On Testing and Conformance Relations for UML Statechart Diagrams Behaviours

Diego Latella, Mieke Massink

Formal Methods and Tools Group
C.N.R./I.S.T.I., Pisa

GMD-CNR Collaboration

ISSTA 2002 (July 2002)

Context

An integrated formal framework and environment for a **behavioural subset** of UML Statechart Diagrams (UMLSDs).

- Formal Operational Core Semantics [FMOODS99]
- LTL model checking [FAC 11(6) 99]
- ACTL model checking [JLAP 51(1)]
- Stochastic Time extension [HASE00]
- Testing Theory [HASE01]
- Conformance Testing for UMLSDs [submitted]

Several funding sources:

EU HIDE(98), EU AGILE (02-04), ASI PRIDE (02-03), GMD-CNR

The Unified Modelling Language (UML)

- a general purpose, object-oriented, visual modelling language
- designed to specify, visualise, construct and document the artifacts of software systems
- a semi-formal language
  - syntax and static semantics defined precisely
  - dynamic semantics described informally

Our Overall Goal and Methodology: SEMANTICS FIRST approach

High Quality assurance of

– **notation definition** and
– **tools implementation**

achieved via

– **formal** definition of notations
  * syntax and
  * semantics
– development of **formal** theories for
  * practical concepts (like, e.g. testing)
– rigorous and whenever convenient **formal** proofs of
  * properties of notations
  * correctness of tool (design)s

‘Semantics First’ Approach

- Abstract Syntax: *Hierarchical Automata*

  $\Rightarrow$ structural induction

- Semantics definition: *recursive logical deduction rules*

  $\Rightarrow$ derivation induction

Similar approach as in [Mikk & alt. 97]

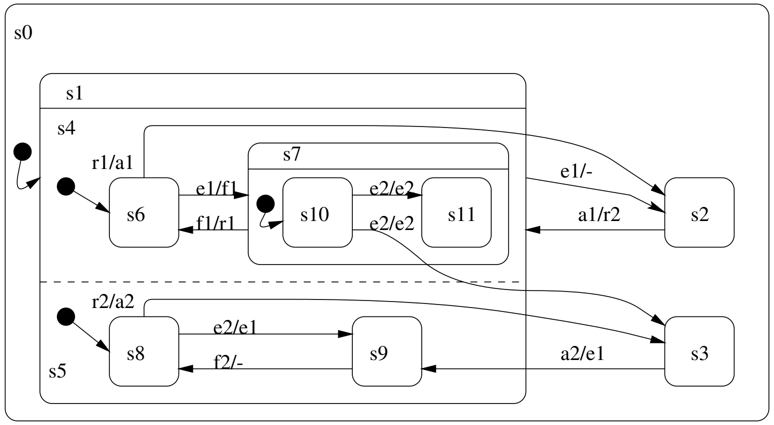
- Semantic domain: *LTSs over I/O-Pairs*

Similar approach as in [Massink 96]

