

# **Modeling biological systems in hybrid concurrent constraint programming**

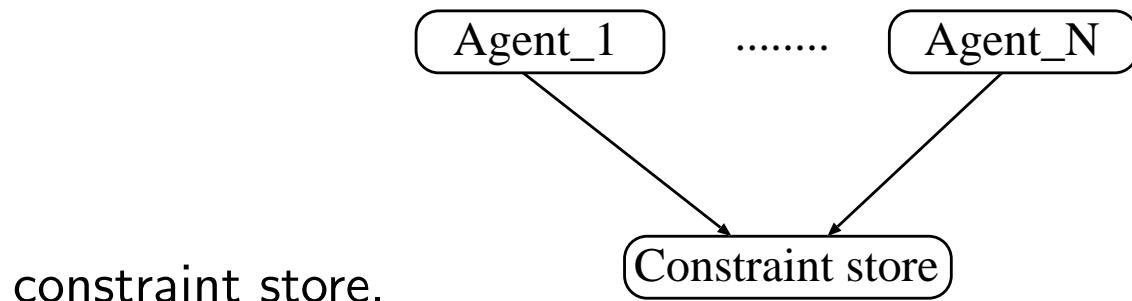
Alexander Bockmayr and Arnaud Courtois

LORIA & Université Henri Poincaré  
Nancy, France

# Concurrent Constraint Programming

Saraswat 93, van Hentenryck/Saraswat 96

- Constraint: Partial information about the state of a system (possible values of variables), e.g.  $x - y \geq 0, x^2 + y^2 \leq 1$ .
- Concurrent constraint program: Set of independent agents interacting through the



- constraint store.
- Each agent may tell the store new constraints or ask if constraints are entailed.
  - Output of a cc program: State of the store when all agents have finished their work.

# Hybrid Concurrent Constraint Programming

Gupta/Jagadeesan/Saraswat/Bobrow 95, Gupta/Jagadeesan/Saraswat 98

- Declarative compositional language for modeling hybrid systems
- Discrete and continuous state changes
- Constraints  $\leadsto$  algebraic and differential equations
- Piecewise continuous systems  $\leadsto$  alternate point and interval phases
- Execute cc program in each phase to determine output
- In interval phase, variables change according to differential equations.  
Phase ends when a transition condition fires.

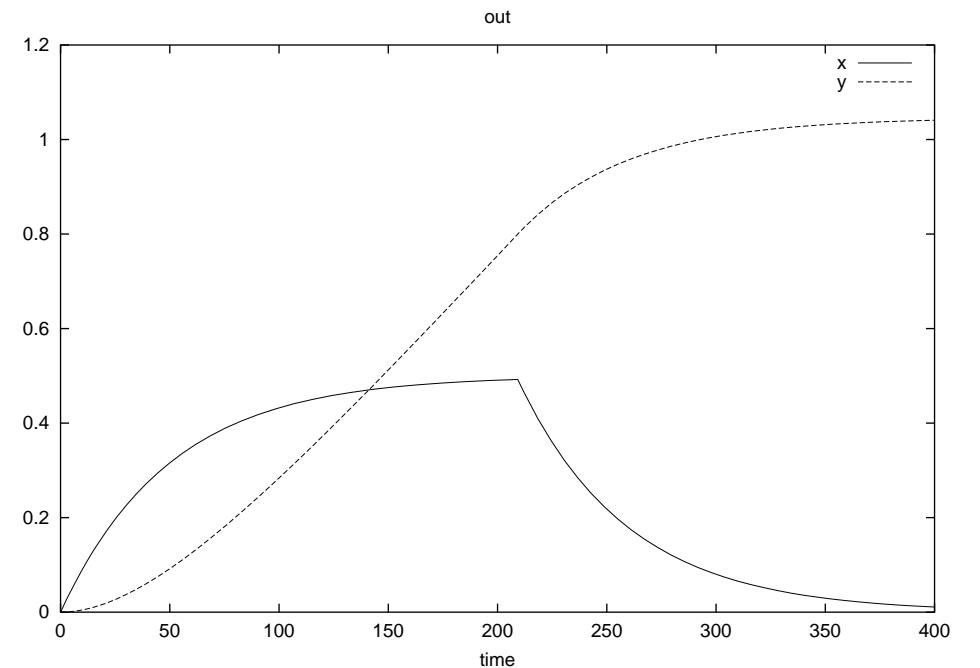
## Example: Interaction between two genes

Gene  $x$  activates gene  $y$ .

