Logical Modeling with Time Delays

Heike Siebert Alexander Bockmayr

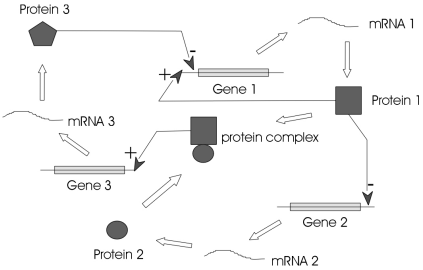
DFG-Research Center MATHEON,
Freie Universität Berlin

Toward Systems Biology
Grenoble
October 2007

Why logical modeling?

- lack of quantitative information on kinetic parameters and molecular concentrations
- biochemical reaction mechanisms underlying interactions not or incompletely known
- resulting systems of differential equations mostly not analytically solvable

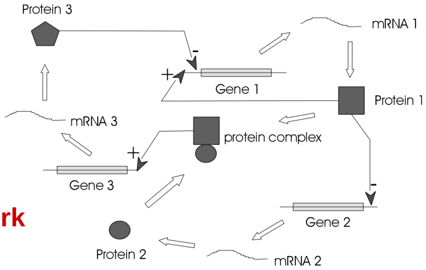
$\Rightarrow$ discrete modeling based on qualitative data



Why logical modeling?

- lack of quantitative information on kinetic parameters and molecular concentrations
- biochemical reaction mechanisms underlying interactions not or incompletely known
- resulting systems of differential equations mostly not analytically solvable

⇒ discrete modeling based on qualitative data



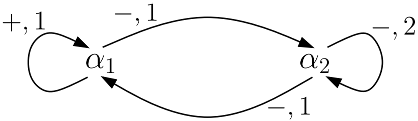
allow for the incorporation of temporal data concerning network processes

Thomas Formalism

[R. Thomas, 1973]

Structure: interaction graph

- discrete variables $\alpha_1, \ldots \alpha_n$
- expression levels $0, \ldots, p_j$ associated with each α_j
- labeled interactions

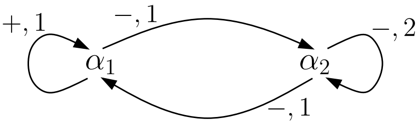


$\alpha_1 \in \{0, 1\},\ \alpha_2 \in \{0, 1, 2\}$

Thomas Formalism
[R. Thomas, 1973]

Structure: interaction graph

- discrete variables $\alpha_1, \dots \alpha_n$
- expression levels $0, \dots, p_j$ associated with each α_j
- labeled interactions



$\alpha_1 \in \{0,1\},\ \alpha_2 \in \{0,1,2\}$

Dynamics: state space and evolution

- state space $S := \{0,\dots,p_1\} \times \dots \times \{0,\dots,p_n\}$
- discrete function $f : S \to S$ determines behavior of the system

$s = (s_1, s_2), f(s) = (f_1(s), f_2(s))$

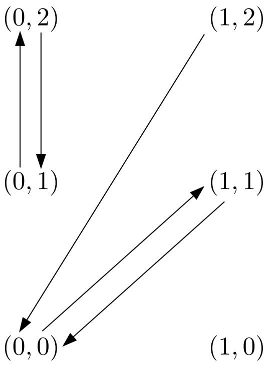
$$f_1(s) = \begin{cases} 1 & , \quad s_2 = 0 \\ 0 & , \quad \text{else} \end{cases}$$

$$f_2(s) = \begin{cases} 2 & , \quad s_1 = 0 \wedge s_2 \le 1 \\ 1 & , \quad s_1 = 0 \wedge s_2 = 2 \\ 0 & , \quad \text{else} \end{cases}$$

Thomas Formalism

Dynamics: state transition graph

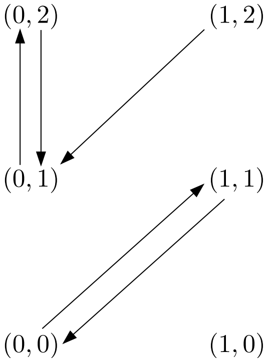
- vertex set S
- edges derived from parameter values



