winner in this case.

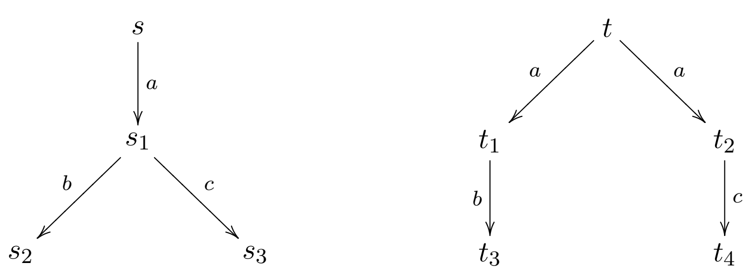
From (s_1, t_1) the attacker has two possible moves. Either $s_1 \xrightarrow{b} s_2$ or $t_1 \xrightarrow{b} t_1$. In the first case the defender answers by $t_1 \xrightarrow{b} t_1$ and in the second case by $s_1 \xrightarrow{b} s_2$. The next configuration is in both cases (s_2, t_1) and we already know that the defender has a winning strategy from this configuration.

Hence we showed that the defender has a universal winning strategy from the configuration (s, t) and according to Proposition 1 this means that $s \sim t$.

□

The game characterization of bisimilarity introduced above is simple, yet powerful. It provides an intuitive understanding of this notion. It can be used both to show that two states are strongly bisimilar as well as that they are not. The technique is particularly useful for showing non-bisimilarity of two states. This is demonstrated by the following examples.

Example 5. Let us consider the following transition system (we provide only its graphical representation).



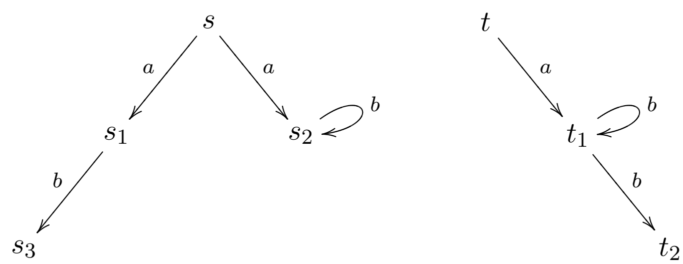
We will show that $s \not\sim t$ by describing a universal winning strategy for the attacker in the bisimulation game starting from (s, t). We will in fact show two different strategies (but of course finding one is sufficient for proving non-bisimilarity).

– In the first strategy, the attacker selects left side, action a and the transition $s \xrightarrow{a} s_1$. Defender can answer by $t \xrightarrow{a} t_1$ or $t \xrightarrow{a} t_2$. This means that we will have to consider two different configurations in the next round, namely (s_1, t_1) and (s_1, t_2). From (s_1, t_1) the attacker wins by playing $s_1 \xrightarrow{c} s_3$ on the left side and the defender cannot answer as there is no c-transition from t_1. From (s_1, t_2) the attacker wins by playing $s_1 \xrightarrow{b} s_2$ and the defender has again no answer from t_2. As we analyzed all different possibilities for the defender and in every one the attacker wins, we have found a universal winning strategy for the attacker and hence s and t are not bisimilar.

– Now we provide another strategy, which is easier to describe and involves switching of sides. Starting from (s,t) the attacker plays on the right side according to the transition $t \stackrel{a}{\longrightarrow} t_1$ and the defender can only answer by $s \stackrel{a}{\longrightarrow} s_1$ on the left side (no more configurations need to be examined as this is the only defender's possibility). The current configuration hence becomes (s_1,t_1). In the next round the attacker plays $s_1 \stackrel{c}{\longrightarrow} s_3$ and wins the game as $t_1 \stackrel{c}{\not\longrightarrow}$.

$\square$

Example 6. Let us consider a more complex transition system.



We will define attacker's universal winning strategy from (s,t) and hence show that $s \not\sim t$.

