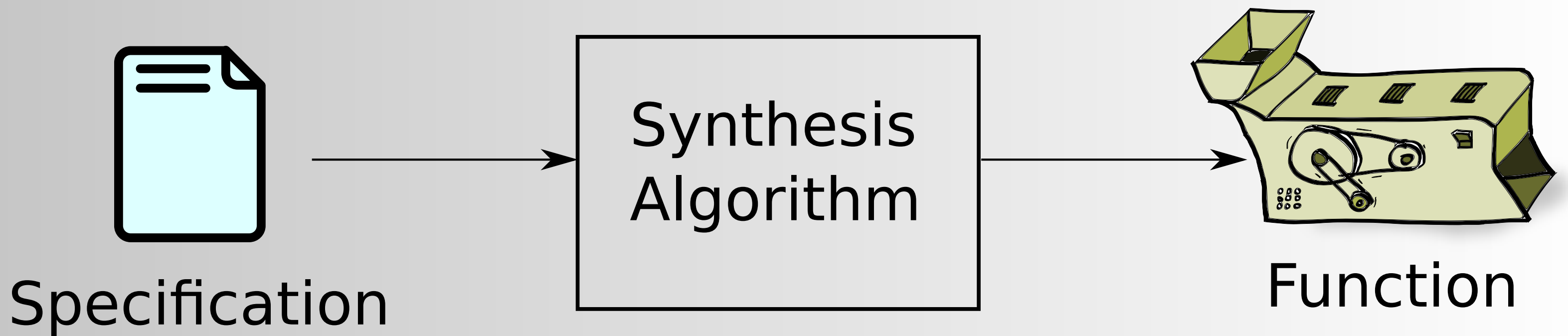
Interpolation for Synthesis on Unbounded Domains

Viktor Kuncak and Régis Blanc
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Switzerland

Synthesis Procedures

Synthesis:



Synthesis Procedures:

- Decidable theories
- Build functions from primitives in the theory

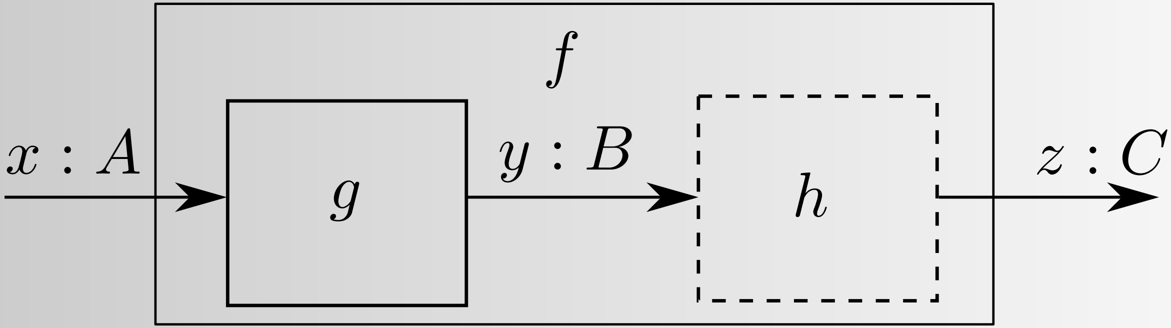
Kuncak, Mayer, Piskac, Suter: *Software Synthesis Procedures*, CACM 2012

Synthesis from Components

Build functionalities using building blocks.

Input: Components and description of functionality.

Output: Connectors between components.