Thomas Formalism

Dynamics: state transition graph

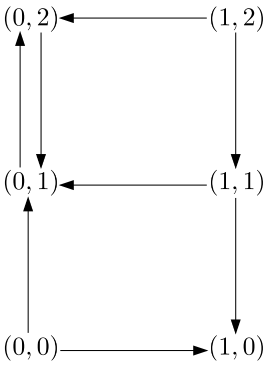
- vertex set *S*
- edges derived from parameter values
    - corresponding component values differ at most by 1



Thomas Formalism

Dynamics: state transition graph

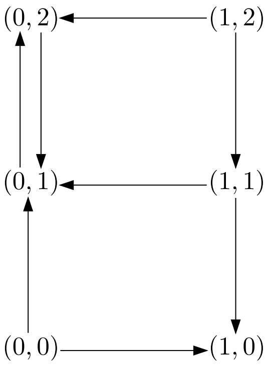
- vertex set S
- edges derived from parameter values
  - corresponding component values differ at most by 1
  - states differ from their successors in one component only
    asynchronous update: sole assumption about time delays



Thomas Formalism

Dynamics: state transition graph

- vertex set S
- edges derived from parameter values
  - corresponding component values differ at most by 1
  - states differ from their successors in one component only
    asynchronous update: sole assumption about time delays

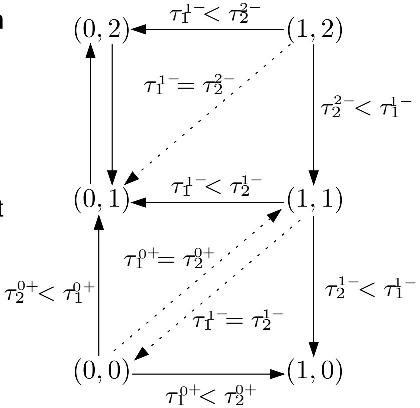


non-deterministic representation of the network dynamics

Considering Time Delays

Command to change for more than one component

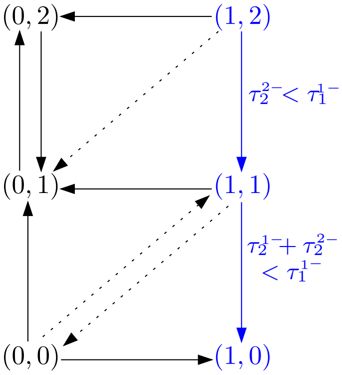
- compare time delays associated with different processes
  - distinguish between components
  - distinguish between production and decay processes
  - take expression levels into account
- allow for the possibility of time delay equality



Considering Time Delays

Command to change for more than one component

- compare time delays associated with different processes
  - distinguish between components
  - distinguish between production and decay processes
  - take expression levels into account
- allow for the possibility of time delay equality
- complexity of time constraints may increase with path length



Introducing Time

Timed Automata [R. Alur, D. Dill, 1994]

- clocks measure time, progress linear and synchronously
- clock constraints are formulated in the grammar

$$\varphi ::= c \leq q \mid c \geq q \mid c < q \mid c > q \mid \varphi_1 \wedge \varphi_2$$

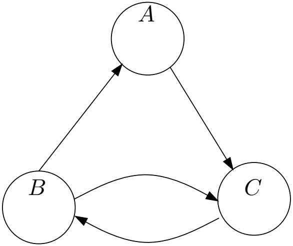
Introducing Time

Timed Automata [R. Alur, D. Dill, 1994]

- clocks measure time, progress linear and synchronously
- clock constraints are formulated in the grammar

$$\varphi ::= c \leq q \mid c \geq q \mid c < q \mid c > q \mid \varphi_1 \wedge \varphi_2$$

- timed automata may be visualized as digraphs where
  - vertices (locations) represent states
  - edges represent (discrete) state changes



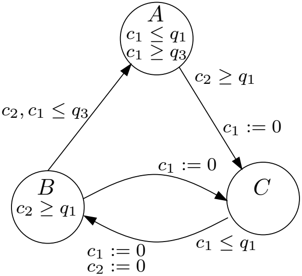
Introducing Time

Timed Automata [R. Alur, D. Dill, 1994]

- clocks measure time, progress linear and synchronously
- clock constraints are formulated in the grammar

$$\varphi ::= c \leq q \,|\, c \geq q \,|\, c < q \,|\, c > q \,|\, \varphi_1 \wedge \varphi_2$$

- timed automata may be visualized as digraphs where
  - vertices (locations) represent states
  - edges represent (discrete) state changes
  - time constraints may be posed on states and edges, clocks may be reset



Modus Operandi

1. Model each component incorporating information on
    - expression levels,
    - interactions,
    - parameter values,
    - time delays.