In the first round the attacker plays on left side the move $s \stackrel{a}{\longrightarrow} s_1$ and the defender can only answer by $t \stackrel{a}{\longrightarrow} t_1$. The current configuration becomes (s_1,t_1). In the second round the attacker plays on right side according to the transition $t_1 \stackrel{b}{\longrightarrow} t_1$ and the defender can only answer by $s_1 \stackrel{b}{\longrightarrow} s_3$. The current configuration becomes (s_3,t_1). Now the attacker wins by playing again the transition $t_1 \stackrel{b}{\longrightarrow} t_1$ (or $t_1 \stackrel{b}{\longrightarrow} t_2$) and the defender loses because $s_3 \not\longrightarrow$. $\square$

4 Weak Bisimilarity

We continue by defining the notion of weak bisimilarity and introduce the weak bisimulation game. The main idea is that weak bisimilarity abstracts away from internal behaviour of the systems, which is modelled by a special action τ.

Let $(Proc, Act, \{\stackrel{a}{\longrightarrow} \mid a \in Act\})$ be a labelled transition system such that Act contains a distinguished *silent* action τ. Actions from $Act \smallsetminus \{\tau\}$ will be called *visible* actions. We define a set of *weak transition relations* $\stackrel{a}{\Longrightarrow} \subseteq Proc \times Proc$ for all $a \in Act$ in such a way that $\stackrel{a}{\Longrightarrow}$ is the relation $\stackrel{a}{\longrightarrow}$ preceded and followed by an arbitrary number of τ transitions. Formally,

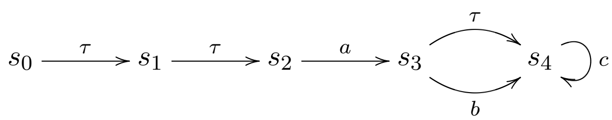
$$\stackrel{a}{\Longrightarrow} \stackrel{\text{def}}{=} \begin{cases} (\stackrel{\tau}{\longrightarrow})^* \circ \stackrel{a}{\longrightarrow} \circ (\stackrel{\tau}{\longrightarrow})^* & \text{if } a \neq \tau \\ (\stackrel{\tau}{\longrightarrow})^* & \text{if } a = \tau. \end{cases}$$

The symbol $\circ$ stands for composition of binary relations and $(\stackrel{\tau}{\longrightarrow})^*$ is the reflexive and transitive closure of the binary relation $\stackrel{\tau}{\longrightarrow}$. In words, if $a \neq \tau$ then $s \stackrel{a}{\Longrightarrow} s'$ means that from s there is a sequence of zero or more transitions

transitions labelled by τ, followed by one transition labelled by a, followed again by zero or more transitions labelled by τ such that we reach the state s'. By writing $s \overset{\tau}{\Longrightarrow} s'$ we understand that we can go from s to s' via zero or more transitions labelled by τ. In particular, for every state s we have $s \overset{\tau}{\Longrightarrow} s$.

As before we extend the weak transition relation to the elements of Act^* and use the notation $\overset{a}{\not\Longrightarrow}$.

Example 7. Let us consider the following transition system.



We can now observe the following facts: $s_0 \overset{a}{\Longrightarrow} s_4$, $s_0 \overset{\tau}{\Longrightarrow} s_2$, $s_2 \overset{\tau}{\Longrightarrow} s_2$, $s_2 \overset{a}{\Longrightarrow} s_3$, $s_0 \overset{ab}{\Longrightarrow} s_4$, $s_0 \overset{abc}{\Longrightarrow} s_4$, $s_0 \overset{ac}{\Longrightarrow} s_4$, and $s_1 \overset{b}{\not\Longrightarrow}$. □

Definition 5 (Weak bisimulation).
Let $(Proc, Act, \{\overset{a}{\longrightarrow} \mid a \in Act\})$ be a labelled transition system (such that Act possibly contains the distinguished silent action τ). A binary relation $R \subseteq Proc \times Proc$ is a weak bisimulation *iff whenever $(s,t) \in R$ then for each $a \in Act$:*

- *if $s \overset{a}{\longrightarrow} s'$ then $t \overset{a}{\Longrightarrow} t'$ for some $t' \in Proc$ such that $(s',t') \in R$, and*
- *if $t \overset{a}{\longrightarrow} t'$ then $s \overset{a}{\Longrightarrow} s'$ for some $s' \in Proc$ such that $(s',t') \in R$.*

We say that two states $p_1, p_2 \in Proc$ are *weakly bisimilar*, and write $p_1 \approx p_2$, if and only if there exists a weak bisimulation R such that $(p_1, p_2) \in R$.

