Specialization for miniKanren and its Translation to a Functional Language

Author: Kate Verbitskaia

JetBrains Programming Languages and Tools Lab

18.12.2021

Recap: Relational Programming

Programs are relations:

$$\begin{array}{l}
\texttt{len}^o\ \texttt{l}\ \texttt{n} = \\
\quad (\texttt{l} \equiv [\,]\ \wedge\ \texttt{n} \equiv 0) \\
\quad \vee\ (\exists\ \texttt{h}\ \texttt{t}\ \texttt{m} \\
\qquad\qquad (\ \texttt{l} \equiv \texttt{h}\ :\ \texttt{t} \\
\qquad\qquad \wedge\ \texttt{n} \equiv 1 + \texttt{m} \\
\qquad\qquad \wedge\ \texttt{len}^o\ \texttt{t}\ \texttt{m}))
\end{array}$$

Execution in different directions

$\texttt{len}^o$ [0,1] ? $\rightsquigarrow$ 2

$\texttt{len}^o$? 2 $\rightsquigarrow$ [_.0, _.1]

Recap: Relational Interpreters for Search

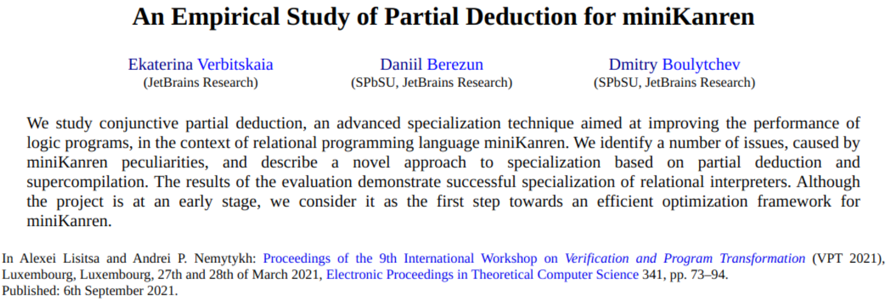
- Implement a verifier (interpreter) in a functional programming language
- Translate the functional interpreter into `miniKanren`
- Run the relational interpreter backwards to get a solver

- Relational interpreter is the most performant in the forward direction and is not as performant in the backwards direction
- Specialization improves performance

Lozov, P., Verbitskaia, E. and Boulytchev, D., 2019.
Relational Interpreters for Search Problems.

The Great Leap

A paper on conservative partial deduction has been published



An Empirical Study of Partial Deduction for miniKanren

Ekaterina Verbitskaia (JetBrains Research)
Daniil Berezun (SPbSU, JetBrains Research)
Dmitry Boulytchev (SPbSU, JetBrains Research)

We study conjunctive partial deduction, an advanced specialization technique aimed at improving the performance of logic programs, in the context of relational programming language miniKanren. We identify a number of issues, caused by miniKanren peculiarities, and describe a novel approach to specialization based on partial deduction and supercompilation. The results of the evaluation demonstrate successful specialization of relational interpreters. Although the project is at an early stage, we consider it as the first step towards an efficient optimization framework for miniKanren.

In Alexei Lisitsa and Andrei P. Nemytykh: Proceedings of the 9th International Workshop on *Verification and Program Transformation* (VPT 2021), Luxembourg, Luxembourg, 27th and 28th of March 2021, Electronic Proceedings in Theoretical Computer Science 341, pp. 73–94.
Published: 6th September 2021.

Translation to Functional Programming Language

Specialization has never been the end goal

The goal was to create solvers out of verifiers

We need to translate `miniKanren` programs back into the functional programming language

Generating Extension

$$p(x, y)$$

$$spec(p, x) = p_x$$

$$gen_p(x) = p_x$$

$$p_x(y) \equiv p(x, y)$$

Generating Extension: Example

```
f (x, y) {
  if x > 0
  then y
  else y+1
}
```

```
gen_f(x) =
  "f (y) {"
  ++ if x > 0 then "y" else "y+1"
  ++ "}"
```

Generating Extension for L

Code

Specialization Efforts for Topsort

Code

The Plan

1. Write a generating extension for `miniKanren`
2. Employ online specialization
3. ???
4. PROFIT

Conclusion

- The Paper
  - Verbitskaia E, Berezun D, Boulytchev D. An Empirical Study of Partial Deduction for miniKanren. (VPT-2021)
- Teaching
  - Theory of Formal Languages (SPbU, HSE, ElTech)
- New research direction
  - Employing generating extension for relational interpreters