Modus Operandi

1. Model each component incorporating information on
   - expression levels,
   - interactions,
   - parameter values,
   - time delays.
2. Combine the components to a model supplying information on
   - the state space of the network,
   - state changes induced by the structure and parameter specification of the network,
   - constraints on time delays associated with state changes.

Modus Operandi

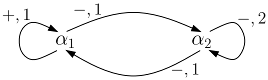
1. Model each component incorporating information on
   - expression levels,
   - interactions,
   - parameter values,
   - time delays.

2. Combine the components to a model supplying information on
   - the state space of the network,
   - state changes induced by the structure and parameter specification of the network,
   - constraints on time delays associated with state changes.

3. Evaluate the data inherent in the network model to obtain a representation of the dynamical behavior in agreement with all given constraints.

Modeling Each Component

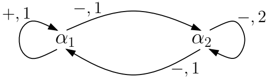
- one clock for each component



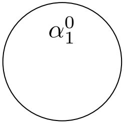
$f : S \to S$

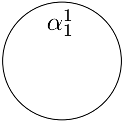
Modeling Each Component

- one clock for each component
- expression levels

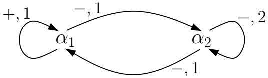


$f : S \to S$



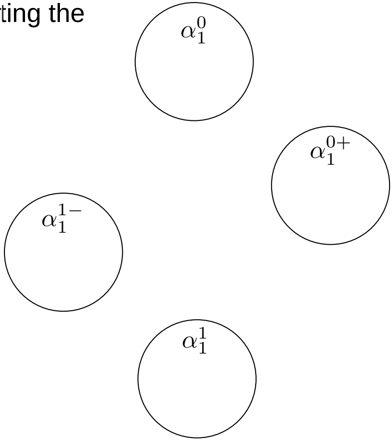


Modeling Each Component

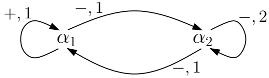


$f : S \rightarrow S$

- one clock for each component
- expression levels – distinction between stationary states and states representing the process of expression level change

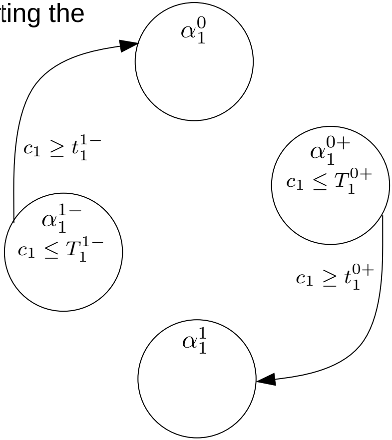


Modeling Each Component

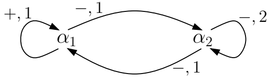


$f : S \to S$

- one clock for each component
- expression levels – distinction between stationary states and states representing the process of expression level change
- maximal and minimal time delays associated with expression level change
- location changes due to elapse of time

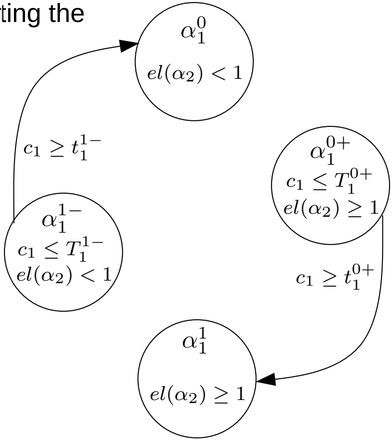


Modeling Each Component

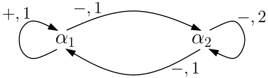


$f : S \to S$

- one clock for each component
- expression levels – distinction between stationary states and states representing the process of expression level change
- maximal and minimal time delays associated with expression level change
- location changes due to elapse of time
- corresponding network interactions and parameters (“switch conditions”),



