Assume-Guarantee Reasoning with Local Specifications

Alessio Lomuscio[1] **Ben Strulo**[2] **Nigel Walker**[2] **Peng Wu**[3]

[1]Department of Computing, Imperial College London, UK

[2]BT Innovate, Adastral Park, UK

[3]Laboratory of Computer Science, Institute of Software

Chinese Academy of Sciences, China

LOCALI, 4 November 2013

Fragrant Hill Hotel, Beijing, China

Modules

System

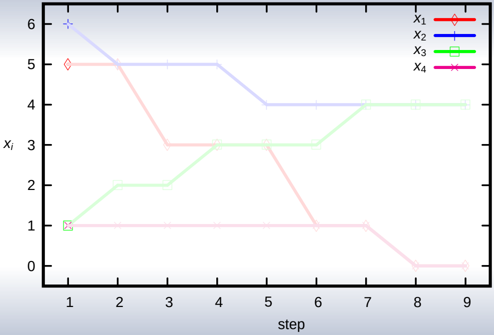
- Each module M_i exclusively controls its own **state variables** X_i (i.e., $X_i \cap X_j = \emptyset$ for any $i \neq j$).
- The state of a module M_i depends on its **input variables**, which are the state variables controlled by other modules.
- Any number of modules can evolve simultaneously.

	X_i	I_i	step function
M_1	$\{x_1\}$	$\{x_2, x_3\}$	$x_1' = x_2 - x_3$
M_2	$\{x_2\}$	$\{x_4\}$	$x_2' = x_2 - x_4$
M_3	$\{x_3\}$	$\{x_4\}$	$x_3' = x_3 + x_4$
M_4	$\{x_4\}$	$\{x_2, x_3\}$	$x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$

- x' is the next value of x.

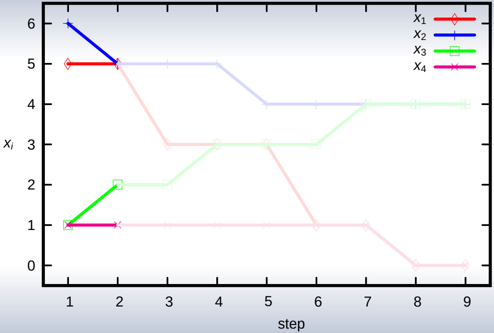
Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$



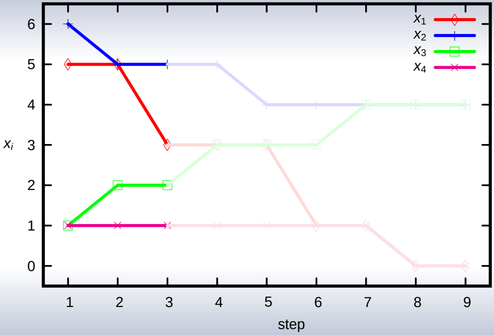
Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$



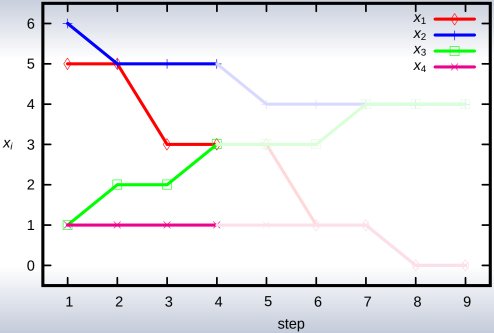
Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$



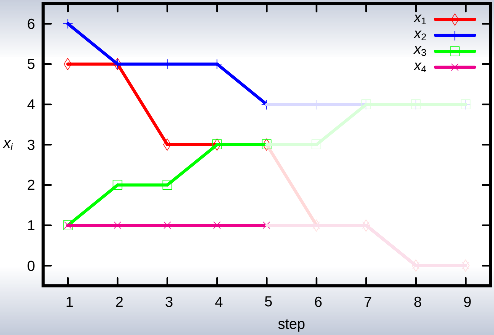
Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' =$
$$\begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$$



