A Compositional Proof Rule for Coordination Logic
Highlights 2013, Paris

joint work with Bernd Finkbeiner

Leander Tentrup
Reactive Systems Group
Saarland University
September 21, 2013
What is Coordination Logic?

Logic of the *Distributed Synthesis Problem*

∃ *implementation* s.t. $\varphi$ holds?
What is Coordination Logic?

Logic of the *Distributed Synthesis Problem*

∃ *implementation* s.t. \( \varphi \) holds?

Set of strategies for output variables
What is Coordination Logic?

Logic of the *Distributed Synthesis Problem*

∃ implementation s.t. \( \varphi \) holds?

Set of strategies for output variables

LTL formula over input/output variables
Syntax

LTL
\[ x \mid \neg x \mid \varphi \lor \varphi \mid \varphi \land \varphi \mid \Box \varphi \mid \varphi U \varphi \mid \varphi \bar{U} \varphi \]

\[ x \in C \cup S \]

Strategic
Quantification

\[ \exists C \triangleright s. \varphi \mid \forall C \triangleright s. \varphi \]

\[ C \subseteq C, s \in S \]

Coordination variables
represent information given by the environment

\[ C \]

Strategy variables
represent strategic choices made based on visible information

\[ S \]

Leander Tentrup, Saarland University
Syntax

LTL

+ 

Strategic Quantification

LTL
\[ x \mid \neg x \mid \varphi \lor \varphi \mid \varphi \land \varphi \mid \varnothing \varphi \mid \varphi U \varphi \mid \varphi U \varphi \]
\[ x \in C \cup S \]

Strategic variables represent strategic choices made based on visible information

Strategy variables

Coordination variables represent information given by the environment

Coordination variables

synthesize strategy

C

S

Leander Tentrup, Saarland University

Highlights 2013, Paris
Syntax

LTL

\[ x \mid \neg x \mid \varphi \lor \varphi \mid \varphi \land \varphi \mid \Diamond \varphi \mid \varphi U \varphi \mid \varphi U \varphi \]

\[ x \in C \cup S \]

coordinate variables represent information given by the environment

strategy variables represent strategic choices made based on visible information

synthesize strategy

no control about strategy
Decidability

- Distributed Synthesis is undecidable
  - Coordination Logic is **undecidable**

A special case is decidable
- Syntactic restricted fragment of CL

Many practical synthesis problems are not in the fragment
- Goal: Complete Proof Framework for CL
Decidability

- Distributed Synthesis is undecidable
  - Coordination Logic is **undecidable**

- Special cases are decidable
  - Syntactic restricted fragment of CL
- Distributed Synthesis is undecidable
  - Coordination Logic is **undecidable**

- Special cases are decidable
  - Syntactic restricted fragment of CL

- Many practical synthesis problems are not in the fragment
  - **Goal:** Complete Proof Framework for CL
A Compositional Proof Rule

- CL formula $\Phi = \mathcal{H}(S) \cdot \varphi = QC_1 \triangleright s_1 \ldots QC_n \triangleright s_n \cdot \varphi$ in PNF
- Suitable cut-set $S_{cut} = \{s_1, \ldots, s_k\} \subseteq S$

\begin{align*}
(R_1) & \vdash \mathcal{H}(S_{cut}) \cdot \psi \\
(R_2) & \vdash \mathcal{H}(S \setminus S_{cut}) \cdot \varphi' \\
(R_3) & \vdash \mathcal{H}(\bigvee(S)) \cdot \psi \land \varphi' \rightarrow \varphi \\
\hline \\
& \vdash \Phi
\end{align*}
A Compositional Proof Rule

- CL formula $\Phi = \mathcal{H}(S). \varphi = QC_1 \triangleright s_1 \ldots QC_n \triangleright s_n \cdot \varphi$ in PNF
- Suitable cut-set $S_{cut} = \{s_1, \ldots, s_k\} \subseteq S$

\begin{align*}
(R_1) & \vdash \mathcal{H}(S_{cut}) \cdot \psi \\
(R_2) & \vdash \mathcal{H}(S \setminus S_{cut}) \cdot \varphi' \\
(R_3) & \vdash \mathcal{H}_\forall(S) \cdot \psi \land \varphi' \rightarrow \varphi \\
\hline
\vdash \Phi
\end{align*}
A Compositional Proof Rule

