

# eBCSgen 2.0: Modelling and Analysis of Regulated Rule-based Systems

Matej Troják, David Šafránek, Branislav Brozmann, and  
Luboš Brim

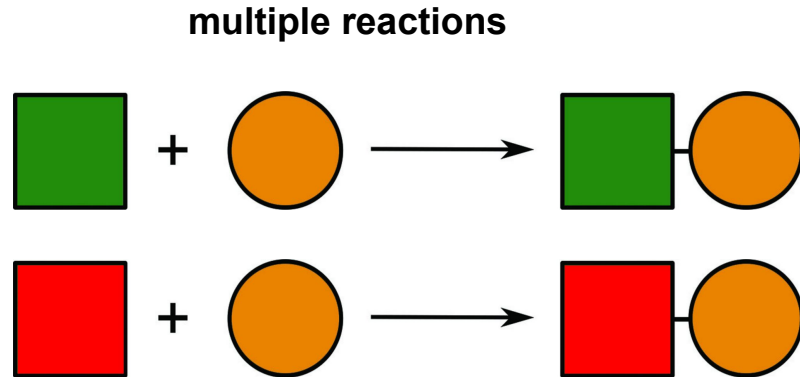


# eBCSgen

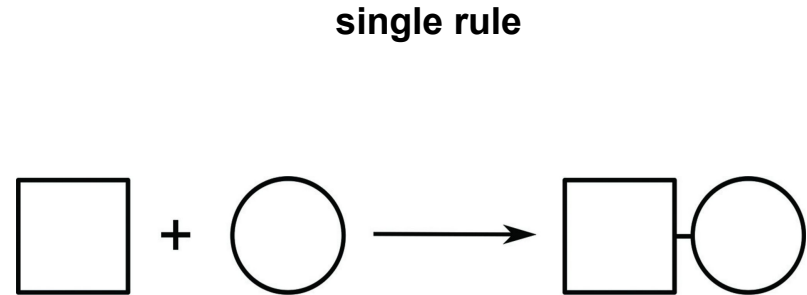
- Support tool for BioChemical Space Language (BCSL)
  - Rule-based formalism
- Introduced new features
  - SBML-multi export
  - Support for early stage model development
    - No mechanistic or quantitative details known
    - E.g complete and incomplete combustion
      - Complete:  $2 \text{CH}_4 + 4 \text{O}_2 \rightarrow 2 \text{CO}_2 + 4 \text{H}_2\text{O}$
      - Incomplete:  $2 \text{CH}_4 + 3 \text{O}_2 \rightarrow 2 \text{CO} + 4 \text{H}_2\text{O}$
    - Signalling pathways
      - Inhibition without mechanistic details
  - Explicit CTL model checking

# Rule-based modeling

- Use patterns (*rules*) to compactly represent reactions
- Indirectly specify a mathematical model



vs.



# BioChemical Space Language

```
#! rules
```

```
r1_S ~ P(S{i})::cell => P(S{a})::cell
```

```
r1_T ~ P(T{i})::cell => P(T{a})::cell
```

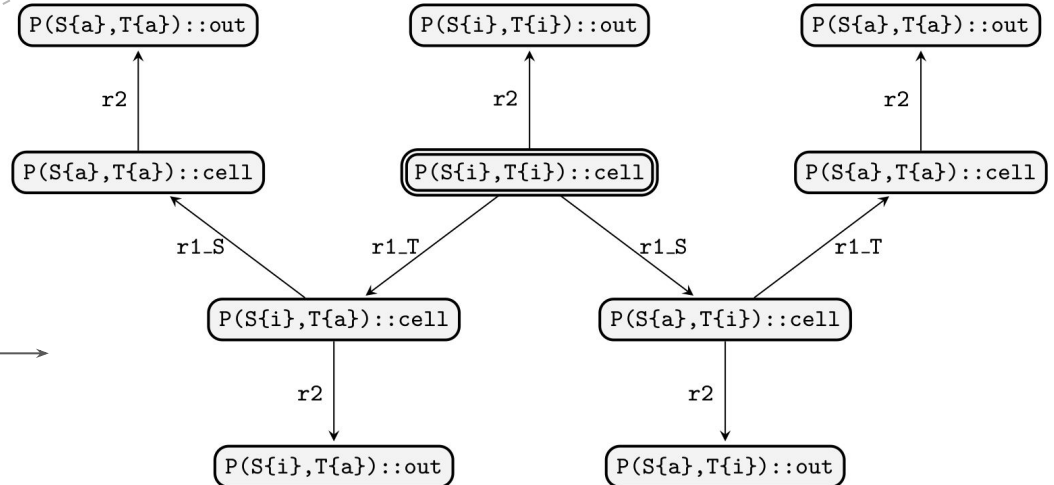
```
r2 ~ P()::cell => P()::out
```

```
#! inits
```

```
1 P(S{i},T{i})::cell
```