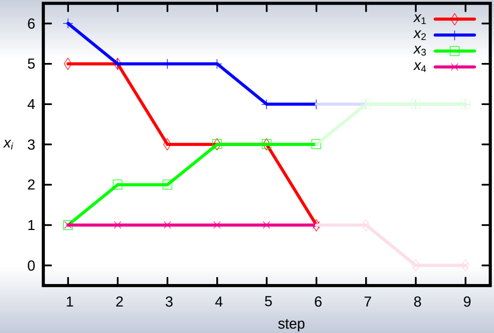
Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$



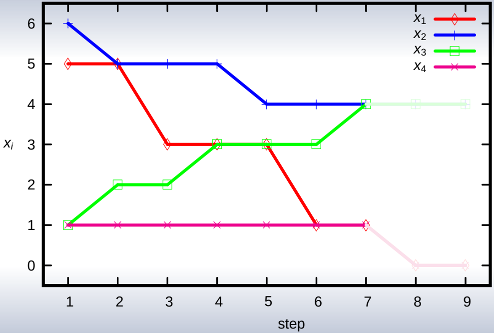
Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$



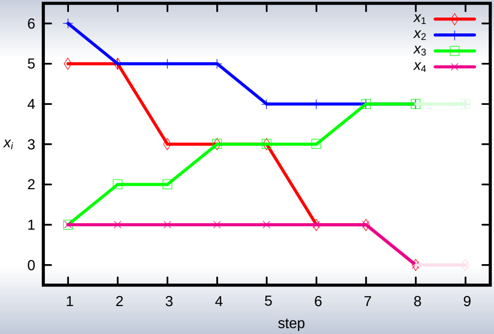
Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$



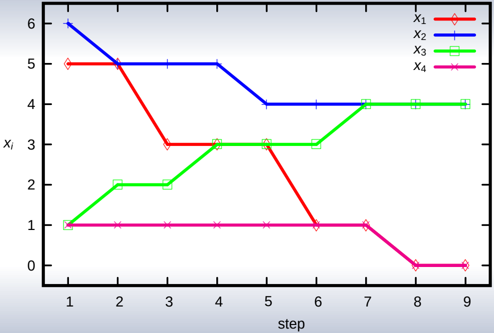
Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' =$
$$\begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$$



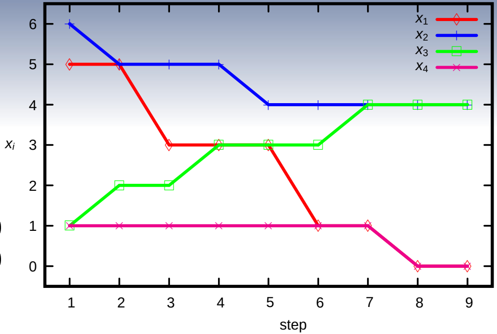
Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$



Example

- $x_1' = x_2 - x_3$
- $x_2' = x_2 - x_4$
- $x_3' = x_3 + x_4$
- $x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$



System-wide property

- Global specification on $\bigcup_{i \in [1,4]} X_i$
  - stability - $FG \bigwedge_{i \in [1,4]} x_i' = x_i$

Assume-Guarantee Reasoning

General rules

$$\text{SYM} \quad \frac{\forall 1 \le i \le n,\ M_i|A_i \ \vDash \psi \qquad \mathcal{L}(coA_1|\cdots|coA_n) = \emptyset}{M_1|\cdots|M_n \ \vDash \psi} \qquad \text{ASYM} \quad \frac{M_1|A_1 \ \vDash \psi \qquad M_2|\cdots|M_n \vDash A_1}{M_1|\cdots|M_n \ \vDash \psi}$$

- A_i - assumption; coA_i - the complement of A_i.
- Rules SYM and ASYM are sound and complete.
  - Giannakopoulou, Păsăreanu, et.al., Formal Methods in System Design, 32(3):175-205, 2008
- Assumptions are built upon the global specification ψ.
  - $M_1|A_1 \vDash FG \bigwedge_{i\in[1,4]} x_i' = x_i$
  - but M_1 depends only on x_2 and x_3 directly, but not on x_4.

