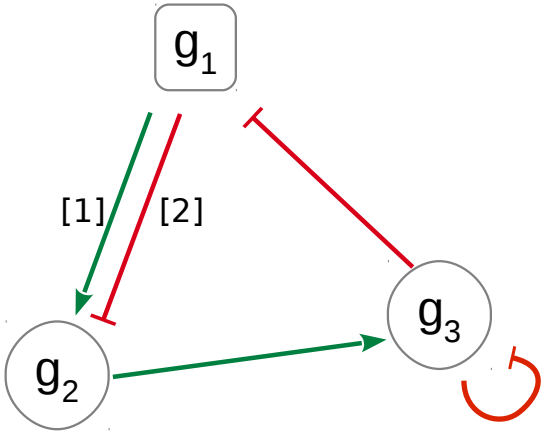
CoLoMoTo, SBML qual and LogicalModel: building bridges for qualitative models

Aurélien Naldi

Logical modeling



- Nodes: genes, proteins...

  Activity level (Boolean, MV)

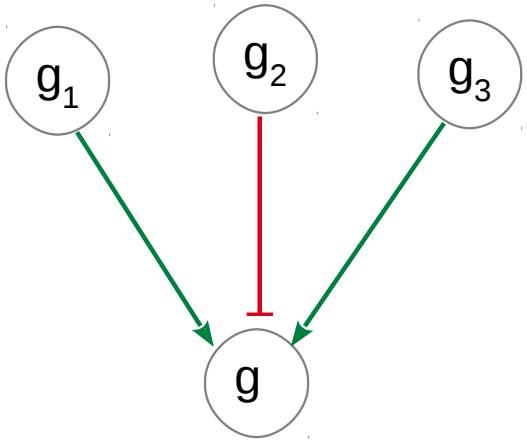
- Edges: interactions

  Threshold for MV source

- Logical rules

  next(G3) = G2 and not G3

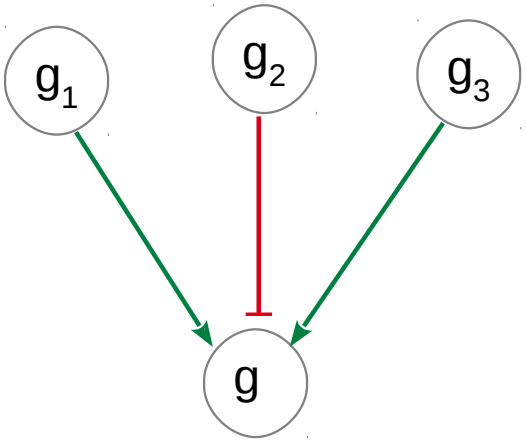
Discrete modelling methods



$(g_1 \mid g_3) \,\&\, !\, g_2$

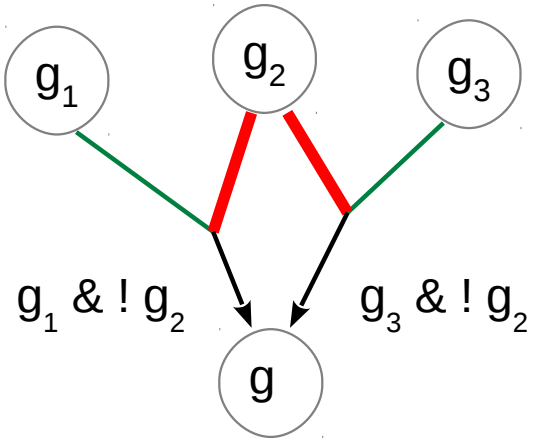
g1	g2	g3	g
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Discrete modelling methods

