Causality in Membrane Systems

Nadia Busi

University of Bologna

Overview

- Aim
- Causality in P systems
  - Events in P systems
  - An informal description of causality
  - Retrievability and diamond properties
- Maximal parallelism semantics
- Causal semantics
- Future work

Overview

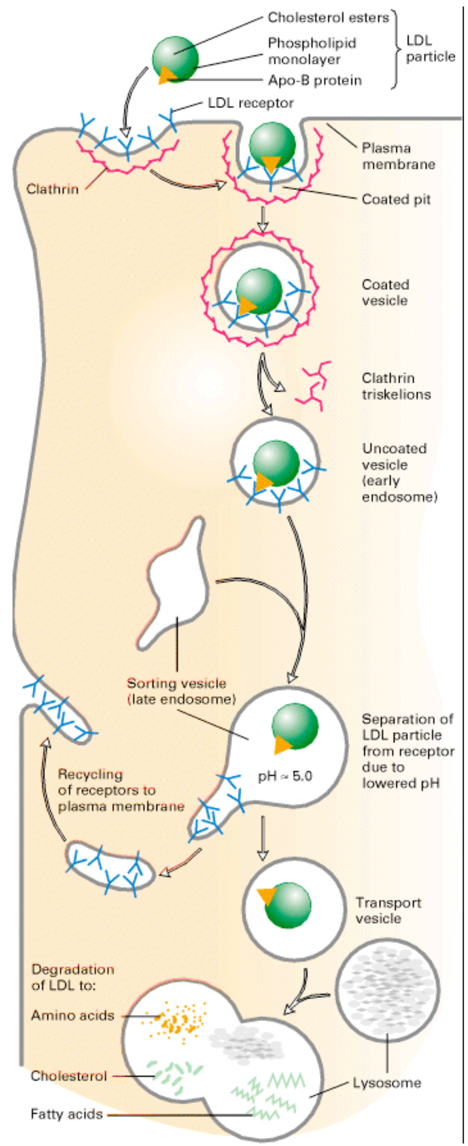
- Aim
- Causality in P systems
  - Events in P systems
  - An informal description of causality
  - Retrievability and diamond properties
- Maximal parallelism semantics
- Causal semantics
- Future work

Causality

- Identify the causal dependencies among the events of a system
- Causality in concurrency theory
  - Petri nets ~'80
  - CCS-like process algebras ~'90
  - Pi-calculus ~'95
  - Mobile ambients ???
  - Bio-inspired calculi (Beta Binders, Brane Calculi) ongoing work

Causality in biology

- Identify dependencies between two events in a pathway
- Analysis: limit the search space in case an unpredictable behaviour occurs



LDL Cholesterol degradation pathway

Overview

- Aim
- **Causality in P systems**
  - Events in P systems
  - An informal description of causality
  - Retrievability and diamond properties
- Maximal parallelism semantics
- Causal semantics
- Future work

Causality in P systems

- We consider (basic) P systems with cooperative rules
  - ab -> c, (d,in), (e,out)
- The used semantics (sequential, maximal parallelism, …) doesn't matter
  - … even if the definitions of the (maximally) parallel semantics and of the causal semantics are intimately connected

What is an event in a P system?

- An event is the application of a single reaction rule
  - Alternatives: e.g., an event is a maximal parallelism computational step
  - Closer to the intuition of what is an event
  - More faithful to the biological reality
  - Independent from the adopted semantics

Causal semantics for P system

- Given a “sequential” execution of the system, for each event identify its set of causes, i.e., the set of previously occurred events on which it depends
- Mixed ordering (vs. partial ordering) semantics

Causal semantics: example

[a, a, b, a->c, bc->d]

Causal semantics: example

[a, a, b, a->c, bc->d]

Causal semantics: example

[a, a, b, a->c, bc->d] --e1-->

[a, b, c, a->c, bc->d]

Causal semantics: example

[a, a, b, a->c, bc->d] --e1-->

[a, b, c, a->c, bc->d]

Causal semantics: example

[a, a, b, a->c, bc->d] --e1-->

[a, b, c, a->c, bc->d] --e2-->

[a, d, a->c, bc->d]

Causal semantics: example

[a, a, b, a->c, bc->d] --e1-->

[a, b, c, a->c, bc->d] --e2-->

[a, d, a->c, bc->d]

Causal semantics: example

[a, a, b, a->c, bc->d] --e1-->

[a, b, c, a->c, bc->d] --e2-->

[a, d, a->c, bc->d] --e3-->

[c, d, a->c, bc->d]

