A Framework for Contract-Based Reasoning: Motivation and Application

Sophie Quinton and Susanne Graf

VERIMAG, Université Joseph Fourier

FLACOS, Malta, November 28th, 2008

Outline

Introduction

Interface (or contract-based) theories

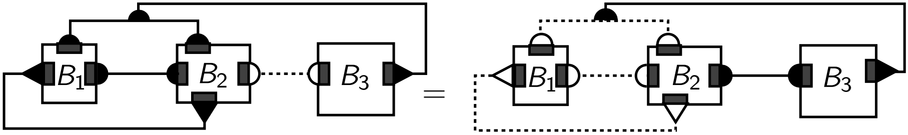
- A huge number of interface (or contract-based) theories have been developed (Henzinger, Larsen etc.)
- Specific to a notion of behavior
- Specific to a notion of interaction between components

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.

Contract-based verification framework

Definition (Contract-based verification framework)

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

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

Contract-based verification framework

Definition (Contract-based verification framework)

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

- $\mathcal{B}$ is a set of *behaviors*;
  each behavior $B \in \mathcal{B}$ has as interface a set of *ports* denoted $\mathcal{P}_B$
- $\mathcal{P} = \bigcup_{B \in \mathcal{B}} \mathcal{P}_B$
- Γ is a set of BIP *composition operators* on subsets of $\mathcal{P}$
- $\| . \| \ : \ \Gamma \times 2^{\mathcal{B}} \longrightarrow \mathcal{B}$ is a partial function defining a *behavior semantics* for the composition of behaviors
  $\|(\gamma, (B_1, \dots, B_n))\|$, denoted $\gamma(B_1, \dots, B_n)$, is defined iff γ is defined on $\bigsqcup_{i=1}^{n} \mathcal{P}_{B_i}$
  $\| . \|$ 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.)
- $\theta \ : \ \mathcal{B} \times \Gamma \longrightarrow 2^{\mathcal{B} \times \mathcal{B}}$ is a *refinement under context*

Contract-based verification framework

Definition (Contract-based verification framework)

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

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

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

Refinement under context

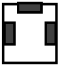
Definition (Context for an interface)

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

Refinement under context

Definition (Context for an interface)

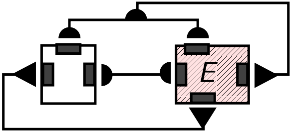
Let $P \in 2^{\mathcal{P}}$ be an interface. A context for P is a pair (E, γ) where E is such that $P \cap \mathcal{P}_E = \emptyset$ and γ is a composition operator defined on $P \sqcup \mathcal{P}_E$.



Refinement under context

Definition (Context for an interface)

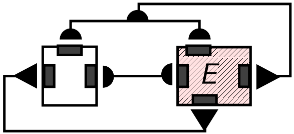
Let $P \in 2^{\mathcal{P}}$ be an interface. A context for P is a pair (E, γ) where E is such that $P \cap \mathcal{P}_E = \emptyset$ and γ is a composition operator defined on $P \sqcup \mathcal{P}_E$.



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, γ) where E is such that $P \cap \mathcal{P}_E = \emptyset$ and γ is a composition operator defined on $P \sqcup \mathcal{P}_E$.



Definition (Refinement under context)

A *refinement under context* $\theta \ : \ \mathcal{B} \times \Gamma \longrightarrow 2^{\mathcal{B} \times \mathcal{B}}$ is a partial function s.t.

- For each context (E, γ) 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 $\mathcal{P}_B$.
- θ is *monotonic w.r.t composition* as defined on the next slide.

Monotony of refinement under context

Definition (Monotony of refinement under context)

θ is monotonic w.r.t. composition iff the following holds for any interface P and any context (E, γ) 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 γ_1 and γ_2 are calculated from γ and γ_E for respectively $P \sqcup \mathcal{P}_{E_1}$ and $P \sqcup \mathcal{P}_{E_1} \sqcup \mathcal{P}_{E_2}$.

