Efficient analysis on very large models

Maxime FOLSCHETTE
MeForBio / IRCCyN / École Centrale de Nantes (Nantes, France)
maxime.folschette@irccyn.ec-nantes.fr
http://maxime.folschette.name/
Efficient analysis on very large models

Introduction

Context and Aims

MeForBio team:
Qualitative modelling to study large dynamical biological systems
Efficient analysis on very large models

Introduction

Context and Aims

MeForBio team:
Qualitative modelling to study large dynamical biological systems

1) The object: **Gene regulations**

   *Large discrete models* to study gene interactions
Efficient analysis on very large models

Introduction

Context and Aims

MeForBio team:
Qualitative modelling to study large dynamical biological systems

1) The object: Gene regulations
   Large discrete models to study gene interactions

2) The method: Static analysis
   Efficient methods thanks to the Process Hitting framework
Efficient analysis on very large models

Introduction

Context and Aims

MeForBio team:
Qualitative modelling to study large dynamical biological systems

1) The object: Gene regulations
   Large discrete models to study gene interactions

2) The method: Static analysis
   Efficient methods thanks to the Process Hitting framework

3) The result: Applications
   The example of gene therapies
Efficient analysis on very large models - Gene regulations

Gene regulations

DNA

Transcription

RNA

Translation

Protein

NUCLEUS

CYTOPLASM

© 2012 Pearson Education, Inc.
Efficient analysis on very large models • Gene regulations

Gene regulations

a
Usual biological algebraic models


Modelling interacting genes/proteins: **Boolean Networks**

\[ a + b \rightarrow -z \]

\[ -b + z \rightarrow +a \]
Efficient analysis on very large models

Gene regulations

Usual biological algebraic models


Modelling interacting genes/proteins: **Boolean Networks**

Questions:

- How does $z$ behave?
- Is it **possible** to make $a$ inactive?
- If I **knock-out** $b$, what changes?
The combinatorial explosion

→ Problem: easy to understand but hard to study
  exponential number of states

<table>
<thead>
<tr>
<th>Model</th>
<th>Possible states</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram of a and b" /></td>
<td>4</td>
</tr>
</tbody>
</table>
The combinatorial explosion

→ Problem: easy to understand but hard to study

exponential number of states

<table>
<thead>
<tr>
<th>Model</th>
<th>Possible states</th>
</tr>
</thead>
<tbody>
<tr>
<td>a (\rightarrow) b (\rightarrow) c (\rightarrow) a (\rightarrow) b</td>
<td>4, 8</td>
</tr>
</tbody>
</table>
The combinatorial explosion

→ Problem: easy to understand but hard to study

*exponential number of states*

<table>
<thead>
<tr>
<th>Model</th>
<th>Possible states</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>4</td>
</tr>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>8</td>
</tr>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>1024</td>
</tr>
</tbody>
</table>
The combinatorial explosion

→ Problem: easy to understand but hard to study

*exponential number of states*

<table>
<thead>
<tr>
<th>Model</th>
<th>Possible states</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>4</td>
</tr>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>8</td>
</tr>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>1024</td>
</tr>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>1048576</td>
</tr>
</tbody>
</table>
The combinatorial explosion

→ Problem: easy to understand but hard to study

exponential number of states

<table>
<thead>
<tr>
<th>Model</th>
<th>Possible states</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>4</td>
</tr>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>8</td>
</tr>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>1024</td>
</tr>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>1048576</td>
</tr>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td>1267650600000000000000000000000</td>
</tr>
</tbody>
</table>

Efficient analysis on very large models ○ Gene regulations
The Process Hitting modelling

[Paulevé et al., *Transactions on Computational Systems Biology*, 2011]

**Sorts**: components $a$, $b$, $z$
The Process Hitting modelling

[Paulevé et al., Transactions on Computational Systems Biology, 2011]

Sorts: components $a$, $b$, $z$

Processes: local states / levels of expression $z_0$, $z_1$, $z_2$
Efficient analysis on very large models ○ Studying large models ○ The Process Hitting framework