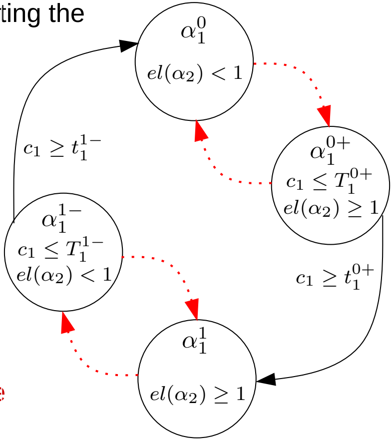
Modeling Each Component



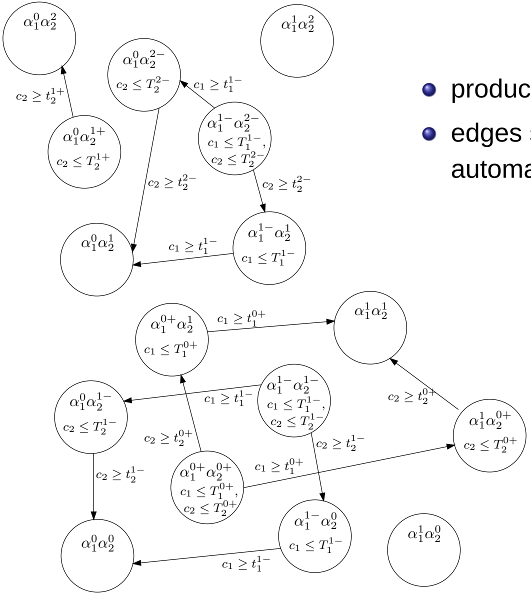
$f : S \to S$

- one clock for each component
- expression levels – distinction between stationary states and states representing the process of expression level change
- maximal and minimal time delays associated with expression level change
- location changes due to elapse of time
- corresponding network interactions and parameters (“switch conditions”),

  induced location changes can only be evaluated in the network context

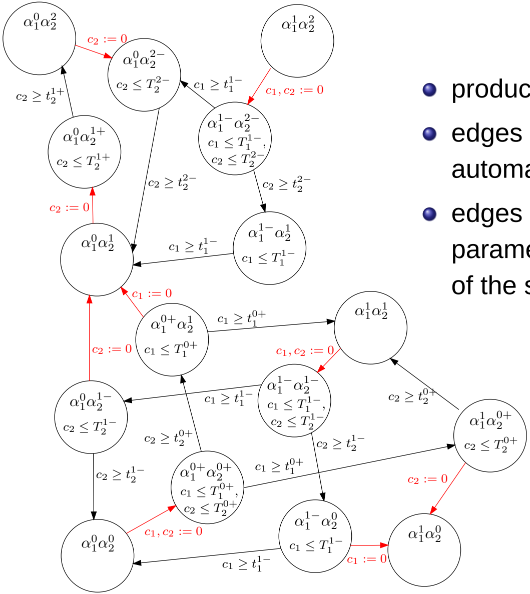


Connecting the Parts



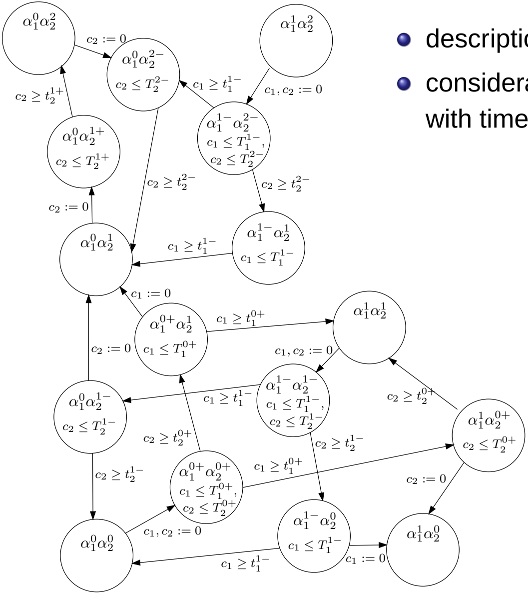
- product locations
- edges specified in component automata

Connecting the Parts



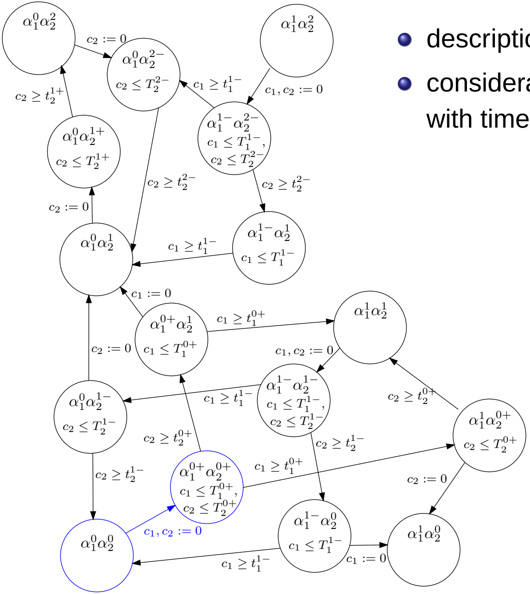
- product locations
- edges specified in component automata
- edges due to network interactions, parameters and current state of the system