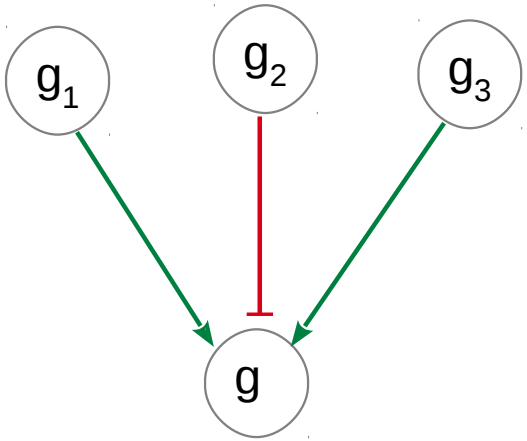


$(g_1 \mid g_3) \,\&\, !\, g_2$

g1	g2	g3	g
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

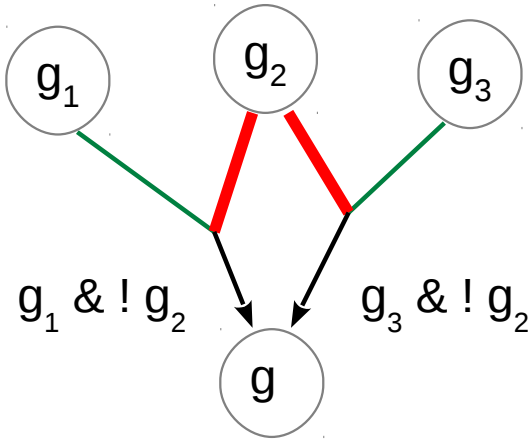


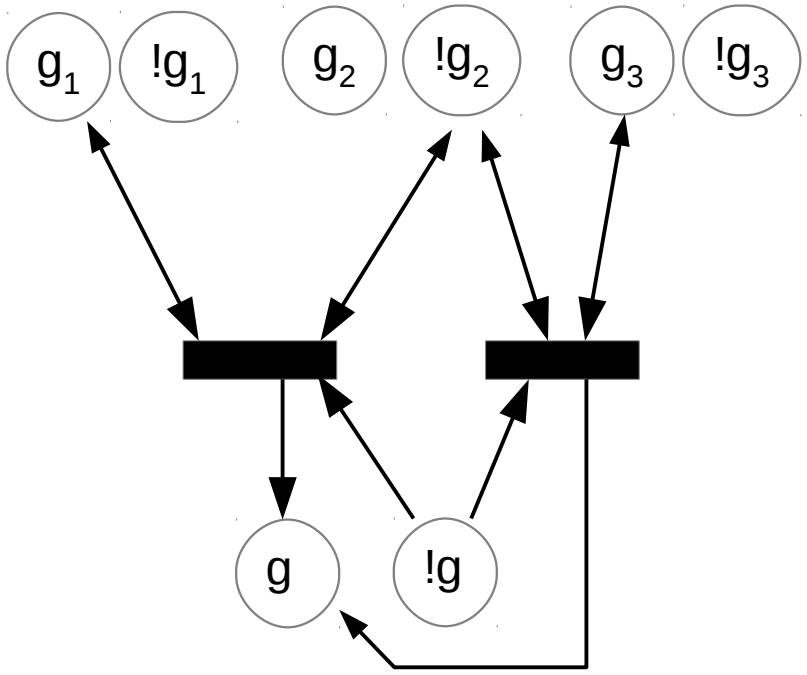
Discrete modelling methods



$(g_1 \mid g_3) \& ! g_2$

g1	g2	g3	g
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

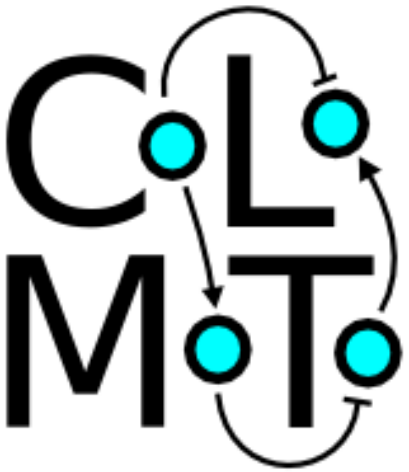


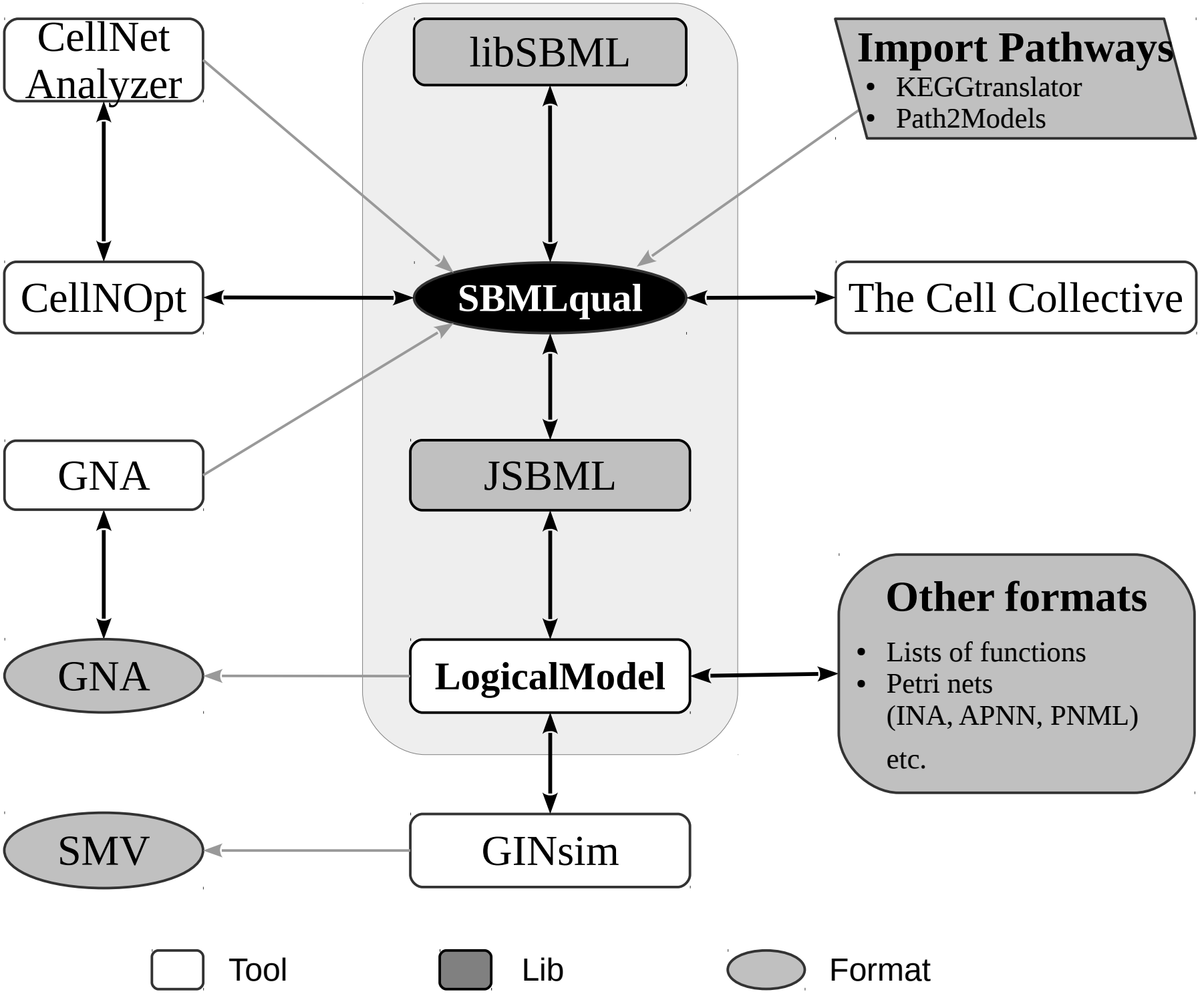


Petri net

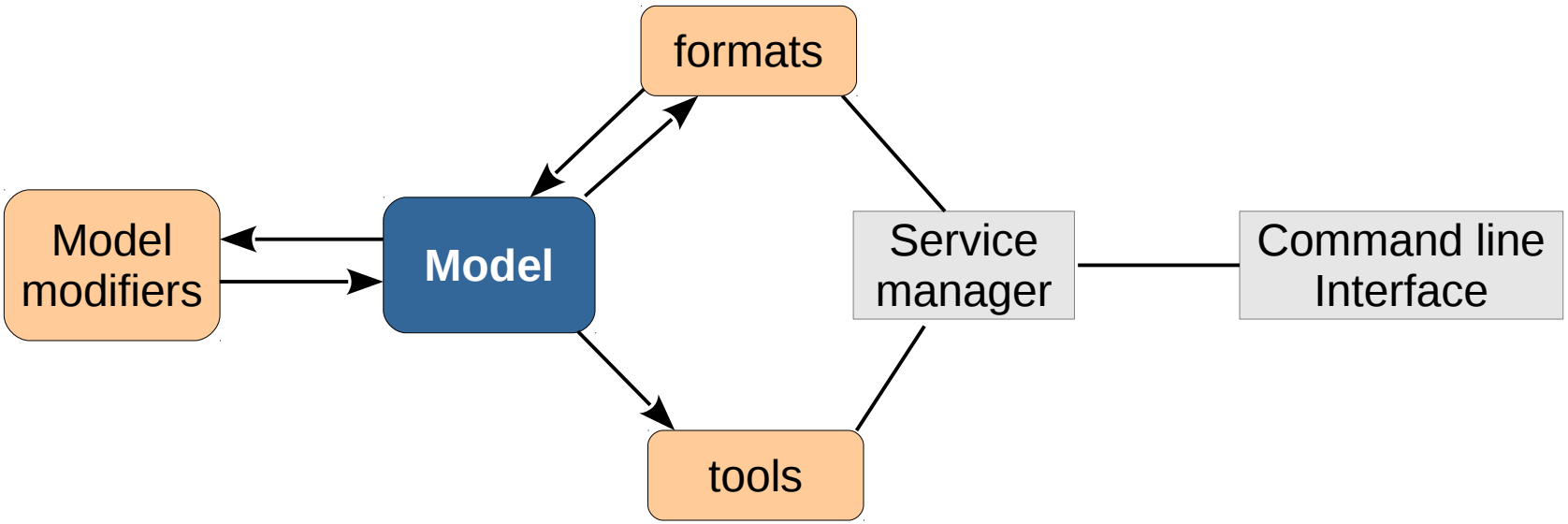
CoLoMoTo: exchanging models

- SBML qual extension: common exchange format
- Conversion tool
  - Read/write SBML qual and a few simple formats
  - Write several other formats
  - Can be extended
- Existing tools
  - Write SBML qual




CellNet Analyzer
libSBML
Import Pathways
• KEGGtranslator
• Path2Models
CellNOpt
SBMLqual
The Cell Collective
GNA
JSBML
Other formats
• Lists of functions
• Petri nets (INA, APNN, PNML)
etc.
GNA
LogicalModel
SMV
GINsim
Tool
Lib
Format

Organisation



Implemented

Model

Core and extra components
Multivalued Decision Diagrams

Model modifiers

Perturbations

Reduction
Output stripping

Formats

B <> boolsim
B <> Raw functions

M <> SBML qual

M > GINML
M > GNA
M > Petri net

M < Truth table

Tools

Stable state search

How to use?

