System Validation

Mohammad Mousavi

2. Strong bahavioral equivalences and weak behavioral equivalences part 2.

Weak Behavioral Equivalences

Mohammad Mousavi

TU/Eindhoven

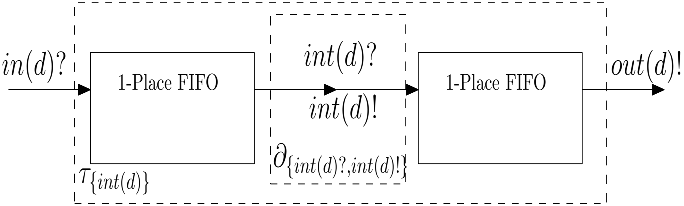
System Validation, 2012-2013
TU Delft

Overview

- Motivation
- Labelled Transition Systems,
- Strong equivalences:
  - √ trace equivalence,
  - √ language (completed trace) equivalence,
  - √ strong bisimilarity,
  - → exercises.
- → Weak equivalences:
  - weak trace equivalence,
  - branching bisimilarity,
  - root condition.
  - exercises

Motivating Example

Verifying a Two Place Buffer



$=_?$

Recap

Strong Equivalences

- Traces: sequences of actions originating from the initial state,

Recap

Strong Equivalences

- Traces: sequences of actions originating from the initial state,
- Language: sequences of actions originating from the initial state and ending in either termination or deadlock,

Recap

Strong Equivalences

- Traces: sequences of actions originating from the initial state,
- Language: sequences of actions originating from the initial state and ending in either termination or deadlock,
- Bisimulation relation: related states can mimic each others' transitions such that the targets are related by the same relation

Weak Equivalences

Idea

- Internal actions should be invisible to the outside world.

Weak Equivalences

Idea

- Internal actions should be invisible to the outside world.
- τ: The collective name for all invisible actions.

Weak Equivalences

Idea

- Internal actions should be invisible to the outside world.
- τ: The collective name for all invisible actions.
- Adapt behavioral equivalence to neglect τ

Labeled Transition Systems

An LTS is a 5-tuple $\langle S, Act, \rightarrow, s, T\rangle$:

- S is a set of *states*,
- Act is a set of *(multi-)actions* (assumption: $\tau \in Act$)
- $\rightarrow \subseteq S \times Act \times S$ is the *transition relation*.
- $s \in S$ is the *initial* state,
- $T \subseteq S$ is the set of *terminating* states,

Trace equivalence

Traces of a State

For state $t \in S$, Traces(t) is the minimal set satisfying:

1. $\epsilon \in$ Traces(t),
2. $\surd \in$ Traces(t) when $t \in T$,
3. $\forall_{t'_0 \in S,\ a \in Act\ ,\ \sigma \in Act_{\surd}^{*}}\ a\sigma \in$ Traces(t) when $\exists_{t' \in S}\ t \xrightarrow{a} t'$ and $\sigma \in$ Traces(t').

Trace Equivalence

For states t, t', t is trace equivalent to t' iff Traces(t) = Traces(t').

Weak Trace equivalence

Weak Traces of a State

For state $t \in S$, Traces(t) is the minimal set satisfying:

1. $\epsilon \in W\text{Traces}(t)$,
2. $\surd \in W\text{Traces}(t)$ when $t \in T$,
3. $\forall_{t'_0 \in S,\ a \in Act \setminus \{\tau\},\ \sigma \in Act_{\surd}{}^*}\ a\sigma \in W\text{Traces}(t)$ when $\exists_{t' \in S}\ t \xrightarrow{a} t'$ and $\sigma \in W\text{Traces}(t')$.
4. $\forall_{t'_0 \in S,\ \sigma \in Act_{\surd}{}^*}\ \sigma \in W\text{Traces}(t)$ when $\exists_{t' \in S}\ t \xrightarrow{\tau} t'$ and $\sigma \in W\text{Traces}(t')$.

Trace Equivalence

For states t, t', t is trace equivalent to t' iff
$\text{Traces}(t) = \quad \text{Traces}(t')$.