Above a certain threshold, gene  $y$  inhibits expression of gene  $x$ .

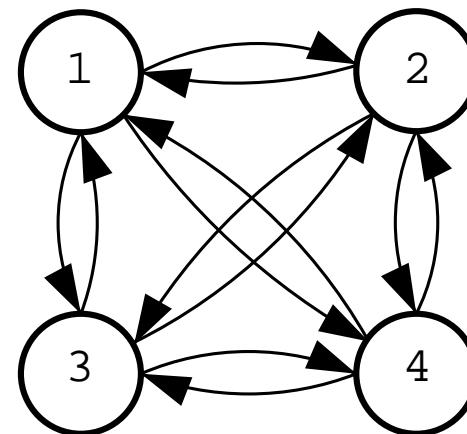
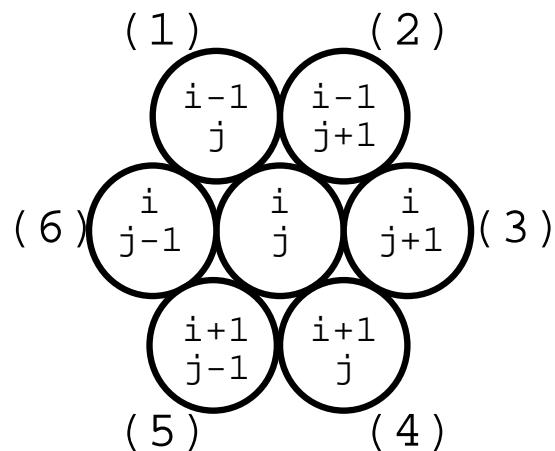
Hybrid cc program

```
x=0; y=0;  
always {  
    if (y < 0.8) x' = -0.02*x+0.01;  
    if (y >= 0.8) x' = -0.02*x;  
    y' = 0.01*x;  
}  
sample(x,y);
```



## Example: Cell differentiation

- Lateral inhibition through Delta-Notch Signaling (Ghosh/Tomlin 2001)
- Hexagonal lattice of cells



- For each cell: Delta and Notch concentrations  $v_D, v_N$
- Notch production capacity  $u_N = \sum_{i=1}^k v_D^i$ , with  $k \in \{2, 3, 4, 6\}$ ,  
Delta production capacity  $u_D = -v_N$ , and thresholds  $h_D, h_N$ .

- Four possible states for each cell, 12 transitions

**State 1:** Delta and Notch inhibited:  $u_D < h_D$  and  $u_N < h_N$

**State 2:** Notch inhibited, Delta expressed:  $u_D \geq h_D$  and  $u_N < h_N$

**State 3:** Delta inhibited, Notch expressed:  $u_D < h_D$  and  $u_N \geq h_N$

**State 4:** Delta and Notch expressed:  $u_D \geq h_D$  and  $u_N \geq h_N$

- Production rules for proteins, depending on the state of the cell

**State 1:**  $v'_D = -\lambda_D \cdot v_D, \quad v'_N = -\lambda_N \cdot v_N$

**State 2:**  $v'_D = R_D - \lambda_D \cdot v_D, \quad v'_N = -\lambda_N \cdot v_N$

**State 3:**  $v'_D = -\lambda_D \cdot v_D, \quad v'_N = R_N - \lambda_N \cdot v_N$

**State 4:**  $v'_D = R_D - \lambda_D \cdot v_D, \quad v'_N = R_N - \lambda_N \cdot v_N$

| Biology             | Hybrid cc              |
|---------------------|------------------------|
| reaching thresholds | discrete events        |
| time, concentration | continuous variables   |
| kinetics            | differential equations |
| gene interaction    | concurrency            |
| stochastic behavior | random numbers         |