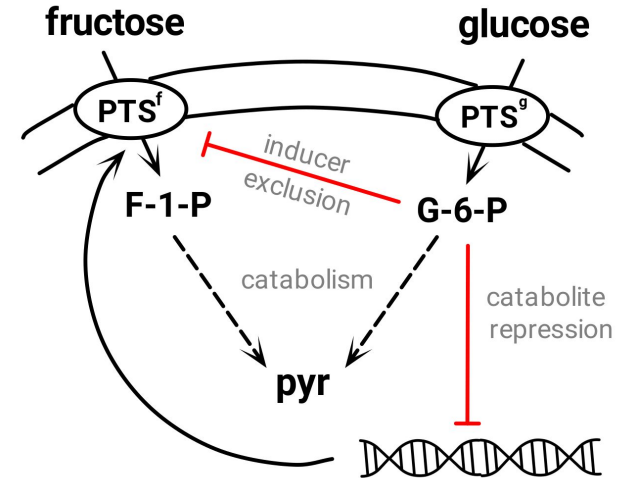
Model

Transition system



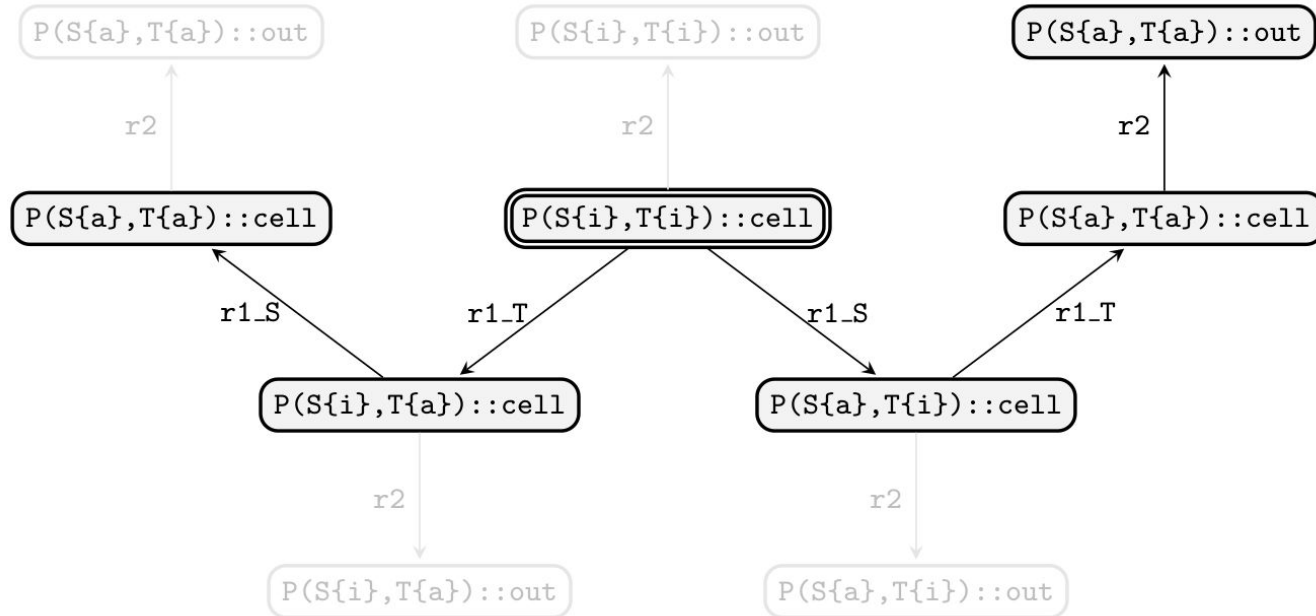
# Early stage model development

- Limited/sketched knowledge about modelled system's parts
  - E.g. carbon transport in E.Coli
- Employed **regulations**
  - Additional conditions on the rewriting process
  - Express uncertain details on the abstract level
  - Several types
    - Regular
    - Ordered
    - Programmed
    - Conditional
    - Concurrent-free