Weak Trace equivalence

Weak Traces of a State

For state $t \in S$, Traces(t) is the minimal set satisfying:

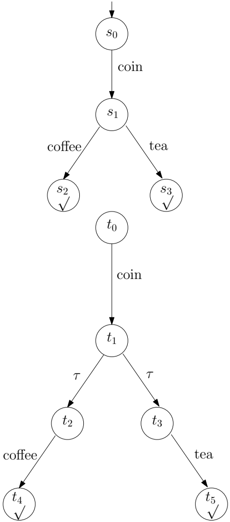
1. $\epsilon \in W\text{Traces}(t)$,
2. $\surd \in W\text{Traces}(t)$ when $t \in T$,
3. $\forall_{t'_0 \in S,\ a \in Act \setminus \{\tau\},\ \sigma \in Act_{\surd}^{*}}\ a\sigma \in W\text{Traces}(t)$ when $\exists_{t' \in S}\ t \xrightarrow{a} t'$ and $\sigma \in W\text{Traces}(t')$.
4. $\forall_{t'_0 \in S,\ \sigma \in Act_{\surd}^{*}}\ \sigma \in W\text{Traces}(t)$ when $\exists_{t' \in S}\ t \xrightarrow{\tau} t'$ and $\sigma \in W\text{Traces}(t')$.

Weak Trace Equivalence

For states t, t', t is trace equivalent to t' iff $W\text{Traces}(t) = W\text{Traces}(t')$.

1. $\epsilon \in \text{Traces}(t)$,
2. $\surd \in \text{Traces}(t)$ when $t \in T$,
3. $a\sigma \in \text{Traces}(t)$ when $t \xrightarrow{a} t'$ and $\sigma \in \text{Traces}(t')$,
4. $\sigma \in \text{Traces}(t)$ when $t \xrightarrow{\tau} t'$ and $\sigma \in \text{Traces}(t')$.

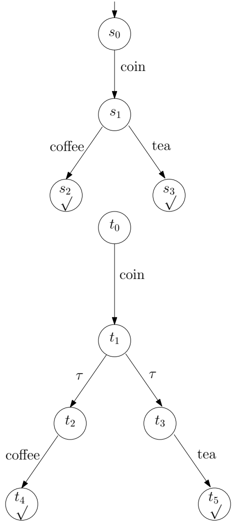
Weak Traces: An Example



1. $\epsilon \in \text{Traces}(t)$,
2. $\surd \in \text{Traces}(t)$ when $t \in T$,
3. $a\sigma \in \text{Traces}(t)$ when $t \xrightarrow{a} t'$ and $\sigma \in \text{Traces}(t')$,
4. $\sigma \in \text{Traces}(t)$ when $t \xrightarrow{\tau} t'$ and $\sigma \in \text{Traces}(t')$.

Weak Traces: An Example

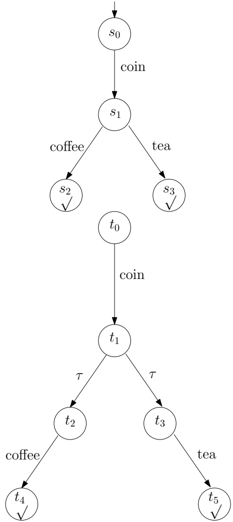
- $\text{WTr}(s_2) = \text{WTr}(s_3) = \text{WTr}(t_4) = \text{WTr}(t_5) = \{\epsilon, \surd\}$,



1. $\epsilon \in \text{Traces}(t)$,
2. $\surd \in \text{Traces}(t)$ when $t \in T$,
3. $a\sigma \in \text{Traces}(t)$ when $t \xrightarrow{a} t'$ and $\sigma \in \text{Traces}(t')$,
4. $\sigma \in \text{Traces}(t)$ when $t \xrightarrow{\tau} t'$ and $\sigma \in \text{Traces}(t')$.

