Boolean Synthesis via Decomposition

Lucas M. Tabajara

Joint work with Supratik Chakraborty, Dror Fried and Moshe Y. Vardi

lucasmt@rice.edu
Rice University

Boolean Synthesis

f $\qquad\qquad$ g

$$(x_0 \oplus y_0 = 0) \wedge (x_1 \oplus y_1 \oplus (x_0 \wedge y_0) = 0) \longrightarrow \begin{array}{l} y_0 := x_0 \\ y_1 := x_1 \oplus x_0 \end{array}$$

Specification of a system encoded as a Boolean formula

Boolean function implementing system behavior

Our goal: To decompose the specification into formulas that are easier to synthesize.

Decomposition using Factored Formulas

$$F(\vec{x}, y_1, y_2, y_3, y_4) = F_1(\vec{x}, y_2, y_4) \wedge F_2(\vec{x}, y_1, y_2, y_3) \wedge F_3(\vec{x}, y_3)$$

- Easy to perform decomposition.
- Has been shown to significantly improve synthesis algorithms.

However: Dependences between factors prevent us from taking full advantage of the decomposition.

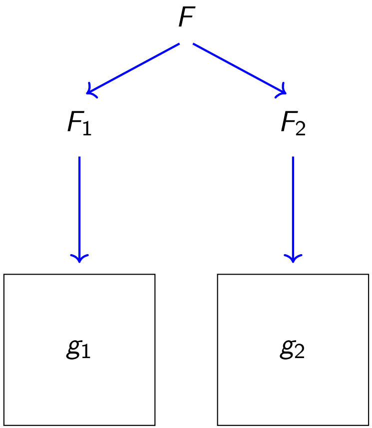
Towards Sequential Decomposition

Sequential Decomposition

Given: A Boolean formula $F(\vec{x}, \vec{y})$ between input variables $\vec{x}$ and output variables $\vec{y}$.

Return: Two Boolean formulas $F_1(\vec{x}, \vec{z})$ and $F_2(\vec{z}, \vec{y})$ that can be composed back into F.

Sequential Decomposition



Sequential Decomposition