Synthesis of Boolean Networks from Biological Dynamical Constraints using Answer-Set Programming

Stéphanie Chevalier
LRI, CNRS, U. Paris-Sud
U. Paris-Saclay, France
stephanie.chevalier@lri.fr

Christine Froidevaux
LRI, CNRS, U. Paris-Sud
U. Paris-Saclay, France
christine.froidevaux@lri.fr

Loïc Paulevé
LaBRI, CNRS, U. Bordeaux
Bordeaux INP, France
loic.pauleve@labri.fr

Andrei Zinovyev
Institut Curie, INSERM
U. PSL, Mines ParisTech, France
andrei.zinovyev@curie.fr

Conception automatique de modèles
(réseaux Booléens)
à partir de connaissances sur le système
(son domaine et sa dynamique)

Problématique: **synthétiser logiquement une fonction**
à partir de contraintes sur ses propriétés

Motivation: **modèle aisément compréhensible**
de **dynamiques complexes**
avec **des milliers d'entités**

Conception automatique de modèles
(réseaux Booléens)
à partir de connaissances sur le système
(son domaine et sa dynamique)

Intérêt:

profil d'expression de gènes + **réseau d'interactions issus des connaissances** (littérature / bases de données)

= contextualisation des connaissances → **modèles**

Boolean network *(BN)*: *a discrete dynamical system*

A **BN of dimension *n***

is a function $f = (f_1, \ldots, f_n)$ with

$\forall i \in \{1, \ldots, n\},\ f_i : \{0, 1\}^n \rightarrow \{0, 1\}$

A **configuration** is a vector $x \in \{0, 1\}^n$

example:
$$
\begin{aligned}
f_1(x) &:= \neg x_2 \\
f_2(x) &:= \neg x_1 \\
f_3(x) &:= \neg x_1 \wedge x_2
\end{aligned}
$$

Boolean network (*BN*): *a discrete dynamical system*

A BN of dimension *n*

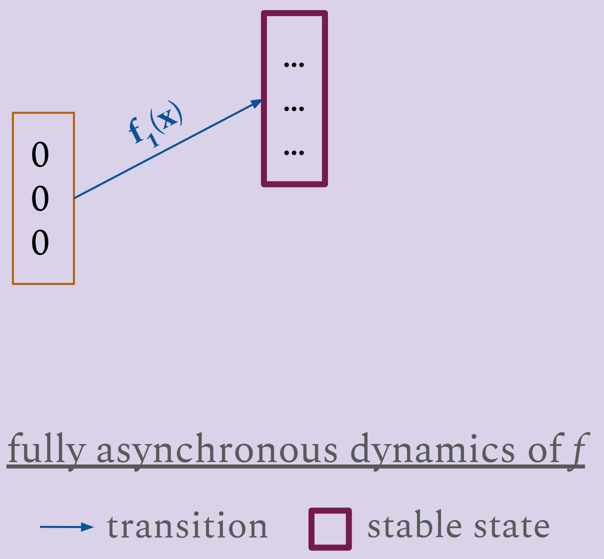
is a function $f = (f_1, \dots, f_n)$ with

$\forall i \in \{1, \dots, n\},\ f_i : \{0, 1\}^n \to \{0, 1\}$

A **configuration** is a vector $x \in \{0, 1\}^n$

example: $$\begin{aligned} f_1(x) &:= \neg x_2 \\ f_2(x) &:= \neg x_1 \\ f_3(x) &:= \neg x_1 \wedge x_2 \end{aligned}$$

Dynamics of a BN:



Boolean network *(BN)*: *a discrete dynamical system*

A BN of dimension *n*

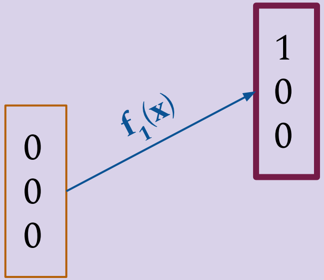
is a function $f = (f_1, \dots, f_n)$ with