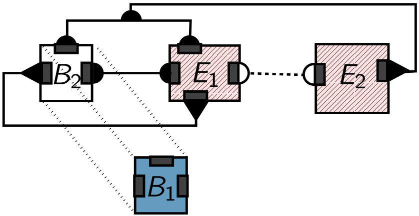
Monotony of refinement under context

Definition (Monotony of refinement under context)

θ is monotonic w.r.t. composition iff the following holds for any interface P and any context (E, γ) 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 γ_1 and γ_2 are calculated from γ and γ_E for respectively $P \sqcup \mathcal{P}_{E_1}$ and $P \sqcup \mathcal{P}_{E_1} \sqcup \mathcal{P}_{E_2}$.



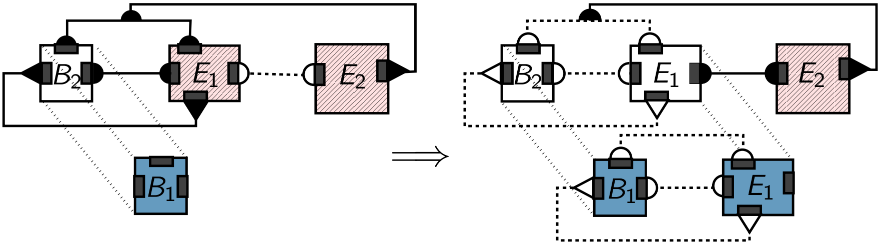
Monotony of refinement under context

Definition (Monotony of refinement under context)

θ is monotonic w.r.t. composition iff the following holds for any interface P and any context (E, γ) 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 γ_1 and γ_2 are calculated from γ and γ_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* $\mathcal{C}$ for an interface P consists of:

- a context (A, γ) 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 $\mathcal{C} = (A, \gamma, G)$ be a contract for an interface P and B a behavior on P. B *satisfies* $\mathcal{C}$, denoted $B \models \mathcal{C}$, iff $B \sqsubseteq_{A,\gamma} G$.

Contract and satisfaction

Definition (Contract for an interface)

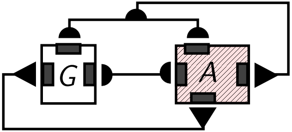
A *contract* $\mathcal{C}$ for an interface P consists of:

- a context (A, γ) 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 $\mathcal{C} = (A, \gamma, G)$ be a contract for an interface P and B a behavior on P. B *satisfies* $\mathcal{C}$, denoted $B \models \mathcal{C}$, iff $B \sqsubseteq_{A,\gamma} G$.



Contract and satisfaction

Definition (Contract for an interface)

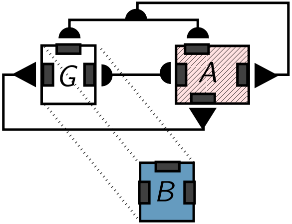
A *contract* $\mathcal{C}$ for an interface P consists of:

- a context (A, γ) 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 $\mathcal{C} = (A, \gamma, G)$ be a contract for an interface P and B a behavior on P. B *satisfies* $\mathcal{C}$, denoted $B \models \mathcal{C}$, iff $B \sqsubseteq_{A,\gamma} G$.



Contract and satisfaction

Definition (Contract for an interface)

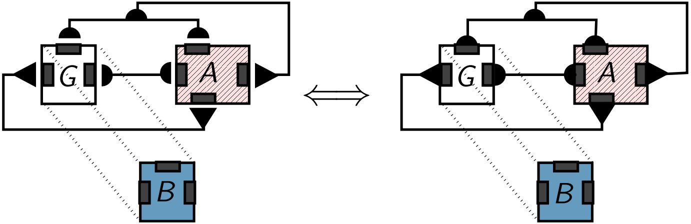
A *contract* $\mathcal{C}$ for an interface P consists of:

- a context (A, γ) 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 $\mathcal{C} = (A, \gamma, G)$ be a contract for an interface P and B a behavior on P. B *satisfies* $\mathcal{C}$, denoted $B \models \mathcal{C}$, iff $B \sqsubseteq_{A,\gamma} G$.



