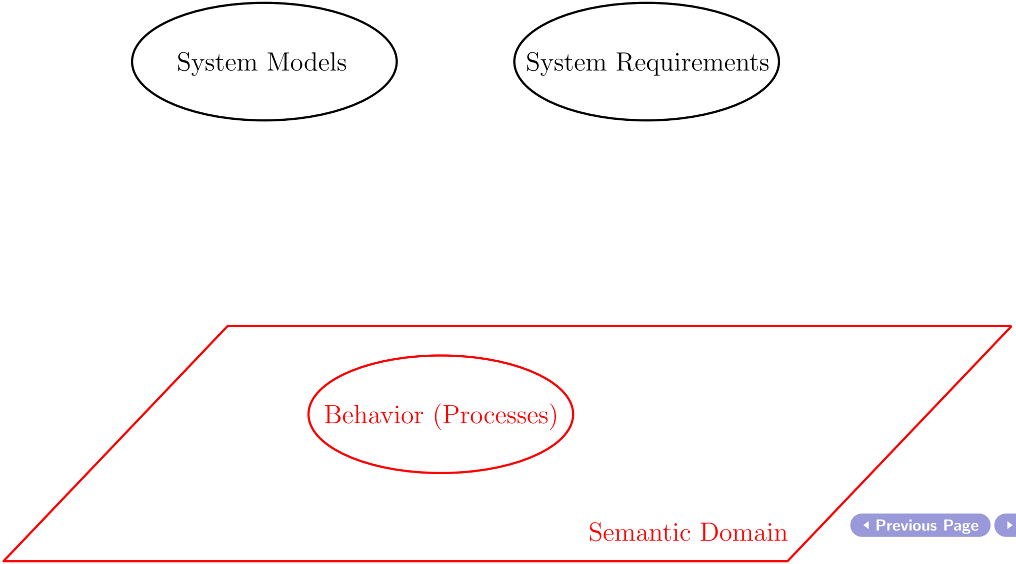
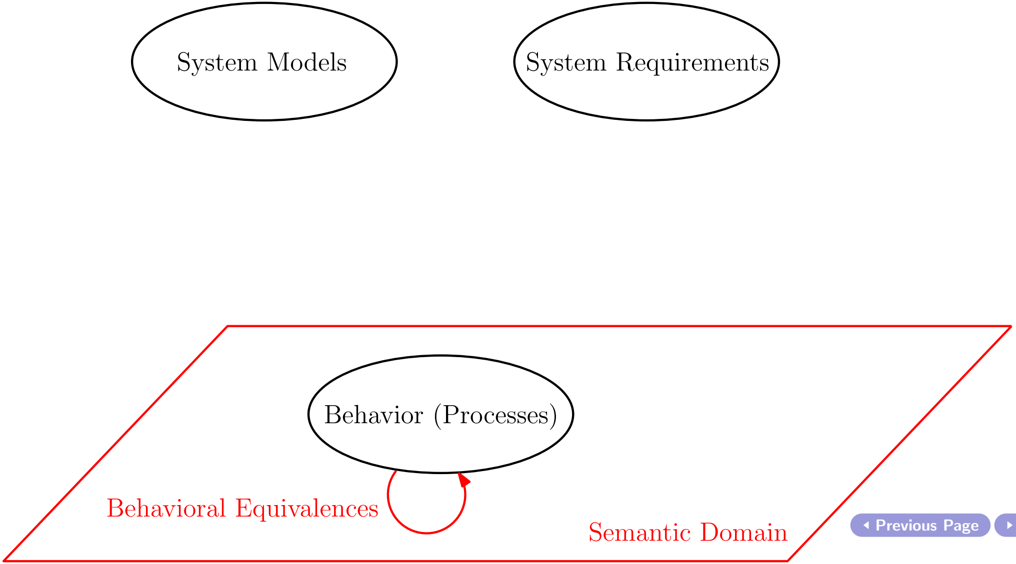
System Validation: Trace Equivalence

Mohammad Mousavi and Jeroen Keiren

General Overview



General Overview



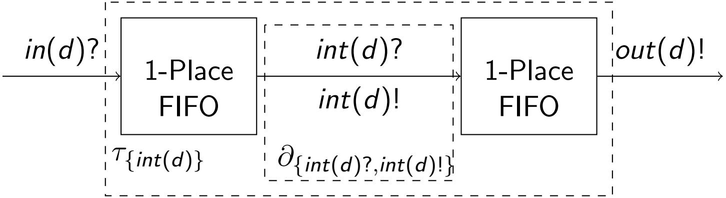
Behavioral Equivalences

Motivation

- Verification: check whether implementation conforms to specification
- Implementation: transition system with more actions added
- Method: abstracting and comparing with specification