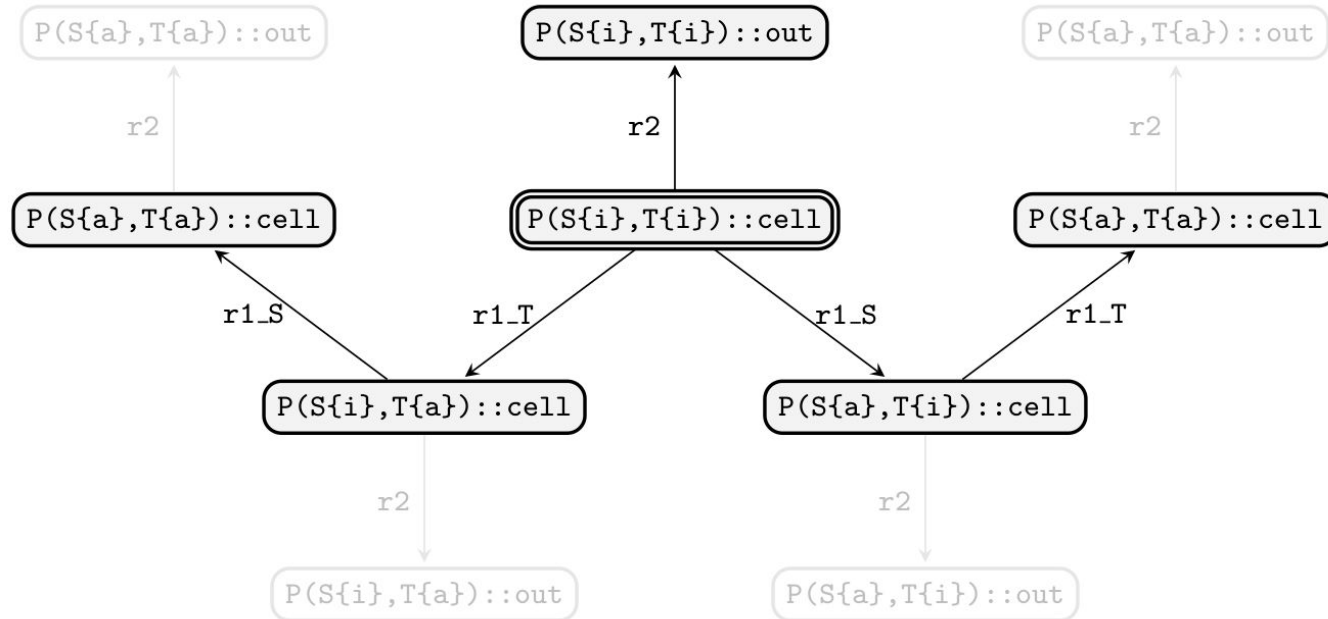
# Regular regulation

- Specify regular expression over rule labels
- E.g.  $r1\_S . r1\_T . r2 \mid r1\_T . r1\_S$