Dominance

Definition (Dominance)

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

$\mathcal{C}$ *dominates* $\{\mathcal{C}_i\}_{i=1}^n$ w.r.t. γ_I iff $\forall B_1, \ldots, B_n \in \mathcal{B}$ on resp. $P_1, \ldots, P_n$:

$$\forall i, B_i \models \mathcal{C}_i \implies \gamma_I(B_1, \ldots, B_n) \models \mathcal{C}$$

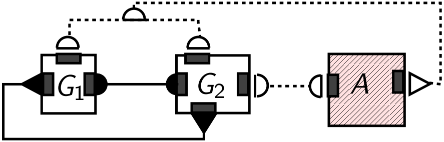
Dominance

Definition (Dominance)

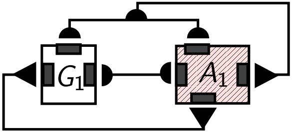
- $\{P_i\}_{i=1}^n \in 2^{\mathcal{P}}$ a family of pairwise disjoint interfaces; $P = \bigsqcup_{i=1}^n P_i$
- $\mathcal{C} = (A, \gamma, G)$ a contract for P
- $\forall i = 1..n$, $\mathcal{C}_i = (A_i, \gamma_i, G_i)$ a contract for P_i
- γ_I a composition operator on P compatible with γ and the γ_i

$\mathcal{C}$ *dominates* $\{\mathcal{C}_i\}_{i=1}^n$ w.r.t. γ_I iff $\forall B_1, \ldots, B_n \in \mathcal{B}$ on resp. $P_1, \ldots, P_n$:

$$\forall i, B_i \models \mathcal{C}_i \implies \gamma_I(B_1, \ldots, B_n) \models \mathcal{C}$$



and



are compatible.

Compositional reasoning

- $$\frac{S_1 \subseteq P_1 \qquad S_2 \subseteq P_2}{S_1 \cap S_2 \subseteq P_1 \cap P_2}$$

Compositional reasoning

- $$\frac{S_1 \subseteq P_1 \qquad S_2 \subseteq P_2}{S_1 \cap S_2 \subseteq P_1 \cap P_2}$$

- $$\frac{S_1 \subseteq P_1 \qquad P_1 \cap S_2 \subseteq P_2}{S_1 \cap S_2 \subseteq P_1 \cap P_2}$$

Compositional reasoning

- $$\frac{S_1 \subseteq P_1 \qquad S_2 \subseteq P_2}{S_1 \cap S_2 \subseteq P_1 \cap P_2}$$

- $$\frac{S_1 \subseteq P_1 \qquad P_1 \cap S_2 \subseteq P_2}{S_1 \cap S_2 \subseteq P_1 \cap P_2}$$

- $$\frac{P_2 \cap S_1 \subseteq P_1 \qquad P_1 \cap S_2 \subseteq P_2}{S_1 \cap S_2 \subseteq P_1 \cap P_2}$$

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, γ) for P and contract $\mathcal{C} = (A, \gamma, G)$ for P we have:

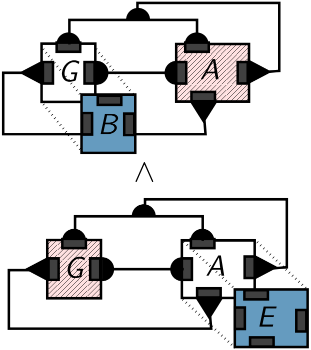
$$B \sqsubseteq_{A,\gamma} G \wedge E \sqsubseteq_{G,\gamma} A \implies B \sqsubseteq_{E,\gamma} G$$

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, γ) for P and contract $\mathcal{C} = (A, \gamma, G)$ for P we have:

$$B \sqsubseteq_{A,\gamma} G \wedge E \sqsubseteq_{G,\gamma} A \implies B \sqsubseteq_{E,\gamma} G$$

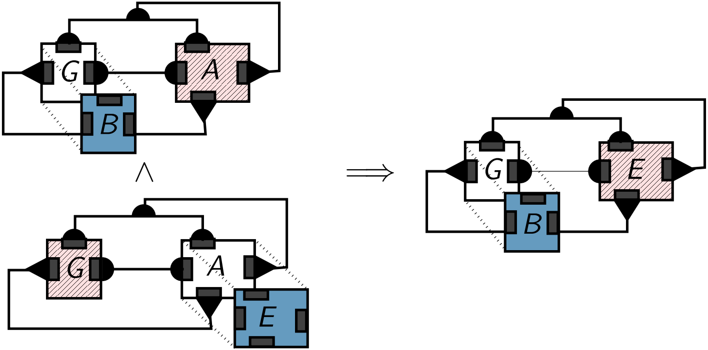


Apparently circular reasoning

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, γ) for P and contract $\mathcal{C} = (A, \gamma, G)$ for P we have:

$$B \sqsubseteq_{A,\gamma} G \wedge E \sqsubseteq_{G,\gamma} A \implies B \sqsubseteq_{E,\gamma} G$$