The Process Hitting modelling

[Paulevé et al., Transactions on Computational Systems Biology, 2011]

Sorts: components $a$, $b$, $z$

Processes: local states / levels of expression $z_0$, $z_1$, $z_2$

States: sets of active processes $\langle a_0, b_1, z_0 \rangle$
The Process Hitting modelling

[Paulevé et al., Transactions on Computational Systems Biology, 2011]

**Sorts**: components \( a, b, z \)

**Processes**: local states / levels of expression \( z_0, z_1, z_2 \)

**States**: sets of active processes \( \langle a_0, b_1, z_0 \rangle \)

**Actions**: dynamics \( b_1 \rightarrow z_0 \uparrow z_1, a_0 \rightarrow a_0 \uparrow a_1, a_1 \rightarrow z_1 \uparrow z_2 \)
Efficient analysis on very large models

Studying large models

The Process Hitting framework

The Process Hitting modelling

[Paulevé et al., Transactions on Computational Systems Biology, 2011]

**Sorted Types:*** components\( a, b, z \)

**Processes:** local states / levels of expression \( z_0, z_1, z_2 \)

**States:** sets of active processes \( \langle a_0, b_1, z_0 \rangle \)

**Actions:** dynamics \( b_1 \rightarrow z_0 \uparrow z_1, a_0 \rightarrow a_0 \uparrow a_1, a_1 \rightarrow z_1 \uparrow z_2 \)
The Process Hitting modelling

[Paulevé et al., Transactions on Computational Systems Biology, 2011]

**Sorts**: components \( a, b, z \)

**Processes**: local states / levels of expression \( z_0, z_1, z_2 \)

**States**: sets of active processes \( \langle a_0, b_1, z_1 \rangle \)

**Actions**: dynamics \( b_1 \rightarrow z_0 \uparrow z_1, a_0 \rightarrow a_0 \uparrow a_1, a_1 \rightarrow z_1 \uparrow z_2 \)
The Process Hitting modelling

[Paulevé et al., Transactions on Computational Systems Biology, 2011]

**Sorts**: components $a, b, z$

**Processes**: local states / levels of expression $z_0, z_1, z_2$

**States**: sets of active processes $\langle a_1, b_1, z_1 \rangle$

**Actions**: dynamics $b_1 \rightarrow z_0 \triangleright z_1$, $a_0 \rightarrow a_0 \triangleright a_1$, $a_1 \rightarrow z_1 \triangleright z_2$
The Process Hitting modelling

[Paulevé et al., Transactions on Computational Systems Biology, 2011]

Sorts: components \( a, b, z \)

Processes: local states / levels of expression \( z_0, z_1, z_2 \)

States: sets of active processes \( \langle a_1, b_1, z_2 \rangle \)

Actions: dynamics \( b_1 \rightarrow z_0 \uparrow z_1, a_0 \rightarrow a_0 \uparrow a_1, a_1 \rightarrow z_1 \uparrow z_2 \)
Static analysis: successive reachability

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

- Initial state

\[ \langle a_1, b_0, c_0, d_0 \rangle \]
Static analysis: successive reachability

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

- Initial state
\[ \langle a_1, b_0, c_0, d_0 \rangle \]
- Objectives
\[ [ \overset{\text{\tiny\textrightarrow}}{d_1} : \overset{\text{\tiny\textrightarrow}}{d_2} ] \]
Efficient analysis on very large models

Studying large models

Static analysis

Static analysis: successive reachability

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

- Initial state
  \[ \langle a_1, b_0, c_0, d_0 \rangle \]

- Objectives
  \[
  \begin{align*}
    & \langle d_1 \rangle \quad \langle d_2 \rangle \\
    \iff & \langle d_1 \rangle \quad \langle b_1 \rangle \quad \langle d_2 \rangle
  \end{align*}
  \]
Static analysis: successive reachability