Weak Traces: An Example

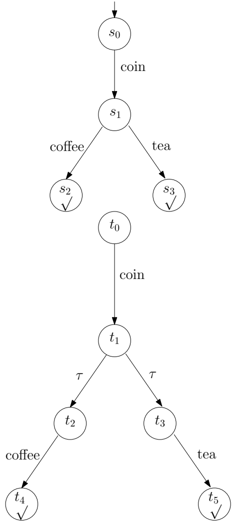
- $\text{WTr}(s_2) = \text{WTr}(s_3) = \text{WTr}(t_4) = \text{WTr}(t_5) = \{\epsilon, \surd\}$,
- $\text{WTr}(s_1) = \{\epsilon, \mathit{coffee}, \mathit{tea}, \mathit{coffee}\surd, \mathit{tea}\surd\}$,



1. $\epsilon \in \text{Traces}(t)$,
2. $\surd \in \text{Traces}(t)$ when $t \in T$,
3. $a\sigma \in \text{Traces}(t)$ when $t \xrightarrow{a} t'$ and $\sigma \in \text{Traces}(t')$,
4. $\sigma \in \text{Traces}(t)$ when $t \xrightarrow{\tau} t'$ and $\sigma \in \text{Traces}(t')$.

Weak Traces: An Example

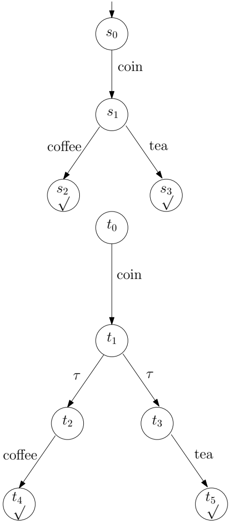
- $\text{WTr}(s_2) = \text{WTr}(s_3) = \text{WTr}(t_4) = \text{WTr}(t_5) = \{\epsilon, \surd\}$,
- $\text{WTr}(s_1) = \{\epsilon, \textit{coffee}, \textit{tea}, \textit{coffee}\surd, \textit{tea}\surd\}$,
- $\text{WTr}(t_2) = \{\epsilon, \textit{coffee}, \textit{coffee}\surd\}$, $\text{WTr}(t_3) = \{\epsilon, \textit{tea}, \textit{tea}\surd\}$,



1. $\epsilon \in \text{Traces}(t)$,
2. $\surd \in \text{Traces}(t)$ when $t \in T$,
3. $a\sigma \in \text{Traces}(t)$ when $t \xrightarrow{a} t'$ and $\sigma \in \text{Traces}(t')$,
4. $\sigma \in \text{Traces}(t)$ when $t \xrightarrow{\tau} t'$ and $\sigma \in \text{Traces}(t')$.

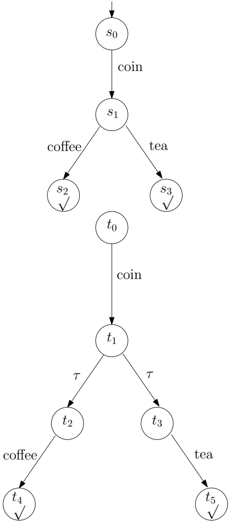
Weak Traces: An Example

- $\text{WTr}(s_2) = \text{WTr}(s_3) = \text{WTr}(t_4) = \text{WTr}(t_5) = \{\epsilon, \surd\}$,
- $\text{WTr}(s_1) = \{\epsilon, \mathit{coffee}, \mathit{tea}, \mathit{coffee}\surd, \mathit{tea}\surd\}$,
- $\text{WTr}(t_2) = \{\epsilon, \mathit{coffee}, \mathit{coffee}\surd\}$, $\text{WTr}(t_3) = \{\epsilon, \mathit{tea}, \mathit{tea}\surd\}$,
- $\text{WTr}(t_1) = \{\epsilon, \mathit{coffee}, \mathit{tea}, \mathit{coffee}\surd, \mathit{tea}\surd\}$,



1. $\epsilon \in \text{Traces}(t)$,
2. $\surd \in \text{Traces}(t)$ when $t \in T$,
3. $a\sigma \in \text{Traces}(t)$ when $t \xrightarrow{a} t'$ and $\sigma \in \text{Traces}(t')$,
4. $\sigma \in \text{Traces}(t)$ when $t \xrightarrow{\tau} t'$ and $\sigma \in \text{Traces}(t')$.

