A Framework for Contract-Based Reasoning: Motivation and Application

Sophie Quinton and Susanne Graf

VERIMAG, Université Joseph Fourier

FLACOS, Malta, November 28th, 2008
Outline

1 Introduction

2 A definition of contract-based verification framework

3 One application: a generic sufficient condition for dominance

4 Application to interface Input/Output automata

5 Conclusion and future work
Introduction

A definition of contract-based verification framework

One application: a generic sufficient condition for dominance

Application to interface Input/Output automata

Conclusion and future work
Introduction

Interface (or contract-based) theories

- A huge number of interface (or contract-based) theories have been developed (Henzinger, Larsen etc.)
- Specific to a notion of behavior
- Specific to a notion of interaction between components

Our approach

- What do these theories have in common?
- The BIP (Behavior, Interaction, Priority) framework clearly separates the notion of behavior from the notion of interaction.
- BIP allows to represent heterogeneous systems of components, from asynchronous to synchronous systems.
- We give a definition of contract-based verification framework.
The BIP framework

- Clearly separates behavior, interaction, priority
- Behaviors are represented as LTSs or Petri nets
- Interactions are represented as sets of ports
- Priorities are a preorder

BIP composition operators are sets of structured connectors which are sets of interactions.
Composition is associative and commutative.
1 Introduction

2 A definition of contract-based verification framework

3 One application: a generic sufficient condition for dominance

4 Application to interface Input/Output automata

5 Conclusion and future work
Contract-based verification framework

Definition (Contract-based verification framework)

A contract-based verification framework is given by a tuple $(B, \mathcal{P}, \Gamma, \parallel . \parallel, \theta)$, where:

- $B$ is a set of behaviors; each behavior $B \in B$ has as interface a set of ports denoted $\mathcal{P}_B$
- $\mathcal{P} = \bigcup_{B \in B} \mathcal{P}_B$
- $\Gamma$ is a set of BIP composition operators on subsets of $\mathcal{P}$
- $\parallel . \parallel : \Gamma \times 2^B \rightarrow B$ is a partial function defining a behavior semantics for the composition of behaviors
- $\theta : B \times \Gamma \rightarrow 2^{B \times B}$ is a refinement under context
A contract-based verification framework is given by a tuple $(\mathcal{B}, \mathcal{P}, \Gamma, \parallel . \parallel, \theta)$, where:

- $\mathcal{B}$ is a set of behaviors; each behavior $B \in \mathcal{B}$ has as interface a set of ports denoted $\mathcal{P}_B$
- $\mathcal{P} = \bigcup_{B \in \mathcal{B}} \mathcal{P}_B$
- $\Gamma$ is a set of BIP composition operators on subsets of $\mathcal{P}$
- $\parallel . \parallel : \Gamma \times 2^\mathcal{B} \rightarrow \mathcal{B}$ is a partial function defining a behavior semantics for the composition of behaviors $\parallel (\gamma, (B_1, \ldots, B_n))\parallel$, denoted $\gamma(B_1, \ldots, B_n)$, is defined iff $\gamma$ is defined on $\bigcup_{i=1}^n \mathcal{P}_{B_i}$
- $\parallel . \parallel$ preserves associativity and commutativity of the BIP composition operators ($\gamma_3(\gamma_{1,2}(B_1, B_2), B_3) = \gamma_{1}(B_1, \gamma_{2,3}(B_2, B_3))$ etc.)
- $\theta : \mathcal{B} \times \Gamma \rightarrow 2^{\mathcal{B} \times \mathcal{B}}$ is a refinement under context
A contract-based verification framework is given by a tuple $(\mathcal{B}, \mathcal{P}, \Gamma, \| \cdot \|, \theta)$, where:

- $\mathcal{B}$ is a set of behaviors; each behavior $B \in \mathcal{B}$ has as interface a set of ports denoted $\mathcal{P}_B$
- $\mathcal{P} = \bigcup_{B \in \mathcal{B}} \mathcal{P}_B$
- $\Gamma$ is a set of BIP composition operators on subsets of $\mathcal{P}$
- $\| \cdot \| : \Gamma \times 2^\mathcal{B} \rightarrow \mathcal{B}$ is a partial function defining a behavior semantics for the composition of behaviors
- $\theta : \mathcal{B} \times \Gamma \rightarrow 2^{\mathcal{B} \times \mathcal{B}}$ is a refinement under context

In the following we suppose given a contract-based verification framework $(\mathcal{B}, \mathcal{P}, \Gamma, \| \cdot \|, \theta)$. 
Refinement under context

