Temporal parameters within π-calculus modeling of gene regulatory networks

Ph.D. supervised by Olivier Roux and Morgan Magnin

Loïc Paulevé

Moves team, IRCCyN, Nantes, FR

November 6, 2008

Context

- Studying dynamics of gene regulatory networks.
- We want models fitting to reality.
- Understand the different dynamics.
- Use of computer science technics.

Objective:

- Introduction of temporal parameters within π-calculus models.

Method:

- Parameters synthesis : where are they needed?
- Parameters value inference : what value they must have?

π-calculus modeling

- Concurrent processes algebra, Milner 89
- Processes communicate using channels.
- P_1 calls (?) on channel a, P_2 answers (!) on channel a.
- Operators : parallelization, replication, message passing, name restriction.
- Turing complete.
- Here we will focus on π-calculus programs having a restricted grammar and operators.

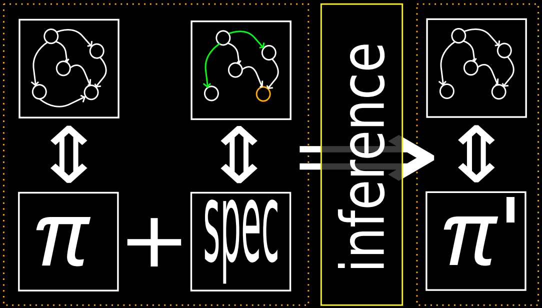
Stochastic π-calculus modeling

- Introduction of delays (τ) and channels rates, Priami 95
- For each channels and delays we may specify a use rate.
- Choice operator : +
- Race between the different channels/delays.

$$
\begin{array}{rcl}
 & & (P_1|P_2) \\
P_1 & := & ?a.P_2 + \\
 & & !b.P_1 + \\
 & & \tau_1 \\
P_2 & := & !a.P_1 + \\
 & & ?b + \\
 & & \tau_2.P_2
\end{array}
$$

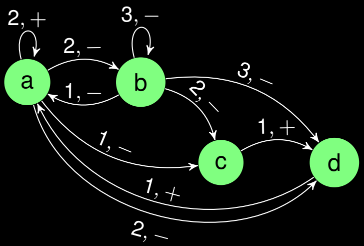
- Relations between rates have to be preserved.

Parameters inference



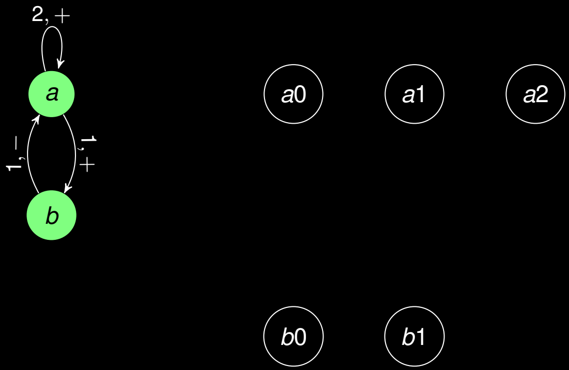
- Input: π-calculus program + set of properties and restrictions for its dynamics.
- Output: π-calculus program respecting specifications.
- Parameters are channel rates. For now, we will consider only two values: 0 (disabling) and 1 (activating)

General GRN dynamics



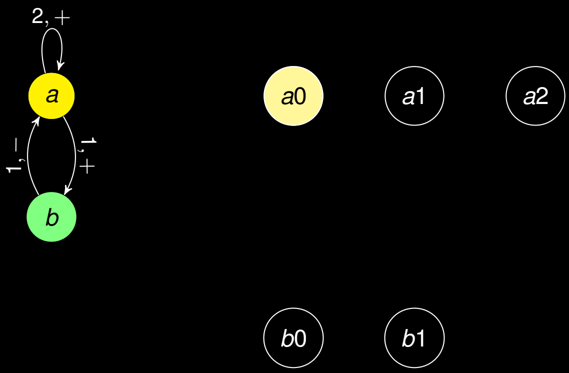
- Gene regulatory networks graph using thresholds.
- Asynchronous transitions: one state at once changes level.
- A gene may see its level increase iff at least one activator is present.
- A gene may see its level decrease iff at least one inhibitor is present.

