A Compositional Proof Rule for Coordination Logic

Highlights 2013, Paris

Leander Tentrup

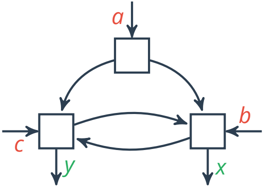
Reactive Systems Group
Saarland University

September 21, 2013

joint work with Bernd Finkbeiner

What is Coordination Logic?

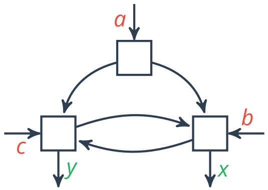
Logic of the *Distributed Synthesis Problem*



$\exists$ *implementation* s.t. φ holds?

What is Coordination Logic?

Logic of the *Distributed Synthesis Problem*

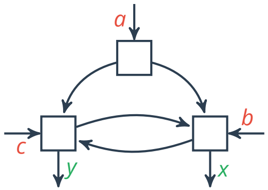


$\exists$ *implementation* s.t. φ holds?

Set of strategies for output variables

What is Coordination Logic?

Logic of the *Distributed Synthesis Problem*



$\exists$ *implementation* s.t. φ holds?

Set of strategies for output variables

LTL formula over input/output variables

Syntax

LTL $\quad x \mid \neg x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc \varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\overline{\mathcal{U}}\, \varphi$

$x \in \mathcal{C} \cup \mathcal{S}$

+

Strategic Quantification $\quad \exists C \triangleright s.\, \varphi \mid \forall C \triangleright s.\, \varphi$

$C \subseteq \mathcal{C}, s \in \mathcal{S}$

Coordination variables

represent information given by the environment

$\mathcal{C}$

Strategy variables

represent strategic choices made based on visible information

$\mathcal{S}$

Syntax

LTL $\quad x \mid \neg x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc \varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\overline{\mathcal{U}}\, \varphi$

$x \in \mathcal{C} \cup \mathcal{S}$

+

Strategic Quantification $\quad \exists C \triangleright s.\, \varphi \mid \forall C \triangleright s.\, \varphi$

$C \subseteq \mathcal{C}, s \in \mathcal{S}$

synthesize strategy

Coordination variables

represent information given by the environment

$\mathcal{C}$

Strategy variables

represent strategic choices made based on visible information

$\mathcal{S}$

Syntax

LTL

$$x \mid \neg x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc \varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\overline{\mathcal{U}}\, \varphi$$

$$x \in \mathcal{C} \cup \mathcal{S}$$

+

Strategic Quantification

$$\exists C \triangleright s.\, \varphi \mid \forall C \triangleright s.\, \varphi$$

$$C \subseteq \mathcal{C}, s \in \mathcal{S}$$

synthesize strategy

no control about strategy

Coordination variables

represent information given by the environment

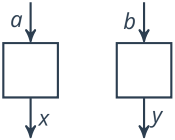
$\mathcal{C}$

Strategy variables

represent strategic choices made based on visible information

$\mathcal{S}$

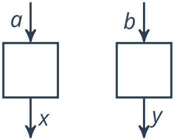
Decidability



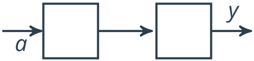
- Distributed Synthesis is undecidable
- ➤ Coordination Logic is **undecidable**

Decidability

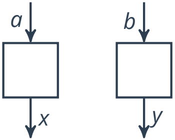
- Distributed Synthesis is undecidable
- ➤ Coordination Logic is **undecidable**



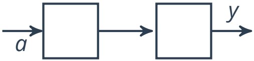
- Special cases are decidable
- ➤ Syntactic restricted fragment of CL



Decidability



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

$$
\frac{
\begin{array}{lll}
(R_1) & \models & \mathcal{H}(S_{cut})\,.\,\psi \\
(R_2) & \models & \mathcal{H}(\mathcal{S} \setminus S_{cut})\,.\,\varphi' \\
(R_3) & \models & \mathcal{H}_{\forall}(\mathcal{S}).\,\psi \wedge \varphi' \rightarrow \varphi
\end{array}
}{\models \Phi}
$$