**Definition (Context for an interface)**

Let $P \in 2^P$ be an interface. A context for $P$ is a pair $(E, \gamma)$ where $E$ is such that $P \cap \mathcal{P}_E = \emptyset$ and $\gamma$ is a composition operator defined on $P \sqcup \mathcal{P}_E$. 
Refinement under context

**Definition (Context for an interface)**

Let $P \in 2^P$ be an interface. A context for $P$ is a pair $(E, \gamma)$ where $E$ is such that $P \cap \mathcal{P}_E = \emptyset$ and $\gamma$ is a composition operator defined on $P \sqcup \mathcal{P}_E$. 
Refinement under context

**Definition (Context for an interface)**

Let $P \in 2^P$ be an interface. A context for $P$ is a pair $(E, \gamma)$ where $E$ is such that $P \cap \mathcal{P}_E = \emptyset$ and $\gamma$ is a composition operator defined on $P \sqcup \mathcal{P}_E$. 
Refinement under context

**Definition (Context for an interface)**

Let \( P \in 2^\mathcal{P} \) be an interface. A context for \( P \) is a pair \((E, \gamma)\) where \( E \) is such that \( P \cap \mathcal{P}_E = \emptyset \) and \( \gamma \) is a composition operator defined on \( P \sqcup \mathcal{P}_E \).

**Definition (Refinement under context)**

A refinement under context \( \theta : \mathcal{B} \times \Gamma \rightarrow 2^{\mathcal{B} \times \mathcal{B}} \) is a partial function s.t.

- For each context \((E, \gamma)\) for an interface \( P \), \( \theta(E, \gamma) \), denoted \( \sqsubseteq_{E,\gamma} \), is a reflexive and transitive binary relation over the set of behaviors with associated set of ports \( \mathcal{P}_B \).
- \( \theta \) is monotonic w.r.t composition as defined on the next slide.
Monotony of refinement under context

Definition (Monotony of refinement under context)

\( \theta \) is monotonic w.r.t. composition iff the following holds for any interface \( P \) and any context \((E, \gamma)\) for \( P \) such that \( E \) is of the form \( \gamma_E(E_1, E_2) \). For all \( B_1, B_2 \) behaviors on \( P \):

\[ B_1 \sqsubseteq_E,\gamma B_2 \implies \gamma_1(B_1, E_1) \sqsubseteq_{E_2,\gamma_2} \gamma_1(B_2, E_1) \]

where \( \gamma_1 \) and \( \gamma_2 \) are calculated from \( \gamma \) and \( \gamma_E \) for respectively \( P \sqcup \mathcal{P}_{E_1} \) and \( P \sqcup \mathcal{P}_{E_1} \sqcup \mathcal{P}_{E_2} \).
Monotony of refinement under context

Definition (Monotony of refinement under context)

$\theta$ is monotonic w.r.t. composition iff the following holds for any interface $P$ and any context $(E, \gamma)$ for $P$ such that $E$ is of the form $\gamma_E(E_1, E_2)$. For all $B_1, B_2$ behaviors on $P$:

$B_1 \sqsubseteq_{E,\gamma} B_2 \implies \gamma_1(B_1, E_1) \sqsubseteq_{E_2,\gamma_2} \gamma_1(B_2, E_1)$

where $\gamma_1$ and $\gamma_2$ are calculated from $\gamma$ and $\gamma_E$ for respectively $P \uplus \mathcal{P}_{E_1}$ and $P \uplus \mathcal{P}_{E_1} \uplus \mathcal{P}_{E_2}$. 
Monotony of refinement under context

**Definition (Monotony of refinement under context)**

\( \theta \) is monotonic w.r.t. composition iff the following holds for any interface \( P \) and any context \( (E, \gamma) \) for \( P \) such that \( E \) is of the form \( \gamma_E(E_1, E_2) \).

For all \( B_1, B_2 \) behaviors on \( P \):

\[ B_1 \sqsubseteq_{E, \gamma} B_2 \implies \gamma_1(B_1, E_1) \sqsubseteq_{E_2, \gamma_2} \gamma_1(B_2, E_1) \]

where \( \gamma_1 \) and \( \gamma_2 \) are calculated from \( \gamma \) and \( \gamma_E \) for respectively \( P \sqcup \mathcal{P}_{E_1} \) and \( P \sqcup \mathcal{P}_{E_1} \sqcup \mathcal{P}_{E_2} \).
Contract and satisfaction

Definition (Contract for an interface)

A contract $C$ for an interface $P$ consists of:
- a context $(A, \gamma)$ for $P$, where $A$ is called the assumption
- a behavior $G$ on $P$ called the guarantee

We write $C = (A, \gamma, G)$ rather than $((A, \gamma), G)$.