A generic sufficient condition for dominance

Theorem

$\mathcal{C}$ dominates $\{\mathcal{C}_i\}_{i=1}^{n}$ w.r.t. γ if:

$$\begin{cases} \gamma_I(G_1, \ldots, G_n) \models \mathcal{C} \\ \forall i, \gamma_{\setminus i}(A, \gamma_{I\setminus i}(G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n)) \models \mathcal{C}_i^{-1} \end{cases}$$

with $\gamma_{I\setminus i}$ standing for the restriction of γ_I to $P\setminus P_i$, $\gamma_{\setminus i}$ for the restriction of γ to $P_E \cup P\setminus P_i$ and $\mathcal{C}_i^{-1} = (G_i, \gamma_i, A_i)$.

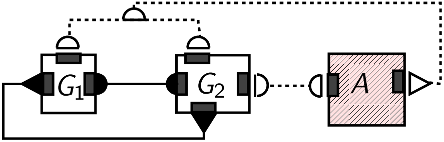
A generic sufficient condition for dominance

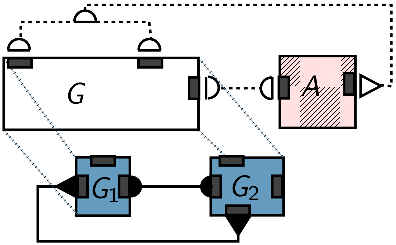
Theorem

$\mathcal{C}$ dominates $\{\mathcal{C}_i\}_{i=1}^{n}$ w.r.t. γ if:

$$\begin{cases} \gamma_I(G_1, \ldots, G_n) \models \mathcal{C} \\ \forall i, \gamma_{\setminus i}(A, \gamma_{I\setminus i}(G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n)) \models \mathcal{C}_i^{-1} \end{cases}$$

with $\gamma_{I\setminus i}$ standing for the restriction of γ_I to $P \setminus P_i$, $\gamma_{\setminus i}$ for the restriction of γ to $P_E \cup P \setminus P_i$ and $\mathcal{C}_i^{-1} = (G_i, \gamma_i, A_i)$.





Interface Input/Output automata

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

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

- $\mathcal{B}$ is a set of LTSs. For each LTS B, $\mathcal{P}_B$ denotes the set of its labels.
- $\mathcal{P} = \bigcup_{B \in \mathcal{B}} \mathcal{P}_B$.
- Γ is the set of composition operators such that every connector has at most one trigger.
- $\| . \|$ 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 γ.

θ as defined here is monotonous w.r.t with composition.
The framework $(\mathcal{B}, \mathcal{P}, \Gamma, \| . \|, \theta)$ allows circular reasoning.

Theorem 3 of LarsenNW06

Theorem (Theorem 3 of LarsenNW06)

$\forall I_1, I_2,$

$$I_1 \sqsubseteq_{E_1,\gamma_1} S_1 \wedge I_2 \sqsubseteq_{E_2,\gamma_2} S_2 \implies \gamma_3(E, I_2) \sqsubseteq_{I_1,\gamma_1} E_1 \wedge \gamma_4(E, I_1) \sqsubseteq_{I_2,\gamma_2} E_2$$

is equivalent to

$$\gamma_3(E, S_2) \sqsubseteq_{S_1,\gamma_1} E_1 \wedge \gamma_4(E, S_1) \sqsubseteq_{S_2,\gamma_2} E_2$$

Proof.

- Left-to-right implication is trivial since $S_1 \sqsubseteq_{E_1,\gamma_1} S_1 \wedge S_2 \sqsubseteq_{E_2,\gamma_2} S_2$ (for all $E, \gamma, \sqsubseteq_{E,\gamma}$ is reflexive).
- Right-to-left implication: Let I_1 and I_2 be fixed. Suppose:

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

We have to prove that $\gamma_3(E, I_2) \sqsubseteq_{I_1,\gamma_1} E_1 \wedge \gamma_4(E, I_1) \sqsubseteq_{I_2,\gamma_2} E_2$.