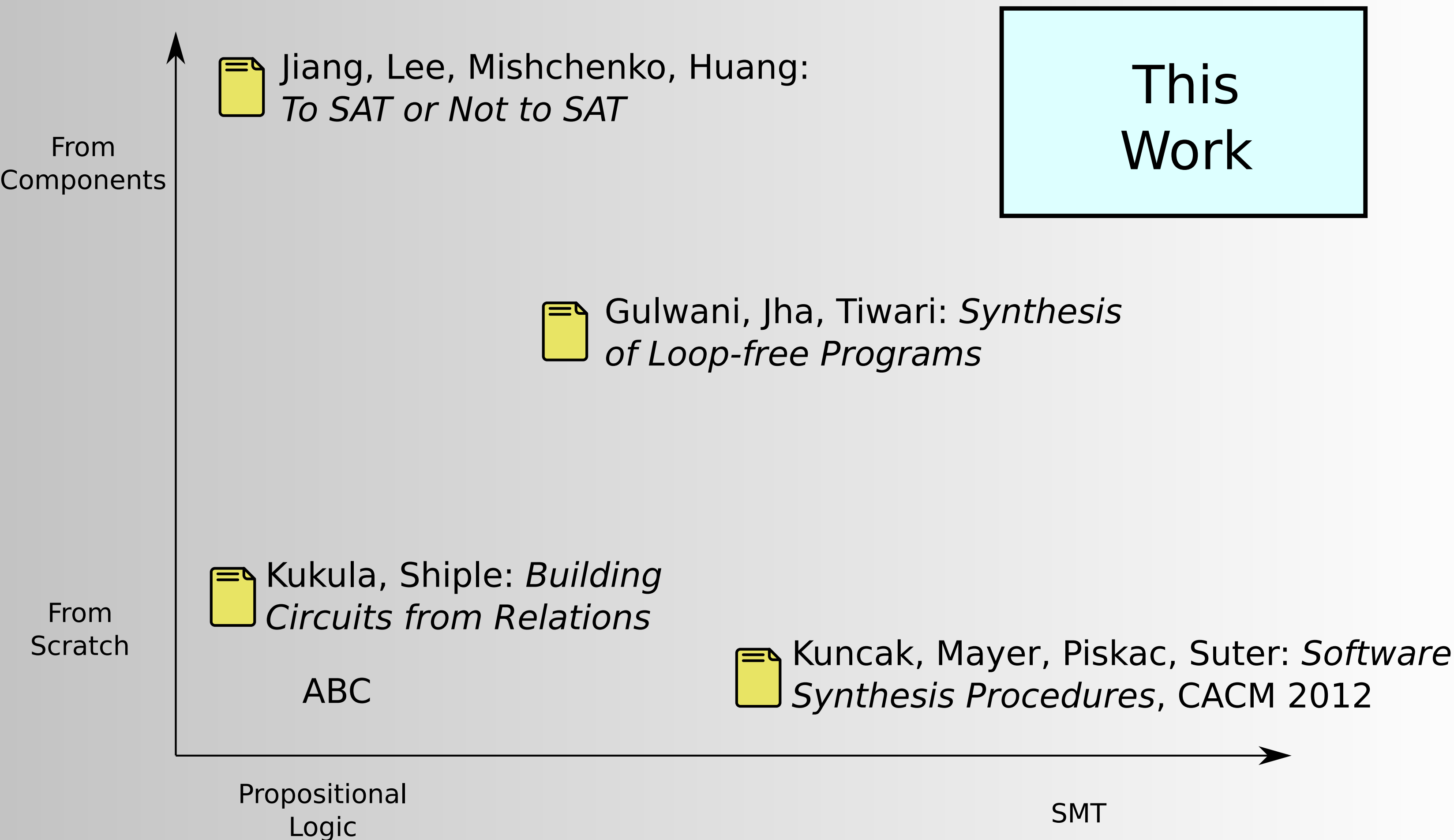
Challenges:

- Force/Allow using components instead of inputs.
- Synthesis procedures only combines primitives.

Motivation:

- Re-use optimized components.
- Scale to complex functionalities.

Existing Work



Interpolation

Given: $A \wedge B \implies \bot$

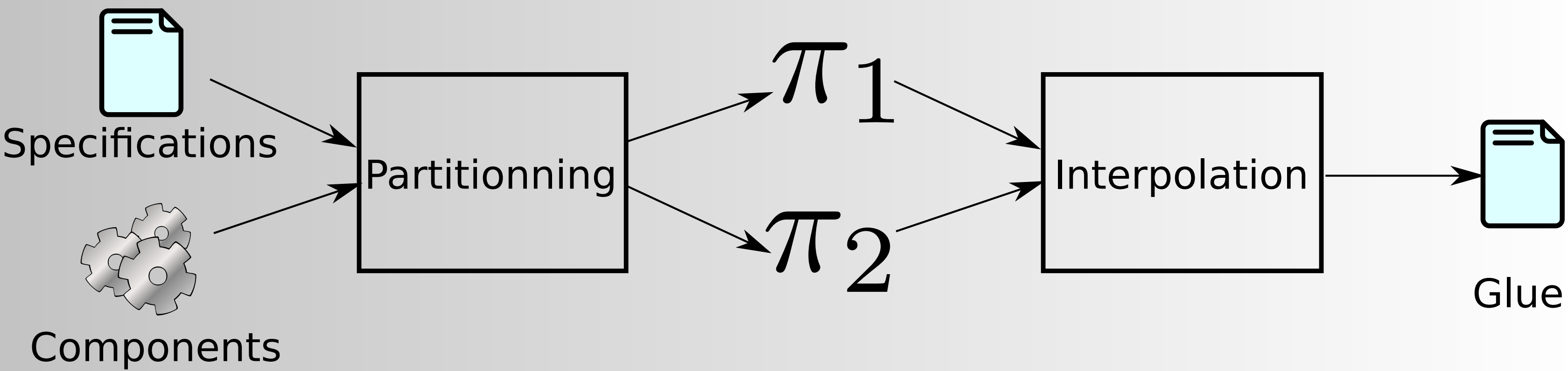
Craig Interpolant I

1) $A \implies I$
2) $I \wedge B \implies \bot$
3) $L(I) \subseteq L(A) \cap L(B)$