Definition (Satisfaction of a contract)

Let $C = (A, \gamma, G)$ be a contract for an interface $P$ and $B$ a behavior on $P$. $B$ satisfies $C$, denoted $B \models C$, iff $B \sqsubseteq_{A, \gamma} G$. 
Contract and satisfaction

Definition (Contract for an interface)

A contract $C$ for an interface $P$ consists of:
- a context $(A, \gamma)$ for $P$, where $A$ is called the assumption
- a behavior $G$ on $P$ called the guarantee

We write $C = (A, \gamma, G)$ rather than $((A, \gamma), G)$.

Definition (Satisfaction of a contract)

Let $C = (A, \gamma, G)$ be a contract for an interface $P$ and $B$ a behavior on $P$. $B$ satisfies $C$, denoted $B \models C$, iff $B \sqsubseteq_{A, \gamma} G$. 
Contract and satisfaction

Definition (Contract for an interface)
A contract $C$ for an interface $P$ consists of:

- a context $(A, \gamma)$ for $P$, where $A$ is called the assumption
- a behavior $G$ on $P$ called the guarantee

We write $C = (A, \gamma, G)$ rather than $((A, \gamma), G)$.

Definition (Satisfaction of a contract)
Let $C = (A, \gamma, G)$ be a contract for an interface $P$ and $B$ a behavior on $P$. $B$ satisfies $C$, denoted $B \models C$, iff $B \sqsubseteq_{A, \gamma} G$. 
Contract and satisfaction

Definition (Contract for an interface)
A contract $C$ for an interface $P$ consists of:
- a context $(A, \gamma)$ for $P$, where $A$ is called the assumption
- a behavior $G$ on $P$ called the guarantee

We write $C = (A, \gamma, G)$ rather than $((A, \gamma), G)$.

Definition (Satisfaction of a contract)
Let $C = (A, \gamma, G)$ be a contract for an interface $P$ and $B$ a behavior on $P$. $B$ satisfies $C$, denoted $B \models C$, iff $B \subseteq_{A, \gamma} G$. 
Dominance

Definition (Dominance)

- \( \{P_i\}_{i=1}^{n} \in 2^P \) a family of pairwise disjoint interfaces; \( P = \bigsqcup_{i=1}^{n} P_i \)
- \( C = (A, \gamma, G) \) a contract for \( P \)
- \( \forall i = 1..n, C_i = (A_i, \gamma_i, G_i) \) a contract for \( P_i \)
- \( \gamma_I \) a composition operator on \( P \) compatible with \( \gamma \) and the \( \gamma_i \)

\( C \) dominates \( \{C_i\}_{i=1}^{n} \) w.r.t. \( \gamma_I \) iff \( \forall B_1, ..., B_n \in B \) on resp. \( P_1, ..., P_n \):

\[ \forall i, B_i \models C_i \implies \gamma_I(B_1, ..., B_n) \models C \]
Dominance

**Definition (Dominance)**

- \( \{P_i\}_{i=1}^n \in 2^P \) a family of pairwise disjoint interfaces; \( P = \bigcup_{i=1}^n P_i \)
- \( C = (A, \gamma, G) \) a contract for \( P \)
- \( \forall i = 1..n, C_i = (A_i, \gamma_i, G_i) \) a contract for \( P_i \)
- \( \gamma_I \) a composition operator on \( P \) compatible with \( \gamma \) and the \( \gamma_i \)

\( C \) dominates \( \{C_i\}_{i=1}^n \) w.r.t. \( \gamma_I \) iff \( \forall B_1, ..., B_n \in \mathcal{B} \) on resp. \( P_1, ..., P_n \):

\[
\forall i, B_i \models C_i \implies \gamma_I(B_1, ..., B_n) \models C
\]

are compatible.
Introduction

A definition of contract-based verification framework

One application: a generic sufficient condition for dominance

Application to interface Input/Output automata

Conclusion and future work
Compositional reasoning

\[
\begin{align*}
S_1 \subseteq P_1 & \quad S_2 \subseteq P_2 \\
S_1 \cap S_2 \subseteq P_1 \cap P_2
\end{align*}
\]
Compositional reasoning