Assume-Guarantee Reasoning

General rules

$$\text{SYM}\ \frac{\begin{array}{c}\forall 1 \le i \le n,\ M_i | A_i \ \models \psi \\ \mathcal{L}(coA_1 | \cdots | coA_n) = \emptyset\end{array}}{M_1 | \cdots | M_n \ \models \psi} \qquad \text{ASYM}\ \frac{\begin{array}{c}M_1 | A_1 \ \models \psi \\ M_2 | \cdots | M_n \models A_1\end{array}}{M_1 | \cdots | M_n \ \models \psi}$$

- A_i - assumption; coA_i - the complement of A_i.
- Rules SYM and ASYM are sound and complete.
    - Giannakopoulou, Păsăreanu, et.al., Formal Methods in System Design, 32(3):175-205, 2008
- Assumptions are built upon the global specification ψ.
    - $M_1 | A_1 \models FG \bigwedge_{i \in [1,4]} x'_i = x_i$
    - but M_1 depends only on x_2 and x_3 directly, but **not** on x_4.

Motivation

Observations

- Redundancy
  - A_1 has to incorporate the irrelevant state variable x_4.
- Non-reusability
  - A_1 is **not** reusable for systems extended with more modules/variables.

Solution - reasoning with local specifications

- Local specifications concern only the state and input variables of individual modules;
- for global specifications that can be regarded as conjunctions of local specifications
  - $FG \bigwedge_{i \in [1,4]} x_i' = x_i$ is equivalent to $\bigwedge_{i \in [1,4]} FG \bigwedge_{i \in X_i \cup I_i} x_i' = x_i$.

Contributions

- A sound and complete assume-guarantee rule
  - allowing reasoning about individual modules for local specifications
  - drawing conclusions on global specifications
- A case study from the field of network congestion control
  - reasoning about any number of modules, any initial state, and any network topology of bounded degree.

Tentative Rule

$$\text{SYM}\ \frac{\forall 1 \leq i \leq n,\ M_i | A_i \ \models \psi \qquad \mathcal{L}(coA_1 | \cdots | coA_n) = \emptyset}{M_1 | \cdots | M_n \ \models \psi}$$

Local specifications

- φ_i is a local specification on $X_i \cup I_i$.
- ψ is equivalent to $\bigwedge_i \varphi_i$.
- Each assumption A_i concerns only on $X_i \cup I_i$.
- Assumption A_i may admit too much interactions with module M_i.

Tentative Rule

$$\mathbf{R_0}\quad \frac{\begin{array}{c}\forall 1 \leq i \leq n,\ M_i | A_i \ \models \varphi_i \\ \mathcal{L}(coA_1 | \cdots | coA_n) = \emptyset\end{array}}{M_1 | \cdots | M_n \ \models \bigwedge_i \varphi_i}$$

Local specifications

- φ_i is a local specification on $X_i \cup I_i$.
- ψ is equivalent to $\bigwedge_i \varphi_i$.
- Each assumption A_i concerns only on $X_i \cup I_i$.
- Assumption A_i may admit too much interactions with module M_i.

Tentative Rule

$$\mathbf{R_0} \quad \frac{\begin{array}{c}\forall 1 \leq i \leq n,\ M_i | A_i \ \models \varphi_i \\ \mathcal{L}(coA_1 | \cdots | coA_n) = \emptyset\end{array}}{M_1 | \cdots | M_n \ \models \bigwedge_i \varphi_i}$$

Local specifications

- φ_i is a local specification on $X_i \cup I_i$.
- ψ is equivalent to $\bigwedge_i \varphi_i$.
- Each assumption A_i concerns only on $X_i \cup I_i$.
- Assumption A_i may admit too much interactions with module M_i.

Tentative Rule R_0

A counterexample for soundness

	X_i	I_i	step function
M_1	$\{x_1\}$	$\{x_2, x_3\}$	$x_1' = x_2 - x_3$
M_2	$\{x_2\}$	$\{x_4\}$	$x_2' = x_2 + x_4$
M_3	$\{x_3\}$	$\{x_4\}$	$x_3' = x_3 - x_4$
M_4	$\{x_4\}$	$\{x_2, x_3\}$	$x_4' = 1$

