University of Leicester

Stochastic Analysis of Graph Transformation Systems:
A Case Study in P2P Networks

Reiko Heckel, University of Leicester
Joint work with G. Lajios and S. Menge, Dortmund

CS Colloquium, Swansea, 15/11/05

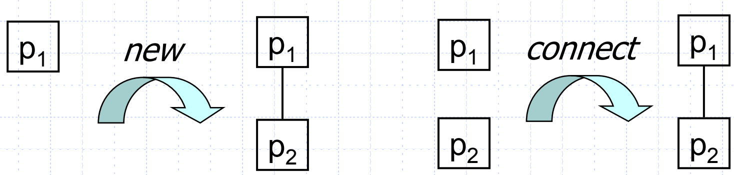
Intl. Conference on Theoretical Aspects of Computing, October 2005, Hanoi, Vietnam

Why it is fun: Modelling By Example

StageCast (www.stagecast.com): visual programming environment for kids

- behavioural rules associated to graphical objects
- visual pattern matching
- control structures, ….

➔ intuitive rule-based behaviour modelling



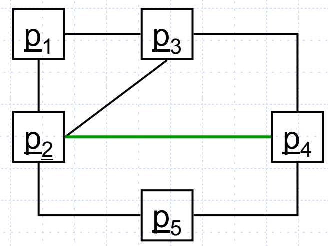
Graph Transformation: abstract from concrete visual presentation

Case Study

Problem:

- no central infrastructure
- unreliable components

➔ removing nodes may disconnect network



Idea: introduce redundancy!

Question: Which links should be added to guarantee given level of reliability ?

a) at random, up to a limit of **n** links
b) so that deletion of a node does not increase length of communication paths

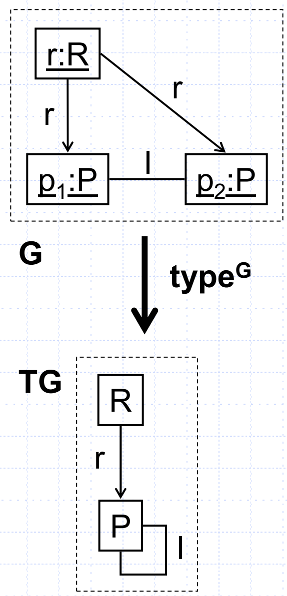
Outline

- Graph transformation in a nutshell
- Stochastic graph transformation
- Stochastic temporal logic
- Application
- Conclusion

How it works: Typed Graphs

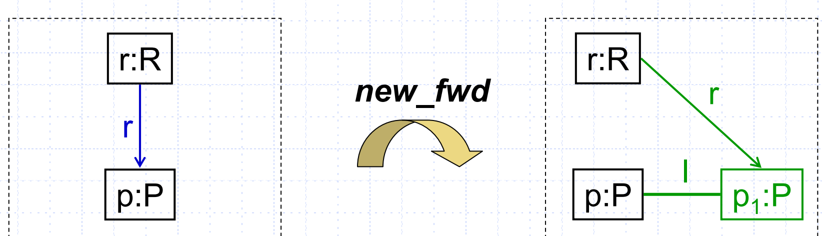
- Directed graphs as algebraic structures
  G = (V, E, src, tar)
  with **src, tar: E → V**
- Graph homomorphism as pair of mappings
  h = (h_V , h_E): G_1 → G_2 with
  - **h_V : V_1 → V_2**
  - **h_E : E_1 → E_2**

  preserving **src** and **tar**
- Typed graphs given by
  - fixed type graph **TG**
  - instance graphs **G** typed over **TG** by homomorphism **g: G → TG**