\[
\begin{align*}
S_1 &\subseteq P_1 & S_2 &\subseteq P_2 \\
\Rightarrow & & S_1 \cap S_2 &\subseteq P_1 \cap P_2
\end{align*}
\]

\[
\begin{align*}
S_1 &\subseteq P_1 & P_1 \cap S_2 &\subseteq P_2 \\
\Rightarrow & & S_1 \cap S_2 &\subseteq P_1 \cap P_2
\end{align*}
\]
Compositional reasoning

\[ S_1 \subseteq P_1 \quad S_2 \subseteq P_2 \]
\[ S_1 \cap S_2 \subseteq P_1 \cap P_2 \]

\[ S_1 \subseteq P_1 \quad P_1 \cap S_2 \subseteq P_2 \]
\[ S_1 \cap S_2 \subseteq P_1 \cap P_2 \]

\[ P_2 \cap S_1 \subseteq P_1 \quad P_1 \cap S_2 \subseteq P_2 \]
\[ S_1 \cap S_2 \subseteq P_1 \cap P_2 \]
Apparently circular reasoning

Definition (Apparent circular reasoning)

A framework \((\mathcal{B}, \mathcal{P}, \Gamma, \| . \|, \theta)\) allows apparent circular reasoning iff for any given interface \(P\), behavior \(B\) on \(P\), context \((E, \gamma)\) for \(P\) and contract \(C = (A, \gamma, G)\) for \(P\) we have:

\[
B \sqsubseteq_{A, \gamma} G \land E \sqsubseteq_{G, \gamma} A \implies B \sqsubseteq_{E, \gamma} G
\]
Apparent circular reasoning

Definition (Apparent circular reasoning)

A framework \((B, P, \Gamma, \| . \|, \theta)\) allows apparent circular reasoning iff for any given interface \(P\), behavior \(B\) on \(P\), context \((E, \gamma)\) for \(P\) and contract \(C = (A, \gamma, G)\) for \(P\) we have:

\[
B \sqsubseteq_{A,\gamma} G \land E \sqsubseteq_{G,\gamma} A \implies B \sqsubseteq_{E,\gamma} G
\]
Apparently circular reasoning

Definition (Apparent circular reasoning)

A framework \((B, \mathcal{P}, \Gamma, \parallel . \parallel, \theta)\) allows apparent circular reasoning iff for any given interface \(P\), behavior \(B\) on \(P\), context \((E, \gamma)\) for \(P\) and contract \(C = (A, \gamma, G)\) for \(P\) we have:

\[
B \sqsubseteq_{A, \gamma} G \land E \sqsubseteq_{G, \gamma} A \implies B \sqsubseteq_{E, \gamma} G
\]
A generic sufficient condition for dominance

**Theorem**

$C$ dominates $\{C_i\}_{i=1}^n$ w.r.t. $\gamma$ if:

\[
\begin{cases}
\gamma_I(G_1, \ldots, G_n) \models C \\
\forall i, \, \gamma_{\setminus i}(A, \gamma_{\setminus i}(G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n)) \models C_i^{-1}
\end{cases}
\]

with $\gamma_{\setminus i}$ standing for the restriction of $\gamma_I$ to $P \setminus P_i$,
$\gamma_{\setminus i}$ for the restriction of $\gamma$ to $P_E \cup P \setminus P_i$ and $C_i^{-1} = (G_i, \gamma_i, A_i)$. 

A generic sufficient condition for dominance

Theorem

$C$ dominates $\{C_i\}_{i=1}^n$ w.r.t. $\gamma$ if:

\[
\begin{cases}
\gamma_I(G_1, \ldots, G_n) \models C \\
\forall i, \gamma_{\setminus i}(A, \gamma_{\setminus i}(G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n)) \models C_i^{-1}
\end{cases}
\]

with $\gamma_{\setminus i}$ standing for the restriction of $\gamma_I$ to $P \setminus P_i$, $\gamma_{\setminus i}$ for the restriction of $\gamma$ to $P_E \cup P \setminus P_i$ and $C_i^{-1} = (G_i, \gamma_i, A_i)$. 
1 Introduction

2 A definition of contract-based verification framework

3 One application: a generic sufficient condition for dominance

4 Application to interface Input/Output automata

5 Conclusion and future work
Interface Input/Output automata

- Paper written by Larsen, Nyman and Wasowski (FM’06)
- Behaviours are I/O automata
- Interfaces are pairs of I/O automata \((E, S)\)
- Notion of refinement under context
- Composition of interfaces, comparison with interface automata