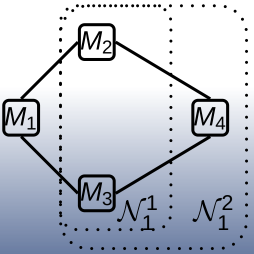
Dependency Relation

Notations

- M_i **directly depends on** M_j if $I_i \cap X_j \neq \emptyset$.
  - $\mathcal{D} = \{(M_i, M_j) \mid I_i \cap X_j \neq \emptyset\}$.
- k-Dependency - $(M_i, M_j) \in \mathcal{D}^k$ $(k \geq 1, i \neq j)$ iff
  - $(M_i, M_j) \in \mathcal{D}$ or
  - there exists M such that $(M_i, M) \in \mathcal{D}$ and $(M, M_j) \in \mathcal{D}^{k-1}$.
- Irreflexive transitive dependency closure
  - the smallest π such that $\mathcal{D}^\pi = \mathcal{D}^{\pi+1}$
- $\mathcal{N}_i^k$ - the set of all the modules M_j such that $(M_i, M_j) \in \mathcal{D}^k$
- $\mathcal{C}_i^k$ - the composition of all the modules in $\mathcal{N}_i^k$

Example

	X_i	I_i	step function
M_1	$\{x_1\}$	$\{x_2, x_3\}$	$x_1' = x_2 - x_3$
M_2	$\{x_2\}$	$\{x_4\}$	$x_2' = x_2 - x_4$
M_3	$\{x_3\}$	$\{x_4\}$	$x_3' = x_3 + x_4$
M_4	$\{x_4\}$	$\{x_2, x_3\}$	$x_4' = \begin{cases} 1 & x_2 > x_3 \wedge x_4 > 0 \\ -1 & x_2 < x_3 \wedge x_4 < 0 \\ 0 & \text{otherwise} \end{cases}$



M_i	$\mathcal{C}_i^1$	$\mathcal{C}_i^2$
M_1	$M_2 \vert M_3$	$M_2 \vert M_3 \vert M_4$
M_2	M_4	$M_3 \vert M_4$
M_3	M_4	$M_2 \vert M_4$
M_4	$M_2 \vert M_3$	$M_2 \vert M_3$

Sound Rules

$$\text{ASYM}\ \frac{\boxed{M_1|A_1 \ \models \psi} \qquad M_2|\ldots|M_n \models A_1}{M_1|\cdots|M_n \ \models \psi}$$

Completeness

- If $\mathcal{C}_i^k \models A_i$, then $\mathcal{C}_i^{k+1} \models A_i$.
- $\mathbf{R_k}$ is not complete in general.
- $\mathbf{R}_\pi$ is complete.
  - $\mathcal{D}_i^\pi = \mathcal{D}_i^{\pi+1}$ for any i.

Sound Rules

$$\mathbf{R_k}\quad \frac{\forall 1 \leq i \leq n, \quad \begin{array}{r} M_i | A_i \;\vDash \varphi_i \\ \mathcal{C}_i^k \vDash A_i \end{array}}{M_1 | \cdots | M_n \;\vDash \bigwedge_i \varphi_i}$$

Completeness

- If $\mathcal{C}_i^k \vDash A_i$, then $\mathcal{C}_i^{k+1} \vDash A_i$.
- $\mathbf{R_k}$ is not complete in general.
- $\mathbf{R}_\pi$ is complete.
  - $\mathcal{D}_i^\pi = \mathcal{D}_i^{\pi+1}$ for any i.

Sound and Complete Rule

Bounded rule

$$\mathbf{R}^{\pi} \quad \frac{\forall 1 \leq i \leq n, \quad \begin{array}{r} M_i | A_i \ \models \varphi_i \\ \exists 1 \leq d_i \leq \pi, \ \mathcal{C}_i^{d_i} \models A_i \end{array}}{M_1 | \cdots | M_n \ \models \bigwedge_{i=1}^{n} \varphi_i}$$

Weakest assumptions

- For module M_i, WA_i is the weakest assumption with respect to the local specification φ_i, i.e.,
  - $\mathcal{L}(WA_i) \subseteq \mathcal{I}(M_i)$ and $M_i | WA_i \models \varphi_i$;
  - $\mathcal{L}(A_i) \subseteq \mathcal{L}(WA_i)$ for any A_i such that $\mathcal{L}(A_i) \subseteq \mathcal{I}(M_i)$ and $M_i | A_i \models \varphi_i$.
- The weakest assumption WA_i admits as many as possible sequences of inputs to module M_i without violating the local specification φ_i.