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

- Initial state
  $\langle a_1, b_0, c_0, d_0 \rangle$

- Objectives
  $[ \uparrow d_1 \uparrow d_2 ]$
  $[ \uparrow d_1 \uparrow b_1 \uparrow d_2 ]$
  $[ \uparrow d_2 ]$
Static analysis: successive reachability

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

- Initial state

\[ \langle a_1, b_0, c_0, d_0 \rangle \]

- Objectives

\[ \begin{align*}
\overset{\rightarrow}{d_1} &:: \overset{\rightarrow}{d_2} \\
\overset{\rightarrow}{d_1} &:: \overset{\rightarrow}{b_1} :: \overset{\rightarrow}{d_2} \\
\overset{\rightarrow}{d_2} &
\end{align*} \]

\[ \overset{\rightarrow}{\text{Concretization of the objective \()=\text{ scenario}} \]

\[ a_0 \rightarrow c_0 \overset{\rightarrow}{c_1} :: b_0 \rightarrow d_0 \overset{\rightarrow}{d_1} :: c_1 \rightarrow b_0 \overset{\rightarrow}{b_1} :: b_1 \rightarrow d_1 \overset{\rightarrow}{d_2} \]
Static analysis: successive reachability

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

- Initial state
  \[ \langle a_1, b_0, c_0, d_0 \rangle \]
- Objectives
  \[ \uparrow d_1 :: \uparrow d_2 \]
  \[ \uparrow d_1 :: \uparrow b_1 :: \uparrow d_2 \]
  \[ \uparrow d_2 \]

→ Concretization of the objective = scenario
\[ a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2 \]
Static analysis: successive reachability

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

- Initial state
  \( \langle a_1, b_0, c_0, d_0 \rangle \)
- Objectives
  \[
  \begin{align*}
  & [ \uparrow d_1 \:: \uparrow d_2 ] \\
  & [ \uparrow d_1 \:: \uparrow b_1 \:: \uparrow d_2 ] \\
  & [ \uparrow d_2 ]
  \end{align*}
  \]

→ Concretization of the objective = scenario

\[
\begin{align*}
  & a_0 \rightarrow c_0 \uparrow c_1 :: b_0 \rightarrow d_0 \uparrow d_1 :: c_1 \rightarrow b_0 \uparrow b_1 :: b_1 \rightarrow d_1 \uparrow d_2
\end{align*}
\]
Static analysis: successive reachability

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

- Initial state
  \[ \langle a_1, b_0, c_0, d_0 \rangle \]

- Objectives
  \[
  \begin{array}{c}
  \vdash d_1 :: \vdash d_2 \\
  \vdash d_1 :: \vdash b_1 :: \vdash d_2 \\
  \vdash d_2
  \end{array}
  \]

\[ \rightarrow \text{Concretization of the objective = scenario} \]
\[ a_0 \rightarrow c_0 \vdash c_1 :: b_0 \rightarrow d_0 \vdash d_1 :: c_1 \rightarrow b_0 \vdash b_1 :: b_1 \rightarrow d_1 \vdash d_2 \]
Static analysis: successive reachability

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

- Initial state
  \( \langle a_1, b_0, c_0, d_0 \rangle \)

- Objectives
  \[
  \begin{align*}
  &\downarrow d_1 :: \downarrow d_2 \\ &\downarrow d_1 :: \downarrow b_1 :: \downarrow d_2 \\ &\downarrow d_2
  \end{align*}
  \]

→ Concretization of the objective = scenario
\[
\begin{align*}
  a_0 &\rightarrow c_0 \downarrow c_1 :: b_0 \rightarrow d_0 \downarrow d_1 :: c_1 \rightarrow b_0 \downarrow b_1 :: b_1 \rightarrow d_1 \downarrow d_2
\end{align*}
\]
Over- and Under-approximations

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

→ Directly checking $R$ is hard (exponential)
→ Rather check approximations $P$ and $Q$ so that: $P \Rightarrow R \Rightarrow Q$
Efficient analysis on very large models ◦ Studying large models ◦ Static analysis