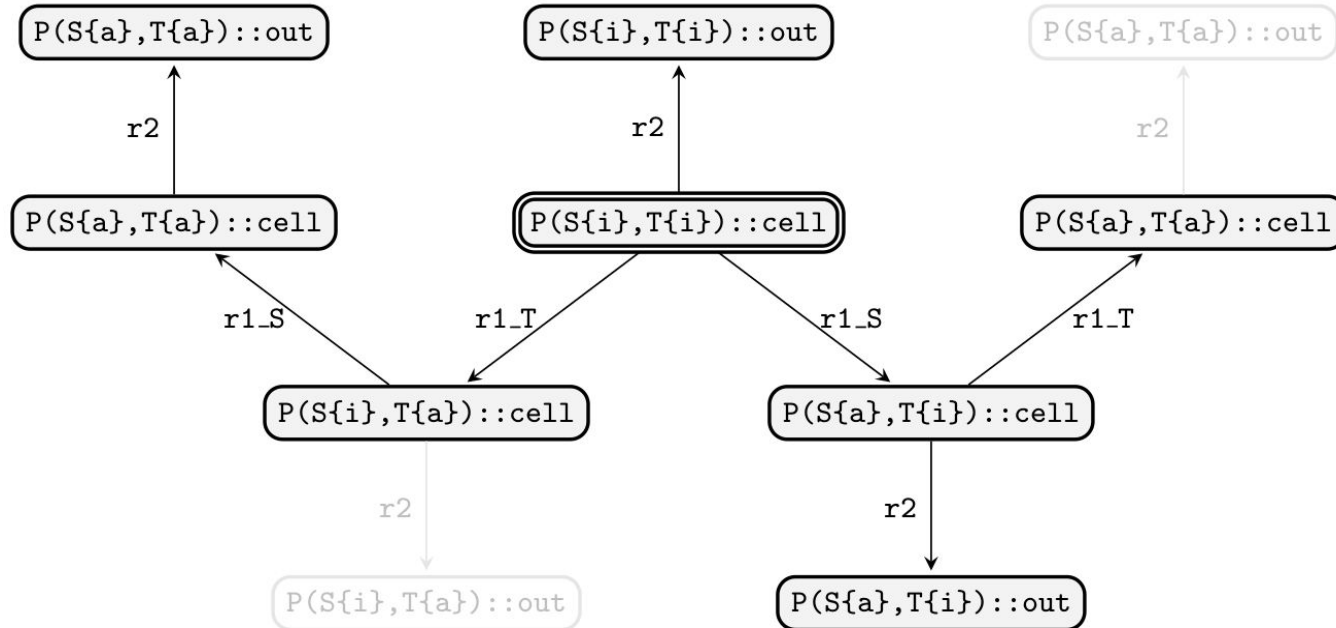
# Ordered regulation

- Partial order over the rules
- E.g.  $r1\_S > r2$ ,  $r1\_T > r2$



# Programmed regulation

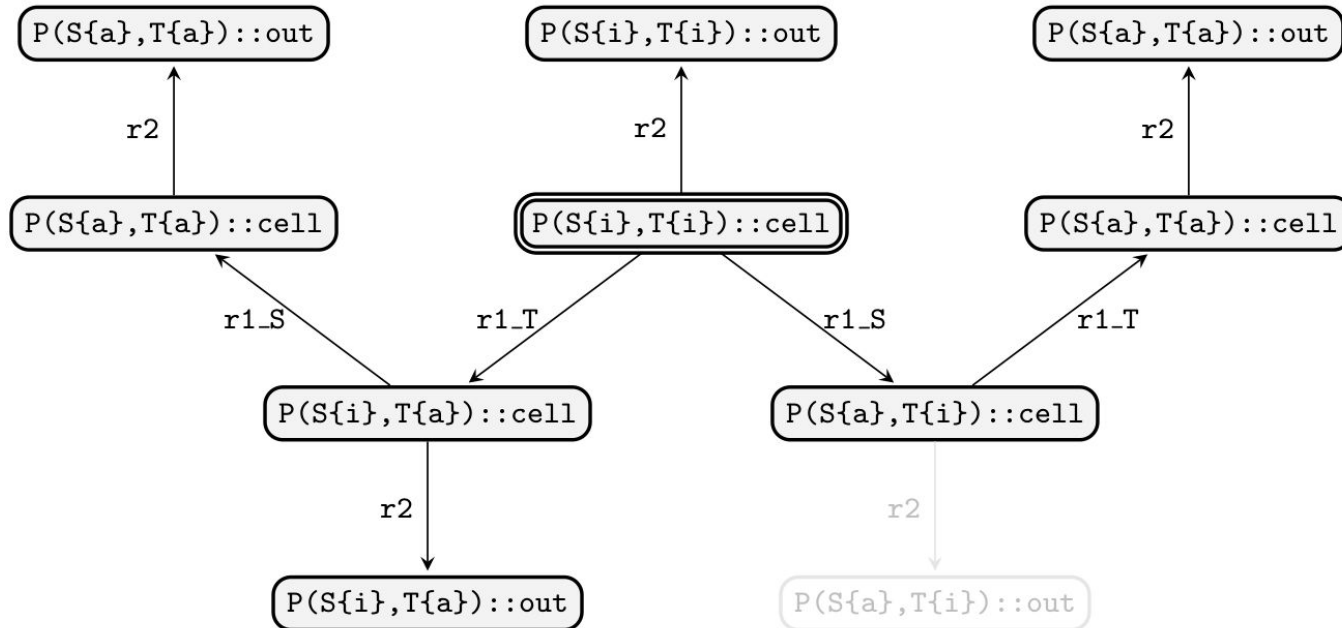
- Explicit successors for every rule
- E.g.  $r1\_S \rightarrow \{r2, r1\_T\}$ ,  $r1\_T \rightarrow \{r1\_S\}$ ,  $r2 \rightarrow \emptyset$