Incremental Algorithm

Generate the weakest assumption WA_i from the local specification φ_i;
$d_i \leftarrow 1$;
while $\mathcal{C}_i^{d_i} \not\models WA_i$ **do**
    if $\mathcal{N}_i^{d_i} \neq \mathcal{N}_i^{d_i+1}$ **then**
        $d_i \leftarrow d_i + 1$;
    else
        return false;
return true;

Multi-Path Congestion/Rate Control

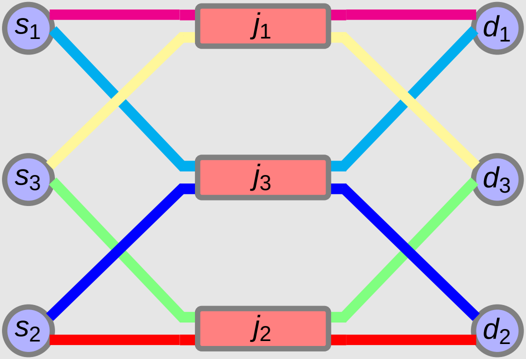
Optimisation-based network control

- Synchronous fluid-flow models by Frank Kelly et al. in 2005
- Stability of the algorithms has been well proved.

Nondeterminism

- Degeneracy (non-unique local equilibrium)
  - multiple equal-cost paths
  - random (re-)routing
- Process fairness

Network



- Sources s
- **Shared** resources (i.e. links) j
- Routes r

Distributed System

State variables

- $\{x_r \mid r \in s\}$ - the flow rate on route $r \in s$

Input variables

- $\{x_{r'} \mid \text{route } r' \text{ shares resources with some route } r \in s\}$

Step function

$$\begin{aligned}
\Delta x_r &= \kappa_r x_r \left(1 - \frac{1}{\alpha_s} \sum_{j \in r} y_j \sum_{r' \in s(r)} x_{r'}\right)^{+}_{x_r} \\
y_j &= \beta_j \sum_{j \in r} x_r
\end{aligned}$$

Global Specification

Stability of route flow rates

$$FG \bigwedge_{s_i} \Big(\bigwedge_{r \in s_i} x'_r = x_r \Big)$$

$$\bigwedge_{s_i} FG \bigwedge_{r \in \gamma(s_i)} x'_r = x_r$$

- x'_r is the next value of x_r.
- $r \in \gamma(s)$ if
  - $r \in s$, or
  - there exists j such that $j \in r$ and $j \in r'$ for some $r' \in s$.

Global Specification

Stability of route flow rates

$$FG \bigwedge_{s_i} \left(\bigwedge_{r \in s_i} x'_r = x_r \right)$$

$$\bigwedge_{s_i} FG \bigwedge_{r \in \gamma(s_i)} x'_r = x_r$$

- x'_r is the next value of x_r.
- $r \in \gamma(s)$ if
  - $r \in s$, or
  - there exists j such that $j \in r$ and $j \in r'$ for some $r' \in s$.

Assume-Guarantee Reasoning

System stability

$$\textbf{SS}\ \frac{\forall 1 \leq i \leq n, \quad \begin{array}{c} M_i | A_i \ \vDash FG \bigwedge_{x_r \in X_i \cup I_i} x_r' = x_r \\ \exists 1 \leq d_i \leq \pi,\ \mathcal{C}_i^{d_i} \vDash A_i \end{array}}{M_1 | \cdots | M_n \ \vDash \ FG \bigwedge_{s_i} (\bigwedge_{r \in s_i} x_r' = x_r)}$$

- Source s_i is represented as module M_i.

Parameterised Model Checking

Network topology of bounded degree

- Each source has a bounded number of routes.
- Each source shares resources with a bounded number of other sources.

Any number of modules, any initial state

- Each source is associated with a bounded number of state and input variables.
    - each module is of the general form $M_{\vec{u}}$, where $\vec{u}$ ranges over the set of initial states.
- The number of modules is not relevant any more.
- All possible initial states can be taken into account together.