$\forall i \in \{1, \dots, n\},\ f_i : \{0, 1\}^n \to \{0, 1\}$

A **configuration** is a vector $x \in \{0, 1\}^n$

example:
$$f_1(x) := \neg x_2$$
$$f_2(x) := \neg x_1$$
$$f_3(x) := \neg x_1 \wedge x_2$$

Dynamics of a BN:



fully asynchronous dynamics of *f*

transition — stable state

Boolean network *(BN)*: *a discrete dynamical system*

A **BN of dimension *n***

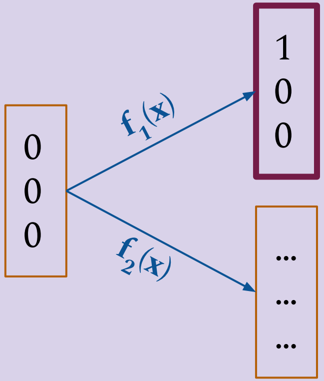
is a function $f = (f_1, \dots, f_n)$ with

$\forall i \in \{1, \dots, n\},\ f_i : \{0,1\}^n \rightarrow \{0,1\}$

A **configuration** is a vector $x \in \{0,1\}^n$

example:
$$\begin{aligned} f_1(x) &:= \neg x_2 \\ f_2(x) &:= \neg x_1 \\ f_3(x) &:= \neg x_1 \wedge x_2 \end{aligned}$$

Dynamics of a BN:



fully asynchronous dynamics of *f*

→ transition   □ stable state

Boolean network *(BN)*: *a discrete dynamical system*

A BN of dimension *n*

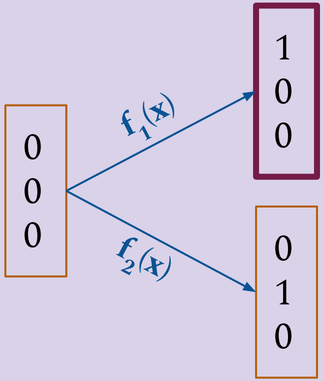
is a function $f = (f_1, \dots, f_n)$ with

$\forall i \in \{1, \dots, n\},\ f_i : \{0, 1\}^n \rightarrow \{0, 1\}$

A **configuration** is a vector $x \in \{0, 1\}^n$

example:
$$
\begin{aligned}
f_1(x) &:= \neg x_2 \\
f_2(x) &:= \neg x_1 \\
f_3(x) &:= \neg x_1 \wedge x_2
\end{aligned}
$$

Dynamics of a BN:



fully asynchronous dynamics of *f*

transition stable state

Boolean network *(BN)*: *a discrete dynamical system*

A BN of dimension *n*

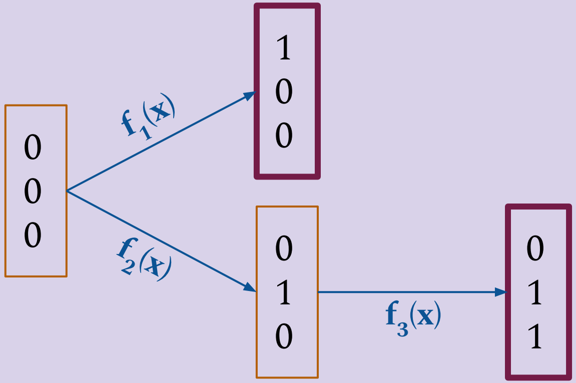
is a function $f = (f_1, \dots, f_n)$ with

$\forall i \in \{1, \dots, n\},\ f_i : \{0, 1\}^n \to \{0, 1\}$

A **configuration** is a vector $x \in \{0, 1\}^n$

example:
$$f_1(x) := \neg x_2$$
$$f_2(x) := \neg x_1$$
$$f_3(x) := \neg x_1 \wedge x_2$$

Dynamics of a BN:

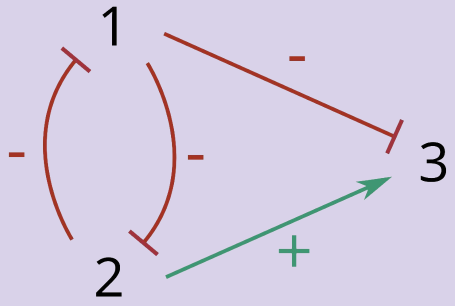


fully asynchronous dynamics of *f*

transition
stable state

Boolean network *(BN)*: *a discrete dynamical system*

Domain of BNs:



Influence graph *(IG)*: specifies the domain of compatible BNs

activating
inhibiting

Règles possible pour gène 3 :

$f_3(x) = 0$

$f_3(x) = 1$

$f_3(x) = \neg x_1$

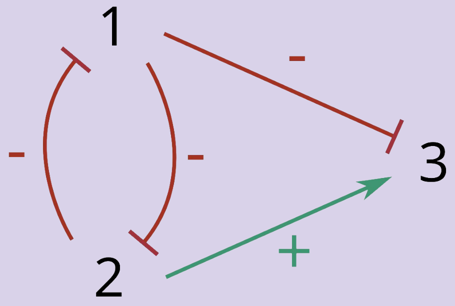
$f_3(x) = x_2$

$f_3(x) = \neg x_1 \wedge x_2$

$f_3(x) = \neg x_1 \vee x_2$

Boolean network *(BN)*: *a discrete dynamical system*

Domain of BNs:



<u>Influence graph *(IG)*:</u> specifies the domain of compatible BNs

activating
inhibiting

BN compatible with an IG:
uses at most influences in the IG

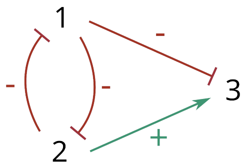
example:
2 BNs compatible with the same IG

BN f :	BN g :
$f_1(x) = \neg x_2$	$g_1(x) = 1$
$f_2(x) = \neg x_1$	$g_2(x) = \neg x_1$
$f_3(x) = \neg x_1 \wedge x_2$	$g_3(x) = \neg x_1 \vee x_2$

dynamics(f) ≠ dynamics(g)

Synthesis problem: find $f : \{0, 1\}^n \to \{0, 1\}^n$

Given a domain

$G =$ 

find BN f compatible with G

Combinatorial problem:

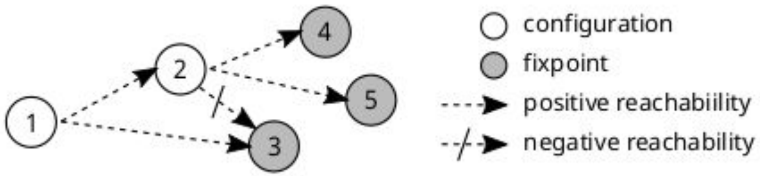
indegree	*# monotonic Boolean functions*
0	2
2	6
4	168
6	7,828,354
8	56,130,437,228,687,557,907,788

Given dynamics

of *configurations compatible* with *partial observations* of the system

obs. n°2 → *compatible conf.*

obs. n°2		compatible conf.	
$x_1 = 1$, $x_2 = 0$	→	$x_1 = 1$, $x_2 = 0$, $x_3 = 0$	$x_1 = 1$, $x_2 = 0$, $x_3 = 1$



Answer-Set Programming (ASP)

brief overview of ASP syntax

A Logic Program in ASP is a set of logical rules of the form:

$$a_0 \leftarrow a_1, \ \ldots, \ a_n, \ \text{not} \ a_{n+1}, \ \ldots, \ \text{not} \ a_{n+k}.$$

with integrity constraints as:

$$\leftarrow a_1, \ \ldots, \ a_n, \ \text{not} \ a_{n+1}, \ \ldots, \ \text{not} \ a_{n+k}.$$

Suitable for solving combinatorial satisfaction problem

Computes **<u>stable models</u>: minimal sets of a_i satisfying the rules**

[Gelfond and Lifschitz, 1988]

Contribution: ASP encoding

Boolean function: expressed in propositional logic under Disjunctive Normal Form

encoded as clause(N,C,L,S) specifying that:
atom L with sign S (-1, 1) is included in the Cth clause of f_N

example: $f_a(x) = \boxed{x_c} \lor \boxed{(\neg x_a \land x_b)}$

is encoded: clause(a,1,c,1).