Weak Traces: An Example



- $\text{WTr}(s_2) = \text{WTr}(s_3) = \text{WTr}(t_4) = \text{WTr}(t_5) = \{\epsilon, \surd\}$,
- $\text{WTr}(s_1) = \{\epsilon, \textit{coffee}, \textit{tea}, \textit{coffee}\surd, \textit{tea}\surd\}$,
- $\text{WTr}(t_2) = \{\epsilon, \textit{coffee}, \textit{coffee}\surd\}$,
  $\text{WTr}(t_3) = \{\epsilon, \textit{tea}, \textit{tea}\surd\}$,
- $\text{WTr}(t_1) = \{\epsilon, \textit{coffee}, \textit{tea}, \textit{coffee}\surd, \textit{tea}\surd\}$,
- $\text{WTr}(s_0) = \text{WTr}(t_0) = \{\epsilon, \textit{coin}, \textit{coin coffee}, \textit{coin tea}, \textit{coin coffee}\surd, \textit{coin tea}\surd\}$.

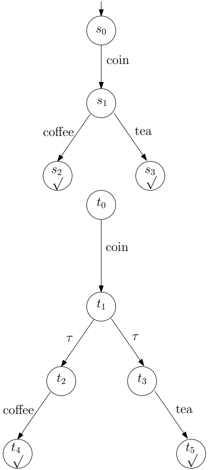
Weak Trace Equivalence: An Observation

Observation

$\mathsf{WTr}(s_0) = \mathsf{WTr}(t_0) =$
$\{\epsilon, coin, coin\ coffee, coin\ tea, coin\ coffee\surd, coin\ tea\surd\}$

Moral of the Story

Weak Trace equivalence is usually too coarse for interacting (open) systems (neglects important differences).



Weak Bisimulations

Idea

1. An a-transition should be mimicked by the same transition possibly with before and/or after (stuttering) τ-transitions;
2. A τ-transition can be mimicked by remaining in the same state (making no transition).

(Weak, Branching) Bisimulation

Formal Definition: Bisimulation

- $R \subseteq S \times S$ is a bisimulation relation when for all $\forall_{(t_0,t_1) \in R}$
  - $\forall_{t_0' \in S, a \in A} \, t_0 \xrightarrow{a} t_0' \Rightarrow$
    - $\exists \; t_2' \in S \; t_1 \xrightarrow{a} t_2'$ $\wedge \, (t_0', t_2') \in R$,
  - $t_0 \in T \Rightarrow t_1 \in T$,

and vice versa.

(Weak, Branching) Bisimulation

Formal Definition: Weak Bisimulation

- $R \subseteq S \times S$ is a weak bisimulation relation when for all $\forall_{(t_0, t_1) \in R}$
  - $\forall_{t_0' \in S, a \in A} \, t_0 \xrightarrow{a} t_0' \Rightarrow$
    - $a = \tau \land (t_0', t_1) \in R$ or
    - $\exists_{t_1', t_2', t_3' \in S} \; t_1 \xrightarrow{\tau}{}^{*} t_1' \xrightarrow{a} t_2' \xrightarrow{\tau}{}^{*} t_3' \land (t_0', t_3') \in R$,
  - $t_0 \in T \Rightarrow \exists_{t_1' \in S} \; t_1 \xrightarrow{\tau}{}^{*} t_1' \land t_1' \in T$,

and vice versa.