Experimental Results

Network settings

- Each source has two routes, each using one shared resource.
- Each resource is shared by two sources.
- $D = [1, 6]$

$d_i \leftarrow 1$

For any $\vec{u}, \vec{v}, \vec{w}_1, \vec{w}_2 \in D^2$,

$$\begin{array}{c} M_{\vec{u}} | A_{\vec{v}} \models FG \bigwedge_{x_r \in X_i \cup I_i} x'_r = x_r \\ M_{\vec{w}_1} | M_{\vec{w}_2} \models A_{\vec{v}} \end{array}$$

Experimental Results

u_0	v_0	$A^{\psi}_{u_0,v_0}$			A_{u_0,v_0}			coA_{u_0,v_0}	
		#st.	#trans.	time(s)	#st.	#trans.	time(s)	#st.	#trans.
1	1	1332	49248	1511.0	37	108	3.3	73	2628
1	2	1332	49248	1475.1	37	108	1.9	73	2628
1	3	1332	49248	1415.8	37	108	1.7	73	2628
1	4	2016	97200	3292.9	56	180	3.5	110	3960
1	5	2268	144288	4247.5	63	228	4.9	123	4428
1	6	2304	190944	5693.8	64	264	5.0	124	4464
2	2	1332	49248	1477.2	37	108	1.6	73	2628
2	3	4752	195840	14207.2	132	400	19.3	114	4104
2	4	5760	291024	21088.4	160	524	31.5	168	6048
2	5	5796	337680	25180.2	161	560	33.1	169	6084
2	6	6084	431424	-	169	644	34.6	183	6588
3	3	8532	389736	-	237	746	77.3	174	6264
3	4	9648	531720	-	268	910	103.5	233	8388
3	5	9684	578376	-	269	946	106.4	234	8424
3	6	9756	671688	-	271	1018	105.9	236	8496
4	4	8568	436392	-	238	782	74.7	175	6300
4	5	9684	578376	-	269	946	108.1	234	8424
4	6	9684	578376	-	269	946	104.1	234	8424
5	5	10836	767016	-	301	1146	145.1	294	10584
5	6	10836	767016	-	301	1146	138.0	294	10584
6	6	10836	767016	-	301	1146	138.1	294	10584

Behavioural Constraint Propagation

Generalized Arc Consistency

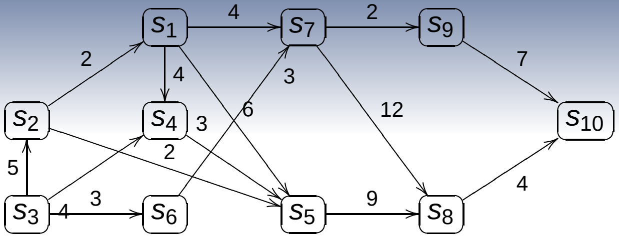
For a constraint $C_{x_1,\ldots,x_k}$, $GAC(C_{x_1\ldots,x_k}) : \forall 1 \leq i \leq k, u \in D_i \models C_{x_1,\ldots,x_k}(v_1,\ldots,v_{i-1},u,v_{i+1}\ldots,v_k)$ *for* $v_j \in D_j, j = 1,\ldots,i-1, i+1,\ldots,k$

Compositional State Space Reduction

For $M_{x_1},\ldots,M_{x_i},\ldots,M_{x_k}$ such that M_{x_i} depends on M_{x_j} for $j = 1,\ldots,i-1, i+1,\ldots,k$,

For each run w of M_{x_i}, $w \upharpoonright_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_k}$ is admitted by $M_{x_1}|\cdots|M_{x_{i-1}}|M_{x_{i+1}}|\cdots|M_{x_k}$.

Experimental Results



i	A_i		coA_i		A'_i		coA'_i	
	#state	#trans	#state	#trans	#state	#trans	#state	#trans
1	27	78608	27	132651	9	126	6	216
2	31	5202	31	8959	13	59	8	120
3	29	83521	29	142477	21	452	12	1344
4	29	289	29	493	7	12	5	15
5	17	187	17	289	5	8	4	8
6	29	289	29	493	7	12	5	15
7	31	5202	31	8959	5	12	4	16
8	27	272	27	459	3	4	1	0
9	21	221	21	357	3	4	1	0

1. Introduction
2. Reasoning with Local Specifications
3. Case Study
4. Current Work
5. **Conclusions**

Summary

- A sound and complete assume-guarantee rule for local specification
- An incremental compositional model checking approach

Work in progress

- Behavioural constraint propagation