π-calculus modeling of GRN



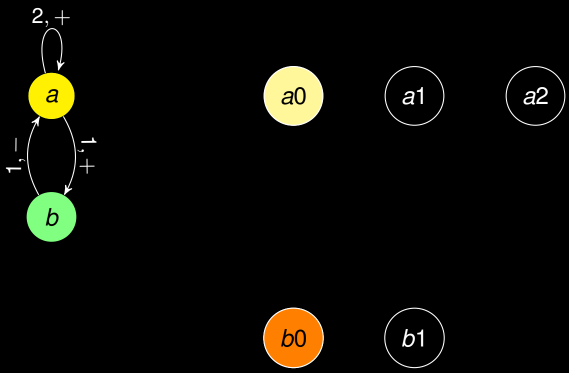
- One process per gene and per level.
- Answers to the call from its activators/inhibitors and goes to corresponding following/preceding process.

π-calculus modeling of GRN



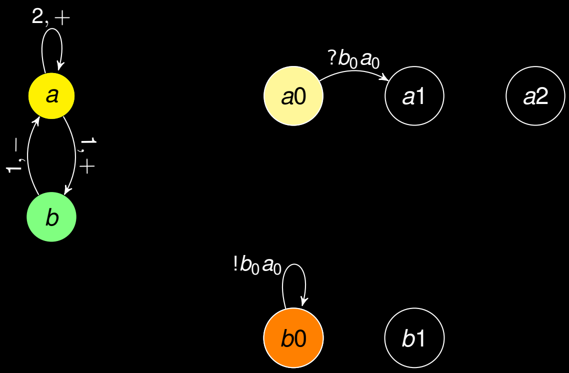
- One process per gene and per level.
- Answers to the call from its activators/inhibitors and goes to corresponding following/preceding process.

π-calculus modeling of GRN



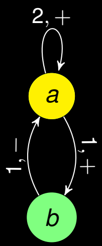
- One process per gene and per level.
- Answers to the call from its activators/inhibitors and goes to corresponding following/preceding process.

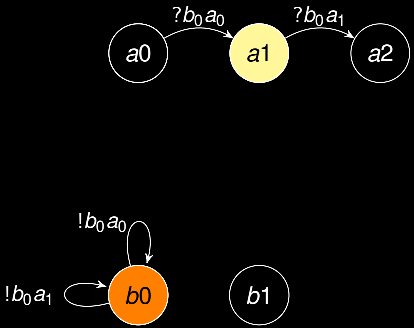
π-calculus modeling of GRN



- One process per gene and per level.
- Answers to the call from its activators/inhibitors and goes to corresponding following/preceding process.

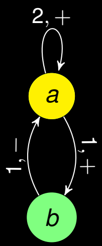
π-calculus modeling of GRN

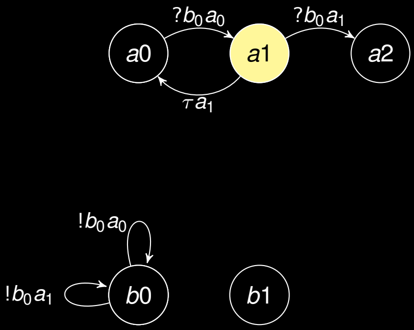




- One process per gene and per level.
- Answers to the call from its activators/inhibitors and goes to corresponding following/preceding process.

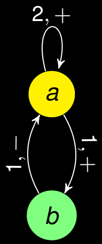
π-calculus modeling of GRN

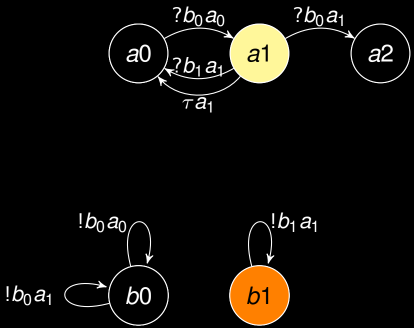




- One process per gene and per level.
- Answers to the call from its activators/inhibitors and goes to corresponding following/preceding process.

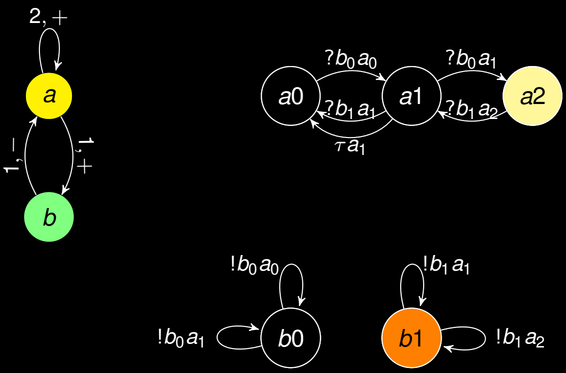
π-calculus modeling of GRN





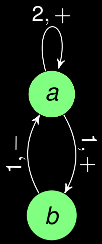
- One process per gene and per level.
- Answers to the call from its activators/inhibitors and goes to corresponding following/preceding process.