Our approach

- We encode output ports as triggers and input ports as synchrons.
- We show that the corresponding framework allows circular reasoning.
- We provide simple proofs to the first theorems of the paper.
Interface I/O automata as a contract-based verification framework

\((\mathcal{B}, \mathcal{P}, \Gamma, \| \cdot \|, \theta)\) is defined as:

- \(\mathcal{B}\) is a set of LTSs. For each LTS \(B\), \(\mathcal{P}_B\) denotes the set of its labels.
- \(\mathcal{P} = \bigcup_{B \in \mathcal{B}} \mathcal{P}_B\).
- \(\Gamma\) is the set of composition operators such that every connector has at most one trigger.
- \(\| \cdot \|\) is the standard BIP composition semantics for LTSs.
- For \(E, B_1, B_2 \in \mathcal{B}\) such that \(\mathcal{P}_{B_1} = \mathcal{P}_{B_2}\) and \(\gamma \in \Gamma\) defined on \(\mathcal{P}_E \sqcup \mathcal{P}_{B_1}\), 
  \(B_1 \sqsubseteq_{E, \gamma} B_2\) is defined as \(Tr(\gamma(B_1, E)) \upharpoonright \gamma \subseteq Tr(\gamma(B_2, E)) \upharpoonright \gamma\),
  where \(Tr(B)\) denotes the set of traces of \(B\) and \(\upharpoonright \gamma\) is the projection of a set of traces onto ports of \(\gamma\).

\(\theta\) as defined here is monotonous w.r.t with composition.

The framework \((\mathcal{B}, \mathcal{P}, \Gamma, \| \cdot \|, \theta)\) allows circular reasoning.
Theorem 3 of LarsenNW06

\[ \forall I_1, I_2, \]
\[ I_1 \sqsubseteq_{E_1, \gamma_1} S_1 \land I_2 \sqsubseteq_{E_2, \gamma_2} S_2 \implies \gamma_3(E, I_2) \sqsubseteq_{I_1, \gamma_1} E_1 \land \gamma_4(E, I_1) \sqsubseteq_{I_2, \gamma_2} E_2 \]

is equivalent to

\[ \gamma_3(E, S_2) \sqsubseteq_{S_1, \gamma_1} E_1 \land \gamma_4(E, S_1) \sqsubseteq_{S_2, \gamma_2} E_2 \]

Proof.

- Left-to-right implication is trivial since \( S_1 \sqsubseteq_{E_1, \gamma_1} S_1 \land S_2 \sqsubseteq_{E_2, \gamma_2} S_2 \) (for all \( E, \gamma, \sqsubseteq_{E, \gamma} \) is reflexive).

- Right-to-left implication: Let \( I_1 \) and \( I_2 \) be fixed. Suppose:

\[
\begin{align*}
\gamma_3(E, S_2) & \sqsubseteq_{S_1, \gamma_1} E_1 & (1) \\
\gamma_4(E, S_1) & \sqsubseteq_{S_2, \gamma_2} E_2 & (2) \\
I_1 & \sqsubseteq_{E_1, \gamma_1} S_1 & (3) \\
I_2 & \sqsubseteq_{E_2, \gamma_2} S_2 & (4)
\end{align*}
\]

We have to prove that \( \gamma_3(E, I_2) \sqsubseteq_{I_1, \gamma_1} E_1 \land \gamma_4(E, I_1) \sqsubseteq_{I_2, \gamma_2} E_2 \).
Proof of theorem 3 of LarsenNW06

Suppose:

\[
\begin{align*}
\gamma_3(E, S_2) & \sqsubseteq_{S_1, \gamma_1} E_1 \quad (1) \\
\gamma_4(E, S_1) & \sqsubseteq_{S_2, \gamma_2} E_2 \quad (2) \\
I_1 & \sqsubseteq_{E_1, \gamma_1} S_1 \quad (3) \\
I_2 & \sqsubseteq_{E_2, \gamma_2} S_2 \quad (4)
\end{align*}
\]

Goal: \( \gamma_3(E, I_2) \sqsubseteq_{I_1, \gamma_1} E_1 \land \gamma_4(E, I_1) \sqsubseteq_{I_2, \gamma_2} E_2 \).
Proof of theorem 3 of LarsenNW06

Suppose:

\[
\begin{align*}
\gamma_3(E,S_2) &\sqsubseteq_{S_1,\gamma_1} E_1 \quad (1) \\
\gamma_4(E,S_1) &\sqsubseteq_{S_2,\gamma_2} E_2 \quad (2) \\
l_1 &\sqsubseteq_{E_1,\gamma_1} S_1 \quad (3) \\
l_2 &\sqsubseteq_{E_2,\gamma_2} S_2 \quad (4)
\end{align*}
\]

Goal: \(\gamma_3(E,l_2) \sqsubseteq_{l_1,\gamma_1} E_1 \land \gamma_4(E,l_1) \sqsubseteq_{l_2,\gamma_2} E_2\).