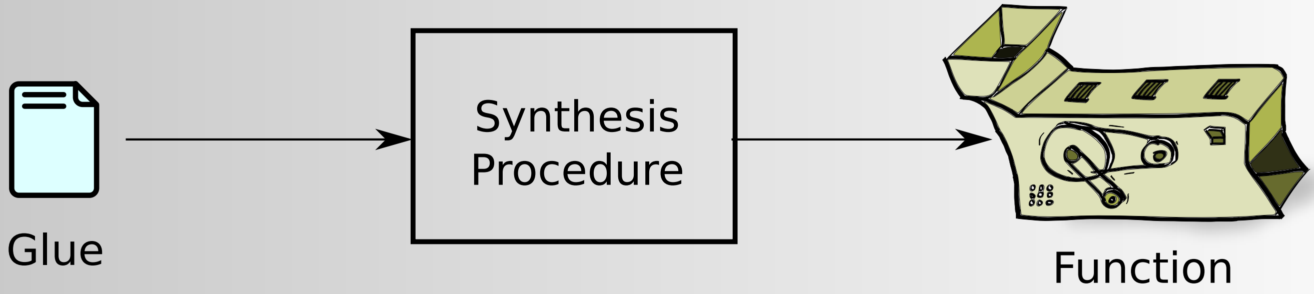
Holds in propositional logic.
Holds in some first order theories.

Our Approach: Two-step Process

(i) Interpolation finds a specification:



(ii) Functional synthesis turns the spec into a function:



Partitionning

$h(g(x)) = f(x)$

$$f(x) = z \quad g(x) = y$$

$$f(x') = z' \quad g(x') = y'$$

Partitionning

$h(g(x)) = f(x)$

$$\frac{f(x) = z \quad g(x) = y \qquad y = y' \\ f(x') = z' \quad g(x') = y'}{z = z'}$$

Partitionning

$h(g(x)) = f(x)$

$$\frac{f(x) = z \quad g(x) = y \quad y = y' \quad f(x') = z' \quad g(x') = y'}{z = z'}$$

By rearranging, equivalent:

$$\frac{f(x) = z \quad g(x) = y \quad y = y' \quad f(x') = z' \quad g(x') = y' \quad z \neq z'}{\bot}$$

Partitionning

$h(g(x)) = f(x)$

$$\frac{f(x) = z \quad g(x) = y \quad y = y' \qquad f(x') = z' \quad g(x') = y'}{z = z'}$$

By rearranging, equivalent:

A

$$f(x) = z \quad g(x) = y \quad y = y'$$

B

$$f(x') = z' \quad g(x') = y' \quad z \neq z'$$

$$\overline{\qquad\qquad\qquad}$$

$$\bot$$

Interpolants and Relational Connectors

By definition of interpolant:

1)

2)

Interpolants and Relational Connectors

By definition of interpolant:

1)

$$f(x) = z \wedge g(x) = y \wedge y = y' \models I(y', z)$$

2)

Interpolants and Relational Connectors

By definition of interpolant:

1)

$$f(x) = z \wedge g(x) = y \wedge y = y' \models I(y', z)$$

$$f(x) = z \models I(g(x), z)$$

2)

Interpolants and Relational Connectors

By definition of interpolant:

1)

$$f(x) = z \land g(x) = y \land y = y' \models I(y', z)$$

$$f(x) = z \models I(g(x), z)$$

2)

$$I(y', z) \land g(x') = y' \land f(x') = z' \models z = z'$$

Interpolants and Relational Connectors

By definition of interpolant:

1)

$$f(x) = z \wedge g(x) = y \wedge y = y' \models I(y', z)$$

$$f(x) = z \models I(g(x), z)$$

2)

$$I(y', z) \wedge g(x') = y' \wedge f(x') = z' \models z = z'$$

$$I(g(x'), z) \models z = f(x')$$

Interpolants and Relational Connectors

By definition of interpolant:

1)

$$f(x) = z \wedge g(x) = y \wedge y = y' \models I(y', z)$$

$$f(x) = z \models I(g(x), z)$$

2)

$$I(y', z) \wedge g(x') = y' \wedge f(x') = z' \models z = z'$$

$$I(g(x'), z) \models z = f(x')$$

$$\boxed{I(g(x), z) \iff z = f(x)}$$

Conclusion

If a theory admits interpolation and synthesis procedures, then it admits synthesis from components.

- Synthesis from components is challenging and important.
- Interpolation can be used to force the use of components in a specification.
- Synthesis procedures can be used to complete the process.
- More generalizations on the paper.