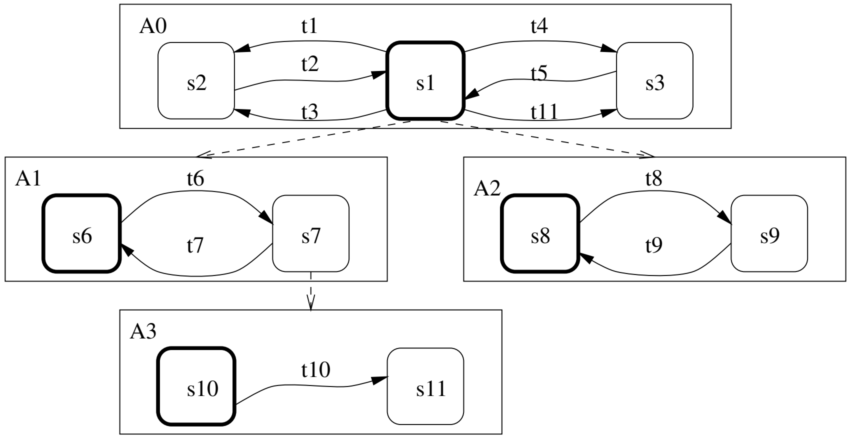
Related Work

- Testing
  - (Cavarra et al. 2002) Agedis Project: Automatic test gen. from UML Statechart Diagrams. Emphasis on industrial applicability more than on formal semantics.
  - (Pretschner et al. 2001) Autofocus: a subset of UML-RT transition diagrams with specific formal semantics (link to UML-RT not clear).
  - (Bogdanov et al. 93) Classic Statecharts
  - (Lee et al. 96; Bosik et al. 91) FSM: Mainly DFSM; Strongly Connected
- Semantics
  - (Schäfer, Knapp & Merz 2001) Both UML Statechart and Collaboration Diagrams are covered. Model-checking (via PROMELA) No proofs.
  - (Lilius & Paltor 99) All interesting aspects of UML Statechart Diagrams semantics are covered. A completely flat representation and semantics. Model-checking (via PROMELA) No proofs.
  - (Broersen & Wieringa 97) A flat modal logics approach (modal μ-calculus). No state refinement. No transition priorities.
  - (Boerger et al.) Based on ASMs.