Step 1: applying circular reasoning to (3) and (1), and to (4) and (2):

\[
\begin{align*}
l_1 &\sqsubseteq_{\gamma_3(E,S_2),\gamma_1} S_1 \quad (5) \\
l_2 &\sqsubseteq_{\gamma_4(E,S_1),\gamma_2} S_2 \quad (6)
\end{align*}
\]
Proof of theorem 3 of LarsenNW06

- Suppose:
  \[
  \begin{align*}
  \gamma_3(E, S_2) & \triangleq S_1, \gamma_1 E_1 \quad (1) \\
  \gamma_4(E, S_1) & \triangleq S_2, \gamma_2 E_2 \quad (2) \\
  I_1 & \triangleq E_1, \gamma_1 S_1 \quad (3) \\
  I_2 & \triangleq E_2, \gamma_2 S_2 \quad (4)
  \end{align*}
  \]

- Goal: \( \gamma_3(E, I_2) \triangleq I_1, \gamma_1 E_1 \land \gamma_4(E, I_1) \triangleq I_2, \gamma_2 E_2. \)

- Step 1: applying circular reasoning to (3) and (1), and to (4) and (2):
  \[
  \begin{align*}
  I_1 & \triangleq \gamma_3(E, S_2), \gamma_1 S_1 \quad (5) \\
  I_2 & \triangleq \gamma_4(E, S_1), \gamma_2 S_2 \quad (6)
  \end{align*}
  \]

- Step 2: monotony w.r.t. with composition, from (5) and (6):
  \[
  \begin{align*}
  \gamma_4(E, I_1) & \triangleq S_2, \gamma_2 \gamma_4(E, S_1) \quad (7) \\
  \gamma_3(E, I_2) & \triangleq S_1, \gamma_1 \gamma_3(E, S_2) \quad (8)
  \end{align*}
  \]
Proof of theorem 3 of LarsenNW06

- Suppose:

\[
\begin{align*}
\gamma_3(E, S_2) &\sqsubseteq_{S_1, \gamma_1} E_1 \quad (1) \\
\gamma_4(E, S_1) &\sqsubseteq_{S_2, \gamma_2} E_2 \quad (2) \\
I_1 &\sqsubseteq_{E_1, \gamma_1} S_1 \quad (3) \\
I_2 &\sqsubseteq_{E_2, \gamma_2} S_2 \quad (4)
\end{align*}
\]

- Goal: \(\gamma_3(E, I_2) \sqsubseteq_{I_1, \gamma_1} E_1 \land \gamma_4(E, I_1) \sqsubseteq_{I_2, \gamma_2} E_2\).

- Step 2: monotony w.r.t. with composition, from (5) and (6):

\[
\begin{align*}
\gamma_4(E, I_1) &\sqsubseteq_{S_2, \gamma_2} \gamma_4(E, S_1) \quad (7) \\
\gamma_3(E, I_2) &\sqsubseteq_{S_1, \gamma_1} \gamma_3(E, S_2) \quad (8)
\end{align*}
\]
Proof of theorem 3 of LarsenNW06

Suppose:

\[
\begin{align*}
\gamma_3(E, S_2) \sqsupseteq S_1, \gamma_1 \ E_1 & \quad (1) \\
\gamma_4(E, S_1) \sqsupseteq S_2, \gamma_2 \ E_2 & \quad (2) \\
I_1 \sqsubseteq E_1, \gamma_1 \ S_1 & \quad (3) \\
I_2 \sqsubseteq E_2, \gamma_2 \ S_2 & \quad (4)
\end{align*}
\]

Goal: \( \gamma_3(E, I_2) \sqsubseteq I_1, \gamma_1 \ E_1 \land \gamma_4(E, I_1) \sqsubseteq I_2, \gamma_2 \ E_2. \)