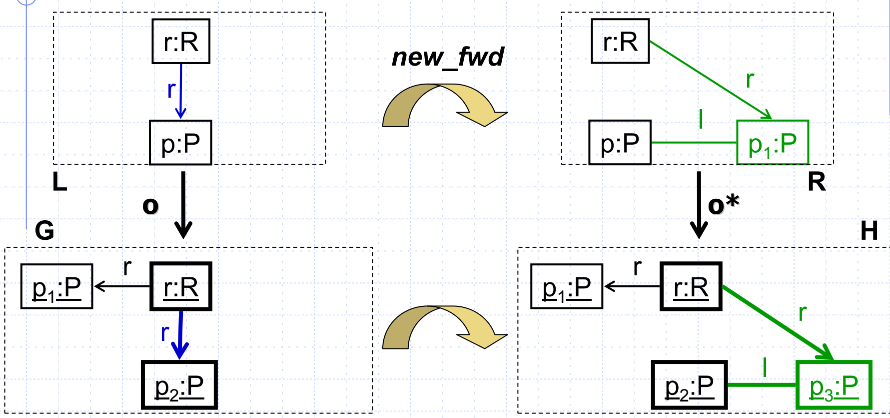
Rules

p: L → R with **L ∩ R** well-defined, in different presentations
- with *L ∩ R* explicit [DPO]: **L ← K → R**
- as partial homomorphism [SPO]: **L ⊇ dom(p) → R**



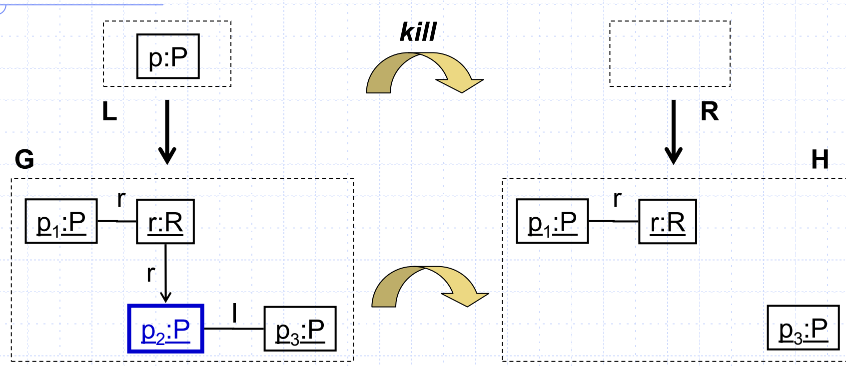
Adds node p1 to the network, connecting it to a recently added node, and changes the reference.

Transformation: Operationally



1. select rule **p : L → R** ; occurrence **o : L → G**
2. remove from *G* the occurrence of **L \ R**
3. add to result a copy of **R \ L**

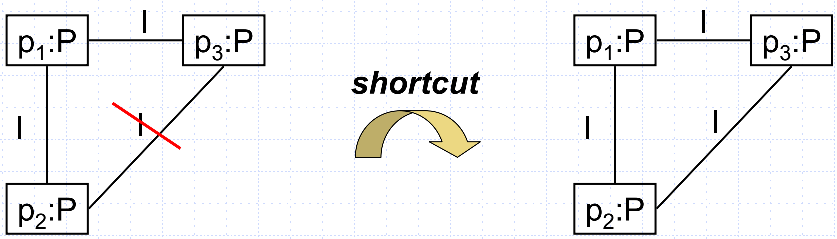
Transformation: More interesting



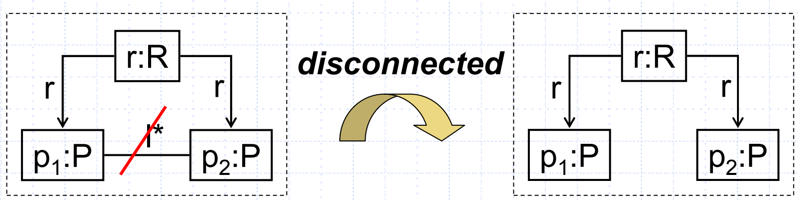
- × conservative solution [DPO]: application is forbidden
  - invertible transformations, no side-effects
- ✓ radical solution [SPO]: delete dangling edges
  - more complex behavior, requires more control

Application Conditions

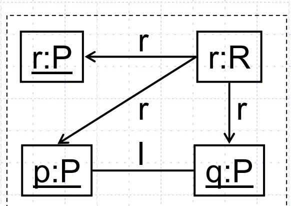
Add redundant link, if it does not exists yet.