Formal Operational Semantics: from UMLSDs to HAs



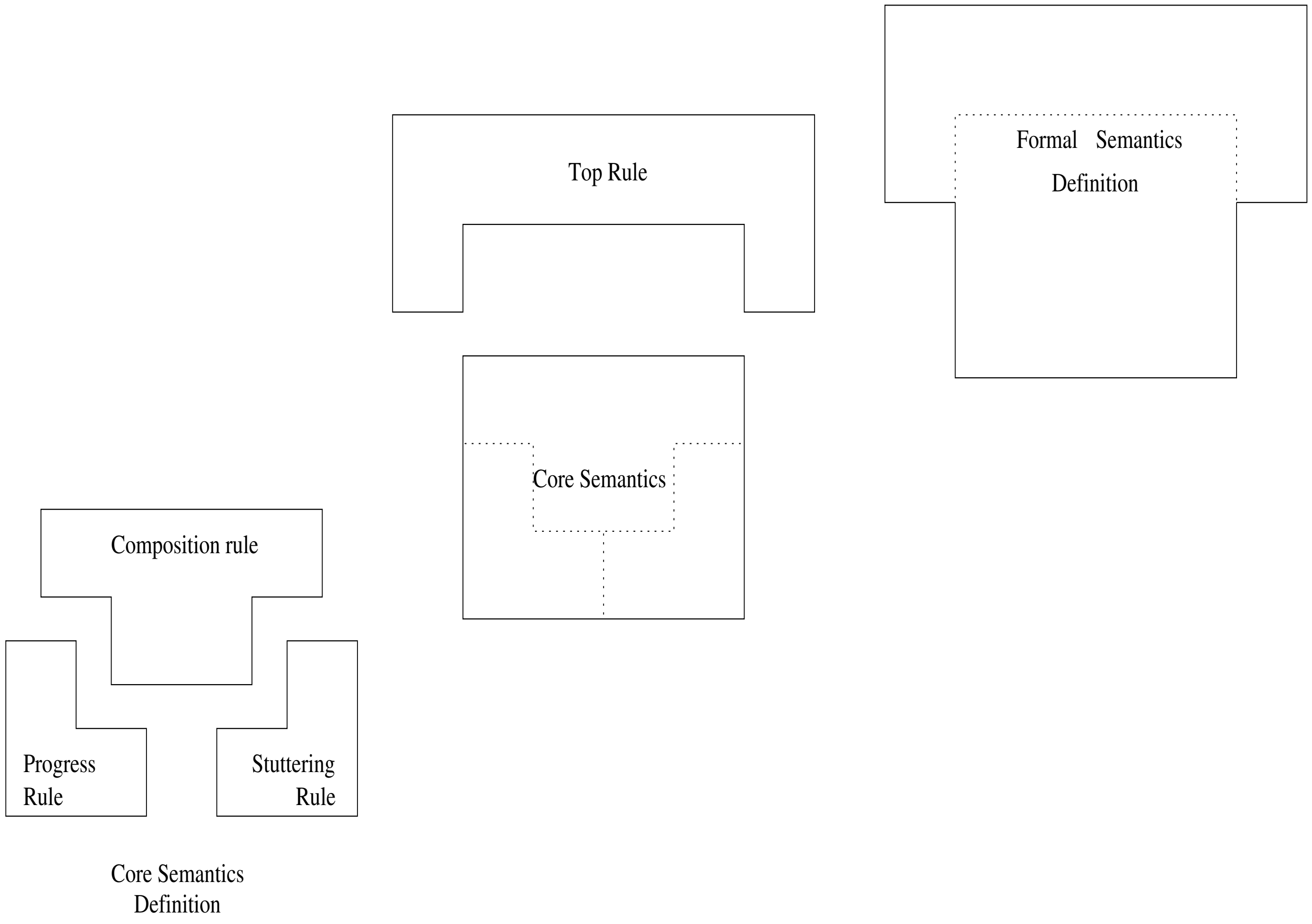
UMLSD



Hierarchical Automaton

t	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$	$t7$	$t8$	$t9$	$t10$	$t11$
$SR\ t$	$\{s6\}$	$\emptyset$	$\emptyset$	$\{s8\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{s10\}$
$EV\ t$	$r1$	$a1$	$e1$	$r2$	$a2$	$e1$	$f1$	$e2$	$f2$	$e2$	$e2$
$AC\ t$	$a1$	$r2$	ϵ	$a2$	$e1$	$f1$	$r1$	$e1$	ϵ	$e2$	$e2$
$TD\ t$	$\emptyset$	$\{s6, s8\}$	$\emptyset$	$\emptyset$	$\{s6, s9\}$	$\{s10\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Formal Operational Semantics: Architecture



A Testing theory for HAs

Latella, Massink [HASE01]

- Use of classical De Nicola & Hennessy framework
  (extensional/intensional characterizations; MUST/MAY preorders, etc)
- adapted to TS with input/output labeled transitions [Massink 96]
  (weaker than classical testing, stronger than trace; impact on experimenters)
- Modified semantics w.r.t. the original one [FMOODS 99]
  – Explicit separation of configurations and environment (= experimenter)
  – No reaction = deadlock
  – Core semantics **unchanged**. TOP Rule replaced.
  – Consistency with original semantics formally proven
- A proven correct algorithm for verifying testing equivalence of HAs
  (using bisimulation verification algorithms over *Finite Acceptance Automata*)

Relevant Testing Theory Definitions (1)

Def.3.1 *Experimenter.* An *experimenter* $\mathcal{O}$ on $\mathcal{I} \times \mathcal{U}$ is a tuple $(\omega_{\mathcal{O}}, \omega^0_{\mathcal{O}}, \iota_{\mathcal{O}}, \mathcal{I}, \mathcal{U}, \longrightarrow)$ where $\omega_{\mathcal{O}}$ is a set of *output* states, with $\omega^0_{\mathcal{O}} \in \omega_{\mathcal{O}}$ being the *initial* (output) state, $\iota_{\mathcal{O}} \subseteq \mathcal{U} \mapsto \omega_{\mathcal{O}}$ is the set of *input* states, each input state being a *total* function from $\mathcal{U}$ to output states. Finally $\longrightarrow \subseteq (\omega_{\mathcal{O}} \times \mathcal{I} \times \iota_{\mathcal{O}}) \cup (\omega_{\mathcal{O}} \times \{\tau, \mathbf{W}\} \times \omega_{\mathcal{O}})$ is the *transition relation*, with $(\mathcal{I} \cup \mathcal{U}) \cap \{\tau, \mathbf{W}\} = \emptyset$. ∘

Def.3.2 *Experimental System.* For all LTSs $s = (S, s^0, \mathcal{I} \times \mathcal{U}, \longrightarrow)$ over $\mathcal{I} \times \mathcal{U}$ and experimenters $\mathcal{O} = (\omega_{\mathcal{O}}, \omega^0_{\mathcal{O}}, \iota_{\mathcal{O}}, \mathcal{I}', \mathcal{U}', \longrightarrow)$ on $\mathcal{I}' \times \mathcal{U}'$, with $\mathcal{U} \subseteq \mathcal{U}'$, the experimental system $< \mathcal{O}, s >$ is the transition system $(\omega_{\mathcal{O}} \times S, (\omega^0_{\mathcal{O}}, s^0), \rightsquigarrow)$. The transition relation $\rightsquigarrow \subseteq (\omega_{\mathcal{O}} \times S) \times (\omega_{\mathcal{O}} \times S)$ is the smallest relation on $(\omega_{\mathcal{O}} \times S)$ induced by the deduction system below, where $\mathcal{O}_1, \mathcal{O}_2 \in \omega_{\mathcal{O}}$, $\mathcal{J} \in \iota_{\mathcal{O}}$, $s_1, s_2 \in S$ and for $((\mathcal{O}_1, s_1), (\mathcal{O}_2, s_2)) \in \rightsquigarrow$ we write $\mathcal{O}_1 \mid\mid s_1 \rightsquigarrow \mathcal{O}_2 \mid\mid s_2$