A Compositional Proof Rule

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

decidable

$$
\frac{
\begin{array}{lll}
(R_1) & \models & \boxed{\mathcal{H}(S_{cut})}\,.\,\psi \\
(R_2) & \models & \mathcal{H}(\mathcal{S} \setminus S_{cut})\,.\,\varphi' \\
(R_3) & \models & \mathcal{H}_\forall(\mathcal{S}).\,\psi \wedge \varphi' \rightarrow \varphi
\end{array}
}{\models \Phi}
$$

A Compositional Proof Rule

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

decidable

simplified

$$
\begin{array}{lll}
(R_1) & \models & \boxed{\mathcal{H}(S_{cut})}\,.\,\psi \\
(R_2) & \models & \boxed{\mathcal{H}(\mathcal{S} \setminus S_{cut})}\,.\,\varphi' \\
(R_3) & \models & \mathcal{H}_{\forall}(\mathcal{S}).\,\psi \wedge \varphi' \rightarrow \varphi \\
\hline
 & \models \Phi &
\end{array}
$$

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

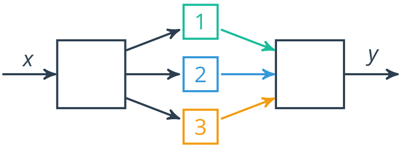
- in the *universal-hierarchical fragment* of Coordination Logic, and
- in Prenex Normal Form (PNF)

Example

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

$$\{a\} \subseteq \{a,c\}, \{a\} \subseteq \{a,d\}$$

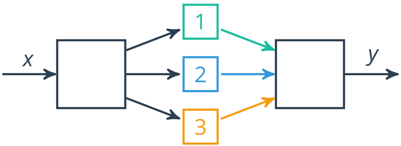
Example



$$\varphi := \big(y = f(x)\big)$$

$$\begin{aligned} & \left(\text{operational}_{2,3} \rightarrow \square\, \varphi\right) \\ \wedge & \left(\text{operational}_{1,3} \rightarrow \square\, \varphi\right) \\ \wedge & \left(\text{operational}_{1,2} \rightarrow \square\, \varphi\right) \end{aligned}$$

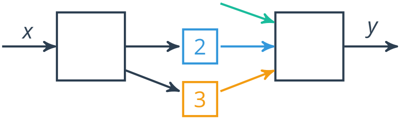
Example



$$\square\,(y = \text{majority vote})$$
$$\wedge\ \square\,(p_1 = f(x))$$
$$\wedge\ \square\,(p_2 = f(x))$$
$$\wedge\ \square\,(p_3 = f(x))$$

Example

$\square\,(p_1 = f(x))$

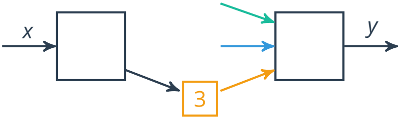


$$\begin{aligned} &\square\,(y = \text{majority vote}) \\ \wedge\ &\square\,(p_2 = f(x)) \\ \wedge\ &\square\,(p_3 = f(x)) \end{aligned}$$