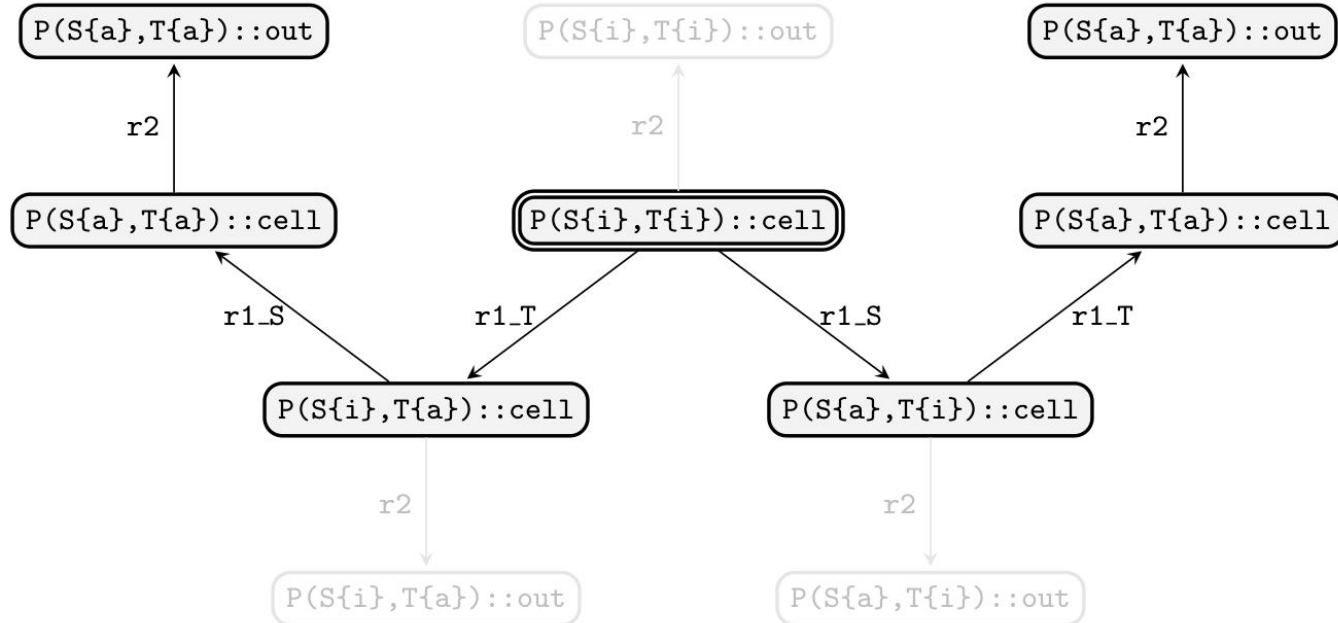
# Conditional regulation

- Requires absence of molecules
- E.g.  $r2 \rightarrow \{ P(S\{a\}, T\{i\})::cell \}$



# Concurrent-free regulation

- Assign priority to concurrent processes
- E.g.  $r1\_S > r2$ ,  $r1\_T > r2$



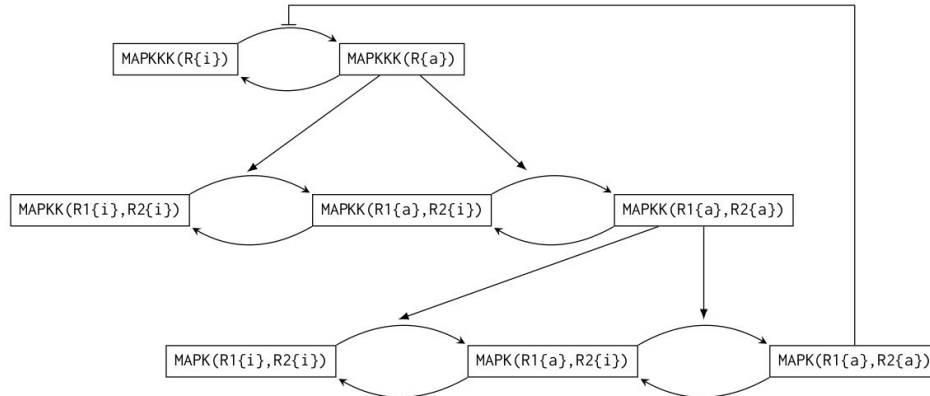
# Back to the examples...

