Basic Problems in Multi-View Modeling

Christos Stergiou
Jan Reineke, Stavros Tripakis

Saarland University, Aalto University,
University of California, Berkeley, University of Pennsylvania
Multi-View Modeling

Complex system ➔ many design teams ➔ many viewpoints ➔ many perspectives ➔ many models = views
Problem: View Consistency

Partially overlapping content $\rightarrow$ potential for contradictions

Industry: “system integration is the biggest issue”
Example

Inconsistent
Goals

• **Verification**: is a set of views consistent
• **Synthesis**: construct a system that produces a set of views
• We focus on behavioral/dynamic views
Outline

• An abstract formal framework for multi-view modeling
• Instantiation of the framework for discrete systems
• Instantiation for regular and $\omega$-regular languages
Outline

• An abstract formal framework for multi-view modeling
• Instantiation of the framework for discrete systems
• Instantiation for regular and \(\omega\)-regular languages
What are views, formally?

• **Semantically:** systems & views are **sets of behaviors**

• **Syntactically:** they can be any formal model that generates behaviors
  – e.g., automata, transition systems, differential equations, …

• Views are **derived** from systems
  – View = system “aspect”
How are views derived from systems?

- Intuition: view = projection of a system
- Generalization: views are defined by abstraction functions
  - System behavior domain: U
  - View behavior domain: D
  - Abstraction function: $a: U \rightarrow D$
- Example
  - System has 5 state variables: $x, y, z, a, b$
  - Abstraction keeps only 3: $x, y, z$
  - Abstraction function: projection (variable elimination)
Potential for Inconsistencies

• What if we don’t have the system, but only have some views?

• Example:
  – System has 5 variables: x, y, z, a, b
  – View V1 is over only 3 variables: x, y, z
  – View V2 is over x, a, b

  Overlapping views
  $\rightarrow$

  Potential for inconsistencies
What is View Consistency?

A set of views are consistent = \( \exists \text{witness system} \) that could generate those views
View Consistency, formally

• Given a set of abstraction functions $a_1, a_2, \ldots, a_n$

• Given a set of views $V_1, V_2, \ldots, V_n$

• The views are consistent if there exists witness system $S$ such that:
  – For all $i = 1, \ldots, n$: $V_i = a_i(S)$
Conformance

• Sometimes requiring “=” may be too strict
• Generalize to conformance relations
• Examples:
  – Top view must **over-approximate**: drop something without hitting the structure.
  – Top view must **under-approximate**: land a helicopter on the structure.
Outline

• An abstract formal framework for multi-view modeling
• Instantiation of the framework for discrete systems
• Instantiation for regular and $\omega$-regular languages
Discrete Systems = Symbolic Transition Systems

- **Fully observable** discrete system (FOS): \((X, \theta, \phi)\)
  - \(X\): set of variables
  - \(\theta\): formula on \(X\) characterizing initial states
  - \(\phi\): formula on \(X, X'\) characterizing transition relation
- Semantics: set of generated infinite behaviors
- Problem: **FOS not closed under projection**
  - E.g., system with two variables, \(x, y\), where \(x\) counts modulo 5, and \(y\) boolean \(x = 0\)
  - Projection on \(y\) is not representable as a FOS
Discrete Systems = Symbolic Transition Systems

- Discrete system with internal variables: $(X, Z, \theta, \phi)$
  - $X$ : set of variables
  - $Z$ : set of internal **unobservable** variables
  - $\theta$ : formula on $X$, $Z$ characterizing initial states
  - $\phi$ : formula on $X$, $Z$, $X'$, $Z'$ characterizing transition relation

- Trivially closed under projection (=hiding)
  - Move hidden variables from $X$ to $Z$

- Also closed under union & intersection
Views for Discrete Systems

• Abstraction functions: projections (hiding variables)
• Conformance relations: $\subseteq$, $\supseteq$, $=$
• View consistency checking:
  \[ V_1 = (Y_1, W_1, \theta_1, \phi_1), V_2 = (Y_2, W_2, \theta_2, \phi_2) \]
  – Trivial for $\subseteq$, $\supseteq$: empty and “true” systems
  – Witness system
    \[ S = (Y_1 \cup Y_2, W_1 \cup W_2, \theta_1 \land \theta_2, \phi_1 \land \phi_2) \]
  – This solve also synthesis of witness system
Outline

• An abstract formal framework for multi-view modeling
• Instantiation of the framework for discrete systems
• Instantiation for regular and ω-regular languages
Views for Languages

• Abstraction function: alphabet projection

\[ \Sigma_1 = \{a, b, c\}, \Sigma_2 = \{a, b\} \]

\[ w = abccac \]

\[ w' = h_{\Sigma_2}(w) = aba \]

• View consistency: given \( L_1, L_2 \) on \( \Sigma_1, \Sigma_2 \), find \( L \) such that

\[ h_{\Sigma_1}(L) = L_1, h_{\Sigma_2}(L) = L_2 \]
Language View Consistency (1/2)

• Inverse projection of \( L \) on \( \Sigma \), on alphabet \( \Sigma' \supseteq \Sigma \)
  \[
h^{-1}_{\Sigma'}(L) = \bigcup \{ L' \text{ language on } \Sigma' \mid h_{\Sigma}(L') = L \}
  \]

• \( L_1 \) and \( L_2 \) are consistent iff:
  \[
  L = h^{-1}_{\Sigma_1 \cup \Sigma_2}(L_1) \cap h^{-1}_{\Sigma_1 \cup \Sigma_2}(L_2)
  \]
  \[
  h_{\Sigma_1}(L) = L_1 \text{ and } h_{\Sigma_2}(L) = L_2
  \]

• How can we compute \( h^{-1}_{\Sigma} \) ?
Language View Consistency (2/2)

- $h_{\Sigma'}^{-1}$ for regular languages: self-loops

$$\sum = \{a\} \quad \Sigma' = \{a, b\}$$

- $h_{\Sigma'}^{-1}$ for $\omega$-regular languages: new states

$$L = a^\omega \quad L' = (a + b)^\omega \quad h_{\{a\}}(L') = a^* + a^\omega$$

$$L' = (b^*a)^\omega \quad h_{\{a\}}(L') = a^\omega$$
Conclusion

• First steps in formalizing multi-view topic
• “Basic Problems in Multi-View Modeling” TACAS 2014

• Future Work
  – Mixed notions of consistency
    • Combine over and under approximations
  – Different types of abstraction functions
    • Masking
    • Round abstraction
  – Heterogeneous views
    • Discrete vs. continuous vs. hybrid
Thank you

• Questions?