(Weak, Branching) Bisimulation

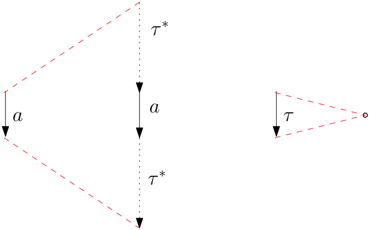
Formal Definition: Branching Bisimulation

- $R \subseteq S \times S$ is a branching bisimulation relation when for all $\forall_{(t_0,t_1) \in R}$
  - $\forall_{t_0' \in S, a \in A} t_0 \xrightarrow{a} t_0' \Rightarrow$
    - $a = \tau \wedge (t_0', t_1) \in R$ or
    - $\exists_{t_1', t_2' \in S}\ t_1 \xrightarrow{\tau}{}^{*} t_1' \xrightarrow{a} t_2' \wedge (t_0', t_2') \in R \wedge (t_0, t_1') \in R$,
  - $t_0 \in T \Rightarrow \exists_{t_1' \in S}\ t_1 \xrightarrow{\tau}{}^{*} t_1' \wedge (t_0, t_1') \in R \wedge t_1' \in T$,

and vice versa.

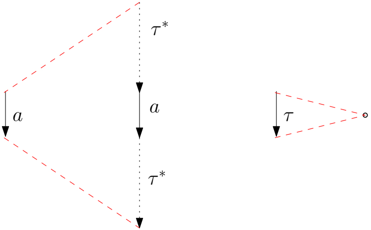
Weak vs. Branching Bisimulation

Weak Bisimulation

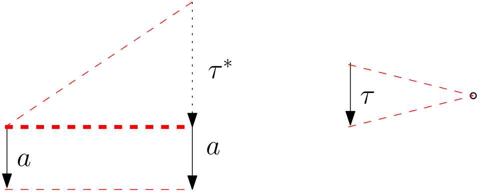


Weak vs. Branching Bisimulation

Weak Bisimulation

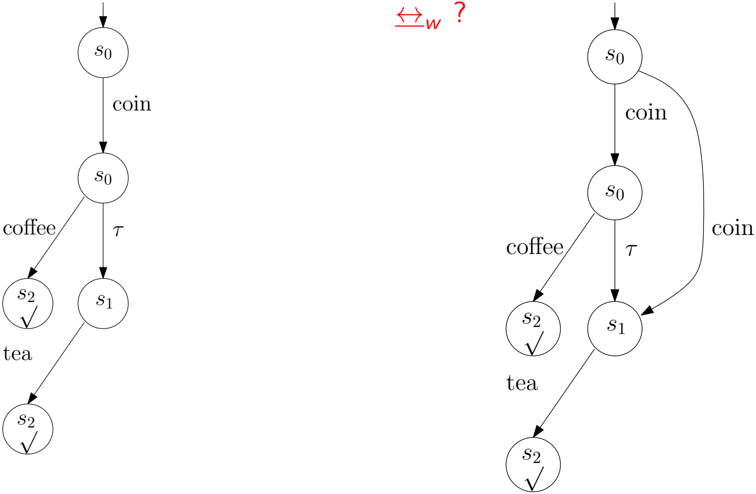


Branching Bisimulation

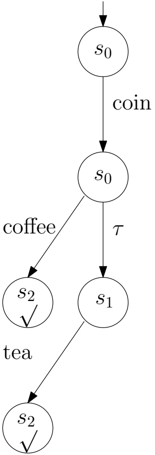


Weak Vending Machines

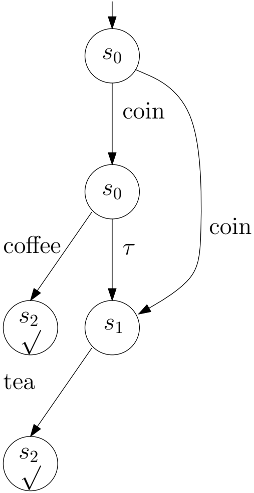
$\underline{\leftrightarrow}_w$?



