AVACS Phase 2 Paradigm: "Design Meets Verification"

- Example: Enhanced Fischer's Mutex for 2 CS
- Abstract: Extended Timed Automata (Shared) Data

For Extended Timed Automata (ETA)

Interplay of 5 Structural Transformations

- A Workshop Package on "Structure and Hierarchy"

Joint work with Ernst-Rüdiger Oldenbourg

Muni: Swaminathan, University of Oldenburg
Extended timed automata: locations + clocks + shared data variables

- Design meets verification for real-time systems
- Complex equivalence simple side-condition
- Distinct parallelism: \( A_1 \uparrow A_2 \equiv A_2 \uparrow A_1 \quad \text{if} \quad A_1 \cap A_2 \neq \emptyset \)
- Simplify verification
to simplify verification
- "Design meets verification": restructure system design

Easier to verify

Transformations:
- Structural
- Examining dependencies
- Streamlining system structure much

Possible solution: reduce parallelism by

- Between system and environment
- Interactions between system components
- Complexity of main source of system complexity

(2) The "Design meets verification" Paradigm
Example: Fisher's Mutex for two critical sections
A parallel context preserving embedding locally ETX separation in

\[ \text{Axiom: Identity and functional (syntactic) conditions}\]

Double Fischer however comprises of all instances of \( \Omega \).

\( \Omega = \text{In} (\text{Lo}_A) \)
\( \Omega = \text{In} (\text{Lo}_B) \)
\( \Omega = \text{In} (\text{Lo}_C) \)
\( \Omega = \text{In} (\text{Lo}_D) \)

Memoryless at \( \text{Lo}_A \) \( \Rightarrow \) initial conditions established \( \text{In} (\text{Lo}_A) \)

w.r.t. \( \text{In} \) initial conditions renaming modulo location renaming not a congruence some sets of reachable states

\[ \{ A \} \]

\[ \{ A \} \]

\[ \{ A \} \]

\[ \{ A \} \]

\[ \{ A \} \]

Separation of ETX: From \( \text{Lo}_A \) to \( \text{Lo}_B \)
Proposed solution for P.R. P.R.

- Local time semantics

- Bengtsson et al. 98, 4 many others

\[ \frac{dx}{dt} = \frac{dy}{dt} = \frac{dz}{dt} = 0 \]

This is not sufficient for \( \psi \) with global time

- In distinguishing clocks for different actions

- \( x \) and \( y \) are synchronous \( \Rightarrow \) \( e \) and \( f \)

- Synchronously evolving clocks \( \Rightarrow \) timing-induced dependencies

- \( (x, y, z) \) in reaching (4, 6)

- (c) do simultaneously

- \( x \) and \( f \) do not disable each other

- \( x \) do in parallel order reduction

- Independence (4) as in partial order reduction

- Dependences (5) in ETA due to shared data, since actions, \( y \) clocks
shared data is a sync action to a syntactic inspection of EHA.

So by wrapping EHA, \( \neg \) reduces true parameters arbitrarily violating \( \neg \).

This \( \forall x, y \in [A, B] \).

\( \text{true parameters arbitrarily violating at } \neg \).

Eliminates timing-imposed dependencies

\( = [A] \) solution: \( \neg \) wrap EHA to mimic local time with sync clocks.

But eliminating timing-imposed \( \neg \) still desirable.

\( \times \) local time semantics not applicable to our case-study.

\( \times \) universal most naturally supports globally sync clocks.

\( \times \) extra reference clocks for syncs.

\( \times \) local time semantics as in p.o.r. introduces.