Contribution: ASP encoding

Boolean function: expressed in propositional logic under Disjunctive Normal Form

encoded as `clause(N,C,L,S)` specifying that:
atom `L` with sign `S` (-1, 1) is included in the `C`th clause of f_N

example: $f_a(x) = \boxed{x_c} \vee \boxed{(\neg x_a \wedge x_b)}$

is encoded:
```
clause(a,1,c,1).
clause(a,2,a,-1).
clause(a,2,b,1).
```

Contribution: ASP encoding

Boolean function: expressed in propositional logic under Disjunctive Normal Form

encoded as clause(N,C,L,S) specifying that:
atom L with sign S (-1, 1) is included in the C[th] clause of f_N

example: $f_a(x) = \boxed{x_c} \vee \boxed{(\neg x_a \wedge x_b)}$

is encoded:
```
clause(a,1,c,1).
clause(a,2,a,-1).
clause(a,2,b,1).
```

Canonicity of *f* : 2 solutions = 2 non-equivalent BNs

enforced by a total ordering between the clauses

Dynamical constraints *(suitable for addressed bio. processes)* **:**

positive & negative reachability, several stable properties

Complexity

with:

- n #nodes
- d #variables
- k the **fixed upper bound** on #DNF clauses per local function (*the max. being* $\binom{d}{\lfloor d/2 \rfloor}$)

Linear:

- **BN encoding without canonicity** (O(ndk) predicates and rules)
- **Pos. reachability** and **stable properties** (O(nk) predicates and O(ndk) rules)

Quadratic:

- **BN encoding with canonicity** ($O(nd^2k^2)$ predicates and $O(ndk^2)$ rules)
- **Neg. reachability** ($O(n^2k)$ predicates and $O(n^2dk)$ rules)

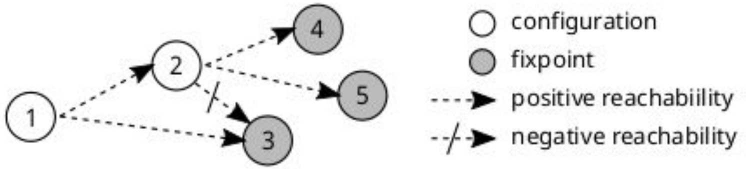
Synthesis with scales and types of knowledge not addressed before

Scalability results

Synthesis on random networks

Domains: random scale-free directed graphs, with different biases on the in-degree of nodes