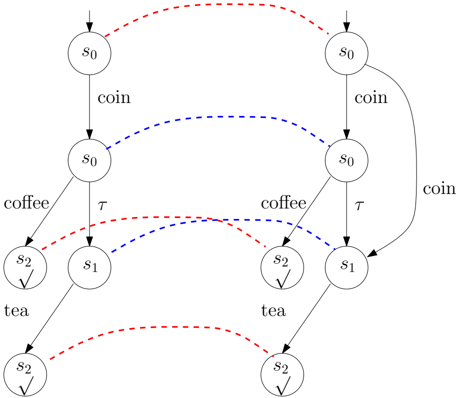
Weak Vending Machines



$\underline{\leftrightarrow}_w$



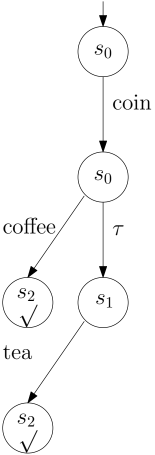
Weak Vending Machines



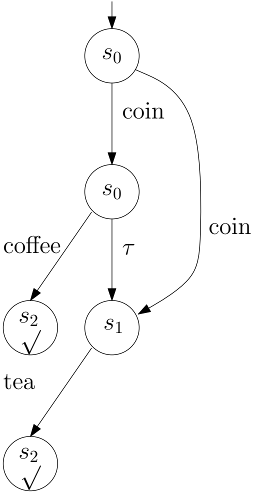
Observation

Weak Bisimulation can be too coarse. It does not preserve the branching structure.

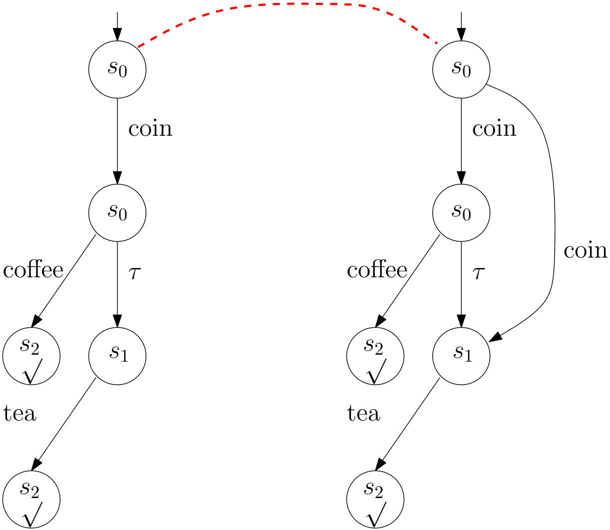
Weak Vending Machines



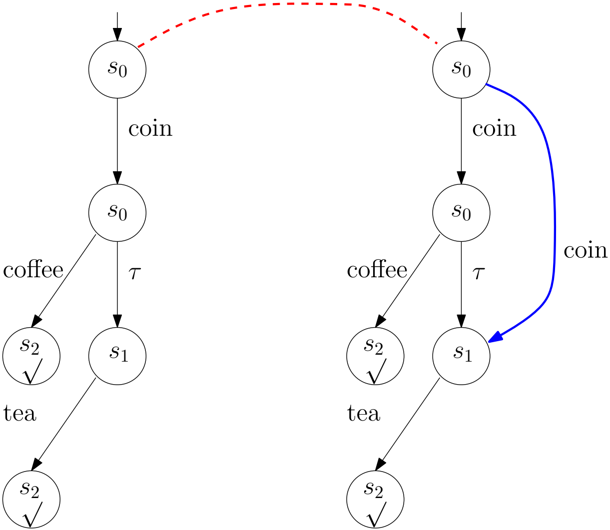
$\underline{\leftrightarrow}_b$?



