E3Solver: Decision Tree Unification by Enumeration

Ammar Ben Khadra
Introduction

• **E3Solver**: an enumerative solver for programming by example

• Participated in SyGuS Comp‘17 (Bitvector subtrack) and won, unexpectedly, the first place!

• Source code is publicly available ([https://github.com/sygus-tools](https://github.com/sygus-tools))
Solving in two phases

Enumerate terminal expressions

\[ E_1(\text{in}_1, \text{out}_1) \rightarrow T_1 \]
\[ E_2(\text{in}_2, \text{out}_2) \rightarrow T_1 \]
\[ E_3(\text{in}_3, \text{out}_3) \rightarrow T_2 \]
\[ E_4(\text{in}_4, \text{out}_4) \rightarrow T_3 \]
\[ E_5(\text{in}_5, \text{out}_5) \rightarrow T_3 \]
\[ E_6(\text{in}_6, \text{out}_6) \rightarrow T_1 \]

Enumerate decision tree conditions

\[ C_1 \rightarrow T_1, T_2 \]
Solving in two phases

Enumerate terminal expressions

\[
\begin{align*}
E_1(\text{in}_1, \text{out}_1) & \rightarrow T_1 \\
E_2(\text{in}_2, \text{out}_2) & \rightarrow T_1 \\
E_3(\text{in}_3, \text{out}_3) & \rightarrow T_2 \\
E_4(\text{in}_4, \text{out}_4) & \rightarrow T_3 \\
E_5(\text{in}_5, \text{out}_5) & \rightarrow T_3 \\
E_6(\text{in}_6, \text{out}_6) & \rightarrow T_1
\end{align*}
\]

Enumerate decision tree conditions

\[
\begin{align*}
C_1 & \rightarrow C_2 \\
C_2 & \rightarrow T_2 \\
T_1 & \rightarrow T_3 \\
T_3 & \rightarrow T_1
\end{align*}
\]
Key properties

• Correct-by-construction

• Incremental

• Cheap unification steps. Bottleneck in terms enumeration

\[ \text{Exp}[\text{in}_i] \neq \text{Exp}[\text{in}_j] \land (\text{Exp}[\text{in}_i] = 1 \lor \text{Exp}[\text{in}_j] = 1) \]

- Non-constant condition
- One true branch
History of E3Solver

- E Solver
- E Solver+
- E2 Solver
- E3 Solver

0 (0%) → 135 (18%) → 138 (18%) 30% faster → 750 (100%)

• Credit’s due to Abhishek Udupa for the well-written and publicly available code of E Solver
Questions?