(6) Wrapping on EHA \([B]\) for \( \neg \) with sync clocks.
DF satisfies (a) \( \triangleright \) preserves DMx

5. Permit reordering of executions so as to preserve

\[ (a) \quad B \triangleright A \quad \text{access only} \]

In DP:

\[ (b) \quad A \triangleright B \quad \text{access only} \]

(c) \( A \triangleright B \quad \text{Innermost wrapped:} \quad [A \triangleright B] \quad [B \triangleright A] \quad [A \triangleright B] \]

\[ (d) \quad A \triangleright B, A \triangleright B \quad \text{are all memoryless at } \forall \alpha \land \beta \]

\[ \forall \alpha \quad \exists \beta \quad \text{SDP} \]

\[ \text{when} \quad \begin{pmatrix} \forall \alpha \triangleright \beta \\ \forall \alpha \triangleright \beta \end{pmatrix} \quad \equiv \quad \begin{pmatrix} \forall \alpha \triangleright \beta \\ \forall \alpha \triangleright \beta \end{pmatrix} \quad \equiv \quad \begin{pmatrix} \forall \alpha \triangleright \beta \\ \forall \alpha \triangleright \beta \end{pmatrix} \]

\[ \text{Theorem} \quad \text{Separation} \]

(4) Local Separation in a Parallel Context: \( U \triangleright \beta \) under \( \forall \alpha \triangleright \beta \)
DF \equiv SDT = (1) 

\text{(8) Local Separation applied to Double Fischer}
\( SDF \) satisfies (b) \(-\omega\) \( \iff (\forall \beta \in \ell) \) preserves \( D \).

Proof intuition: \( \forall \beta \in \ell \) with \( \omega \)-wrapping.

\( (a) \) \( \forall \beta \in \ell \) only \( \iff \forall \beta \in \ell \). 

(1) \( \forall \beta \in \ell \) only.

(2) \( \forall \beta \in \ell \).

(3) \( \forall \beta \in \ell \).

(4) \( \forall \beta \in \ell \).

(5) \( \forall \beta \in \ell \).

(6) \( \forall \beta \in \ell \).

(7) \( \forall \beta \in \ell \).
So it suffices to show $L = L_1 \lor L_2 = L_2$ whenever $L_1 = mxf \lor L_2 = mxf$ are preserved. By $\triangleleft$ between $(\log a, \log b)$ and $\triangleleft L_1 = mxf \lor L_2 = mxf$, $L = mxf$. (10) Lowering under $\not\exists x$ applied to $\Sigma_m$
by $\text{e}_1, \text{e}_2, \text{e}_3$

established at $\text{e}_4$

conditions

Initail\n
$$\text{A}_4 \equiv \text{e}_4$$

Synthetic restrictions of $\text{A}$ from $\text{e}_0$

entering $\text{A}$ having $\text{e}_4$ as their target, while retaining

- $\text{A}_4$ obtained from $\text{A}_4$ by removing those edges

$\text{A}_4, \text{A}_4 \equiv \text{e}_4$

when $\text{A}_4$ is memoryless at $\text{e}_4$ [Flattening Thm]

$\text{A}_4 \equiv \text{e}_4$

possibly eliminates cycle thrgh memoryless locations

Idea: exploit memorylessness to remove superfluous edges

(II) Flattening ETH
CS

- limited resolution of a model a larger

while A* starts in A

B might involve

while B stops in B

no more than once,

no more than once,

if a transition entering A enables a

while B stops in B

no more than once,

no more than once,

- memoryless at A.

- Flattening them under if A ∈ B || B

- not a congruence w.r.t. ||, so more conditions needed

Local Flattening in a Parallel Context
\[ L = \alpha \cdot L \]

Thus, \( L \subseteq \alpha \cdot L \), thus preserves MX.

For local flattening (a), (b), (c), (d) for local flattening

Both \( R' \neq B' \) satisfy conditions

\( L = \alpha \cdot L \) local flattening at layer \( L' \)
\[ L = A \mid B \equiv (A_0 \And A_1) + (B_0 \Or B_1) + (A_0 \Or B_1) + (B_0 \And A_1) \]

\[ \text{Expansion of } A \mid B \equiv A_0 B_1 + A_1 B_0 \text{ into } A_0 + \text{and} \]

\[ R = B_0 \]

\[ A = B_0 \]

\[ B = B_0 \]

\[ x = 0 \]

\[ l = 0 \]

\[ y = 0 \]

\[ w = 1 \]

\[ z = 1 \]

\[ c = 1 \]

\[ d = 1 \]

\[ \text{Expansion of } A \mid B \equiv A_0 B_1 + A_1 B_0 \text{ into } A_0 + \text{and} \]
- Checking under $\delta$ does not apply. $\forall i, R_i, B_i, L_{\bar{L}} \forall i$.