Path Expressions



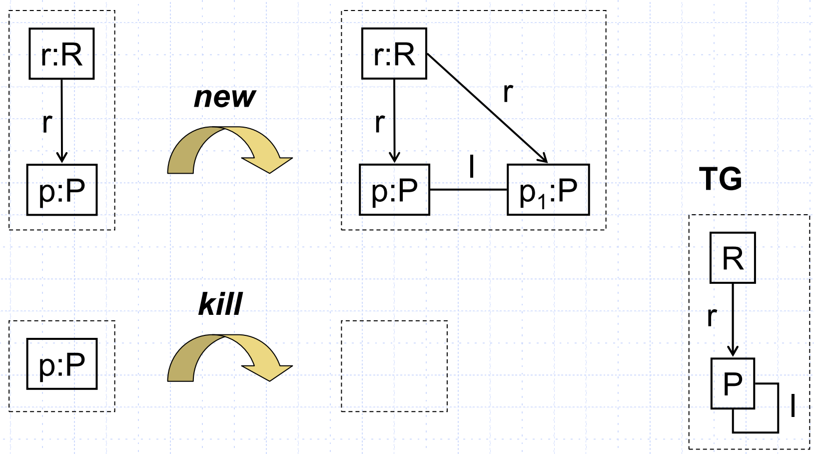
- applicable if there is *no path labelled **l**** between the images of p_1 and p_2 ; no side-effect
- used as graphical predicate *disconnected(p1, p2)*



disconnected(r, q),
but not disconnected(p, q)

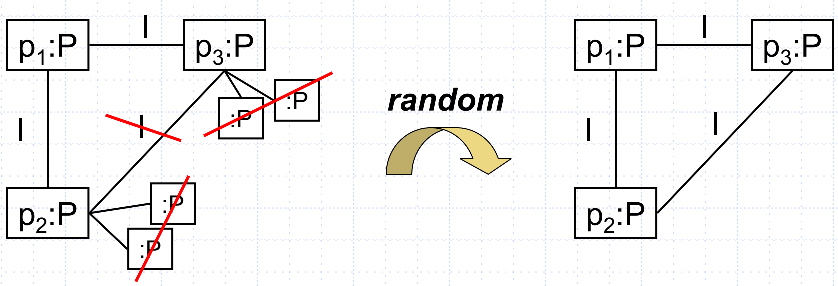
Graph Transformation System

G = ⟨ TG, P ⟩



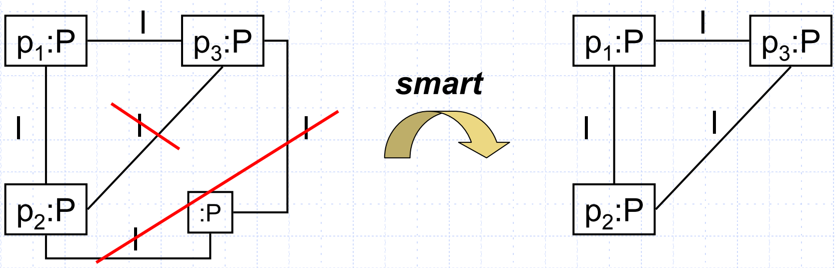
Which links should be added?

a) At random, up to a limit of **n = 3** links!



Which links should be added?

b) So that deletion of a node does not increase length of paths!



L. Mariani. Fault-tolerant routing for p2p systems with unstructured topology. Proc. International Symposium on Applications and the Internet (SAINT 2005), Trento, Italy.

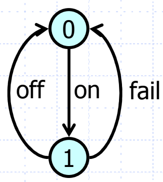
Outline

- ✓ Graph transformation in a nutshell
- ✖ Stochastic graph transformation
- ✖ Stochastic temporal logic
- ✖ Application
- ✖ Conclusion

Idea (what everyone does)

- generate labelled transition system
  LTS = ⟨ S, L, → ⟩
- associate labels with rates of exponential probability distribution **ρ: L → R_+**
- derive cont.-time Markov chain (CTMC)

read: 100 ×
(e.g. per day)



t	**ρ(t)**
on	100
off	100
fail	1

S × **S** matrix **Q**

	0	1
0	-100	100
1	101	-101

Generate Labelled Transition System

Assume
- **G = ⟨ TG, P ⟩**
- G_0 start graph, typed over **TG**

Labelled transition system ⟨ **S, L, →** ⟩
- Labels **L**: production (names) **P**
- States **S**: iso classes of graphs reachable from G_0
- Transitions: $[G] \rightarrow_p [H]$ iff $G \rightarrow_{p(o)} H$

Stochastic Graph Transformation Systems

Stochastic GTS **SG = ⟨ TG, P, ρ ⟩**

- GTS ⟨ **TG, P** ⟩
- mapping ρ**:** **P** → $\mathbf{R}_+$ associating production rate ρ**(p)**