As command-line tool

java -jar LogicalModel.jar model.sbml model.pnt -r stable model.sbml

- Search stable states
- Format conversion

As (Java) library

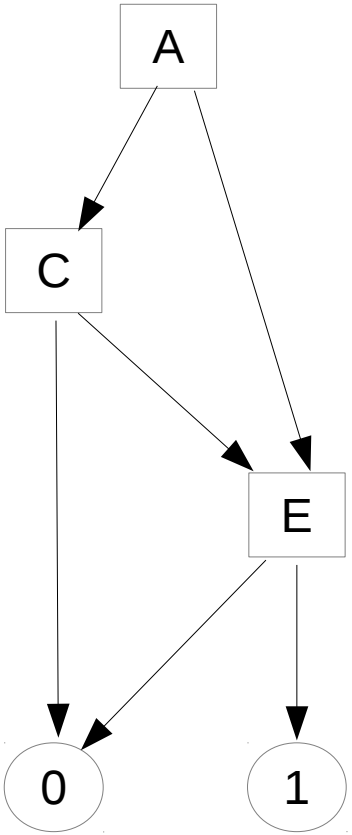
- GINsim (general qualitative modelling tool)
  - Data structure for simulation
  - Import/export backend (adds SBML layout markup)
  - Some tools backend (stable search, reduction)
- EpiLog (specialised for networks of epithelial cells)

Adding a format: minimal boilerplate

```
@ProviderFor(LogicalModelFormat.class)
public class YourFormat extends AbstractFormat {

	public YourFormat() {
		super("extension", "your description", true);
	}

	@Override
	public void export(LogicalModel model, OutputStream out) throws IOException {
		new YourFormatWriter(model).export(out);
	}

	@Override
	public LogicalModel importFile(File f) throws IOException {
		Return new YourFormatImporter.import(f);
	}
}
```

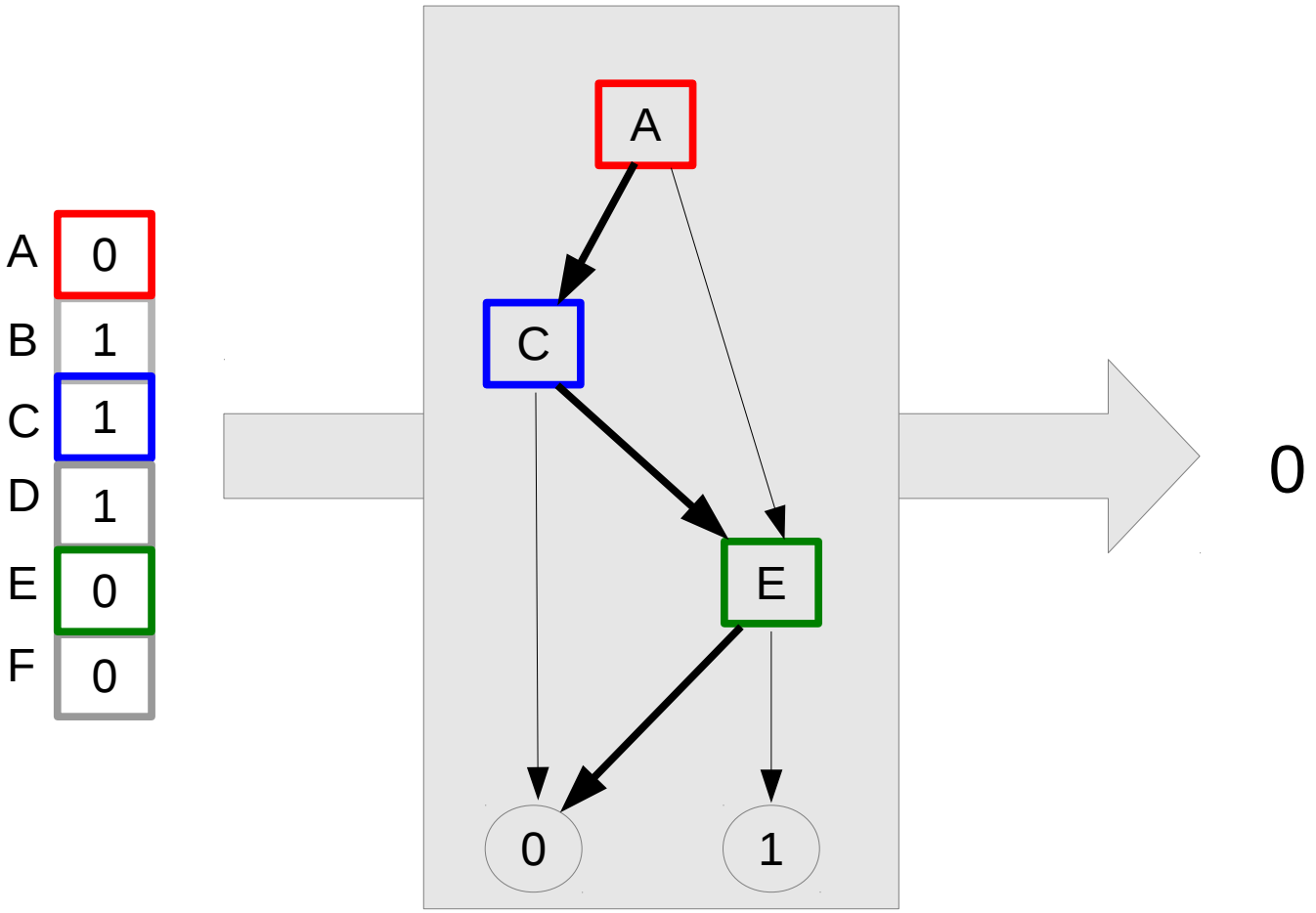
Decision Diagrams (MDD)