Step 2: monotony w.r.t. with composition, from (5) and (6):

\[
\begin{align*}
\gamma_4(E, I_1) \sqsubseteq S_2, \gamma_2 \gamma_4(E, S_1) & \quad (7) \\
\gamma_3(E, I_2) \sqsubseteq S_1, \gamma_1 \gamma_3(E, S_2) & \quad (8)
\end{align*}
\]

Step 3: applying transitivity of \( \sqsubseteq S_2, \gamma_2 \) (resp. \( \sqsubseteq S_1, \gamma_1 \)) to (7) and (2) (resp. (8) and (1)).

\[
\begin{align*}
\gamma_4(E, I_1) \sqsubseteq S_2, \gamma_2 \ E_2 & \quad (9) \\
\gamma_3(E, I_2) \sqsubseteq S_1, \gamma_1 \ E_1 & \quad (10)
\end{align*}
\]
Proof of theorem 3 of LarsenNW06

Suppose:

\[
\begin{align*}
\gamma_3(E, S_2) & \sqsubseteq_{S_1, \gamma_1} E_1 \quad (1) \\
\gamma_4(E, S_1) & \sqsubseteq_{S_2, \gamma_2} E_2 \quad (2) \\
I_1 & \sqsubseteq_{E_1, \gamma_1} S_1 \quad (3) \\
I_2 & \sqsubseteq_{E_2, \gamma_2} S_2 \quad (4)
\end{align*}
\]

Goal: \(\gamma_3(E, I_2) \sqsubseteq_{I_1, \gamma_1} E_1 \land \gamma_4(E, I_1) \sqsubseteq_{I_2, \gamma_2} E_2\).

Step 3: applying transitivity of \(\sqsubseteq_{S_2, \gamma_2}\) (resp. \(\sqsubseteq_{S_1, \gamma_1}\)) to (7) and (2) (resp. (8) and (1)).

\[
\begin{align*}
\gamma_4(E, I_1) & \sqsubseteq_{S_2, \gamma_2} E_2 \quad (9) \\
\gamma_3(E, I_2) & \sqsubseteq_{S_1, \gamma_1} E_1 \quad (10)
\end{align*}
\]
Proof of theorem 3 of LarsenNW06

- Suppose:
  \[
  \begin{align*}
  &\gamma_3(E, S_2) \sqsubseteq_{S_1, \gamma_1} E_1 \quad (1) \\
  &\gamma_4(E, S_1) \sqsubseteq_{S_2, \gamma_2} E_2 \quad (2) \\
  &I_1 \sqsubseteq_{E_1, \gamma_1} S_1 \quad (3) \\
  &I_2 \sqsubseteq_{E_2, \gamma_2} S_2 \quad (4)
  \end{align*}
  \]

- Goal: \( \gamma_3(E, I_2) \sqsubseteq_{I_1, \gamma_1} E_1 \land \gamma_4(E, I_1) \sqsubseteq_{I_2, \gamma_2} E_2. \)

- Step 3: applying transitivity of \( \sqsubseteq_{S_2, \gamma_2} \) (resp. \( \sqsubseteq_{S_1, \gamma_1} \)) to (7) and (2) (resp. (8) and (1)).

  \[
  \begin{align*}
  &\gamma_4(E, I_1) \sqsubseteq_{S_2, \gamma_2} E_2 \quad (9) \\
  &\gamma_3(E, I_2) \sqsubseteq_{S_1, \gamma_1} E_1 \quad (10)
  \end{align*}
  \]

- Step 4: applying circular reasoning to (9) and (4), and to (10) and (3):

  \[
  \begin{align*}
  &\gamma_4(E, I_1) \sqsubseteq_{I_2, \gamma_2} E_2 \quad (11) \\
  &\gamma_3(E, I_2) \sqsubseteq_{I_1, \gamma_1} E_1 \quad (12)
  \end{align*}
  \]
Theorem (Theorem 4 of LarsenNW06)

\[ \gamma_3(E, S_2) \sqsubseteq_{S_1, \gamma_1} E_1 \land \gamma_4(E, S_1) \sqsubseteq_{S_2, \gamma_2} E_2 \]

implies

\[ \forall I_1, I_2, I_1 \sqsubseteq_{E_1, \gamma_1} S_1 \land I_2 \sqsubseteq_{E_2, \gamma_2} S_2 \implies \gamma_5(I_1, I_2) \sqsubseteq_{E, \gamma} \gamma_5(S_1, S_2) \]
Theorem 4 of LarsenNW06