Proof of theorem 3 of LarsenNW06

- Suppose:

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

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

Proof of theorem 3 of LarsenNW06

- Suppose:

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

- Goal: $\gamma_3(E, I_2) \sqsubseteq_{I_1,\gamma_1} E_1 \wedge \gamma_4(E, I_1) \sqsubseteq_{I_2,\gamma_2} E_2$.
- Step 1: applying circular reasoning to (3) and (1), and to (4) and (2):

$$\begin{cases} I_1 \sqsubseteq_{\gamma_3(E,S_2),\gamma_1} S_1 & (5) \\ I_2 \sqsubseteq_{\gamma_4(E,S_1),\gamma_2} S_2 & (6) \end{cases}$$

Proof of theorem 3 of LarsenNW06

- Suppose:

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

- Goal: $\gamma_3(E, I_2) \sqsubseteq_{I_1,\gamma_1} E_1 \wedge \gamma_4(E, I_1) \sqsubseteq_{I_2,\gamma_2} E_2$.
- Step 1: applying circular reasoning to (3) and (1), and to (4) and (2):

$$\begin{cases} I_1 \sqsubseteq_{\gamma_3(E,S_2),\gamma_1} S_1 & (5) \\ I_2 \sqsubseteq_{\gamma_4(E,S_1),\gamma_2} S_2 & (6) \end{cases}$$

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

$$\begin{cases} \gamma_4(E, I_1) \sqsubseteq_{S_2,\gamma_2} \gamma_4(E, S_1) & (7) \\ \gamma_3(E, I_2) \sqsubseteq_{S_1,\gamma_1} \gamma_3(E, S_2) & (8) \end{cases}$$

Proof of theorem 3 of LarsenNW06

- Suppose:

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

- Goal: $\gamma_3(E, I_2) \sqsubseteq_{I_1,\gamma_1} E_1 \wedge \gamma_4(E, I_1) \sqsubseteq_{I_2,\gamma_2} E_2$.
- Step 2: monotony w.r.t. with composition, from (5) and (6):

$$\begin{cases} \gamma_4(E, I_1) \sqsubseteq_{S_2,\gamma_2} \gamma_4(E, S_1) & (7) \\ \gamma_3(E, I_2) \sqsubseteq_{S_1,\gamma_1} \gamma_3(E, S_2) & (8) \end{cases}$$

Proof of theorem 3 of LarsenNW06

- Suppose:

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

- Goal: $\gamma_3(E, I_2) \sqsubseteq_{I_1,\gamma_1} E_1 \wedge \gamma_4(E, I_1) \sqsubseteq_{I_2,\gamma_2} E_2$.
- Step 2: monotony w.r.t. with composition, from (5) and (6):

$$\begin{cases} \gamma_4(E, I_1) \sqsubseteq_{S_2,\gamma_2} \gamma_4(E, S_1) & (7) \\ \gamma_3(E, I_2) \sqsubseteq_{S_1,\gamma_1} \gamma_3(E, S_2) & (8) \end{cases}$$

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

$$\begin{cases} \gamma_4(E, I_1) \sqsubseteq_{S_2,\gamma_2} E_2 & (9) \\ \gamma_3(E, I_2) \sqsubseteq_{S_1,\gamma_1} E_1 & (10) \end{cases}$$

Proof of theorem 3 of LarsenNW06

- Suppose:

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

- Goal: $\gamma_3(E, I_2) \sqsubseteq_{I_1,\gamma_1} E_1 \wedge \gamma_4(E, I_1) \sqsubseteq_{I_2,\gamma_2} E_2$.
- Step 3: applying transitivity of $\sqsubseteq_{S_2,\gamma_2}$ (resp. $\sqsubseteq_{S_1,\gamma_1}$) to (7) and (2) (resp. (8) and (1)).

$$\begin{cases} \gamma_4(E, I_1) \sqsubseteq_{S_2,\gamma_2} E_2 & (9) \\ \gamma_3(E, I_2) \sqsubseteq_{S_1,\gamma_1} E_1 & (10) \end{cases}$$