(A or C) and E



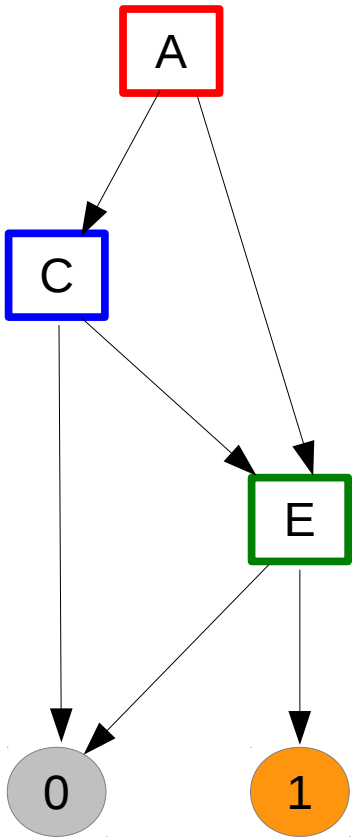
- A,B,C,D,E,F: ordered
- Fast (at most 6 tests, here 3)
- Predictable
- Picking order is hard

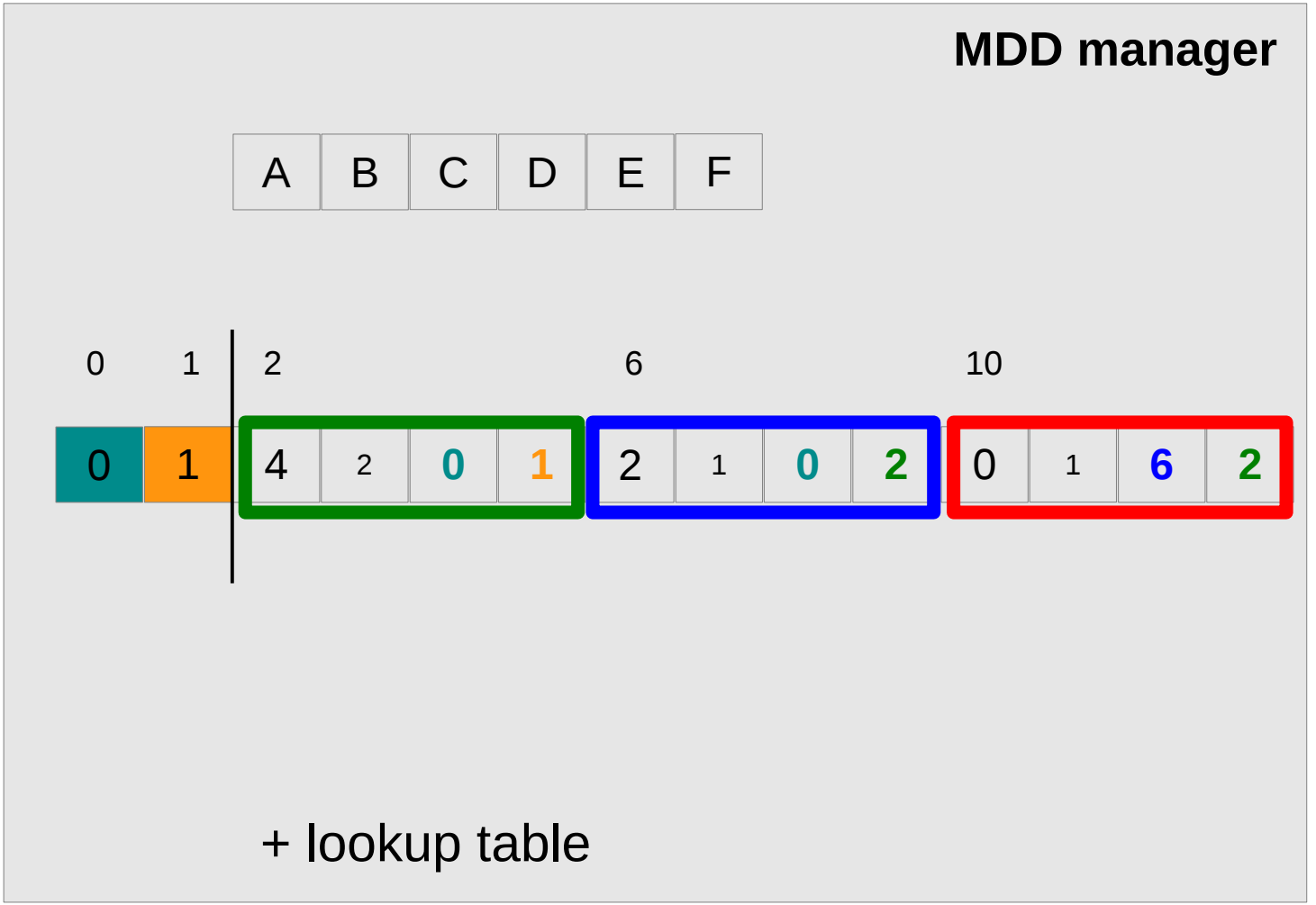