- Verification: check whether implementation conforms to specification
- Implementation: transition system with more actions added
- Method: abstracting and comparing with specification

Behavioral equivalence needed to compare behavioral models



$in(d)?$
1-Place FIFO
$int(d)?$
$int(d)!$
1-Place FIFO
$out(d)!$
$\tau_{\{int(d)\}}$
$\partial_{\{int(d)?,int(d)!\}}$

$\stackrel{?}{=}$

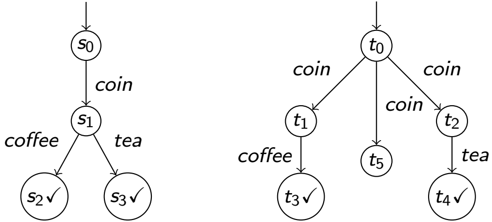
$in(d)?$
2-Place FIFO
$out(d)!$

Behavioral Equivalences
Requirements

Behavioral equivalence should:

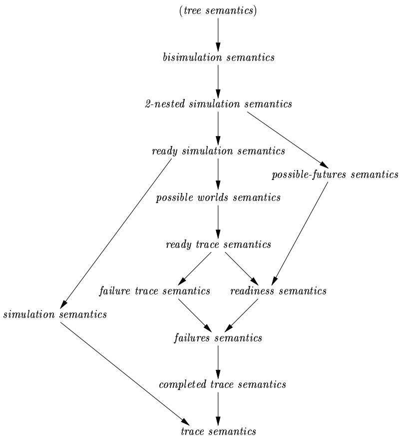
- neglect immaterial differences (not too fine)
- note important differences (not too coarse)
- should be preserved under context (congruence)

depends on the particular application domain



s_0
coin
s_1
coffee
tea
s_2 ✓
s_3 ✓
t_0
coin
coin
coin
t_1
t_5
t_2
coffee
tea
t_3 ✓
t_4 ✓

Linear-Time Branching-Time Spectrum
Strong fragment



Traces of a State

For $t \in S$, $\mathit{Traces}(t)$ is minimal set satisfying:

1. $\epsilon \in \mathit{Traces}(t)$

Trace Equivalence

Traces of a State

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

1. $\epsilon \in Traces(t)$
2. $\surd \in Traces(t)$ when $t \in T$

Trace Equivalence

Traces of a State

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

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

Trace Equivalence

Traces of a State

For $t \in S$, $\mathit{Traces}(t)$ is minimal set satisfying:

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

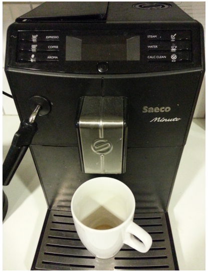
Trace Equivalence

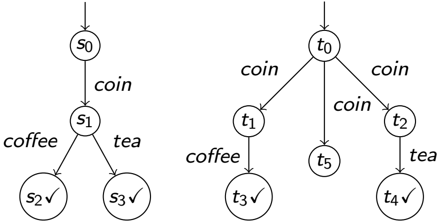
For states t, t', t is trace equivalent to t' iff $\mathit{Traces}(t) = \mathit{Traces}(t')$.

Trace equivalence
Example

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

What are the sets of traces?



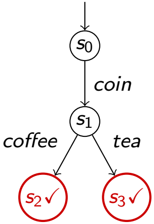


Trace Equivalence

An Exercise

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

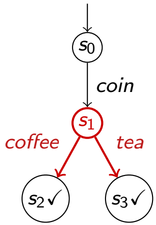
$$Traces(s_2) = Traces(s_3) = \{\epsilon, \surd\}$$



Trace Equivalence
An Exercise

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

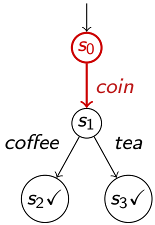
$Traces(s_2) = Traces(s_3) = \{\epsilon, \surd\}$
$Traces(s_1) =$
$\{\epsilon, coffee, tea, coffee\surd, tea\surd\}$