$$\frac{\mathcal{O}_1 \xrightarrow{i} \mathcal{J},\ s_1 \xrightarrow{(i,u)} s_2}{\mathcal{O}_1 \mid\mid s_1 \rightsquigarrow (\mathcal{J}\ u) \mid\mid s_2} \qquad \frac{\mathcal{O}_1 \xrightarrow{\tau} \mathcal{O}_2}{\mathcal{O}_1 \mid\mid s_1 \rightsquigarrow \mathcal{O}_2 \mid\mid s_1}$$

The Success set of the Experimental System is the set:

$\{\mathcal{O}_1 \in \omega_{\mathcal{O}} \mid \exists \mathcal{O}_2 \in \omega_{\mathcal{O}}.\ \mathcal{O}_1 \xrightarrow{\mathbf{W}} \mathcal{O}_2\}$ ∘

Relevant Testing Theory Definitions (2)

Def.3.3 *Computations.*
A *computation* for experimental system $< \mathcal{O}, s >$ is a sequence of the form:

$$\mathcal{O}_0 \mid\mid s_0 \leadsto \mathcal{O}_1 \mid\mid s_1 \leadsto \mathcal{O}_2 \mid\mid s_2 \leadsto \ldots \mathcal{O}_k \mid\mid s_k \leadsto \ldots$$

which is *maximal*, i.e. either it is infinite or it is finite with terminal element $\mathcal{O}_n \mid\mid s_n$ which has the property that $\mathcal{O}_n \mid\mid s_n \leadsto \mathcal{O}' \mid\mid s'$ for no pair $\mathcal{O}'$, s'. $\mathcal{O}_0$ and s_0 are the initial states of $\mathcal{O}$ and s. ◦

Def.3.4 *Successful Computations.*
A computation is *successful* iff $\mathcal{O}_k \in$ *Success* for some $k \geq 0$. ◦

Relevant Testing Theory Definitions (3)

Def.3.5 *Testing Equivalence.*
Let $\mathrm{Result}(\mathcal{O}, s) \subseteq \{\top, \bot\}$ be defined as follows:

$\top \in \mathrm{Result}(\mathcal{O}, s)$ iff $\mathrm{Comp}(\mathcal{O}, s)$ contains a successful computation

$\bot \in \mathrm{Result}(\mathcal{O}, s)$ iff $\mathrm{Comp}(\mathcal{O}, s)$ contains an unsuccessful computation

We say that LTSs s over $\mathcal{I} \times \mathcal{U}$ and s' over $\mathcal{I}' \times \mathcal{U}'$ are *testing equivalent*, written $s \sim s'$, iff for all experimenters $\mathcal{O}$ on $(\mathcal{I} \cup \mathcal{I}') \times (\mathcal{U} \cup \mathcal{U}')$ $\mathrm{Result}(\mathcal{O}, s) = \mathrm{Result}(\mathcal{O}, s')$ $\circ$

Relevant Testing Theory Definitions (4)

Def.3.6 *Testing Preorders.*
For all LTSs s over $\mathcal{I} \times \mathcal{U}$ and s' over $\mathcal{I}' \times \mathcal{U}'$ we let

i) $s \sqsubset\!\!\!\sim_{MAY} s'$ iff for every $\mathcal{O}$ on $(\mathcal{I} \cup \mathcal{I}') \times (\mathcal{U} \cup \mathcal{U}')$,
if $\top \in \mathrm{Result}(\mathcal{O}, s)$ then $\top \in \mathrm{Result}(\mathcal{O}, s')$

ii) $s \sqsubset\!\!\!\sim_{MUST} s'$ iff for every $\mathcal{O}$ on $(\mathcal{I} \cup \mathcal{I}') \times (\mathcal{U} \cup \mathcal{U}')$,
if $\bot \notin \mathrm{Result}(\mathcal{O}, s)$ then $\bot \notin \mathrm{Result}(\mathcal{O}, s')$

iii) $s \sqsubset\!\!\!\sim s'$ iff $(s \sqsubset\!\!\!\sim_{MAY} s') \wedge (s \sqsubset\!\!\!\sim_{MUST} s')$ $\circ$