Causal semantics: example

[a, a, b, a->c, bc->d] --e1-->

[a, b, c, a->c, bc->d] --e2-->

[a, d, a->c, bc->d] --e3-->

[c, d, a->c, bc->d]

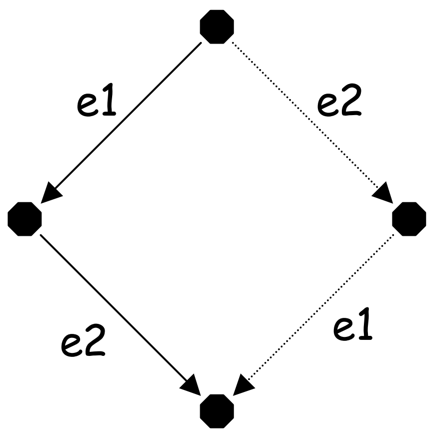
- e2 causally depends on e1 (if e1 does not occur, e2 cannot happen)
- e3 is independent from both e1 and e2

Causal semantics: properties

- Retrievability of (sequential and) maximal parallelism semantics
  - We can produce the maximal parallelism semantics by looking only at the causal moves
  - We do not need to look inside the state of the system

Causal semantics: properties

- Diamond property
  - If two independent (i.e, not causally related) events can occur one after the other, then they can also happen in the reverse ordering
  - The two different orderings lead to the same system



Overview

- Aim
- Causality in P systems
  - Events in P systems
  - An informal description of causality
  - Retrievability and diamond properties
- **Maximal parallelism semantics**
- Causal semantics
- Future work

Maximal parallelism semantics: a formal definition

- A maximal parallelism computational step is obtained as a maximal sequence of simple evolution steps, each one consisting in the application of a single rule
- To represent the intermediate states of the systems (reached after the firing of a non-maximal sequence of rules) we introduce the notion of partial configuration.

Maximal parallelism semantics: partial configuration

- In a partial configuration the contents of each region is represented by two multisets:
  - Active objects:
    - objects that were in the region at the beginning of the current maximally parallel computational step
    - Can be used in the application of the next rule
  - Frozen objects:
    - Objects that have been produced in the region during the current maximally parallel computational step
    - Will be available only in the next maximally parallel computational step

Maximal parallelism semantics: example

[a, a->(b,in) [c, c, bc->(a,out), c->d]]

Maximal parallelism semantics: example

[a, **a->(b,in)** [c, c, bc->(a,out), c->d]]

Maximal parallelism semantics: example

[a, **a->(b,in)** [c, c, bc->(a,out), c->d]] -->

[a->(b,in) [b, c, c, bc->(a,out), c->d]]

Maximal parallelism semantics: example

[a, **a->(b,in)** [c, c, bc->(a,out), c->d]] -->

[a->(b,in) [b, **c**, c, **bc->(a,out)**, c->d]] -/->

Maximal parallelism semantics: example

[a, **a->(b,in)** [c, c, bc->(a,out), c->d]] -->

[a->(b,in) [b, **c**, c, bc->(a,out), **c->d**]]

Maximal parallelism semantics: example

[a, **a->(b,in)** [c, c, bc->(a,out), c->d]] -->

[a->(b,in) [b, c, c, bc->(a,out), **c->d**]] -->

[a->(b,in) [b, d, c, bc->(a,out), c->d]]

Maximal parallelism semantics: example

[a, **a->(b,in)** [c, c, bc->(a,out), c->d]] -->

[a->(b,in) [b, **c**, c, bc->(a,out), **c->d**]] -->

[a->(b,in) [b, d, **c**, bc->(a,out), **c->d**]]

Maximal parallelism semantics: example

[a, **a->(b,in)** [c, c, bc->(a,out), c->d]] -->

[a->(b,in) [b, **c**, c, bc->(a,out), **c->d**]] -->

[a->(b,in) [b, d, **c**, bc->(a,out), **c->d**]] -->

[a->(b,in) [b, d, **d**, bc->(a,out), c->d]] -/->

Maximal parallelism semantics: example

[**a**, **a->(b,in)** [c, c, bc->(a,out), c->d]] -->

[a->(b,in) [b, **c**, c, bc->(a,out), **c->d**]] -->

[a->(b,in) [b, d, **c**, bc->(a,out), **c->d**]] -->

[a->(b,in) [b, d, **d**, bc->(a,out), c->d]] -/->

[a, a->(b,in) [c, c, bc->(a,out), c->d]] ==>

Heated([a->(b,in) [b, d, d, bc->(a,out), c->d]]) =

[a->(b,in) [b, d, d, bc->(a,out), c->d]]