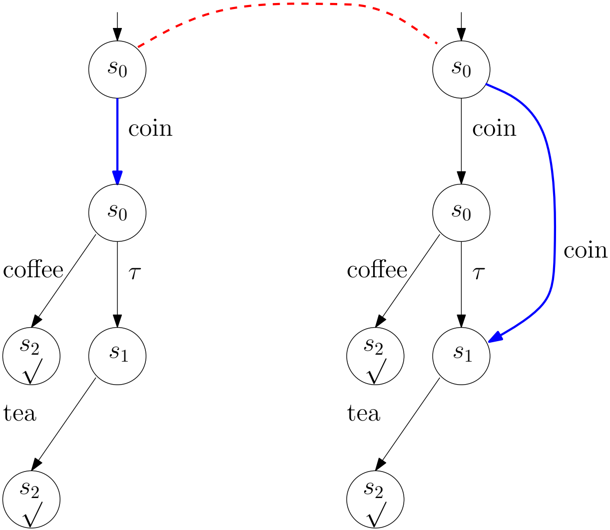
Weak Vending Machines



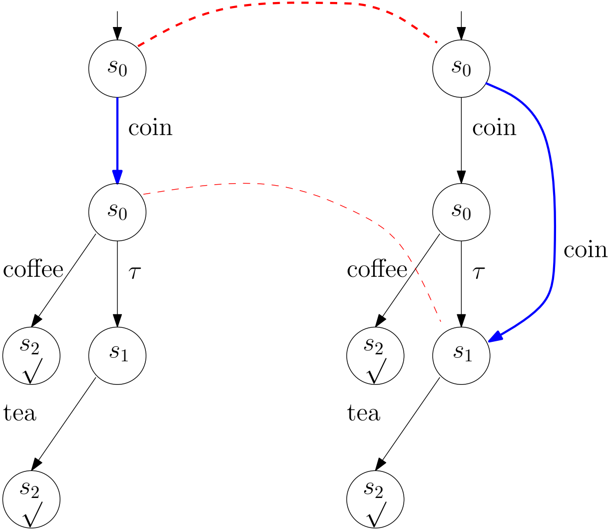
Weak Vending Machines



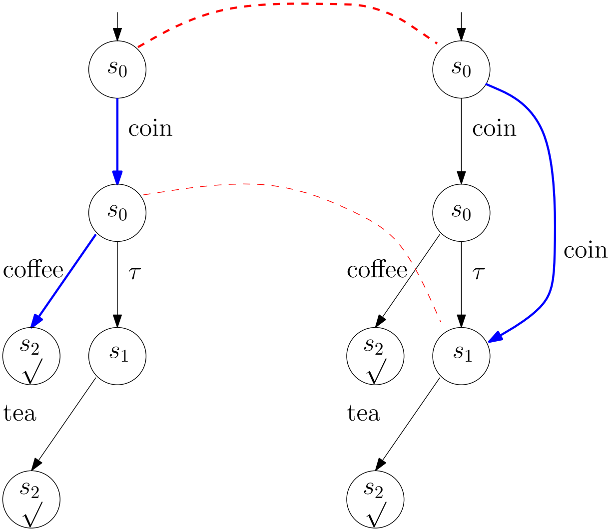
Weak Vending Machines



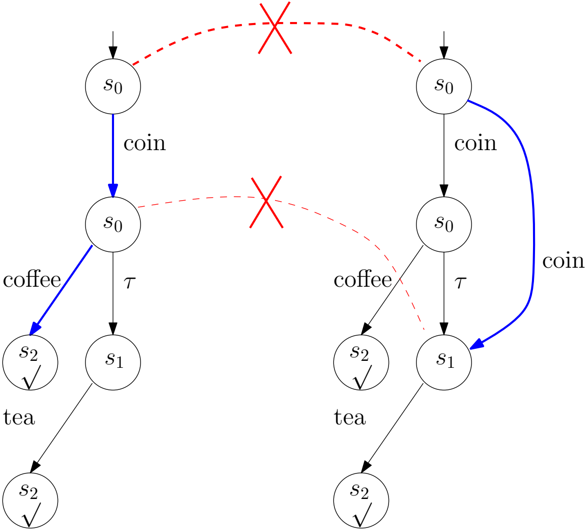
Weak Vending Machines



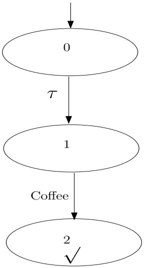
Weak Vending Machines



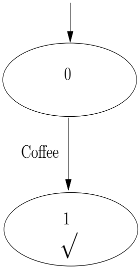
Weak Vending Machines



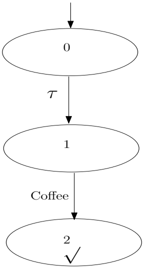
Weak Bisimulations and Choice



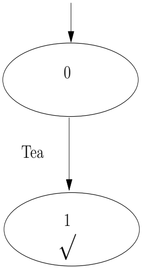
$\underline{\leftrightarrow}_{w,b}$



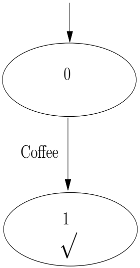
Weak Bisimulations and Choice



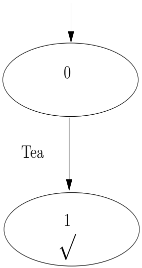
+



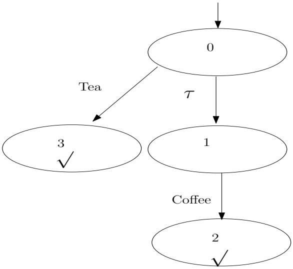
Weak Bisimulations and Choice



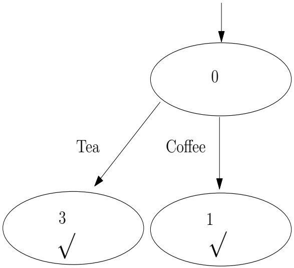
+



Weak Bisimulations and Choice



$\neq$



Weak Bisimulation Equivalence

Conclusions

Weak bisimulation

1. does not preserve branching structure (solution: branching bisimulation);