Dynamics



- description includes time component
- consideration of behavior in agreement with time constraints

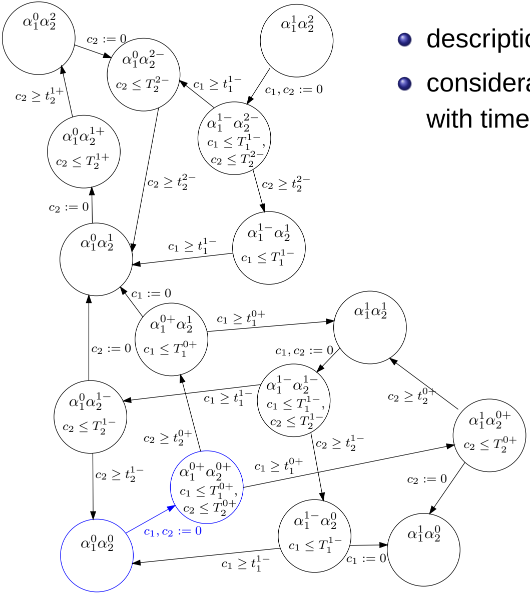
Dynamics



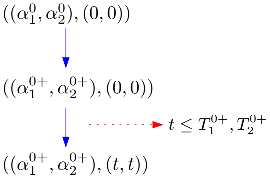
- description includes time component
- consideration of behavior in agreement with time constraints

$$((\alpha_1^0, \alpha_2^0), (0, 0))$$
$$\downarrow$$
$$((\alpha_1^{0+}, \alpha_2^{0+}), (0, 0))$$

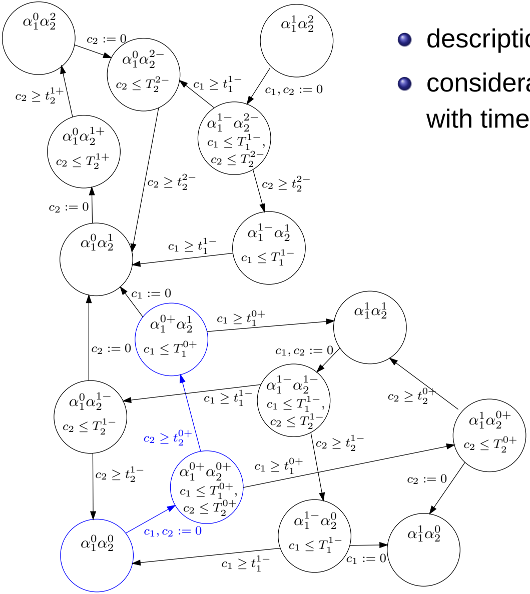
Dynamics



- description includes time component
- consideration of behavior in agreement with time constraints



Dynamics



- description includes time component
- consideration of behavior in agreement with time constraints

$$((\alpha_1^0, \alpha_2^0), (0,0))$$

$$\downarrow$$

$$((\alpha_1^{0+}, \alpha_2^{0+}), (0,0))$$

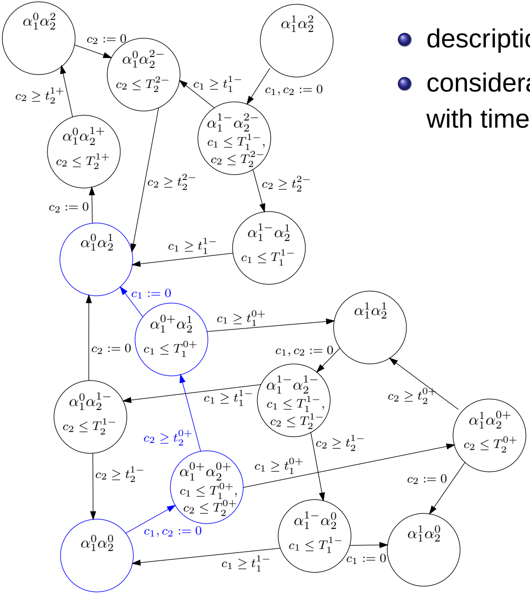
$$\downarrow \quad \cdots\cdots \rightarrow t \leq T_1^{0+}, T_2^{0+}$$