Maximal parallelism semantics

- Reaction relation --> between partial configurations
- Heating function: transforms frozen objects into active objects
- Maximal parallelism computational step ==>
  - Maximal sequence of reactions -->
  - Application of the heating function to the last configuration

Overview

- Aim
- Causality in P systems
  - Events in P systems
  - An informal description of causality
  - Retrievability and diamond properties
- Maximal parallelism semantics
- **Causal semantics**
- Future work

Causal semantics

- Denumerable set of cause names
- Each event is decorated with two data:
  - A fresh cause name k, that identifies the event
  - A set of cause names H containing all the names - associated with previously occurred events - that are a cause for the current event
- When an event occurs, all the objects produced by the event are decorated with the cause name k associated to the event
- The set of causes of an event is obtained as the union of the sets of causes of the objects that it consumes

Causal semantics: example

[(a,0), (e,0), a->bc, c->d, e->f]

Causal semantics: example

[**(a,0)**, (e,0), **a->bc**, c->d, e->f]

Causal semantics: example

[**(a,0)**, (e,0), **a->bc**, c->d, e->f]

-- k1,0 -->

[(b,k1), (c,k1), (e, 0), a->bc, c->d, e->f]

Causal semantics: example

[**(a,0)**, (e,0), **a->bc**, c->d, e->f]

-- k1,0 -->

[(b,k1), (c,k1), **(e, 0)**, a->bc, c->d, **e->f**]

Causal semantics: example

[**(a,0)**, (e,0), **a->bc**, c->d, e->f]

-- k1,0 -->

[(b,k1), (c,k1), **(e, 0)**, a->bc, c->d, **e->f**]

-- k2, 0 -->

[(b,k1), (c,k1), **(f, k2)**, a->bc, c->d, e->f]

Causal semantics: example

[**(a,0)**, (e,0), **a->bc**, c->d, e->f]

-- k1,0 -->

[(b,k1), (c,k1), **(e, 0)**, a->bc, c->d, **e->f**]

-- k2, 0 -->

[(b,k1), (c,k1), **(f, k2)**, a->bc, c->d, e->f]

Events labeled with k1 and k2 are independent and can be swapped.

Causal semantics: example

[**(a,0)**, (e,0), **a->bc**, c->d, e->f]

-- k1,0 -->

[(b,k1), (c,k1), **(e, 0)**, a->bc, c->d, **e->f**]

-- k2, 0 -->

[[(b,k1), **(c,k1)**, (f, k2), a->bc, **c->d**, e->f]

Causal semantics: example

[**(a,0)**, (e,0), **a->bc**, c->d, e->f]

-- k1,0 -->

[(b,k1), (c,k1), **(e, 0)**, a->bc, c->d, **e->f**]

-- k2, 0 -->

[[(b,k1), **(c,k1)**, (f, k2), a->bc, **c->d**, e->f]

-- k3, {k1} -->

[(b,k1), (c, k1), **(d, k3)**, (f, k2), a->bc, c->d, e->f]

Event k3 causally depends on event k1

Causal semantics: properties

- The following properties hold for P systems with cooperative rules:
  - retrievability of the maximal parallelism semantics
  - diamond property

Causal semantics vs maximal parallelism semantics

- Causal semantics is "finer" than maximal parallelism semantics, as it permits to exactly identify which events are a cause for another event

Causal semantics vs maximal parallelism semantics

- [a, e, a->bc, c->d, e->f]
- [a, e, a->bc, cf->d, e->f]

Causal semantics vs maximal parallelism semantics

- P1 = [a, e, **a->bc**, c->d, **e->f**]
- P2 = [a, e, **a->bc**, cf->d, **e->f**]
- According to the maximal parallelism semantics, the two systems have the same behaviour
- According to the causal semantics
  - Event “c->d” in P1 causally depends on “a->bc” only
  - Event “cf->d” in P2 causally depends on both “a->bc” and “e->f”

Future work

- P systems with promoters and inhibitors
  - Different choices for the definition of the semantics
  - Some of the properties enjoyed by P systems with cooperative rules (may) no longer hold
- P systems with a dynamically evolving membrane structure
  - E.g., dissolution, duplication, brane-like operations

Thank you!

Bibliography

- [MBC] Alberts et al., Molecular Biology of the Cell, Garland.
- [MCB] Lodish et al., Molecular Cell Biology, Freeman.