- So $L_1 = \text{MAX}$. If $L_{\bar{L}} \neq L_{\bar{L}}$, then $M_1$ becomes trivial.

- So $R_1 = \text{MAX}$ if $A = \text{MAX}$, $A', \text{MAX} \lor (L_{\bar{L}}, L_{\bar{L}})$ where $L_0 = R_0 \lor L_{\bar{L}} \lor A_1\lor B_1 \lor \ldots$.

- $F_1 \equiv A + B + (L_{\bar{L}}, L_{\bar{L}})$ where $L_0 = R_0 \lor L_{\bar{L}} \lor A_1\lor B_1 \lor \ldots$

(15) Satisfaction of $M_1$ under expansion of $F_1$.
Theorem L-1: If $A > B$ and $A \not\perp\!\!\!\perp B$, then $A \gg B$. 

Proof: If $A > B$, then $A$ completes before $B$. Otherwise, if $A \not\perp\!\!\!\perp B$, then $A$ and $B$ are symmetric, so $B$ cannot complete before $A$. Therefore, $A \gg B$. 

Example:

1. $A = 1$, $B = 2$, $C = 3$, $D = 4$
2. $A = 2$, $B = 1$, $C = 3$, $D = 4$
\[ l^2 = \text{Max} \]

\[ l_1 = l \text{ Max} \]

\[ (A^* \cup B^*) = 1 \]

\[ (A^* \cap B^*) = \text{Max} \]

\[ D_{\text{Max}} \]

 Altogether, \( D^* = \text{Max} \)

 Similarly, \( l^2 = (A^* \cup B^*) = \text{Max} \)

\[ l_1 = (A^* \cap B^*) = \text{Max} \]

Thus, \( l^1 = (A^* \cup B^*) = \text{Max} \)

\[ l^1 \text{ is contained in } l^2 \]

Trivial, as \( l^1 \) is contained in \( l^2 \)

\[ l^1 \text{ Max} \]

\[ (A^* \cup B^*) = l \]

\[ (A^* \cap B^*) = \text{Max} \]

So \( l^1 \text{ Max} \)

\[ \text{By Transitive Law of Max} \]

\[ l^1 \text{ Max} \]

\[ (A^* \cup B^*) = l^1 \text{ Max} \]

\[ (A^* \cap B^*) = \text{Max} \]

\[ \text{Preservation of Max by Double Pricer} \]
Verifikation meets Design

- Shorter traces — easier debugging & diagnostics
- Of state space
- Comprehensive & Conceptual Simplification

Specifications [RWTH Aachen]

- Learning for (probabilistic) models
- Weak & strict sequencing [Univ. of Oxford]
- Learning extended to unl. sequence diagrams

- Randomized mutual exclusion
- Learning extended to probabilistic automata [with J.-P. Kaltofen]

- Of extended timed automata [real-value clocks + shared data]
- Flat learning, expansion, timed learning for the model
- Structural Transformations: Separation, Learning,

Design meets Verification „by means of“

(19) Extensions and Perspectives
A form of non-local flattening applied to

Timed Model Specifications

Incher's matrix analyzed using [Larsson, Steffen, Weise, '96]

Incher's matrix analyzed using [Muniz; Westphal, Pedelski, '12]

"Tr with disjoint activity" studied by [Muniz; Westphal, Pedelski, '12]

Kaherging studied by [Elrad & Frenze, '82;] [Jonsness et al., '90]

Kaherging studied by [Elrad & Frenze, '82]

Non-local flattening for TR by [Commoner, Fummi, '99]

Separation studied by [Cohen, '00] for Kleene Algebras

(20) Other Related Work
- AVACS DB.

- Our results published in Formal Asp. Comp. [12, 15].

- For your interest and attention:
  - to the Autumn School Organizers, and
  - to the Funding Agency (DFG).

(2) Many thanks