MDD query



Decision Diagrams (MDD)

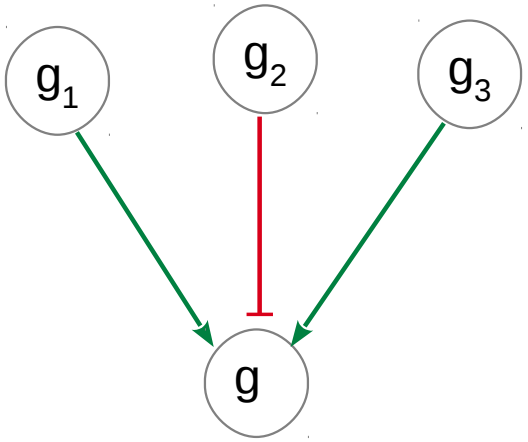
(A or C) and E





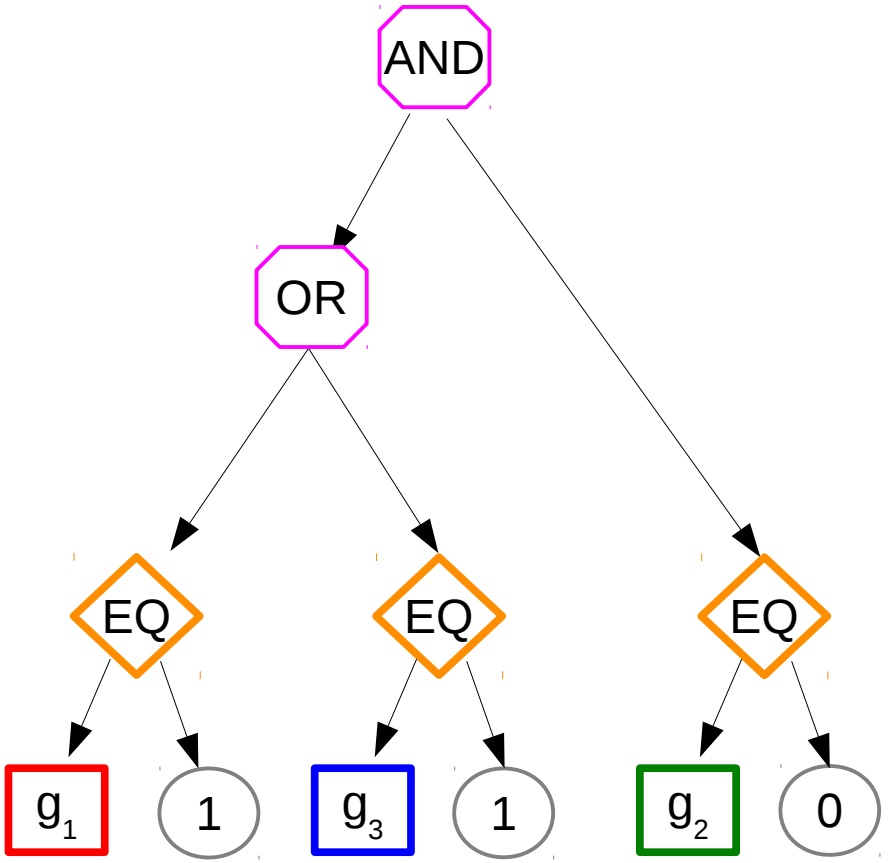
Multi-valued: some variables have more than two children