- CL formula $\Phi = H(S). \varphi = QC_1 \triangleright s_1 \ldots QC_n \triangleright s_n. \varphi$ in PNF
- Suitable cut-set $S_{cut} = \{s_1, \ldots, s_k\} \subseteq S$

$$
\begin{align*}
(R_1) & \vdash H(S_{cut}) . \psi & \text{simplified} \\
(R_2) & \vdash H(S \setminus S_{cut}) . \varphi' \\
(R_3) & \vdash H_{\lor}(S) . \psi \land \varphi' \rightarrow \varphi \\
\hline \\
\vdash \Phi
\end{align*}
$$

Leander Tentrup, Saarland University
Completeness

The proof rule is complete for formulas

- in the *universal-hierarchical fragment* of Coordination Logic, and
- in Prenex Normal Form (PNF)

Example

\[
\exists \{b, c\} \triangleright x_1. \forall \{a\} \triangleright y_1. \exists \{a, c\} \triangleright x_2. \exists \{a, d\} \triangleright x_3. \forall \{a, c\} \triangleright y_2. \varphi
\]
Completeness

The proof rule is complete for formulas

- in the *universal-hierarchical fragment* of Coordination Logic, and
- in Prenex Normal Form (PNF)

Example

\[
\exists \{b, c\} \triangleright x_1. \forall \{a\} \triangleright y_1. \exists \{a, c\} \triangleright x_2. \exists \{a, d\} \triangleright x_3. \forall \{a, c\} \triangleright y_2. \varphi
\]

\[
\{a\} \subseteq \{a, c\}
\]
Completeness

The proof rule is complete for formulas

- in the *universal-hierarchical fragment* of Coordination Logic, and
- in Prenex Normal Form (PNF)

Example

\[
\exists \{b, c\} \triangleright x_1. \forall \{a\} \triangleright y_1. \exists \{a, c\} \triangleright x_2. \exists \{a, d\} \triangleright x_3. \forall \{a, c\} \triangleright y_2. \varphi
\]

\[
\{a\} \subseteq \{a, c\}, \{a\} \subseteq \{a, d\}
\]
Example

\[ \varphi := (y = f(x)) \]

\[ (\text{operational}_{2,3} \rightarrow \square \varphi) \]
\[ \land (\text{operational}_{1,3} \rightarrow \square \varphi) \]
\[ \land (\text{operational}_{1,2} \rightarrow \square \varphi) \]
\( \square (y = \text{majority vote}) \)
\[
\land \square (p_1 = f(x)) \\
\land \square (p_2 = f(x)) \\
\land \square (p_3 = f(x))
\]
Example

\[ (p_1 = f(x)) \]

\[ (y = \text{majority vote}) \]

\[ (p_2 = f(x)) \]

\[ (p_3 = f(x)) \]
Example

\[ (p_2 = f(x)) \]

\[ (p_1 = f(x)) \]

\[ (y = \text{majority vote}) \]

\[ \land (p_3 = f(x)) \]
Example

\[
\begin{align*}
\square (p_3 &= f(x)) \\
\square (p_2 &= f(x)) \\
\square (p_1 &= f(x)) \\
\square (y &= \text{majority vote})
\end{align*}
\]
Improvements

CL

universal-hierarchical in PNF

decidable
Improvements

CL

universal-hierarchical

universal-hierarchical in PNF

decidable
Prenex Normal Form

**Theorem**

*Every CL formula can be transformed into an equivalent CL formula with only prenex quantification.*

- Unlike FOL and other logics, prenex normal form transformation is not trivial

**Example**

\[ \forall\{a, b\} \circ x. \exists\{a\} \circ y. \varphi \]
**Theorem**

*Every CL formula can be transformed into an equivalent CL formula with only prenex quantification.*

Unlike FOL and other logics, prenex normal form transformation is not trivial.

**Example**

\[
\forall \{a, b\} \triangleright x. \bigcirc \exists \{a\} \triangleright y. \varphi
\]
Theorem

Every CL formula can be transformed into an equivalent CL formula with only prenex quantification.

Unlike FOL and other logics, prenex normal form transformation is not trivial.

Example

$$\forall \{a, b\} \rightarrow x. \exists \{a, b\} \rightarrow s_y. \bigcirc \exists \{a\} \rightarrow y. \varphi$$
Theorem

Every CL formula can be transformed into an equivalent CL formula with only prenex quantification.

Unlike FOL and other logics, prenex normal form transformation is not trivial

Example

\[
\forall \{a, b\} \Rightarrow x. \exists \{a, b, b'\} \Rightarrow s_y. \exists \{a, b'\} \Rightarrow y. \varphi'
\]
Conclusion and Future Work

- A complete proof system for CL formulas with hierarchical universal quantification
- This includes all distributed synthesis problems with Pnueli/Rosner architectures
- Open Problem: complete proof system for non-hierarchical universal quantification?