$$((\alpha_1^{0+}, \alpha_2^{0+}), (t,t))$$

$$\downarrow \quad \cdots\cdots \rightarrow t \geq t_2^{0+},\ t \leq T_1^{0+}$$

$$((\alpha_1^{0+}, \alpha_2^1), (t,t))$$

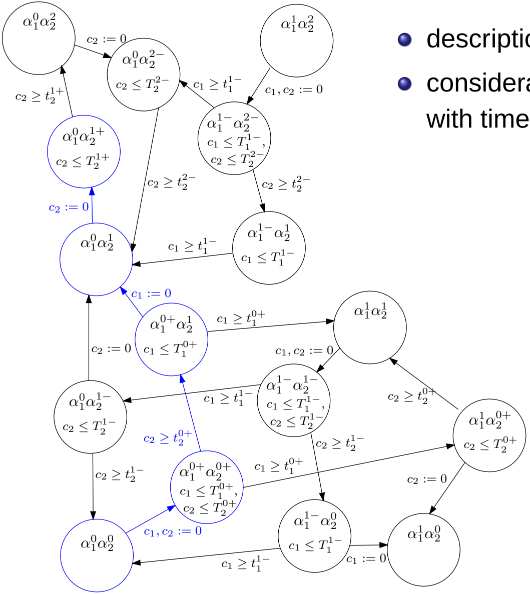
Dynamics



- description includes time component
- consideration of behavior in agreement with time constraints

$$((\alpha_1^0, \alpha_2^0), (0, 0))$$
$$\downarrow$$
$$((\alpha_1^{0+}, \alpha_2^{0+}), (0, 0))$$
$$\downarrow \quad t \leq T_1^{0+}, T_2^{0+}$$
$$((\alpha_1^{0+}, \alpha_2^{0+}), (t, t))$$
$$\downarrow \quad t \geq t_2^{0+},\ t \leq T_1^{0+}$$
$$((\alpha_1^{0+}, \alpha_2^{1}), (t, t))$$
$$\downarrow$$
$$((\alpha_1^{0}, \alpha_2^{1}), (0, t))$$

Dynamics



- description includes time component
- consideration of behavior in agreement with time constraints

$$((\alpha_1^0, \alpha_2^0), (0, 0))$$

$$\downarrow$$

$$((\alpha_1^{0+}, \alpha_2^{0+}), (0, 0))$$

$$\downarrow \quad \cdots\rightarrow t \leq T_1^{0+}, T_2^{0+}$$

$$((\alpha_1^{0+}, \alpha_2^{0+}), (t, t))$$

$$\downarrow \quad \cdots\rightarrow t \geq t_2^{0+},\ t \leq T_1^{0+}$$

$$((\alpha_1^{0+}, \alpha_2^1), (t, t))$$

$$\downarrow$$

$$((\alpha_1^0, \alpha_2^1), (0, t))$$

$$\downarrow$$

$$((\alpha_1^0, \alpha_2^{1+}), (0, 0))$$

Analyzing the Transition System

Dynamics captured in a transition system

- infinite due to time component
- non-deterministic

Analyzing the Transition System

Dynamics captured in a transition system

- infinite due to time component
- non-deterministic

Consistency: state transition graph of the Thomas formalism can be recovered from the dynamics of a suitable timed automata model

Analyzing the Transition System

Dynamics captured in a transition system

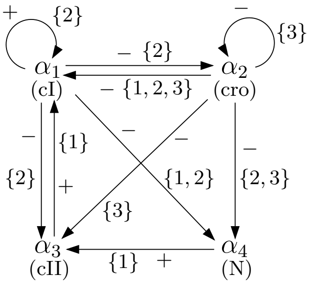
- infinite due to time component
- non-deterministic

Consistency: state transition graph of the Thomas formalism can be recovered from the dynamics of a suitable timed automata model

Possible approach:

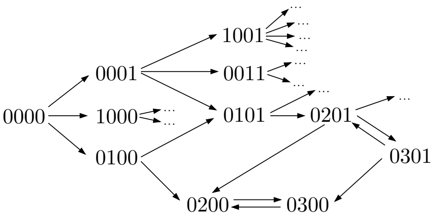
- analysis and verification by means of model checking techniques
- software for editing, simulating and verification of timed automata available
- implementation in UPPAAL