Functions in SBML-qual (and other)

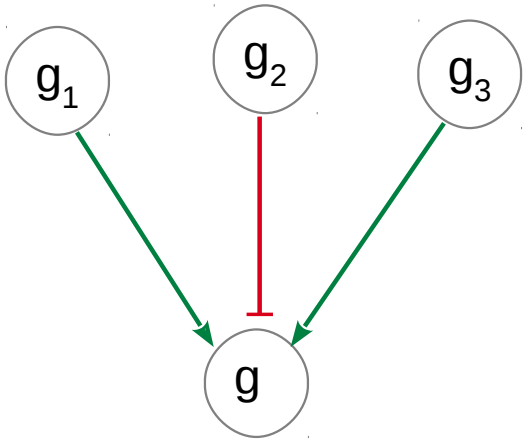


$(g_1 \mid g_3) \,\&\, !\, g_2$

$(g_1{=}1 \mid g_3{=}1) \,\&\, g_2{=}0$

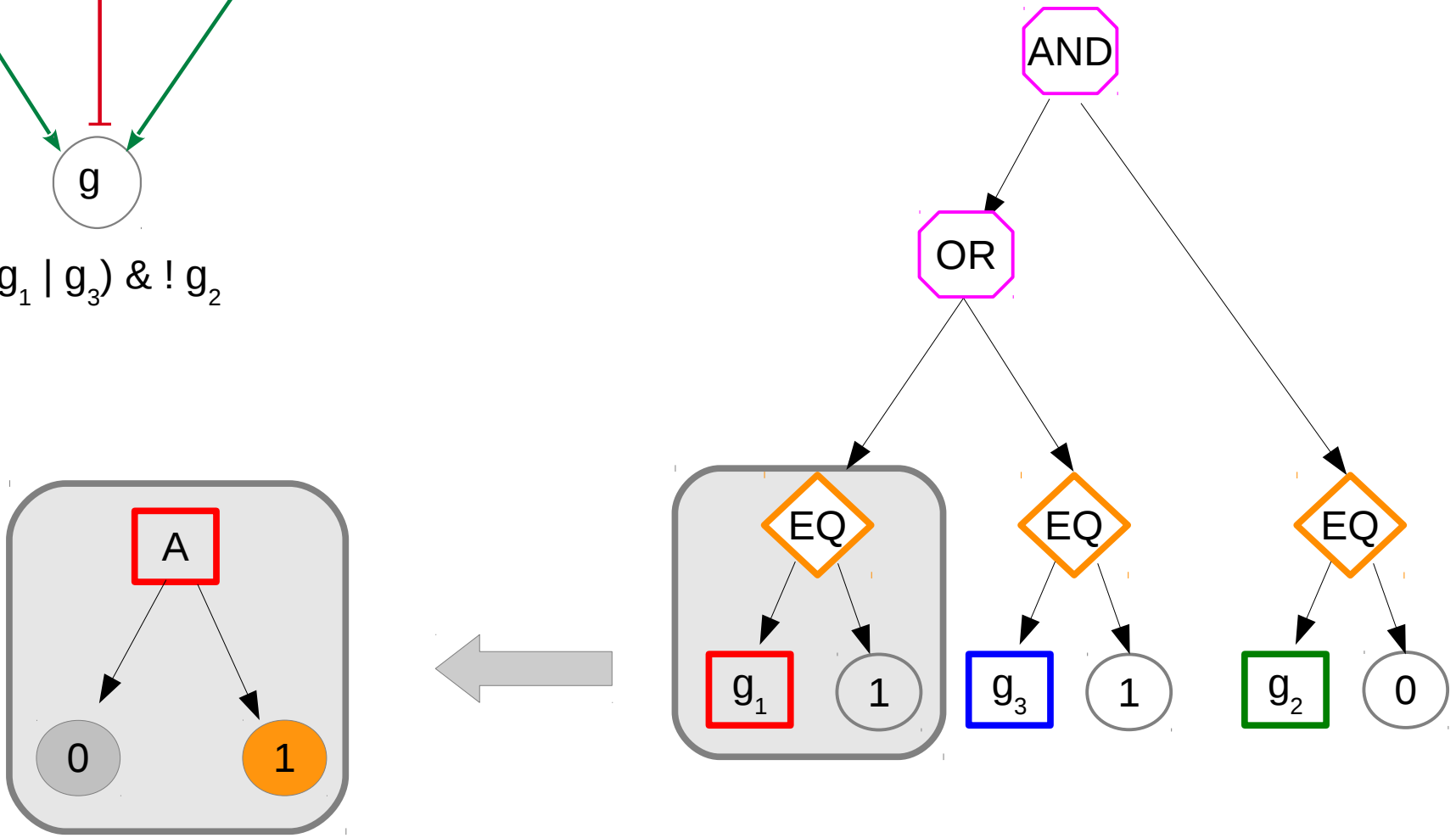


Functions in SBML-qual (and other)



$(g_1 \mid g_3) \;\&\; !\, g_2$

$(g_1{=}1 \mid g_3{=}1) \;\&\; g_2{=}0$



MDD manipulation for “OR”, “AND” operators

Helpers

Little MDDs knowledge required

- Extract list of regulators from function
- Iterate paths in the MDD
- Construct a function?