Example

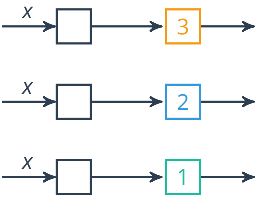
$\Box\,(p_2 = f(x))$

$\Box\,(p_1 = f(x))$



$$\Box\,(y = \text{majority vote})$$
$$\wedge\ \Box\,(p_3 = f(x))$$

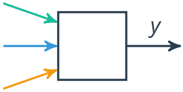
Example



$\Box\, (p_3 = f(x))$

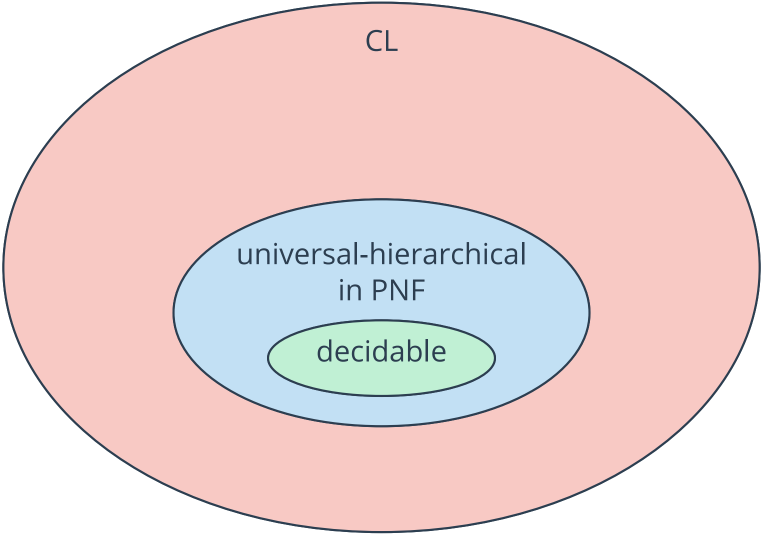
$\Box\, (p_2 = f(x))$

$\Box\, (p_1 = f(x))$

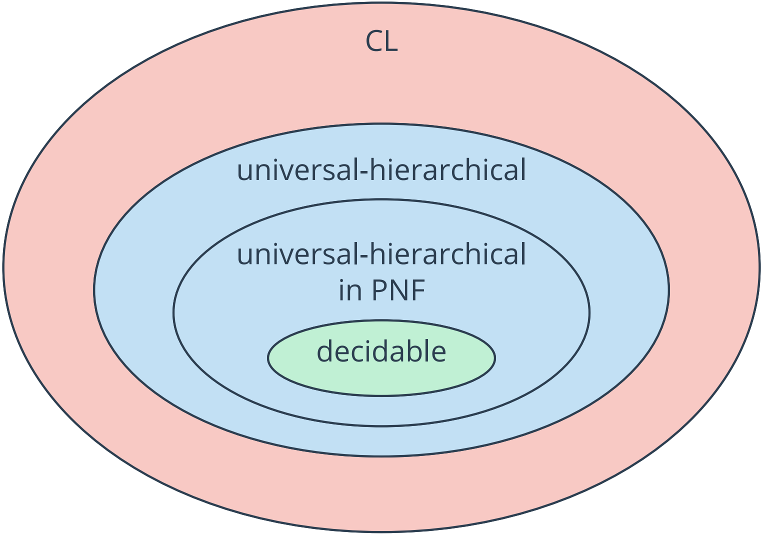


$\Box\, (y = \text{majority vote})$

Improvements



Improvements



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 $\forall\{a,b\} \triangleright x. \bigcirc \exists\{a\} \triangleright y. \varphi$

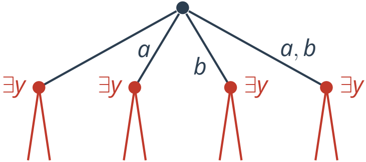
Prenex Normal Form

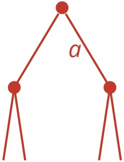
Theorem

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 $\forall\{a,b\} \triangleright x.\, \bigcirc\, \exists\{a\} \triangleright y.\, \varphi$





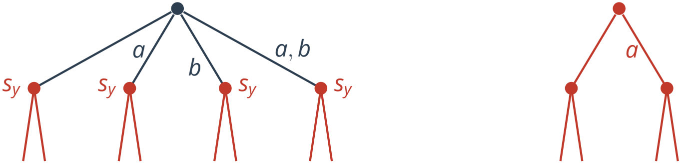
Prenex Normal Form

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



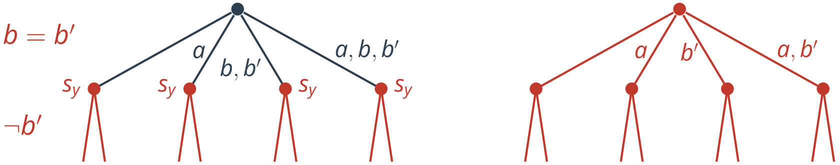
Prenex Normal Form

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 $\forall\{a,b\} \triangleright x.\, \exists\{a,b,b'\} \triangleright s_y.\, \exists\{a,b'\} \triangleright y.\, \varphi'$



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?