$s \eqsim s'$ iff $s \sqsubset\!\!\!\sim s' \wedge s' \sqsubset\!\!\!\sim s$

Relevant Testing Theory Results

Lemma.3.1 *Correspondence.*
For all LTSs s over $\mathcal{I} \times \mathcal{U}$ and s' over $\mathcal{I}' \times \mathcal{U}'$, $s \sim s'$ iff $s \eqsim s'$ $\square$

Lemma.3.2 *MAY, MUST and Language.*
For all LTSs s over $\mathcal{I} \times \mathcal{U}$ and s' over $\mathcal{I}' \times \mathcal{U}'$ the following holds:

i) $s \sqsubseteq\!\!\!\!\sim_{MAY} s'$ iff $(L\ s) \subseteq (L\ s')$

ii) $s \sqsubseteq\!\!\!\!\sim_{MUST} s'$ implies $(L\ s') \subseteq (L\ s)$ $\square$

where the *language* $(L\ s)$ of s is the set

$$L\ s = \{\gamma \in (\mathcal{I} \times \mathcal{U})^* \mid \exists s'.\ s \xrightarrow{\gamma} s'\}$$

Semantics

Latella, Massink [HASE01]

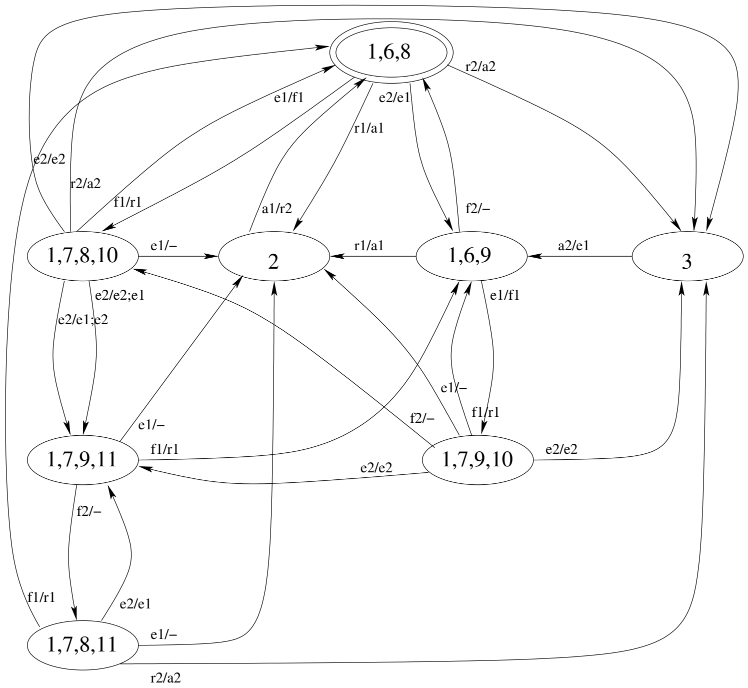
The *labeled transition system* of HA $H = (F, E, \rho)$ is:

$\mathtt{TS_{IO}(H)} = (\mathrm{Conf}_H, \mathcal{C}^0_H, E \times IQ, \longrightarrow)$ with

- Conf_H: the (finite) set of possible *configurations* of H
- $\mathcal{C}^0_H$: the initial *configuration*;
- IQ: the set of possible queues
- $\longrightarrow \subseteq \mathrm{Conf}_H \times (E \times IQ) \times \mathrm{Conf}_H$: the STEP transition relation.

$\mathcal{C} \xrightarrow{(e,\mathcal{E})} \mathcal{C}'$ is defined via a system of deduction rules.