## Program in Hybrid cc

```
interval ud_1_1, un_1_1, vd_1_1, vn_1_1, state_1_1, ud_2_1, ...
interval coeff_rd(interval ud) {
    if (ud<hd) return=0;
    else return=1; }
vd_1_1 = 0.8125; vn_1_1 = 1.0000; ... /* Random noise */

always {
    un_1_1 = vd_1_2 + vd_2_1; ...
    ud_1_1 = -vn_1_1; ...

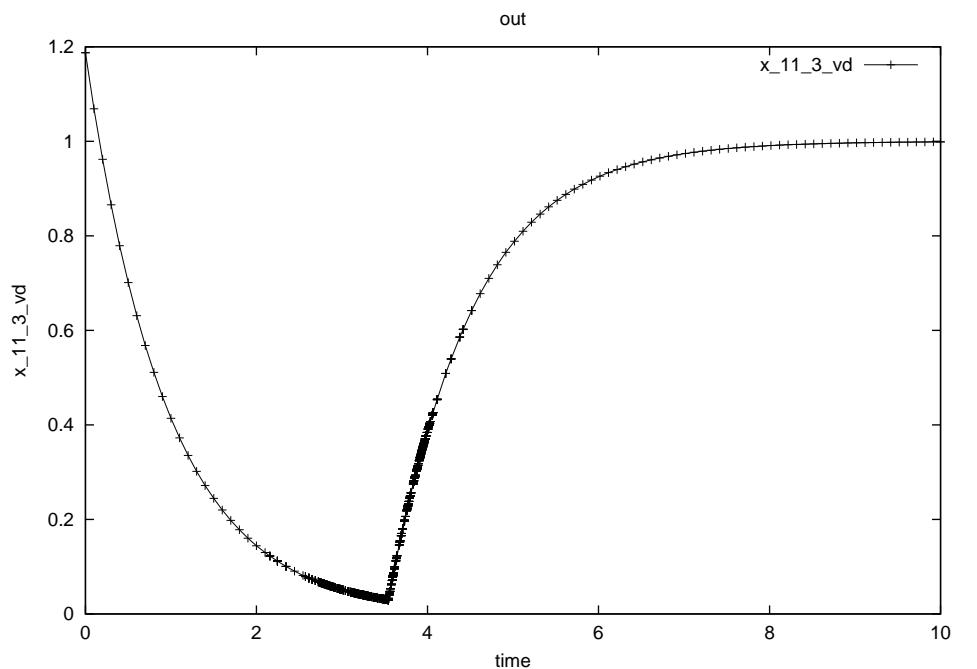
    vd_1_1' = (coeff_rd(ud_1_1)*rd) - lambda_d * vd_1_1; ...
    vn_1_1' = (coeff_rn(un_1_1)*rn) - lambda_n * vn_1_1; ...

    if ((ud_1_1 < hd) && (un_1_1 < hn)) state_1_1 = 1; ...
}

sample(state_1_1, state_1_2, state_2_1, ...);
```

## Example run

```
* o o * o * o * o *
o * o o o o o o o o
o o * o o * o o o o *
* o o * o o * o o o
o o o o * o o * o *
* o o o o * o o o o
o o * o o o o * o o
* o o * o * o o o *
o o o o o o o o o o o
* o * o * o * o o *
```



1. Stable state of a 10x10 lattice:  
each cell is either in State 2 (marked by \*) or State 3 (marked by o).
2. Temporal evolution of Delta concentration in a border cell before and after a transition.

## References

- [1] R. Alur, C. Belta, F. Ivancic, V. Kumar, M. Mintz, G. J. Pappas, H. Rubin, and J. Schug. Hybrid modeling and simulation of biomolecular networks. In *Hybrid Systems: Computation and Control, HSCC 2001*, Springer LNCS 2034, 2001.
- [2] A. Bockmayr, N. Pisaruk, and A. Aggoun. Network flow problems in constraint programming. In *Principles and Practice of Constraint Programming, CP'2001*, Springer LNCS 2239, 2001.
- [3] A. Courtois. Modélisation de systèmes biologiques en programmation par contraintes. Rapport de DEA (in French), Univ. Henri Poincaré, LORIA, July 2001.
- [4] R. Ghosh and C. Tomlin. Lateral inhibition through Delta-Notch signaling: A piecewise affine hybrid model. In *Hybrid Systems: Computation and Control, HSCC 2001*, Springer LNCS 2034, 2001.
- [5] V. Gupta, R. Jagadeesan, and V. Saraswat. Computing with continuous change. *Science of computer programming*, 30(1-2):3–49, 1998.
- [6] V. Gupta, R. Jagadeesan, V. Saraswat, and D. G. Bobrow. Programming in hybrid constraint languages. In *Hybrid Systems II*, Springer, LNCS 999, 1995.
- [7] V. A. Saraswat. *Concurrent constraint programming*. ACM Doctoral Dissertation Awards. MIT Press, 1993.
- [8] P. van Hentenryck and V. Saraswat. Strategic directions in constraint programming. *ACM Computing Surveys*, 28(4):701 – 726, 1996.