

An extension of Hoare logic to identify the parameters of gene regulatory networks

Gilles Bernot

University of Nice sophia antipolis, I3S laboratory, France

Joint work with Jean-Paul Comet, Zohra Khalis,
Adrien Richard and Olivier Roux

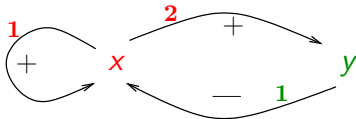
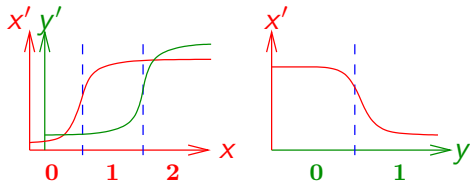


Acknowledgments: Epigenomics Project (Genopole®)

Menu

1. Thomas' multivalued networks
2. Reducing the number of parameters
3. Reminders on Hoare logic
4. Extended assertion and path languages
5. Genetically Modified Hoare Logic

Regulatory Networks (R. Thomas)



No help : K_x

x helps : $K_{x,x}$

Absent y helps : $K_{x,\bar{y}}$

Both : $K_{x,x\bar{y}}$

K_y

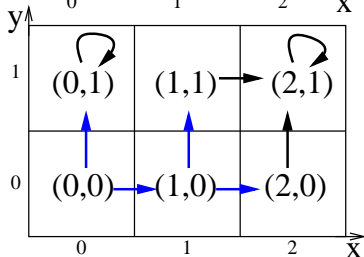
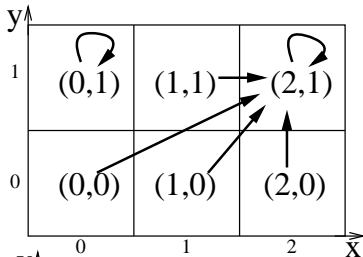
$K_{y,x}$

(x,y)	<i>Focal Point</i>
(0,0)	$(K_{x,\bar{y}}, K_y)$
(0,1)	(K_x, K_y)
(1,0)	$(K_{x,x\bar{y}}, K_y)$
(1,1)	$(K_{x,x}, K_y)$
(2,0)	$(K_{x,x\bar{y}}, K_{y,x})$
(2,1)	$(K_{x,x}, K_{y,x})$

State Graphs

(x,y)	<i>Focal Point</i>
(0,0)	$(K_{x,\bar{y}}, K_y)=(2,1)$
(0,1)	$(K_x, K_y)=(0,1)$
(1,0)	$(K_{x,x\bar{y}}, K_y)=(2,1)$
(1,1)	$(K_{x,x}, K_y)=(2,1)$
(2,0)	$(K_{x,x\bar{y}}, K_{y,x})=(2,1)$
(2,1)	$(K_{x,x}, K_{y,x})=(2,1)$

“desynchronization” \longrightarrow
by units of Manhattan distance



The main problem

Identify the sets of parameters
that cope with known behaviours
from biological experiments,
exhaustively

Solution = formal methods.

- ▶ 2003: enumeration + CTL + model checking (Bernot, Comet, Pérès, Richard)
- ▶ 2005: path derivatives + model checking (Batt, De Jong)
- ▶ 2005: PROLOG with constraints (Trilling, Corblin, Fanchon)
- ▶ 2007: symbolic execution + LTL (Mateus, Le Gall, Comet)
- ▶ 2011: traces + enumeration + CTL + model checking (Siebert, Bockmayr)

(several other formal approaches define extensions of the theory)

Menu

1. Thomas' multivalued networks
2. Reducing the number of parameters
3. Reminders on Hoare logic
4. Extended assertion and path languages
5. Genetically Modified Hoare Logic

Exponential number of parameters

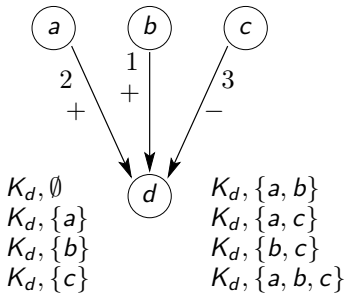
Far more complex than Petri nets:

2^i parameters

where i is the in-degree of the gene

$\prod_{genes} (o + 1)^{2^i}$ possible parameter values

where o is the out degree of the gene



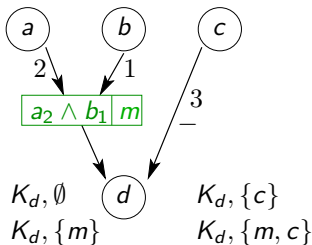
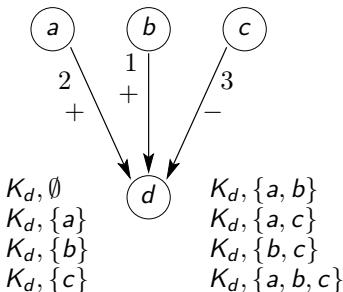
Yeast ≈ 7000 genes

Human ≈ 25000 genes

Rice ≈ 40000 genes

Multiplex: encodes cooperation knowledge

“Proteins of a and b form a complex before acting on d ...”



multiplex name = m

multiplex formula $\equiv a_2 \wedge b_1$

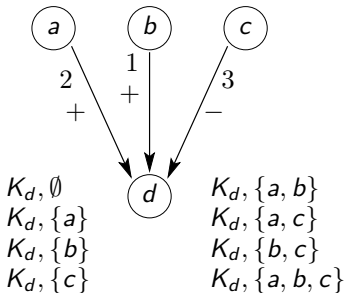
abbreviation:

$v_i \equiv (v \geq i)$

8 \rightarrow 4 parameters

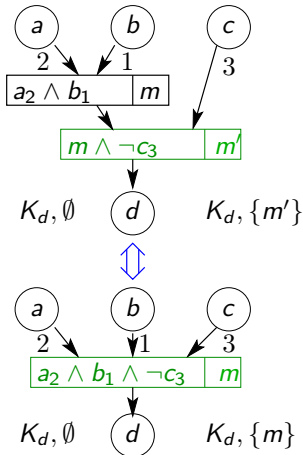
Any propositional formula + remove sign

“... and c inhibits d whatever a or b ”



$8 \rightarrow 2$ parameters,

$(o + 1)^8 \rightarrow (o + 1)^2$ parameterizations



Menu

1. Thomas' multivalued networks
2. Reducing the number of parameters
3. Reminders on Hoare logic
4. Extended assertion and path languages
5. Genetically Modified Hoare Logic

swap(x,y)

```
aux := x;  
x  := y;  
y  := aux
```

→ triple “ $\{P\} \text{pgm} \{Q\}$ ”
precondition P , postcondition Q

swap(x,y)

$\{(x = x_0) \wedge (y = y_0)\}$

aux := x;

x := y;

y := aux

$\{(y = x_0) \wedge (x = y_0)\}$

→ “P \implies (weakest precondition)” ?

swap(x,y)

$\{(x = x_0) \wedge (y = y_0)\}$

aux := x;

x := y;

y := aux

$\{(aux = x_0) \wedge (x = y_0)\}$

$\{(y = x_0) \wedge (x = y_0)\}$

→ backward calculus

swap(x,y)

$\{(x = x_0) \wedge (y = y_0)\}$

aux := x;

x := y; ← $\{(aux = x_0) \wedge (y = y_0)\}$

y := aux ← $\{(aux = x_0) \wedge (x = y_0)\}$

$\{(y = x_0) \wedge (x = y_0)\}$

swap(x,y)

$\{(x = x_0) \wedge (y = y_0)\}$

← $\{(x = x_0) \wedge (y = y_0)\}$
aux := x;
← $\{(aux = x_0) \wedge (y = y_0)\}$
x := y;
← $\{(aux = x_0) \wedge (x = y_0)\}$
y := aux

$\{(y = x_0) \wedge (x = y_0)\}$

swap(x,y)

$\{(x = x_0) \wedge (y = y_0)\}$

← $\{(x = x_0) \wedge (y = y_0)\}$
aux := x;
← $\{(aux = x_0) \wedge (y = y_0)\}$
x := y;
← $\{(aux = x_0) \wedge (x = y_0)\}$
y := aux

$\{(y = x_0) \wedge (x = y_0)\}$

$\frac{}{\{Q[v \leftarrow expr]\} v := expr \{Q\}} :=$

$\frac{\{P\}p_1\{Q'\} \quad \{Q'\}p_2\{Q\}}{\{P\}p_1; p_2\{Q\}} ;$

swap(x,y)

$\{(x = x_0) \wedge (y = y_0)\}$

$\leftarrow \{(x = x_0) \wedge (y = y_0)\}$
 aux := x;
 $\leftarrow \{(aux = x_0) \wedge (y = y_0)\}$
 x := y;
 $\leftarrow \{(aux = x_0) \wedge (x = y_0)\}$
 y := aux