Example



LTS with input/output labels

Conformance Testing HAs

Gnesi, Latella, Massink [TR CNUCE-B04-2001-16]

- Use of classical Tretmans et al. framework
  (quiescence, **ioco** relation)

- adapted to (generic input enabled) TS with input/output labeled transitions (new conformance relation $\sqsubseteq_{\mathrm{co}}$)

- New semantics (*stuttering* semantics)
  - Explicit representation of *stuttering* ($\Sigma \notin E$)
    (analogy to *quiescience*)
  - Core semantics **unchanged**. TOP Rule replaced.
  - Consistency with original semantics formally proven

- A proven complete (sound & exhaustive) algorithm for test case generation from HAs specifications

Relevant Conformance Theory Definitions

Def.4.1 *Conformance Relation.*
For all LTSs s over $\mathcal{I} \times \mathcal{U}$, s' over $\mathcal{I}' \times \mathcal{U}'$: $s' \ \sqsubseteq_{\mathsf{co}} \ s$ iff $\forall \gamma \in (L\ s), i \in \mathcal{I}.\ \mathtt{OUT}\ s'\ \gamma\ i \subseteq \mathtt{OUT}\ s\ \gamma\ i$ $\circ$

where

- $\mathtt{OUT}\ s\ \gamma\ i = \bigcup_{s' \in (s\ \mathtt{after}\ \gamma)} \{u \in \mathcal{U} \mid s' \ \xrightarrow{(i,u)}\}$
- $(s\ \mathtt{after}\ \gamma) = \{s' \mid s \xrightarrow{\gamma} s'\}$

Relevant Conformance Theory Results

Lemma.4.1 $\sqsubseteq_{\mathsf{co}}$ *and Language*
For all finite LTSs s over $\mathcal{I} \times \mathcal{U}$ and s' over $\mathcal{I}' \times \mathcal{U}'$ the following holds:

i) $(L\ s') \subseteq (L\ s)$ implies $s' \ \sqsubseteq_{\mathsf{co}}\ s$

ii) $\mathcal{I}' \subseteq \mathcal{I}$ and $s' \ \sqsubseteq_{\mathsf{co}}\ s$ implies $(L\ s') \subseteq (L\ s)$ $\square$

Stuttering Semantics

Gnesi, Latella, Massink [TR CNUCE-B04-2001-16]

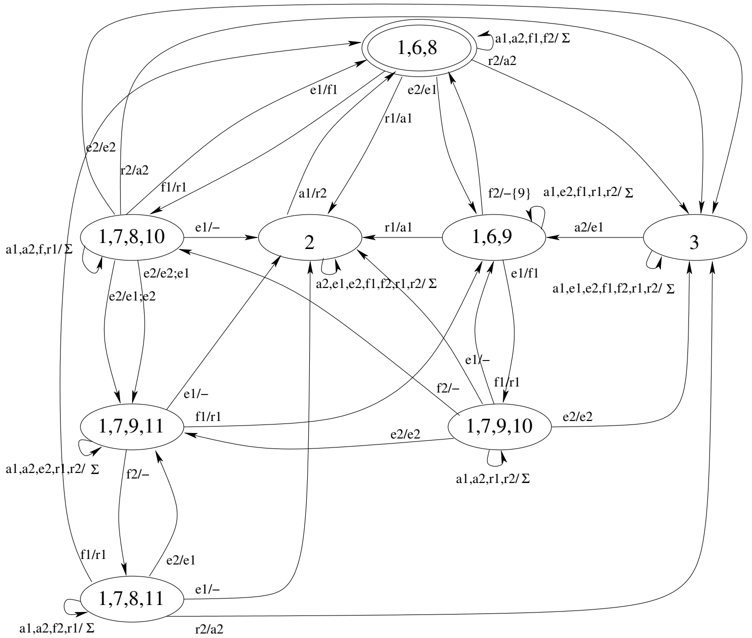
The *labeled transition system* of HA $H = (F, E, \rho)$ is:

${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) = (\mathrm{Conf}_H, \mathcal{C}^0_H, E \times {}^{\Sigma}IQ, \longrightarrow_{\Sigma})$ with

- Conf_H: the (finite) set of possible *configurations* of H
- $\mathcal{C}^0_H$: the initial *configuration*;
- ${}^{\Sigma}IQ$: the set of possible queues
  ($\mathcal{E} \in {}^{\Sigma}IQ \;\Rightarrow\; \Sigma$ can be an element of $\mathcal{E}$)
- $\longrightarrow_{\Sigma} \subseteq \mathrm{Conf}_H \times (E \times (IQ \cup \{\Sigma\})) \times \mathrm{Conf}_H$:
  the STEP transition relation.

