Why Multi-Result Supercompilation Matters: Case Study of Reachability Problems for Transition Systems

Andrei Klimov

Keldysh Institute of Applied Mathematics, Russian Academy of Sciences

July 8, 2012
Third International Valentin Turchin Workshop on Metacomputation, META 2010
Pereslavl-Zalessky, Russia
History and Main Conclusion

- **2005-2007 Andrei Nemytykh and Alexei Lisitsa**
  - have experimentally found a method to solve the coverability problem for (a class of) practical counter systems (models of cache-coherence protocols and other systems) with the Refal Supercompiler SCP4
  - *User + single-result supercompiler = MRSC*

- **2010-2011 Andrei Klimov**
  - have theoretically explained and proved that the coverability problem is solvable for monotonic counter systems by an iterative procedure of applying a domain-specific supercompiler for counter systems varying a parameter of generalization
  - *An optimized MRSC enumerating a small subset of residual graphs*

- **2005-2007 Gilles Geeraerts et al (Belgium)**
  - theory of `Expand, Enlarge and Check` algorithmic schema (ECC) for solving the coverability problem of well-structured transition systems (WSTS)
  - *An MRSC for WSTS with reduced search space*

One thing to remember from this talk

- These are instances of domain-specific multi-result supercompilation (MRSC) with search space reduction based of domain properties and purpose
`User-controlled’ MRSC

- **2005-2007 Andrei Nemytykh and Alexei Lisitsa**
  - have experimentally found a method to solve the coverability problem for (a class of) practical counter systems (models of cache-coherence protocols and other systems) with the Refal Supercompiler SCP4
  - *User + single-result supercompiler = MRSC*

- **Andrei Nemytykh devised two versions of SCP4**
  - SCP4₀ - standard version
  - SCP4₁ - generalization of empty expressions (representing zeros) prohibited

- **The user behavior**
  - When SCP4₀ did not prove the coverability, SCP4₁ was applied
  - No more supercompilers were needed for the considered samples borrowed from the collection by Giorgio Delzanno

- **Questions remained**
  - Were these SCP4 versions sufficient?
  - Might other restrictions of generalization be needed?
  - Had the SCP4 author to invent new modifications of SCP4?
Domain-Specific Special-Purpose MRSC

- **2010-2011 Andrei Klimov**
  - have theoretically explained and proved that the coverability problem is solvable for monotonic counter systems by an iterative procedure of applying a domain-specific supercompiler for counter systems varying a parameter of generalization
  - An optimized MRSC enumerating a small subset of residual graphs

- **Algorithm**
  - Scp\(_l\) – a domain-specific supercompiler for counter systems with parameter \(l\) prohibiting generalization of integers \(n < l\)
    - the simplest version: integers \(n \geq l\) are immediately generalized
  - for \(l=1,2,3...\) do
    - use Scp\(_l\) to build a residual set of configurations
    - if all residual configurations are disjoint with the target set Unsafe
    - then return “Unreachable”

- This algorithm with the simplest supercompiler Scp\(_l\) fits the ECC schema
- My proof of its correctness differs from that of the ECC algorithmic schema
- ...and asserts a stronger termination statement:
  - it terminates for all monotonic counter systems and upper-closed Unsafe sets
ECC as a domain-specific multi-result supercompiler

- **2005-2007 Gilles Geeraerts et al**
  - theory of `Expand, Enlarge and Check` algorithmic schema (ECC) for solving the coverability problem of well-structured transition systems (WSTS)
  - An MRSC for WSTS and its optimized versions

- **Main ideas**
  - The set of all possible configurations $C$ is infinite (as usual)
  - **Def.** A finite $R \subseteq C$ is called a residual set iff it is closed under “driving”:
    - $\text{Post}(\llbracket R \rrbracket) \subseteq \llbracket R \rrbracket$
  - Consider an ascending sequence of finite sets of configurations $C_l$:
    - $C_0 \subset C_1 \subset C_2 \subset C_3 \ldots$
    - $C = \bigcup C_l$
  - Consider residual sets $R \subseteq C_l$
  - The set $\{R | R \subseteq C_l\}$ of all such residual sets is finite as $C_l$ is finite
  - Hence, it is solvable whether there exists a safe $R \subseteq C_l$
    (that is, all configurations in $R$ are disjoint with the target set $\text{Unsafe}$)
  - Iterate for $l=0,1,2,3\ldots$
  - If a safe residual set $R \subseteq C$ exists, then $C_l$ s.t. $R \subseteq C_l$ exists and hence the iterative procedure terminates
  - The notion of MRSC is wider
ECC as a domain-specific multi-result supercompiler

- 2005-2007 Gilles Geeraerts *et al*
  - theory of `Expand, Enlarge and Check’ algorithmic schema (ECC) for solving the coverability problem of well-structured transition systems (WSTS)
  - An MRSC for WSTS and its optimized versions

Main ideas
- The set of all possible configurations $\mathbb{C}$ is infinite (as usual)
- **Def.** A finite $R \subseteq \mathbb{C}$ is called a residual set iff it is closed under "driving":
  - $\text{Post}(\llbracket R \rrbracket) \subseteq \llbracket R \rrbracket$
- Consider an ascending sequence of finite sets of configurations $\mathbb{C}_l$:
  - $C_0 \subset C_1 \subset C_2 \subset \cdots$
  - $\mathbb{C} = \bigcup \mathbb{C}_l$
- Consider residual sets $R \subseteq \mathbb{C}_l$
- The set $\{R \mid R \subseteq \mathbb{C}_l\}$ of all such residual sets is finite as $\mathbb{C}$ is finite
- Hence, it is solvable whether there exists a safe $R \subseteq \mathbb{C}$ (that is, all configurations in $R$ are disjoint with the target set $\text{Unsafe}$)
- Iterate for $l=0,1,2,3...$
  - If a safe residual set $R \subseteq \mathbb{C}$ exists, then $\mathbb{C}_l$ s.t. $R \subseteq \mathbb{C}_l$ exists and hence the iterative procedure terminates

The notion of MRSC is wider

Where is the well-structuredness of a TS used?
(WS = monotonicity + well-quasi-order)
- Existence of safe $R$ when the TS is safe
- Optimizations: reducing the search space

Without the well-structuredness:
If there exists an inductive proof that a TS is safe with the inductive hypotheses in form of a residual set of configurations, then MRSC finds it
Related work: Supercompilation-like algorithms


...and a lot of other works...