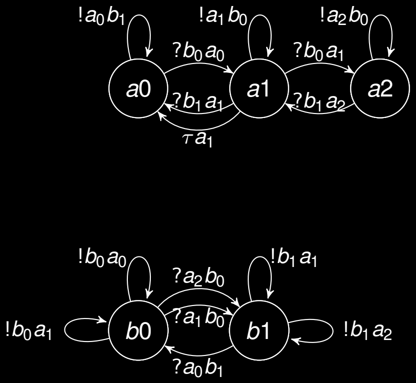
π-calculus modeling of GRN



- One process per gene and per level.
- Answers to the call from its activators/inhibitors and goes to corresponding following/preceding process.

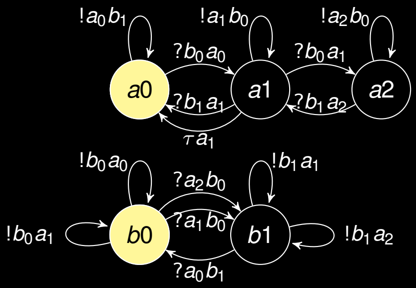
π-calculus modeling of GRN





- One process per gene and per level.
- Answers to the call from its activators/inhibitors and goes to corresponding following/preceding process.

State graph from the π-calculus model

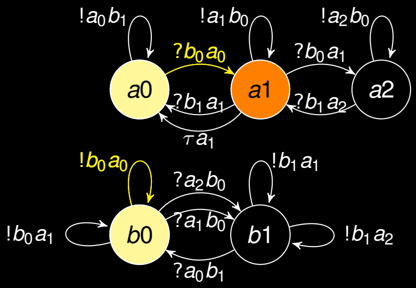


From a given state of the system :

- For each living process, we look for a possible reaction (existence of a call and answer on the same channel)
- Different reactions may result to the same state.

The resulting model is equivalent to the GRN (if all rates > 0).

State graph from the π-calculus model

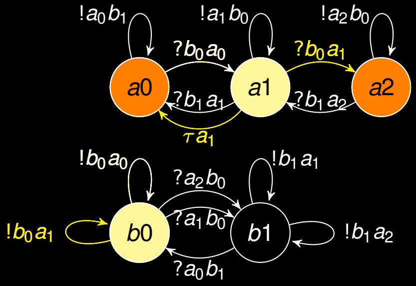


From a given state of the system :

- For each living process, we look for a possible reaction (existence of a call and answer on the same channel)
- Different reactions may result to the same state.

The resulting model is equivalent to the GRN (if all rates > 0).

State graph from the π-calculus model

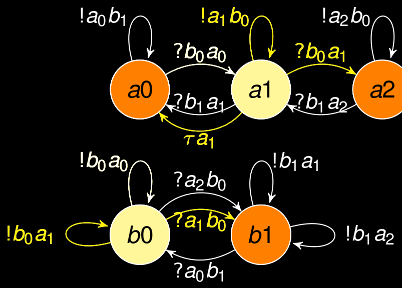


From a given state of the system :

- For each living process, we look for a possible reaction (existence of a call and answer on the same channel)
- Different reactions may result to the same state.

The resulting model is equivalent to the GRN (if all rates > 0).

State graph from the π-calculus model

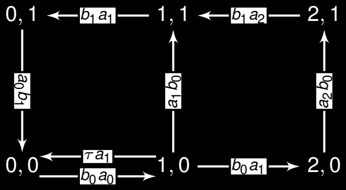


From a given state of the system :

- For each living process, we look for a possible reaction (existence of a call and answer on the same channel)
- Different reactions may result to the same state.

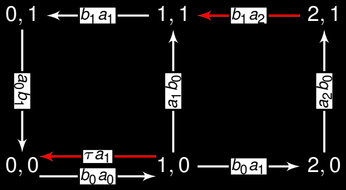
The resulting model is equivalent to the GRN (if all rates > 0).

Dynamic restrictions



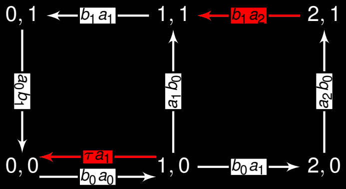
- Eg: stability: we remove transitions going out of a steady state/cycle.
- Within our model, it means disabling the corresponding channels.
- We want to preserve some given properties (reachability, existence of a transition, ...).

