Safe Compositional Equation-based Modeling of Constrained Flow Networks

Nate Soule - Azer Bestavros - Assaf Kfoury - Andrei Lapets
Boston University
Introduction: Constrained Flow Networks

- Flows between nodes regulated by constraints
- Example domain: Vehicular Road Traffic
Introduction: NetSketch

- **Problem:** Traditional analysis of constrained flow networks doesn't scale well, cope with unknowns

- **Solution:** NetSketch
  - Lightweight, efficient, scalable modeling and analysis
  - Formalism/DSL
  - Web-based Tool
Introduction: NetSketch

Whole System Analysis

Compositional Analysis
Compositional Analysis via Type Approximations

```java
int add(int a, int b) {
    return a + b;
}
```

Type system allows the compiler to verify this is safe without knowledge of the exact values or representation of vars.

**Traditional Types**

**NetSketch Types**

EOOLT 2011
NetSketch DSL

- Module
- Connect
- Loop
- Hole
- Let

Connect [(hOut, x)]
Module Fork_0 [x] [y,z] [x=y+z]
Let Hole_1 [Replacement_2...] in
Hole Hole_1 [hIn] [hOut]
Example Model

<table>
<thead>
<tr>
<th>Element</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston_4</td>
<td>Module (untyped network)</td>
</tr>
<tr>
<td>Construction_8</td>
<td>Hole (untyped network)</td>
</tr>
<tr>
<td>CurRotary_9</td>
<td>Replacement (untyped network)</td>
</tr>
<tr>
<td>ProposedRotary_10</td>
<td>Replacement (untyped network)</td>
</tr>
<tr>
<td>Cap_Code_11</td>
<td>Typed Network</td>
</tr>
</tbody>
</table>

Untyped Network (Exact Network)

Typed Network (Approximation)

Untyped Network (Exact Network)
Type System

- Current implementation:
  - Types are open/closed intervals over $\mathbb{IR}$
  - Linear constraints are analyzed to determine:
    - Safe intervals for input ports
    - Safe intervals for output ports

\[ z: [0,10] \]
\[ x: [0,20] \]
Type Inference: Input Types

- Linear constraints form a convex hull
- Input types approximate the feasible region
- Types derived from maximally enclosed axis-aligned hyperrectangle
- Made unique via:
  - Center point
  - Aspect ratio
Type Inference: Output Types

- Again constraints form convex hull
- Must use maximally enclosing hyperrectangle
- Unique without further user input
NetSketch Architecture

- Web Based Front End
  - JavaScript
  - HTML

- Server Back End
  - Haskell
  - C
  - C++

- Asynchronous JavaScript and XML (AJAX) based communication
Harnessing Modelica

Computation Platform

NetSketch (Haskell)

HModelica (Haskell)

Simplex

Open Modelica

Simulation Platform

Conn [(c,x)]
Module Mod1 [a,b] [c] [a+b=c]
Module Mod2 [x] [y,z] [x=y+z]

Package TranslatedNetSketch

Connector OutPort = output Real;
Connector InPort = input Real;

Class Mod1
  InPort a;
  InPort b;
  OutPort c;
Equation
  a + b = c;
End Mod1
...
Current and Future Work

- Constraint and type extensions
  - Adaptive Dynamic Types
  - Variations of constraints
- Tool extensions
  - Bi-directional flow, extended \textit{Let} functionality, internal variables, export to Modelica
- Modelica integration
  - Extend/refine HModelica
  - Modelica -> NetSketch translation
Thank You

More info at: http://www.cs.bu.edu/groups/ibench