$\mathcal{C} \xrightarrow{(e,\mathcal{E})}_{\Sigma} \mathcal{C}'$ is defined via a system of deduction rules.

Example



Stuttering LTS with input/output labels

Relating Testing and Conformance Preorders (Stuttering vs non-stuttering)

For HAs $H = (F, E, \rho)$ and $H' = (F', E', \rho')$

Lemma.5.1
if $E' \subseteq E$ then: ${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}') \sqsubseteq_{\mathsf{co}} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H})$ implies $\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}') \sqsubseteq\!\!\!\!\sim_{MAY} \mathtt{TS}_{\mathtt{IO}}(\mathtt{H})$
proof: Relationship between stuttering and non-stuttering semantics; Lemma 4.1

Lemma.5.2.
if $E' = E$ then: ${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}') \sqsubseteq_{\mathsf{co}} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H})$ implies $\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq\!\!\!\!\sim_{MUST} \mathtt{TS}_{\mathtt{IO}}(\mathtt{H}')$
proof: By contradiction using direct reasoning on computations

Note: *The implications are strictly one-way!*

Relating Testing and Conformance Preorders (in the stuttering framework)

For HAs $H = (F, E, \rho)$ and $H' = (F', E', \rho')$

Lemma.5.3

${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq_{MAY} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H'})$ implies ${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq_{\mathtt{co}} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H'})$

proof: Intensional characterization of $\sqsubseteq_{MAY}$; Lemma 4.1

if $E \subseteq E'$ then: ${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq_{\mathtt{co}} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H'})$ implies ${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq_{MAY} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H'})$

proof: Intensional characterization of $\sqsubseteq_{MAY}$; Lemma 4.1

Lemma.5.4

${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq_{MUST} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H'})$ implies ${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H'}) \sqsubseteq_{\mathtt{co}} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H})$

proof: Lemma 3.2; Lemma 4.1

Relating Testing and Conformance Preorders (in the stuttering framework - cont'd)

For HAs $H = (F, E, \rho)$, $H' = (F', E', \rho')$ and $H'' = (F'', E'', \rho'')$ with $E' \subseteq E$, the following holds:

Theorem.5.1 (Corollary) *Substitutivity properties*
${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}') \eqsim {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}'') \wedge {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}') \sqsubseteq_{\mathtt{co}} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H})$ implies
${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}'') \sqsubseteq_{\mathtt{co}} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H})$
(TE implementations conform to the same specification)
proof: Lemma 3.2; Lemma 4.1; def. of $\eqsim$; Set Theory

${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}') \sqsubseteq_{\mathtt{co}} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \wedge {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}'') \eqsim {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H})$ implies
${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}') \sqsubseteq_{\mathtt{co}} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}'')$
(An implementation conforms to TE specifications)
proof: Lemma 3.2; Lemma 4.1; def. of $\eqsim$; Set Theory

Relating stuttering and non-stuttering semantics (the expected result)

For all UMLSDs H on E and H' on E' the following holds:

${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq\!\!\!\!\sim_{MAY} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}')$ implies $\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq\!\!\!\!\sim_{MAY} \mathtt{TS}_{\mathtt{IO}}(\mathtt{H}')$
proof: Intensional characterization of $\sqsubseteq\!\!\!\!\sim_{MAY}$; Relationship between stuttering and non-stuttering semantics

if $E = E'$ then ${}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq\!\!\!\!\sim_{MUST} {}^{\Sigma}\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}')$ implies $\mathtt{TS}_{\mathtt{IO}}(\mathtt{H}) \sqsubseteq\!\!\!\!\sim_{MUST} \mathtt{TS}_{\mathtt{IO}}(\mathtt{H}')$
proof: Lemma 3.2; Lemma 4.1; Lemma 5.2 (req. $E = E'$)

Future Work

- test selection
- relationship between the proposed semantics and other approaches (like I/O-LTSs)
- implementation and link to existing tools.
- extension to mobility.