Trace Equivalence

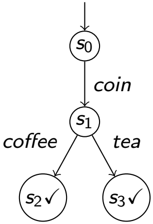
An Exercise

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

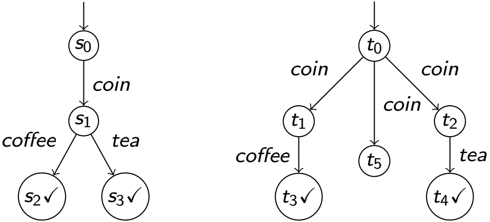


$Traces(s_1) =$
$\{\epsilon, coffee, tea, coffee\surd, tea\surd\}$
$Traces(s_0) =$
$\{\epsilon, coin, coin\ coffee, coin\ tea, coin\ coffee\surd, coin\ tea\surd\}$

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



$Traces(s_0) =$
$\{\epsilon, coin, coin\ coffee, coin\ tea, coin\ coffee\surd, coin\ tea\surd\}$



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

Moral of the Story: Trace equivalence is too coarse (neglects important differences).

Completed Trace Equivalence

$CTraces(t)$:

- $\epsilon \in CTraces(t)$ if $t \notin T$ and $\neg\exists_{t' \in S, a \in Act}$ s.t. $t \xrightarrow{a} t'$
- $\checkmark \in CTraces(t)$ if $t \in T$
- $a\sigma \in CTraces(t)$ if $t \xrightarrow{a} t'$ and $\sigma \in CTraces(t')$

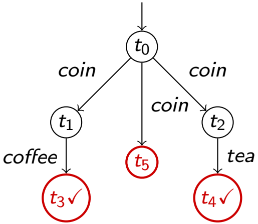
States $t, u \in S$ completed trace equivalent iff

- $Traces(t) = Traces(u)$ and
- $CTraces(t) = CTraces(u)$

Completed Trace Equivalence
An Exercise

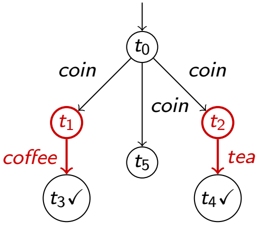
- $\epsilon \in CTraces(t)$ if $t \notin T$ & $\neg\exists_{t' \in S, a \in Act}\ t \xrightarrow{a} t'$
- $\checkmark \in CTraces(t)$ if $t \in T$
- $a\sigma \in CTraces(t)$ if $t \xrightarrow{a} t'$ and $\sigma \in CTraces(t')$

$CTraces(t_3) = CTraces(t_4) = \{\surd\}$, $CTraces(t_5) = \{\epsilon\}$



- $\epsilon \in CTraces(t)$ if $t \notin T$ & $\neg\exists_{t' \in S, a \in Act}\ t \xrightarrow{a} t'$
- $\checkmark \in CTraces(t)$ if $t \in T$
- $a\sigma \in CTraces(t)$ if $t \xrightarrow{a} t'$ and $\sigma \in CTraces(t')$

$CTraces(t_3) = CTraces(t_4) = \{\surd\}$, $CTraces(t_5) = \{\epsilon\}$
$CTraces(t_1) = \{coffee\surd\}$, $CTraces(t_2) = \{tea\surd\}$

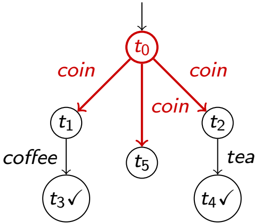


Completed Trace Equivalence

An Exercise

- $\epsilon \in CTraces(t)$ if $t \notin T$ & $\neg\exists_{t' \in S, a \in Act}\ t \xrightarrow{a} t'$
- $\checkmark \in CTraces(t)$ if $t \in T$
- $a\sigma \in CTraces(t)$ if $t \xrightarrow{a} t'$ and $\sigma \in CTraces(t')$

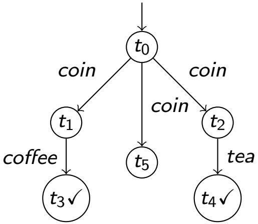
$CTraces(t_1) = \{coffee\surd\}$, $CTraces(t_2) = \{tea\surd\}$
$Traces(t_0) = \{coin, coin\ coffee\surd, coin\ tea\surd\}$