2. is not preserved under choice (solution: rootedness).

Root Condition

Basic Idea

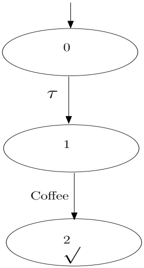
For a weak bisimulation to be a congruence with respect to choice, the first τ-transition should be mimicked by a τ transition.

Formal Definition: Rootedness

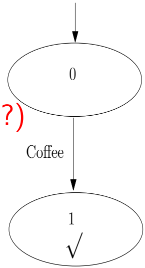
Two state s_0, s_1 are rooted branching (weak) bisimilar if

- there exists a branching (weak) bisimulation relation R such that $(s_0, s_1) \in R$ and
- s_0 and s_1 are only related to each other (and not to any other state).

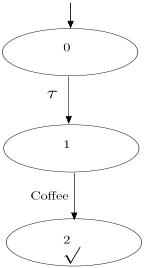
Weak Bisimulations and Choice



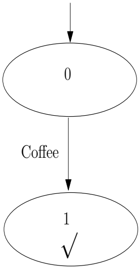
$\underline{\leftrightarrow}_{rw,rb}$ (?)



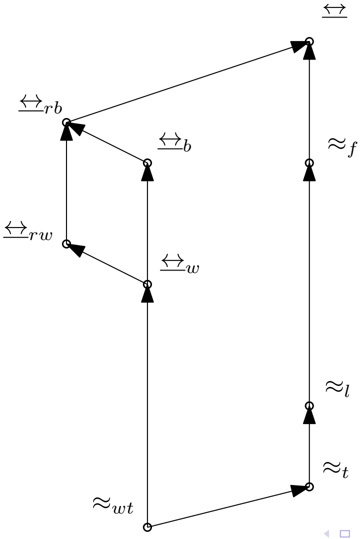
Weak Bisimulations and Choice



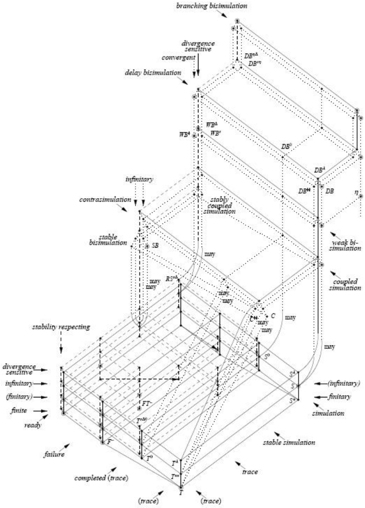
$\neq$



Van Glabbeek's Spectrum (The Treated Part)



Van Glabbeek's Spectrum



Remaining Exercises

- 2.4.6
- 2.4.7