Dynamic restrictions



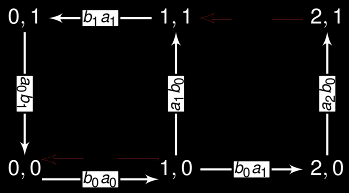
- Eg: stability: we remove transitions going out of a steady state/cycle.
- Within our model, it means disabling the corresponding channels.
- We want to preserve some given properties (reachability, existence of a transition, ...).

Dynamic restrictions



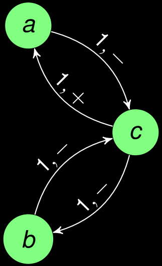
- Eg: stability: we remove transitions going out of a steady state/cycle.
- Within our model, it means disabling the corresponding channels.
- We want to preserve some given properties (reachability, existence of a transition, ...).

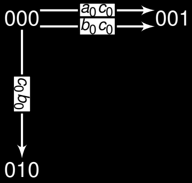
Dynamic restrictions



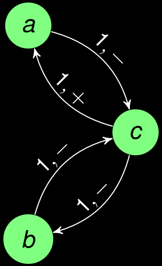
- Eg: stability: we remove transitions going out of a steady state/cycle.
- Within our model, it means disabling the corresponding channels.
- We want to preserve some given properties (reachability, existence of a transition, ...).

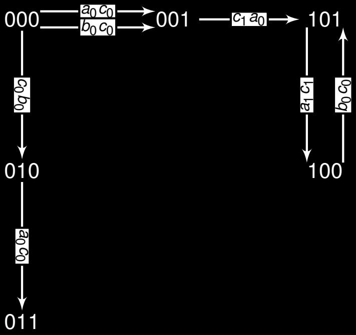
Second example



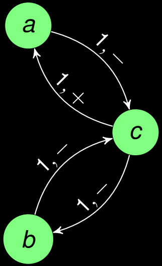


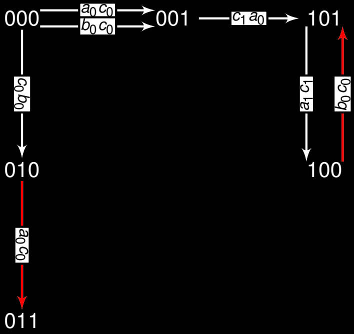
Second example



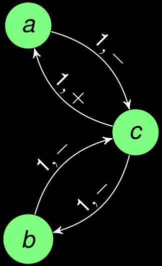


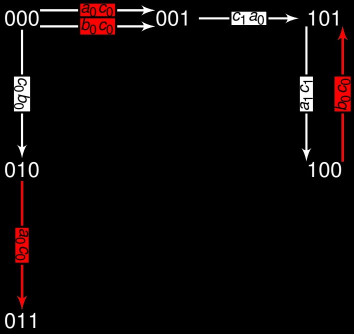
Second example



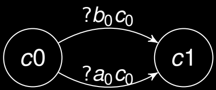


Second example



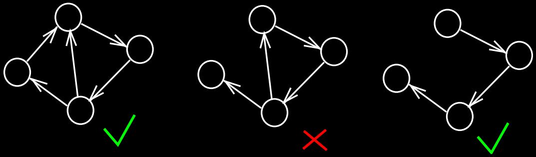


What's wrong?



- By disabling channels, we often remove more than one transition.
- We cannot apply constraints defined using "AND" logical operator.
- A process changes its level as soon as one and only one call is taken.
- When the model has to take a decision about its evolution, it lacks of information about other states.

Classes of dynamics



- Some dynamics are harder than others to reproduce,
- They sometimes require to know extra-information about the global system to take the right decision.
- Agregating information is costly (synchronizations, memory).
- Directly related to the complexity and expressive power of the underlying program.

Work in progress

Make a program satisfying a specification on its dynamics :

- Enrich the program structure by adding components aggregating information about the dynamical state (controllers).

- Starting with a generic model,
- using knowledge about the dynamics,
- infer required information about other components (relations).

- Search for a simple model respecting a specification (Occam's razor).

Perspectives

- Introduce temporal parameters where they are needed.
- Control the presence of a trajectory only by tuning some parameters of the model.
- Control trajectory frequency will be quite easy.
- Tools implementing the parameters synthesis and inference (SPiM extensions, etc.).

Questions ?