Completed Trace Equivalence

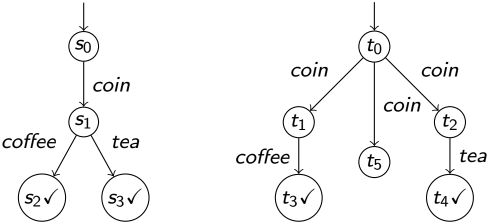
An Exercise

- $\epsilon \in CTraces(t)$ if $t \notin T$ & $\neg\exists_{t' \in S, a \in Act}\ t \xrightarrow{a} t'$
- $\checkmark \in CTraces(t)$ if $t \in T$
- $a\sigma \in CTraces(t)$ if $t \xrightarrow{a} t'$ and $\sigma \in CTraces(t')$



$$Traces(t_0) = \{coin, coin\ coffee\surd, coin\ tea\surd\}$$

Completed Trace Equivalence
An Observation

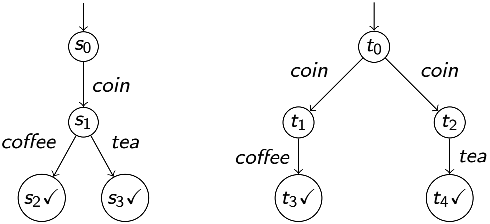


$$Traces(s_0) = Traces(t_0) = \{\epsilon, coin, coin\ coffee, coin\ tea, coin\ coffee\surd, coin\ tea\surd\}$$

$$CTraces(s_0) = \{coin\ coffee\surd, coin\ tea\surd\}$$
$$CTraces(t_0) = \{coin,\ coin\ coffee\surd, coin\ tea\surd\}$$

Completed Trace Equivalence

An Observation

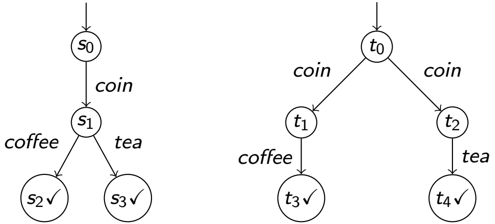


$$Traces(s_0) = Traces(t_0) = \{\epsilon, coin, coin\ coffee, coin\ tea, coin\ coffee\surd, coin\ tea\surd\}$$

$$CTraces(s_0) = \{coin\ coffee\surd, coin\ tea\surd\}$$
$$CTraces(t_0) = \{coin\ coffee\surd, coin\ tea\surd\}$$

Completed Trace Equivalence

An Observation

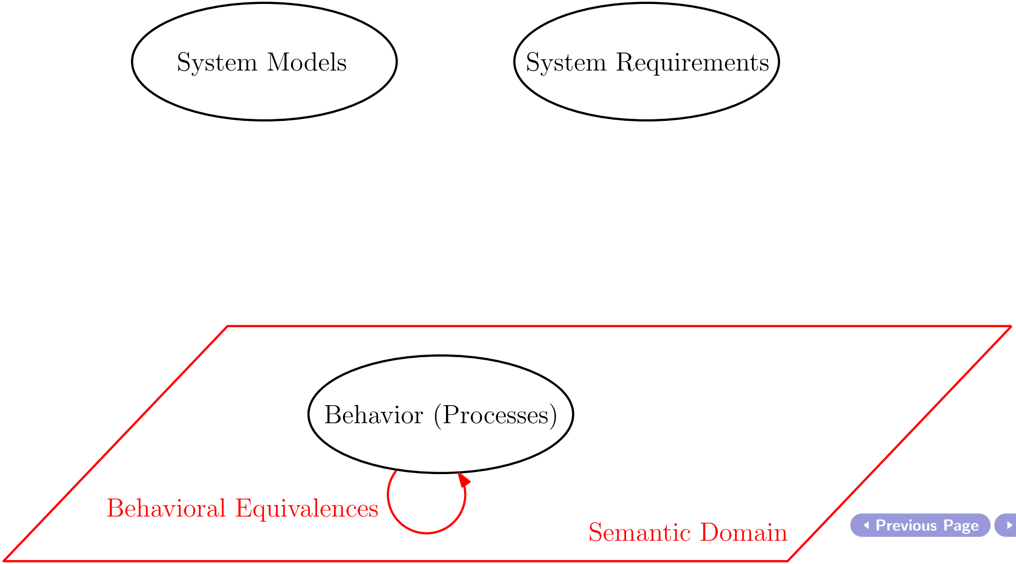


$$Traces(s_0) = Traces(t_0) = \{\epsilon, coin, coin\ coffee, coin\ tea, coin\ coffee\surd, coin\ tea\surd\}$$

$$CTraces(s_0) = \{coin\ coffee\surd, coin\ tea\surd\}$$
$$CTraces(t_0) = \{coin\ coffee\surd, coin\ tea\surd\}$$

Conclusion: Completed trace equivalence is too coarse.

General Overview



Thank you very much.