Theorem (Theorem 4 of LarsenNW06)

\[ \gamma_3(E, S_2) \sqsubseteq_{S_1, \gamma_1} E_1 \land \gamma_4(E, S_1) \sqsubseteq_{S_2, \gamma_2} E_2 \]

implies

\[ \forall I_1, I_2, I_1 \sqsubseteq_{E_1, \gamma_1} S_1 \land I_2 \sqsubseteq_{E_2, \gamma_2} S_2 \implies \gamma_5(I_1, I_2) \sqsubseteq_{E, \gamma} \gamma_5(S_1, S_2) \]

Theorem (Sufficient condition for dominance)

\( C \) dominates \( \{C_i\}_{i=1}^n \) w.r.t. \( \gamma \) if:

\[
\left\{ \begin{array}{l}
\gamma_I(G_1, \ldots, G_n) \models C \\
\forall i, \gamma_{\backslash i}(A, \gamma_{\backslash i}(G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n)) \models C_i^{-1}
\end{array} \right.
\]

with \( \gamma_{\backslash i} \) standing for the restriction of \( \gamma_I \) to \( P \backslash P_i \),
\( \gamma_{\backslash i} \) for the restriction of \( \gamma \) to \( P_E \cup P \backslash P_i \) and \( C_i^{-1} = (G_i, \gamma_i, A_i) \).
Theorem 4 of LarsenNW06

**Theorem (Theorem 4 of LarsenNW06)**

\[ \gamma_3(E, S_2) \sqsubseteq_{S_1, \gamma_1} E_1 \land \gamma_4(E, S_1) \sqsubseteq_{S_2, \gamma_2} E_2 \]

implies

\[ \forall I_1, I_2, I_1 \sqsubseteq_{E_1, \gamma_1} S_1 \land I_2 \sqsubseteq_{E_2, \gamma_2} S_2 \implies \gamma_5(I_1, I_2) \sqsubseteq_{E, \gamma} \gamma_5(S_1, S_2) \]

**Theorem (Sufficient condition for dominance)**

\( (A, \gamma, G) \) dominates \( \{(A_i, \gamma_i, G_i)\}_{i=1}^n \) w.r.t. \( \gamma \) if:

\[
\left\{ \begin{array}{c}
\gamma_I(G_1, \ldots, G_n) \sqsubseteq_{A, \gamma} G \\
\forall i, \gamma \backslash_i (A, \gamma \backslash_i (G_1, \ldots, G_i-1, G_{i+1}, \ldots, G_n)) \sqsubseteq_{G_i, \gamma_i} A_i
\end{array} \right\}
\]

with \( \gamma \backslash_i \) standing for the restriction of \( \gamma_I \) to \( P \backslash P_i \) and \( \gamma \backslash_i \) for the restriction of \( \gamma \) to \( P_E \cup P \backslash P_i \).
Theorem 4 of LarsenNW06

Theorem (Theorem 4 of LarsenNW06)

\[ \gamma_3(E, S_2) \sqsubseteq S_1, \gamma_1 E_1 \land \gamma_4(E, S_1) \sqsubseteq S_2, \gamma_2 E_2 \]

implies

\[ \forall I_1, I_2, I_1 \sqsubseteq E_1, \gamma_1 S_1 \land I_2 \sqsubseteq E_2, \gamma_2 S_2 \implies \gamma_5(I_1, I_2) \sqsubseteq E, \gamma \gamma_5(S_1, S_2) \]

Theorem (Sufficient condition for dominance)

\((A, \gamma, G)\) dominates \(\{(A_i, \gamma_i, G_i)\}_{i=1}^n\) w.r.t. \(\gamma\) if:

\[ \left\{ \begin{array}{l}
\gamma_3(E, S_2) \sqsubseteq S_1, \gamma_1 E_1 \\
\gamma_4(E, S_1) \sqsubseteq S_2, \gamma_2 E_2 \\
\gamma_5(S_1, S_2) \sqsubseteq E, \gamma S 
\end{array} \right. \]
1 Introduction

2 A definition of contract-based verification framework

3 One application: a generic sufficient condition for dominance

4 Application to interface Input/Output automata

5 Conclusion and future work
Conclusion and future work

Conclusion

- a definition of contract-based verification framework
- contracts with a structural part
- separation between assumption and guarantee
- a generic sufficient condition for dominance
- two motivating examples (see Larsen, Nyman, Wasowski, Modal I/O Automata for Interface and Product Line Theories)

Future work

- other proofs can be generalized
- take into account the structure of the set of behaviors
- generalize notions such as compatibility, consistency etc.