Library-Based Scalable Refinement Checking for Contract-Based Design

Antonio Iannopollo
Pierluigi Nuzzo
Stavros Tripakis
Alberto Sangiovanni-Vincentelli

EXCAPE ANNUAL MEETING
10-11 MARCH 2014
UC BERKELEY
Design as composition of library elements

- Libraries of components capture domain knowledge
- System design can be realized by composition of components
- Given a design built using libraries, we need to verify that system specifications are met

Contract-based design allows for compositional reasoning
- Each library component is described by a contract
- System design is a composition of contracts
- Each specification is also represented as a contract

The verification of a system design reduces to verify contract refinements

Our work is focused on making contract refinement check more scalable
Assume/Guarantee Contracts

Set P of ports, $P = I \cup O$
Set A of assumptions
Set G of guarantees

A contract expressed in form of $C = (A, G \vee \neg A)$ is in saturated form

A/G contract theory specifies
- Composition of contracts
- Refinement relations
- Compatibility and Consistency
Assume/Guarantee Contracts

Concrete representation of contracts using Linear Temporal Logic formulas:
• Assumptions and guarantees of a contract represented by a pair of LTL formulas, 
  \((\varphi_e, \varphi_s)\)

Given contracts \(C_1 = (\varphi_{e1}, \varphi_{s1})\) and \(C_2 = (\varphi_{e2}, \varphi_{s2})\)

• Composition is
  \(C_1 \otimes C_2 = ((\varphi_{e1} \land \varphi_{e2}) \lor \neg(\varphi_{s1} \land \varphi_{s2}), (\varphi_{s1} \land \varphi_{s2}))\)

• Refinement is
  \(C_1 \preceq C_2\) iff \(\varphi_{e2} \rightarrow \varphi_{e1}\) and \(\varphi_{s1} \rightarrow \varphi_{s2}\)
Refinement Check Problem (RPC)

Given
• A set of contracts $\mathcal{C}$
• A composition of contracts specifying a system $C_s = (C_1 \otimes C_2 \otimes \ldots \otimes C_n)$, where $C_1, C_2, \ldots, C_n \in \mathcal{C}$
• A property expressed as a contract $C_p$

We want to verify that $C_s \preceq C_p$

For LTL A/G contracts, RCP can be solved as a LTL satisfiability problem

Computationally expensive!
Library of contracts

A **library** is defined as a pair $L = (C, R)$, where

- $C = \{C_1, C_2, \ldots, C_n\}$ is a set of contracts
- $R$ is a **finite set of refinement relations** over contracts in $C$

E.g.

$C = \{C_1, C_2, C_3, C_4\}$,

$R = \{R_1, R_2\}$, where $R_1 = (C_1, C_2 \otimes C_3)$ (meaning that $(C_2 \otimes C_3) < C_1$) and $R_2 = (C_4, C_1 \otimes C_3)$

- Library is **hierarchically organized**
- **Strict refinement relations**, to avoid circular dependencies
- Once a library has been defined (by domain experts) it is possible to verify it as a **sequence of simple RCP instances**
Refinement Check Problem with Libraries (RCPL)

Given

- A **contract library** $L = (C, R)$
- A **composition of contracts** specifying a system $C_S = (C_1 \otimes C_2 \otimes \cdots \otimes C_n)$, where $C_1, C_2, \ldots, C_n \in C$
- A **property** expressed as a contract $C_p$

We want to verify that $C_S \preceq C_p$
Refinement Check Problem with Libraries (RCPL)

Consider the **library**

![Diagram of library with nodes C1, C2, C3, C4 and connections]

And the **design**

![Diagram of design with nodes C2, C3, C4 and connections]

(a connection means shared variables)

**abstract-contract:**

![Diagram of abstract-contract with nodes C2, C3, C1, C4 and connections]

**RCP:**

If $C_4 \preceq C_p$, we are done!

otherwise...

**propagate-no-abstraction:**

![Diagram of propagate-no-abstraction with nodes C2, C3 and connection]

---
The Aircraft Electrical Power System (EPS) Example

- The goal is **synthesize a control software** for the whole plant
- Physical elements consist of generators, buses, contactors, loads

**Library defined as set of subsystem contracts**

<table>
<thead>
<tr>
<th>Input vars(2): failures</th>
<th>Output vars(4): contactor status communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>G(gen_failure -&gt; X(gen_failure))</td>
<td></td>
</tr>
</tbody>
</table>

**Assumptions:**

G(!gen_failure -> sys_health) & G(gen_failure -> !sys_health)
The Aircraft Electrical Power System (EPS) Example

Set of **13 global specifications** expressed as contracts, such as:

- **$C_{p5}$**: If generators $G_L$ and $G_R$ are healthy, contactors $c_5$ and $c_6$ must be opened;
- **$C_{p6}$**: Contactors $c_2$ and $c_3$ cannot be both closed at the same time;
- ...

- **Up to 80% contract size reduction**
- **Up to 2 orders of magnitude performance improvement** *(refinement check execution time)*
Conclusion

• This work addresses the problem of performing scalable refinement checks for contract-based design
• Proposed solution leverages a pre-characterized library of contracts to reduce complexity, solving smaller scale problems
• We applied the proposed algorithm to verify controllers for aircraft electrical power systems

Future works include
  • algorithms for automatic mapping of library contracts to plant architectures
  • definition of benchmarks and quality metrics to estimate the effectiveness of a library
Questions?