Over- and Under-approximations


→ Directly checking $R$ is hard (exponential)
→ Rather check approximations $P$ and $Q$ so that: $P \implies R \implies Q$
Efficient analysis on very large models

- Studying large models
- Static analysis

Over- and Under-approximations

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

→ Directly checking $R$ is hard (exponential)

→ Rather check approximations $P$ and $Q$ so that: $P \Rightarrow R \Rightarrow Q$

---

Over-Approximation

Exact solutions

$R$

$\neg Q$

Maxime FOLSCHETTE

MOVEP'14 — 2014/07/09
Over- and Under-approximations

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

→ Directly checking $R$ is hard (exponential)
→ Rather check approximations $P$ and $Q$ so that: $P \Rightarrow R \Rightarrow Q$

Over-Approximation

Under-Approximation

Exact solutions
Efficient analysis on very large models

Studying large models

Static analysis

Over- and Under-approximations

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

→ Directly checking $R$ is hard (exponential)
→ Rather check approximations $P$ and $Q$ so that: $P \Rightarrow R \Rightarrow Q$
Efficient analysis on very large models

Studying large models

Static analysis

Over- and Under-approximations

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

→ Directly checking $R$ is hard (exponential)

→ Rather check approximations $P$ and $Q$ so that: $P \Rightarrow R \Rightarrow Q$
Efficient analysis on very large models ◦ Studying large models ◦ Static analysis

Over- and Under-approximations

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

→ Directly checking $R$ is hard (exponential)
→ Rather check approximations $P$ and $Q$ so that: $P \Rightarrow R \Rightarrow Q$

Over-Approximation

Under-Approximation

Exact solutions

$R$

$\neg Q$
Efficient analysis on very large models ○ Studying large models ○ Static analysis

Over- and Under-approximations

[Paulevé et al., Mathematical Structures in Computer Science, 2012]

→ Directly checking \( R \) is hard (exponential)
→ Rather check approximations \( P \) and \( Q \) so that: \( P \Rightarrow R \Rightarrow Q \)

Computing \( P \) or \( Q \) is much simpler (roughly polynomial)
→ Efficient for big models → Hundredths of seconds
Efficient analysis on very large models ○ Studying large models ○ Static analysis

Under-approximation

- Under-approximation

**Required process**: $d_2$

**Objective**: $d_0 \not\Rightarrow^* d_2$

○ **Solution to an objective**

Diagram:

- Node $d_2$ points to $b_0$.
- $d_0 \not\Rightarrow^* d_2$ points to $b_1$.
- $b_0 \not\Rightarrow^* b_0$ points to $a_1$.
- $b_1 \not\Rightarrow^* b_1$ points to $c_1$.

- $b_0 \not\Rightarrow^* b_0$ points to $a_1$.
- $b_1 \not\Rightarrow^* b_1$ points to $c_1$.

Maxime FOLSCHETTE

9/16

MOVEP’14 — 2014/07/09
Efficient analysis on very large models

- Studying large models
- Static analysis

Under-approximation

**Sufficient condition:**

- no cycle
- each objective has a solution
Efficient analysis on very large models

Studying large models

Static analysis

Under-approximation

**Sufficient condition:**
- no cycle
- each objective has a solution

\[ P \text{ is true} \implies R \text{ is true} \]
Efficient analysis on very large models

- Studying large models

- Static analysis

Under-approximation

**Sufficient condition:**

- no cycle
- each objective has a solution
Efficient analysis on very large models

- Studying large models
- Static analysis

Under-approximation

Sufficient condition:
- no cycle
- each objective has a solution

\( P \) is false \( \Rightarrow \) Inconclusive
Over-approximation

Necessary condition:
Over-approximation

Necessary condition:
There exists a traversal with no cycle
- objective $\rightarrow$ follow **one** solution
- solution $\rightarrow$ follow **all** processes
- process $\rightarrow$ follow **all** objectives
Efficient analysis on very large models

Studying large models

Static analysis

Over-approximation

Necessary condition:
There exists a traversal with no cycle
- objective → follow one solution
- solution → follow all processes
- process → follow all objectives
Over-approximation