$\{(y = x_0) \wedge (x = y_0)\}$

$$\frac{}{\{Q[v \leftarrow expr]\} \ v := expr \ \{Q\}} :=$$

$$\frac{\{P\}p_1\{Q'\} \quad \{Q'\}p_2\{Q\}}{\{P\}p_1; p_2\{Q\}} ;$$

$$\frac{}{\{Q_3\}a_1\{Q_2\}} := \frac{}{\{Q_2\}a_3\{Q_1\}} :=$$

$$\{P\}a_1; a_2\{Q_1\}$$

$$\frac{}{\{Q_1\}a_3\{Q\}} :=$$

$$\{P\}a_1; a_2; a_3\{Q\}$$

abs(x)

$$\begin{array}{l} \{(x = x_0)\} \\ \text{if } (x < 0) : \\ \quad r := -x; \\ \quad r := x \\ \{(r \geq 0) \wedge (r^2 = x_0^2)\} \end{array} \left\{ \left(\begin{array}{l} (x < 0) \\ (-x \geq 0) \\ ((-x)^2 = x^2) \end{array} \wedge \right) \vee \left(\begin{array}{l} (x \geq 0) \\ (x \geq 0) \\ (x^2 = x^2) \end{array} \wedge \right) \right\}$$

$$\frac{\{Q_1\}p_1\{Q\} \quad \{Q_2\}p_2\{Q\}}{\{(e \wedge Q_1) \vee (\neg e \wedge Q_2)\} \text{ if } e \text{ then } p_1 \text{ else } p_2 \{Q\}} \text{ if}$$

Also:

Empty program: $\frac{P \implies Q}{\{P\}\varepsilon\{Q\}}$ use sparingly: loses *weakest* precondition!

“While sub-specification”:
 $\frac{\{e \wedge I\}p\{I\}}{\{I\}\text{while } e \text{ with } I \text{ do } p\{\neg e \wedge I\}}$

Menu

1. Thomas' multivalued networks
2. Reducing the number of parameters
3. Reminders on Hoare logic
4. Extended assertion and path languages
5. Genetically Modified Hoare Logic

Assertion language (Pre/Post)

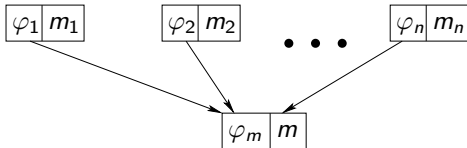
Terms: v gene | $n \in \mathbf{N}$ | $K_{v,\{\dots\}}$ parameter symbols | $+$ | $-$
atoms: comparison or equality of 2 terms

Connectives: \neg | \wedge | \vee | \implies

Example:

$$(a \leq 3 \wedge d+1 < K_{d,\{m,c\}}) \vee (K_{d,\{c\}} < K_{d,\{m,c\}} \wedge c \geq 3)$$

From multiplexes to assertions: flatening



$$\overline{\varphi_m} \equiv \varphi_m[m_i \leftarrow \varphi_i] \text{ for all } i \text{ and recursively}$$

Assertions that formalize Thomas' framework

ω is the set of resources of v :

$$\Phi_v^\omega \equiv \left(\bigwedge_{m \in \omega} \overline{\varphi_m} \right) \wedge \left(\bigwedge_{m \in G^{-1}(v) \setminus \omega} \neg \overline{\varphi_m} \right)$$

v could increase:

$$\Phi_v^+ \equiv (0 \leq v) \wedge (v < b_v) \wedge \left(\bigwedge_{\omega \subset G^{-1}(v)} (\Phi_v^\omega \implies K_{v,\omega} > v) \right)$$

v could decrease:

$$\Phi_v^- \equiv (0 < v) \wedge (v \leq b_v) \wedge \left(\bigwedge_{\omega \subset G^{-1}(v)} (\Phi_v^\omega \implies K_{v,\omega} < v) \right)$$

v cannot change:

$$v= \equiv \left(\bigwedge_{\omega \subset G^{-1}(v)} (\Phi_v^\omega \implies K_{v,\omega} = v) \right)$$

Path language

- ▶ $x+ \mid x- \mid x := n$
- ▶ $p_1; p_2; \dots; p_n$
- ▶ $p_1; \varphi; p_2$ (where φ assertion)
- ▶ *if φ then p_1 else p_2*
- ▶ *while φ with ψ do p*
- ▶ $\forall(p_1, p_2, \dots, p_n)$
- ▶ $\exists(p_1, p_2, \dots, p_n)$

Examples:

- ▶ $b+; c+; b-$
- ▶ $\exists(b+, b-, c+, c-)$
- ▶ *while $(b < 2)$ with $(c > 0)$*
do $\exists(b+, b-, \forall((c-; a-), c+))$ od;
 $b-$

Menu

1. Thomas' multivalued networks
2. Reducing the number of parameters
3. Reminders on Hoare logic
4. Extended assertion and path languages
5. Genetically Modified Hoare Logic

Additional inference rules

Incrementation rule: $\frac{}{\{ \Phi_v^+ \wedge Q[\kappa \leftarrow v+1] \} v+ \{ Q \}}$

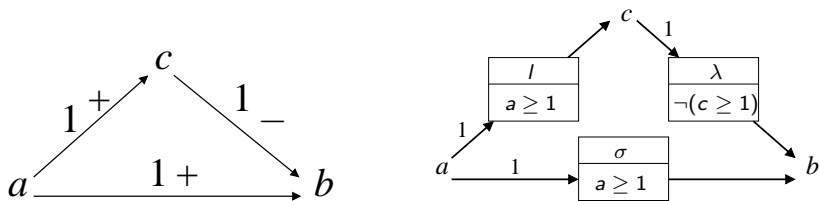
Decrementation rule: $\frac{}{\{ \Phi_v^- \wedge Q[\kappa \leftarrow v-1] \} v- \{ Q \}}$

Assertion rule: $\frac{\{ \varphi \wedge Q \}}{\varphi \{ Q \}}$

Universal quantifier rule: $\frac{\{ P_1 \} p_1 \{ Q \} \quad \{ P_2 \} p_2 \{ Q \}}{\{ P_1 \wedge P_2 \} \forall (p_1, p_2) \{ Q \}}$

Existential quantifier rule: $\frac{\{ P_1 \} p_1 \{ Q \} \quad \{ P_2 \} p_2 \{ Q \}}{\{ P_1 \vee P_2 \} \exists (p_1, p_2) \{ Q \}}$

Example: Feedforward “loop”



Behaviour of b after switching a from off to on ?

Simple off \rightarrow on \rightarrow off behaviour of b with the help of c :

$$\{(a = 1 \wedge b = 0 \wedge c = 0)\} b+ ; c+ ; b- \{b = 0\}$$

possible if and only if: $K_{b,\{\sigma,\lambda\}} = 1 \wedge K_{c,\{l\}} = 1 \wedge K_{b,\{\sigma\}} = 0$

Feedforward example (continued)

off→on→off behaviour of b without the help of c :

$$\{(a = 1 \wedge b = 0 \wedge c = 0)\} b+ ; b- \{b = 0\}$$

$$\left\{ \begin{array}{l} b = 0 \\ ((c \geq 1) \wedge (a < 1)) \implies ((K_b = 1) \wedge (K_b = 0)) \\ ((c \geq 1) \wedge (a \geq 1)) \implies ((K_{b,\sigma} = 1) \wedge (K_{b,\sigma} = 0)) \\ ((c < 1) \wedge (a < 1)) \implies ((K_{b,\lambda} = 1) \wedge (K_{b,\lambda} = 0)) \\ ((c < 1) \wedge (a \geq 1)) \implies ((K_{b,\sigma\lambda} = 1) \wedge (K_{b,\sigma\lambda} = 0)) \end{array} \right\} \text{not satisfiable!}$$

Feedforward example (continued)

Although $b+; c+; b-$ is possible, if c becomes “on” before b , then b will never be able to get “on”

Proof by refutation:

$$\left\{ \begin{array}{l} a = 1 \wedge b = 0 \wedge c = 1 \wedge \\ K_{b,\sigma\lambda} = 1 \wedge K_{c,l} = 1 \wedge K_{b,\sigma} = 0 \end{array} \right\}$$

while $b < 1$ with l do $\exists(b+, b-, c+, c-)$

$$\{ b = 1 \}$$

the triple is inconsistent, whatever the loop invariant l !

Concluding Comments

Pros:

- ▶ simple and elegant
- ▶ easy expression of sequential biological observations, copes well with biological measurements
- ▶ dynamic gene Knock Out within the same model
- ▶ infinite traces can be handled
- ▶ avoids numerous model checkings

Cons:

- ▶ no “holes” in the successive biological observations
- ▶ thresholds are empirical
- ▶ *while* invariant . . . except for proofs by refutation
- ▶ simplification of assertions/constraint solving can be difficult

WP-SMBioNet: proof of feasibility using Choco