Type-Based Structural Analysis for Modular Systems of Equations

Henrik Nilsson

School of Computer Science
University of Nottingham
The Problem (1)

• A core aspect of equation-based modelling: modular description of models through composition of equation system fragments.

• Naturally, we are interested in ensuring composition makes sense, catching any mistakes as early as possible.

• Central question: do the equations have a solution?

• Cannot be answered comprehensively before we have a complete model.

*Not very modular!*
The Problem (2)

- However, it might be possible to check violations of certain necessary conditions for solvability in a modular way!
- One necessary condition for solvability is that a system must not be structurally singular.
- The paper investigates the extent to which the structural singularity of a system of equations can be checked modularly.
We need a notation for modular systems of equations. Note:

- A system of equations specifies a relation among a set of variables.
- Specifically, our interest is relations on time-varying values or signals.
- An equation system fragment needs an interface to distinguish between local variables and variables used for composition with other fragments.
These ideas can be captured through a notion of **typed signal relations**:

\[
\begin{align*}
\text{foo} &:: SR (\text{Real}, \text{Real}, \text{Real}) \\
\text{foo} &= \text{sigrel} (x_1, x_2, x_3) \quad \text{where} \\
&\quad f_1 \; x_1 \; x_2 \; x_3 = 0 \\
&\quad f_2 \; x_2 \; x_3 = 0
\end{align*}
\]
Modular Systems of Equations (3)

Composition can be expressed through \textit{signal relation application}:

$$\text{foo} \circ (u, v, w)$$
$$\text{foo} \circ (w, u + x, v + y)$$

yields

\[
\begin{align*}
    f_1(u, v, w) &= 0 \\
    f_2(v, w) &= 0 \\
    f_1(w) (u + x) (v + y) &= 0 \\
    f_2(u + x) (v + y) &= 0
\end{align*}
\]
Treating signal relations as \textit{first class entities} in a functional setting is a simple way to add essential functionality, such as a way to parameterize the relations:

\begin{align*}
\text{foo2} :: \text{Int} & \rightarrow \text{Real} \rightarrow \text{SR (Real, Real, Real)} \\
\text{foo2} n k &= \text{sigrel} (x_1, x_2, x_3) \text{ where} \\
& \quad f_1 n x_1 x_2 x_3 = 0 \\
& \quad f_2 x_2 x_3 = k
\end{align*}
Example: Resistor Model

twoPin :: SR (Pin, Pin, Voltage)

twoPin = sigrel (p, n, u) where
\[
\begin{align*}
u &= p.v - n.v \\
p.i + n.i &= 0
\end{align*}
\]

resistor :: Resistance \rightarrow SR (Pin, Pin)

resistor r = sigrel (p, n) where
\[
\begin{align*}
twoPin \odot (p, n, u) \\
r \ast p.i &= u
\end{align*}
\]
Equal number of equations and variables is a necessary condition for solvability. For a modular analysis, one might keep track of the balance in the signal relation type:

$$SR(\ldots)n$$
Tracking Variable/Equation Balance?

Equal number of equations and variables is a necessary condition for solvability. For a modular analysis, one might keep track of the balance in the signal relation type:

\[
SR (\ldots) n
\]

But very weak assurances:

\[
\begin{align*}
f(x, y, z) &= 0 \\
g(z) &= 0 \\
h(z) &= 0
\end{align*}
\]
A Possible Refinement (1)

A system of equations is *structurally singular* iff it is not possible to put the variables and equations in a one-to-one correspondence such that each variable occurs in the equation it is related to.
A Possible Refinement (2)

Structural singularities can be discovered by studying the incidence matrix:

Equations | Incidence Matrix
--- | ---
\( f_1(x, y, z) = 0 \) | \[
\begin{pmatrix}
1 & 1 & 1 \\
0 & 0 & 1 \\
0 & 0 & 1 \\
\end{pmatrix}
\]
\( f_2(z) = 0 \)
\( f_3(z) = 0 \)
A Possible Refinement (3)

So maybe we can index signal relations with incidence matrices?

\[
foo :: SR (\text{Real, Real, Real}) \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}
\]

\[
foo = \text{sigrel} (x_1, x_2, x_3) \text{ where }
\]
\[
f_1 \ x_1 \ x_2 \ x_3 = 0
\]
\[
f_2 \ x_2 \ x_3 = 0
\]
The **Structural Type** represents information about which variables occur in which equations.

- Denoted by an incidence matrix.
- Two interrelated instances:
  - Structural type of a system of equations
  - Structural type of a signal relation
The structural type of a system of equations is obtained by *composition* of the structural types of constituent signal relations. *Straightforward*.