Necessary condition:
There exists a traversal with no cycle

- objective $\rightarrow$ follow one solution
- solution $\rightarrow$ follow all processes
- process $\rightarrow$ follow all objectives

$Q$ is false $\Rightarrow R$ is false
Efficient analysis on very large models ◦ Studying large models ◦ Static analysis

Over-approximation

Necessary condition:
There exists a traversal with no cycle
- objective → follow one solution
- solution → follow all processes
- process → follow all objectives
Over-approximation

Necessary condition:
There exists a traversal with no cycle
- objective $\rightarrow$ follow **one** solution
- solution $\rightarrow$ follow **all** processes
- process $\rightarrow$ follow **all** objectives

$R$ is true $\Rightarrow$ Inconclusive
Translation of PH models

[Folschette et al., Computational Methods in Systems Biology, 2012]
Translation of PH models

[Folschette et al., *Computational Methods in Systems Biology*, 2012]

Process Hitting
Efficient but recent

Boolean Networks
Widespread & readable
Efficient analysis on very large models ○ Studying large models ○ Personal contributions

Translation of PH models

[Folschette et al., Computational Methods in Systems Biology, 2012]

Process Hitting
Efficient but recent

Boolean Networks
Widespread & readable
Efficient analysis on very large models ◦ Studying large models ◦ Personal contributions

Translation of PH models

[Folschette et al., Computational Methods in Systems Biology, 2012]

Process Hitting
Efficient but recent

Boolean Networks
Widespread & readable
Enrichment of PH semantics

[Folschette et al., CS2Bio’13, 2013]
Efficient analysis on very large models

Studying large models

Personal contributions

Enrichment of PH semantics

[Folschette et al., CS2Bio'13, 2013]

---

Process Hitting
Accurate behaviour

Boolean Networks
Accurate behaviour
Efficient analysis on very large models  

Applications

Gene therapies

**Modify** DNA to cure a disease

- Replace a mutated gene → remove a **harmful protein**
- Add a new gene → produce a **therapeutic protein**
Efficient analysis on very large models

Applications

Gene therapies

**Modify** DNA to cure a disease

- Replace a mutated gene → remove a **harmful protein**
- Add a new gene → produce a **therapeutic protein**
Gene therapies

**Modify** DNA to cure a disease

- Replace a mutated gene $\rightarrow$ remove a **harmful protein**
- Add a new gene $\rightarrow$ produce a **therapeutic protein**
Back to the Over-approximation

Necessary condition:
There exists a traversal with no cycle
- objective → follow one solution
- solution → follow all processes
- process → follow all objectives
Efficient analysis on very large models

Applications

Back to the Over-approximation

Necessary condition:
There exists a traversal with no cycle
- objective → follow one solution
- solution → follow all processes
- process → follow all objectives

\[ R \text{ is true} \Rightarrow \text{Inconclusive} \]
Back to the Over-approximation

**Necessary condition:**
There exists a traversal with no cycle
- objective → follow **one** solution
- solution → follow **all** processes
- process → follow **all** objectives

\[ R \text{ is true} \Rightarrow \text{Inconclusive} \]
Necessary condition:
There exists a traversal with no cycle
- objective → follow one solution
- solution → follow all processes
- process → follow all objectives
Efficient analysis on very large models

Applications

Back to the Over-approximation

Necessary condition:
There exists a traversal with no cycle
- **objective** → follow **one** solution
- **solution** → follow **all** processes
- **process** → follow **all** objectives

\[ Q \text{ is false} \implies R \text{ is false} \]
Summary & Conclusion

• What is Bio-informatics?
  → Qualitative modelling of gene regulations
  → Large models are hard to study (exponential)

• What do I do?
  → The Process Hitting modelling
  → Very efficient on large-scale models (polynomial)
  → My contribution: reach the expressivity of boolean networks

• What for?
  → Validating & utilizing biological models
  → Gene therapies
Efficient analysis on very large models

Bibliography


Thank you