Remark 3. It is possible to give an alternative definition of weak bisimilarity as follows. A binary relation $R \subseteq Proc \times Proc$ is a *weak bisimulation* iff whenever $(s,t) \in R$ then for each $a \in Act$:

- if $s \overset{a}{\Longrightarrow} s'$ then $t \overset{a}{\Longrightarrow} t'$ for some $t' \in Proc$ such that $(s',t') \in R$, and
- if $t \overset{a}{\Longrightarrow} t'$ then $s \overset{a}{\Longrightarrow} s'$ for some $s' \in Proc$ such that $(s',t') \in R$.

It is straightforward (see e.g. [4]) to realize that this definition of weak bisimulation defines the same notion as Definition 5. □

Several properties of the relation $\approx$ are mentioned in [1], in particular $\approx$ is an equivalence relation and it is the largest weak bisimulation. We call it *weak bisimilarity* or *weak bisimulation equivalence*.

Example 8. Let us consider the following transition system.

$$s \overset{\tau}{\longrightarrow} s_1 \overset{a}{\longrightarrow} s_2 \qquad\qquad t \overset{a}{\longrightarrow} t_1$$

Obviously $s \not\sim t$. On the other hand $s \approx t$ because $R = \{(s,t),(s_1,t),(s_2,t_1)\}$ is a weak bisimulation such that $(s,t) \in R$. It remains to verify that R is indeed a weak bisimulation.

- Let us examine all possible transitions from the components of the pair (s,t). If $s \overset{\tau}{\longrightarrow} s_1$ then $t \overset{\tau}{\Longrightarrow} t$ and $(s_1,t) \in R$. If $t \overset{a}{\longrightarrow} t_1$ then $s \overset{a}{\Longrightarrow} s_2$ and $(s_2,t_1) \in R$.

– Let us examine all possible transitions from (s_1, t). If $s_1 \xrightarrow{a} s_2$ then $t \overset{a}{\Longrightarrow} t_1$ and $(s_2, t_1) \in R$. Similarly if $t \xrightarrow{a} t_1$ then $s_1 \overset{a}{\Longrightarrow} s_2$ and again $(s_2, t_1) \in R$.
– In the last pair (s_2, t_1) neither s_2 nor t_1 can perform any transition, so it is safe to have this pair in R.

Hence we showed that every pair from R satisfies the condition given in Definition 5, which means that R is a weak bisimulation. □

The following proposition points out the relationship between strong and weak bisimilarity.

Proposition 2 ([4]). *If $s \sim t$ then also $s \approx t$.*

Proof. By definition $s \sim t$ means that there is a strong bisimulation R such that $(s, t) \in R$. It is enough to show that R is also a weak bisimulation. This is almost obvious if we realize that for any two states $p, p' \in \mathit{Proc}$ and any action $a \in \mathit{Act}$ (including τ) it holds that $p \xrightarrow{a} p'$ implies $p \overset{a}{\Longrightarrow} p'$. □

Remark 4. Example 8 shows that the opposite direction of Proposition 2 does not hold. Hence the fact that $s \approx t$ does not imply that necessarily $s \sim t$. □

Similarly as for strong bisimilarity, showing that two states are *not* weakly bisimilar is more difficult and means that we have to enumerate all binary relations on states and verify that none of them is a weak bisimulation and at the same time contains the pair of states that we test for equivalence.

Fortunately, the rules of the strong bisimulation game as defined in the previous section need only be slightly modified in order to achieve a characterization of weak bisimilarity in terms of weak bisimulation games.