- Combustion

- Complete:  $2 \text{CH}_4 + 4 \text{O}_2 \rightarrow 2 \text{CO}_2 + 4 \text{H}_2\text{O}$
- Incomplete:  $2 \text{CH}_4 + 3 \text{O}_2 \rightarrow 2 \text{CO} + 4 \text{H}_2\text{O}$
- Concurrent-free - assign priority to Compete combustion rule

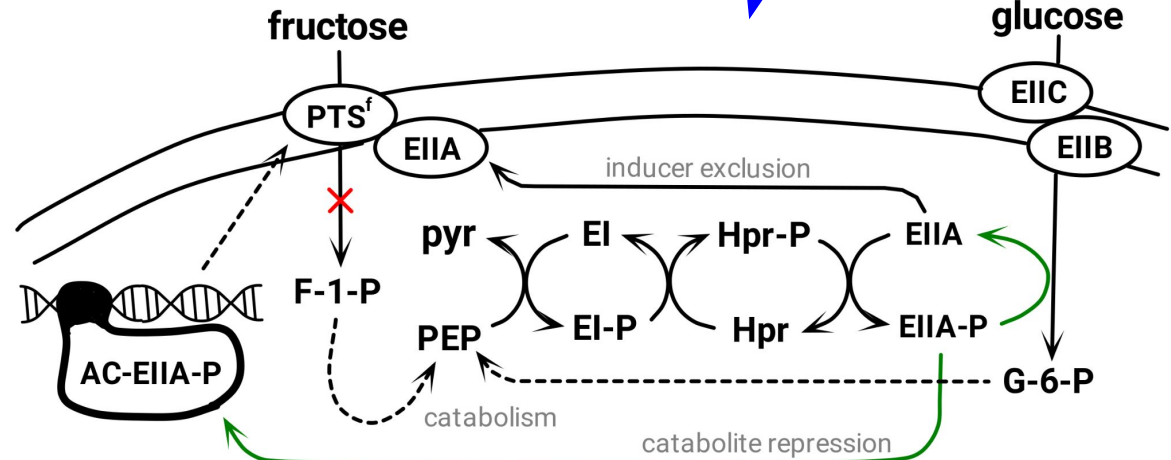
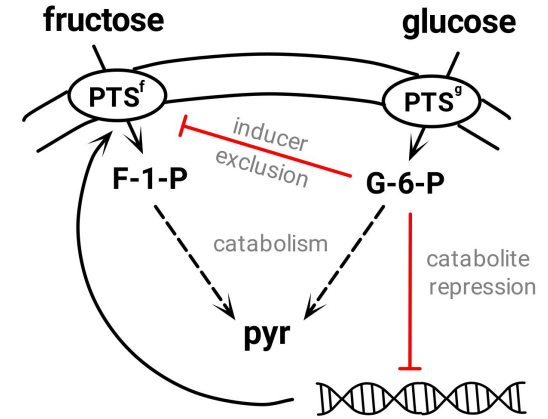
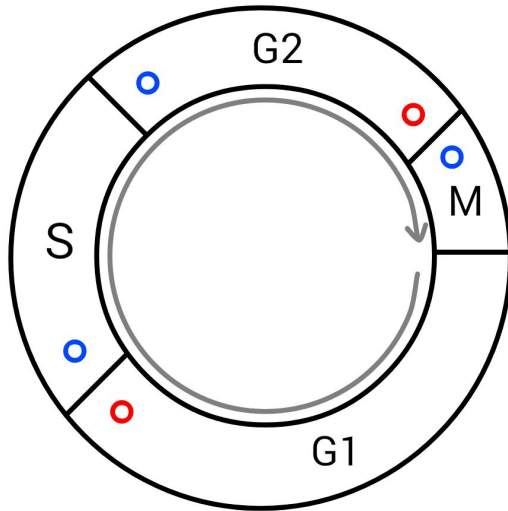
- Signalling pathway (illustrative example on MAPK)

- Conditional regulation, rendering respective rule disabled with  $\text{MAPK}(\text{R1}\{a\}, \text{R2}\{a\})$  present



# Biological case studies

- Decision making in cell cycle (programmed)
- Decision between inducer exclusion and catabolite repression (concurrent-free)



# eBCSgen features summary

- Python command-line tool
- Galaxy user interface
  - <https://biodivine-vm.fi.muni.cz/galaxy>
- Available as bioconda package, docker container and (soon) series of published Galaxy tools
- Main features
  - Interactive editor
  - Simulations
  - PCTL model checking & parameter synthesis
  - Various static analyses
  - Interactive visualisations
  - Support for introduced regulations
  - CTL model checking
  - SBML-multi export

Demo