Iterate paths in a MDD

```
// input: LogicalModel model
MDDManager ddmanager = model.getMDDManager();
PathSearcher searcher = new PathSearcher(ddmanager);

for (int function: model.getLogicalFunctions()) {

   int[] path = searcher.setNode(functions[idx]);
   for (int leaf: searcher) {
      // leaf gives the value of the function
      // path gives the value of the components
   }
}
```

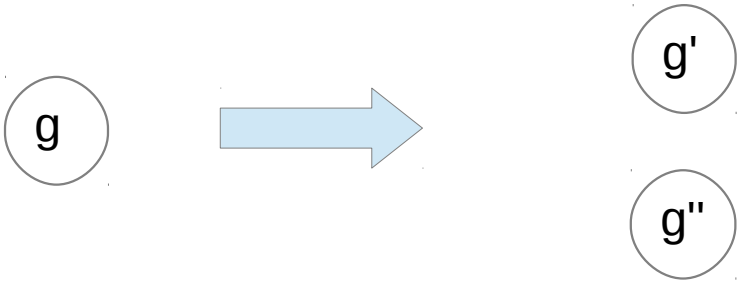
Future extensions

- Model modifiers
  - Transform multi-valued to Boolean
- Data
  - Documentation (SBML, GINML, JSON file)
  - Visual information (positions, styles?)
- Tools
  - state → successors: (a)synchronous, sequential

From multivalued to Boolean

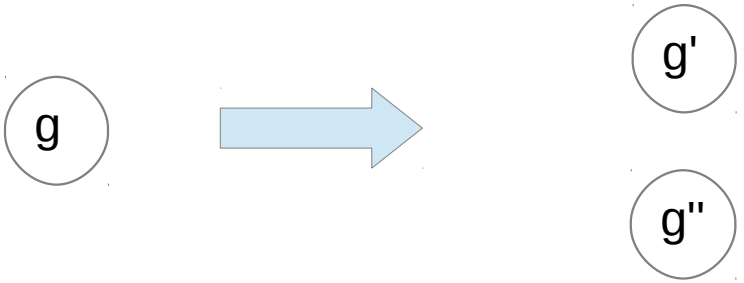
- Some tools support only Boolean models
- Multivalued models getting more common
- “Boleanize” a multivalued model when needed:
  - Multiply components
  - Rewrite functions

From multivalued to Boolean



g'	g''	g'''	g
0			0
1			1
0	1		2
1	1		3
0	0	1	4
1	0	1	5
0	1	1	6
1	1	1	7

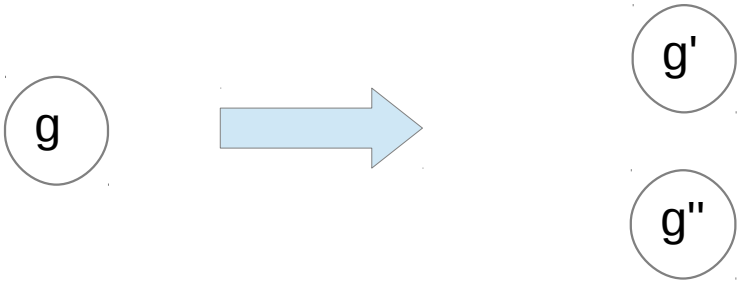
From multivalued to Boolean



g'	g''	g'''	g
0			0
1			1
0	1		2
1	1		3
0	0	1	4
1	0	1	5
0	1	1	6
1	1	1	7

g'	g''	g'''	g
0			0
1			1
1	1		2
0	1		3
0	1	1	4
1	1	1	5
1	0	1	6
0	0	1	7

From multivalued to Boolean



g'	g''	g'''	g
0			0
1			1
0	1		2
1	1		3
0	0	1	4
1	0	1	5
0	1	1	6
1	1	1	7

g'	g''	g'''	g
0			0
1			1
1	1		2
0	1		3
0	1	1	4
1	1	1	5
1	0	1	6
0	0	1	7

g'	g''	g'''	g''''	g
0				0
1				1
1	1			2
1	1	1		3
1	1	1	1	4

Acknowledgements

UNIL | Université de Lausanne

SIB Swiss Institute of Bioinformatics

Béatrice Desvergne group

Ioannis Xenarios
Julien Dorier

CoLoMoTo group

IGC
Claudine Chaouiya
Pedro Monteiro

IBENS
Denis Thieffry

EBI
Sarah Keating
Julio Saez Rodriguez
Nicolas Le Novère

Univ. Nebraska
Tomas Helikar

github.com/colomoto