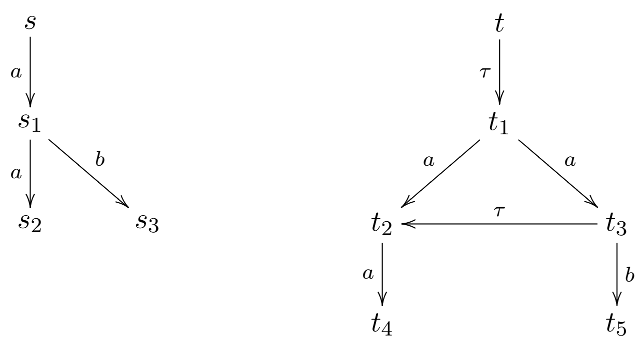
Definition 6 (Weak Bisimulation Game). *A* weak bisimulation game *is defined in the same way as strong bisimulation game in Definition 4, with the only exception that the defender can answer using weak transition relation $\overset{a}{\Longrightarrow}$ instead of only $\xrightarrow{a}$ as in the strong case. The attacker is still allowed to use only the $\xrightarrow{a}$ moves.*

All the definitions of a play and winning strategy are exactly as before and we have a similar proposition as for the strong bisimulation game.

Proposition 3. *States s_1 and t_1 of a labelled transition system are weakly bisimilar if and only if the defender has a universal winning strategy in the weak bisimulation game starting from the configuration (s_1, t_1). The states s_1 and t_1 are not weakly bisimilar if and only if the attacker has a universal winning strategy.*

We remind the reader of the fact that in the weak bisimulation game from the current configuration (s, t), if the attacker chooses a move under the silent action τ (let us say $s \xrightarrow{\tau} s'$) then the defender can (as one possibility) simply answer by doing 'nothing', i.e., by idling in the state t (as we always have $t \overset{\tau}{\Longrightarrow} t$).

Example 9. Consider the following transition system.



We will show that $s \not\approx t$ be defining a universal winning strategy for the attacker in the weak bisimulation game from (s, t).

In the first round, the attacker selects the left side and action a and plays the move $s \xrightarrow{a} s_1$. The defender has three possible moves to answer: (i) $t \overset{a}{\Longrightarrow} t_2$ via t_1, (ii) $t \overset{a}{\Longrightarrow} t_2$ via t_1 and t_3, and (iii) $t \overset{a}{\Longrightarrow} t_3$ via t_1. In case (i) and (ii) the current configuration becomes (s_1, t_2) and in case (iii) it becomes (s_1, t_3).

From the configuration (s_1, t_2) the attacker wins by playing $s_1 \xrightarrow{b} s_3$ and the defender loses because $t_2 \overset{b}{\not\Longrightarrow}$.

From the configuration (s_1, t_3) the attacker plays from the right side the τ move: $t_3 \xrightarrow{\tau} t_2$. Defender's only answer from s_1 is by $s_1 \overset{\tau}{\Longrightarrow} s_1$ because no τ actions are enabled from s_1. The current configuration becomes (s_1, t_2) and as argued above, the attacker has a winning strategy from this pair.

This concludes the proof and shows that $s \not\approx t$ because we found a universal winning strategy for the attacker. $\square$

References

1. Luca Aceto, Kim G. Larsen, and Anna Ingólfsdóttir. *An Introduction to Milner's CCS*. Lecture notes for the Semantics and Verification course at AAU, 2004.
2. J.C.M. Baeten and W.P. Weijland. *Process Algebra*. Number 18 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1990.
3. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
4. R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
5. G. Plotkin. A structural approach to operational semantics. Technical Report Daimi FN-19, Department of Computer Science, University of Aarhus, 1981. Recently appeared in Journal of Logic and Algebraic Programming, volumes 60-61 , pages 3-15, 2004.
6. C. Stirling. Local model checking games. In *Proceedings of the 6th International Conference on Concurrency Theory (CONCUR'95)*, volume 962 of *LNCS*, pages 1–11. Springer-Verlag, 1995.
7. W. Thomas. On the Ehrenfeucht-Fraïssé game in theoretical computer science (extended abstract). In *Proceedings of the 4th International Joint Conference CAAP/FASE, Theory and Practice of Software Development (TAPSOFT'93)*, volume 668 of *LNCS*, pages 559–568. Springer-Verlag, 1993.