Proof of theorem 3 of LarsenNW06

- Suppose:

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

- Goal: $\gamma_3(E, I_2) \sqsubseteq_{I_1,\gamma_1} E_1 \wedge \gamma_4(E, I_1) \sqsubseteq_{I_2,\gamma_2} E_2$.
- Step 3: applying transitivity of $\sqsubseteq_{S_2,\gamma_2}$ (resp. $\sqsubseteq_{S_1,\gamma_1}$) to (7) and (2) (resp. (8) and (1)).

$$\begin{cases} \gamma_4(E, I_1) \sqsubseteq_{S_2,\gamma_2} E_2 & (9) \\ \gamma_3(E, I_2) \sqsubseteq_{S_1,\gamma_1} E_1 & (10) \end{cases}$$

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

$$\begin{cases} \gamma_4(E, I_1) \sqsubseteq_{I_2,\gamma_2} E_2 & (11) \\ \gamma_3(E, I_2) \sqsubseteq_{I_1,\gamma_1} E_1 & (12) \end{cases}$$

Theorem 4 of LarsenNW06

Theorem (Theorem 4 of LarsenNW06)

$$\gamma_3(E, S_2) \sqsubseteq_{S_1,\gamma_1} E_1 \wedge \gamma_4(E, S_1) \sqsubseteq_{S_2,\gamma_2} E_2$$

implies

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

Theorem 4 of LarsenNW06

Theorem (Theorem 4 of LarsenNW06)

$$\gamma_3(E, S_2) \sqsubseteq_{S_1,\gamma_1} E_1 \wedge \gamma_4(E, S_1) \sqsubseteq_{S_2,\gamma_2} E_2$$

implies

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

Theorem (Sufficient condition for dominance)

$\mathcal{C}$ dominates $\{\mathcal{C}_i\}_{i=1}^n$ w.r.t. γ if:

$$\begin{cases} \gamma_I(G_1, \dots, G_n) \models \mathcal{C} \\ \forall i, \gamma_{\setminus i}(A, \gamma_{I\setminus i}(G_1, \dots, G_{i-1}, G_{i+1}, \dots, G_n)) \models \mathcal{C}_i^{-1} \end{cases}$$

with $\gamma_{I\setminus i}$ standing for the restriction of γ_I to $P \setminus P_i$, $\gamma_{\setminus i}$ for the restriction of γ to $P_E \cup P \setminus P_i$ and $\mathcal{C}_i^{-1} = (G_i, \gamma_i, A_i)$.

Theorem 4 of LarsenNW06

Theorem (Theorem 4 of LarsenNW06)

$$\gamma_3(E, S_2) \sqsubseteq_{S_1,\gamma_1} E_1 \wedge \gamma_4(E, S_1) \sqsubseteq_{S_2,\gamma_2} E_2$$

implies

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

Theorem (Sufficient condition for dominance)

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

$$\begin{cases} \gamma_I(G_1, \ldots, G_n) \sqsubseteq_{A,\gamma} G \\ \forall i, \gamma_{\setminus i}(A, \gamma_{I\setminus i}(G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n)) \sqsubseteq_{G_i,\gamma_i} A_i \end{cases}$$

with $\gamma_{I\setminus i}$ *standing for the restriction of* γ_I *to* $P\setminus P_i$ *and* $\gamma_{\setminus i}$ *for the restriction of* γ *to* $P_E \cup P\setminus P_i$.

Theorem 4 of LarsenNW06

Theorem (Theorem 4 of LarsenNW06)

$$\gamma_3(E, S_2) \sqsubseteq_{S_1,\gamma_1} E_1 \wedge \gamma_4(E, S_1) \sqsubseteq_{S_2,\gamma_2} E_2$$

implies

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

Theorem (Sufficient condition for dominance)

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

$$\begin{cases} \gamma_5(S_1, S_2) \sqsubseteq_{E,\gamma} S \\ \gamma_3(E, S_2) \sqsubseteq_{S_1,\gamma_1} E_1 \\ \gamma_4(E, S_1) \sqsubseteq_{S_2,\gamma_2} E_2 \end{cases}$$

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.