The structural type of a signal relation is obtained by *abstraction* over the structural type of a system of equations. *Less straightforward*. 
Recall

\[
\text{foo} :: SR (\text{Real}, \text{Real}, \text{Real}) \begin{pmatrix}
1 & 1 & 1 \\
0 & 1 & 1
\end{pmatrix}
\]

Consider

\[
\text{foo} \Diamond (u, v, w) \\
\text{foo} \Diamond (w, u + x, v + y)
\]

in a context with five variables \(u, v, w, x, y\).
The structural type for the equations obtained by instantiating \( \textit{foo} \) is simply obtained by Boolean matrix multiplication. For \( \textit{foo} \odot (u, v, w) \):

\[
\begin{pmatrix}
1 & 1 & 1 \\
0 & 1 & 1
\end{pmatrix}
\begin{pmatrix}
u & v & w & x & y \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{pmatrix}
= 
\begin{pmatrix}
u & v & w & x & y \\
1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0
\end{pmatrix}
\]
Composition of Structural Types (3)

For $foo \Diamond (w, u + x, v + y)$:

$$
\begin{pmatrix}
1 & 1 & 1 \\
0 & 1 & 1
\end{pmatrix}
\begin{pmatrix}
u & v & w & x & y \\
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1
\end{pmatrix}
= 
\begin{pmatrix}
u & v & w & x & y \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1
\end{pmatrix}
$$
Composition of Structural Types (4)

Complete incidence matrix and corresponding equations:

\[
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 \\
\end{pmatrix}
\]

\[
\begin{align*}
\mathbf{f}_1 & \quad u \cdot v \cdot w \\
\mathbf{f}_2 & \quad v \cdot w \\
\mathbf{f}_1 & \quad w \cdot (u + x) \cdot (v + y) \\
\mathbf{f}_2 & \quad (u + x) \cdot (v + y) \\
\end{align*}
\]

= 0

= 0

= 0

= 0
Abstraction over Structural Types (1)

Now consider encapsulating the equations:

\[ bar = \text{sigrel} (u, y) \text{ where} \]
\[ \text{foo} \diamond (u, v, w) \]
\[ \text{foo} \diamond (w, u + x, v + y) \]

The equations of the body of \( bar \) needs to be partitioned into

- **Local Equations**: equations used to solve for the local variables
- **Interface Equations**: equations contributed to the outside
Abstraction over Structural Types (2)

How to partition?
Abstraction over Structural Types (2)

How to partition?

- *A priori local equations*: equations over local variables only.
Abstraction over Structural Types (2)

How to partition?

• **A priori local equations**: equations over local variables only.

• **A priori interface equations**: equations over interface variables only.
How to partition?

- **A priori local equations**: equations over local variables only.
- **A priori interface equations**: equations over interface variables only.
- **Mixed equations**: equations over local and interface variables.
Abstraction over Structural Types (2)

How to partition?

- **A priori local equations**: equations over local variables only.
- **A priori interface equations**: equations over interface variables only.
- **Mixed equations**: equations over local and interface variables.

Note: too few or too many local equations gives an opportunity to catch locally underdetermined or overdetermined systems of equations.
Abstraction over Structural Types (3)

In our case:
Abstraction over Structural Types (3)

In our case:

- We have 1 a priori local equation, 3 mixed equations
Abstraction over Structural Types (3)

In our case:

- We have 1 a priori local equation, 3 mixed equations
- We need to choose 3 local equations and 1 interface equation
Abstraction over Structural Types (3)

In our case:

- We have 1 a priori local equation, 3 mixed equations
- We need to choose 3 local equations and 1 interface equation
- Consequently, 3 possibilities, yielding the following possible structural types for $bar$:

\[
\begin{pmatrix}
    1 & 0 \\
    1 & 1 \\
    1 & 1 
\end{pmatrix}
\]
Abstraction over Structural Types (4)

The two last possibilities are equivalent. But still leaves two distinct possibilities. How to choose?
The two last possibilities are equivalent. But still leaves two distinct possibilities. How to choose?

- Assume the choice is free
The two last possibilities are equivalent. But still leaves two distinct possibilities. How to choose?

- Assume the choice is free
- Note that a type with more variable occurrences is “better” as it gives more freedom when pairing equations and variables. Thus discard choices that are subsumed by better choices.
The two last possibilities are equivalent. But still leaves two distinct possibilities. How to choose?

- Assume the choice is free
- Note that a type with more variable occurrences is “better” as it gives more freedom when pairing equations and variables. Thus discard choices that are subsumed by better choices.
- As a last resort, approximate.
The two last possibilities are equivalent. But still leaves two distinct possibilities. How to choose?

- Assume the choice is free
- Note that a type with more variable occurrences is “better” as it gives more freedom when pairing equations and variables. Thus discard choices that are subsumed by better choices.
- As a last resort, approximate.

Details in the paper.
Also in the Paper

- A more realistic modelling example:

- Structural types for components of this model

- Example of error in this model that is caught by the proposed method, but would not have been found by just counting equations and variables.