Bacteriophage λ
[D. Thieffry, R. Thomas, 1995]

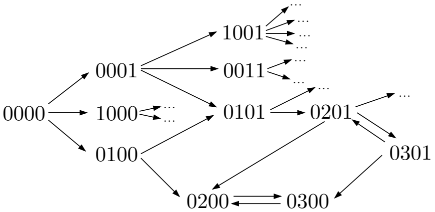


$$
\begin{aligned}
K_{1,\{e_{21}\}} &= 2 & K_{2,\{e_{12}\}} &= 2\\
K_{1,\{e_{31}\}} &= 2 & K_{2,\{e_{12},e_{22}\}} &= 3\\
K_{1,\{e_{11},e_{21}\}} &= 2 & &\\
K_{1,\{e_{11},e_{31}\}} &= 2 & K_{3,\{e_{13},e_{23},e_{43}\}} &= 1\\
K_{1,\{e_{21},e_{31}\}} &= 2 & &\\
K_{1,\{e_{11},e_{21},e_{31}\}} &= 2 & K_{4,\{e_{14},e_{24}\}} &= 1
\end{aligned}
$$

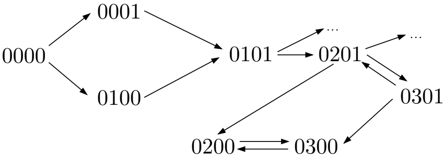
Bacteriophage λ



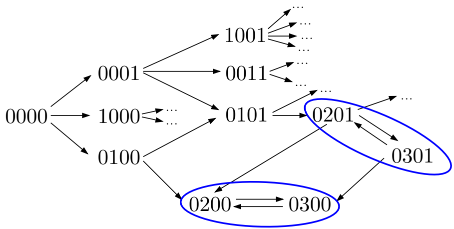
Bacteriophage λ



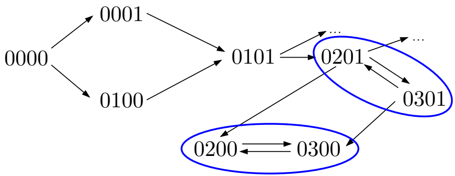
- elimination of pathways violating clock constraints based on temporal data



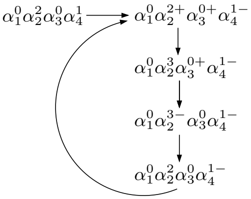
Bacteriophage λ

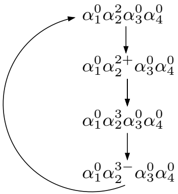


- elimination of pathways violating clock constraints based on temporal data

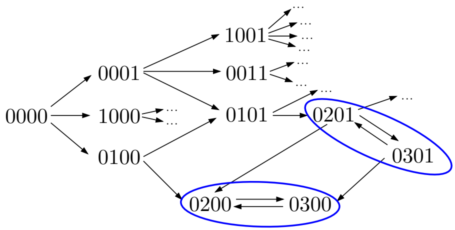


- additional information on the status of component activity

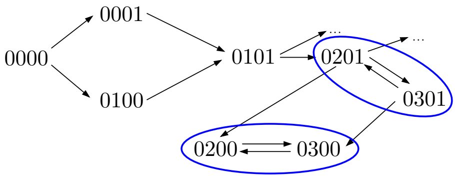




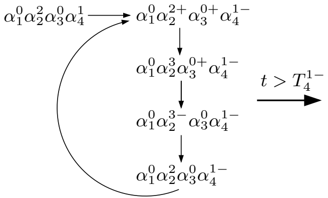
Bacteriophage λ

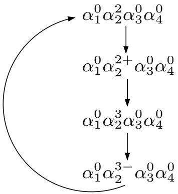


- elimination of pathways violating clock constraints based on temporal data



- additional information on the status of component activity





- evaluation of feasibility and stability of behavior

Conclusion

Modeling formalism

- modular logical modeling of regulatory networks
- incorporating time delays

$\Rightarrow$ refined analysis of the network dynamics

Outlook

- applying the formalism
- developing precise concepts to evaluate feasibility and stability of dynamical behavior
- consideration of more expressive modeling frameworks