Dynamics: a generic dynamical property of a two stages differentiation process

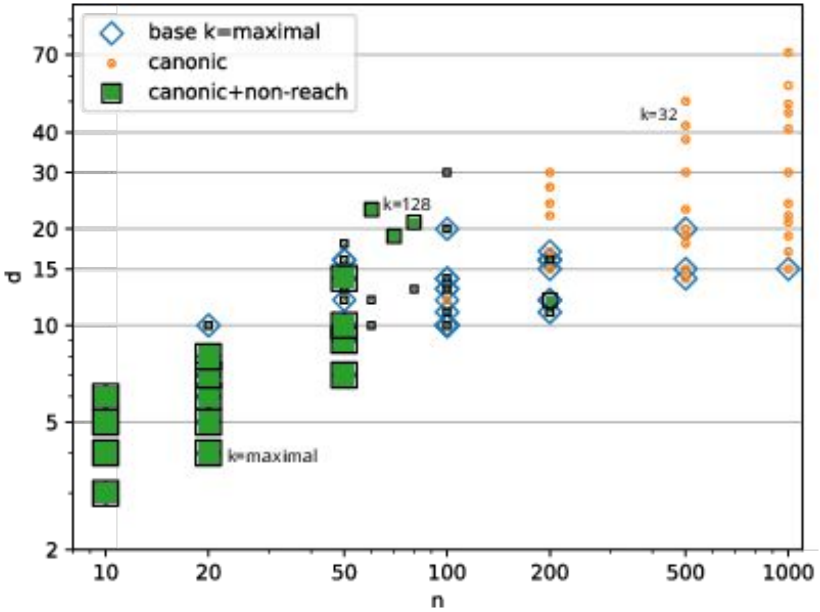


Scalability results

Exhaustive synthesis with all constraints:
➔ up to 50 nodes

Synthesis with a bounded number of clauses:
➔ up to 200 nodes

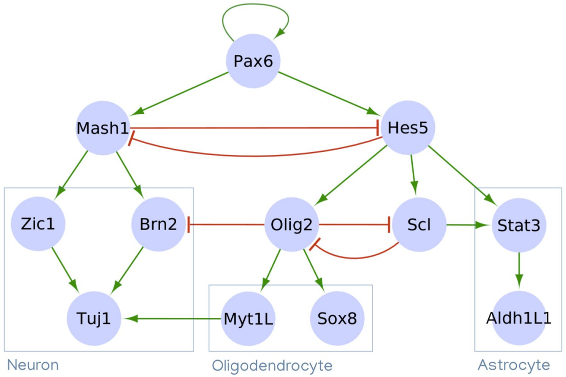
Synthesis with approximate neg. reachability:
➔ up to 1000 nodes

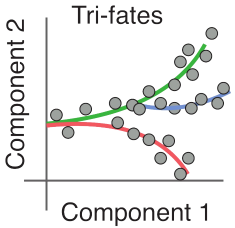


Successfully solved instances within 2h of CPU time (2.5Ghz)

Biological application

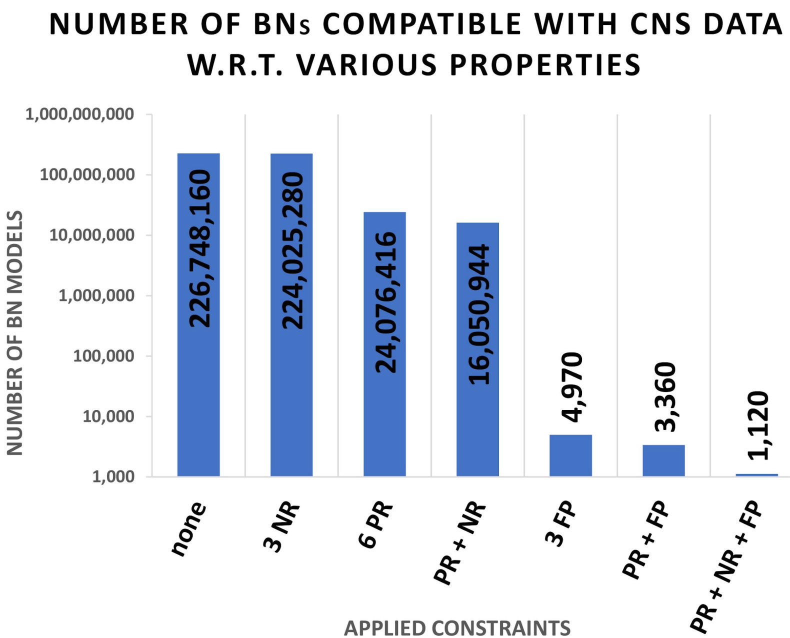
central nervous system development





6 pos. reach (PR)
3 neg. reach (NR)
3 fixpoint (FP)

Impact of the constraints:



Conclusion

Contribution:

Boolean network synthesis method in ASP

Features w.r.t. the state of the art:

- new constraints (negative reachability)
- mix reachability and stable properties
- scalability

Future work:

- Encoding of 2QBF constraints to check universal properties
- Application on blood single-cell differentiation data