$\mathbf{SG}_{\mathbf{random,\ x}}$

rule **p**	rate ρ**(p)**
new	1
kill	1
random	*x*
disconnected	0

$\mathbf{SG}_{\mathbf{smart,\ x}}$

rule **p**	rate ρ**(p)**
new	1
kill	1
smart	*x*
disconnected	0

Continuous Stochastic Logic

Temporal operators

- $\Phi\ U^{I}\ \Psi$: until Ψ, within interval **I**
- $X^{I}\ \Phi$: next Φ, within interval **I**

Stochastic "quantifiers"

- $S_{\Delta p}(\Phi)$: steady-state probability for Φ
- $P_{\Delta p}(\Phi)$: transient probability for Φ

Examples

- $\boldsymbol{P}_{>0.2}$ (*true* $\boldsymbol{U}^{[0,\,10]}$ *disconnected*):
  The probability of reaching a disconnected state within 10 units of time is below 0.2.
- $\boldsymbol{S}_{>0.1}$ (*disconnected*):
  The long-term probability of being in a disconnected state is below 0.1.
- $\boldsymbol{S}_{>?}$ (*disconnected*):
  What is the long-term probability of being in a disconnected state?

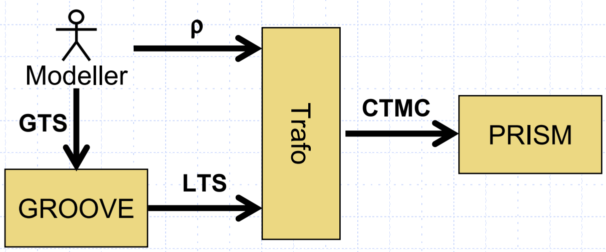
➔ applicability of rules yields interpretation of prop. vars

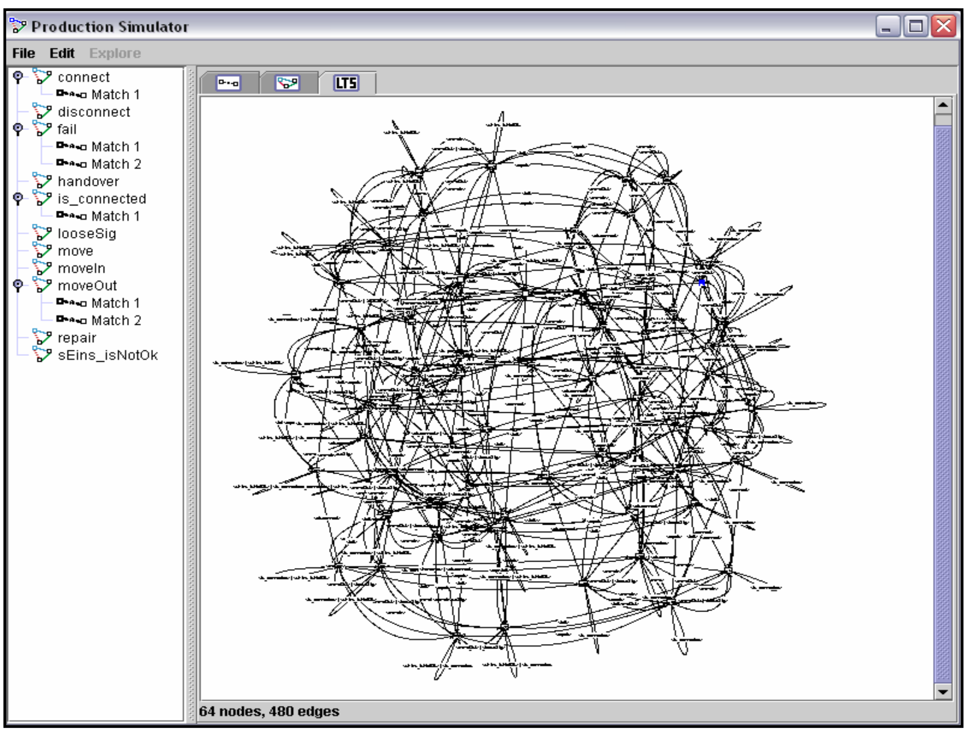
Outline

- ✓ Graph transformation in a nutshell
- ✓ Stochastic graph transformation
- ✓ Stochastic temporal logic
- ✖ Application
- ✖ Conclusion

Experimental Tool Chain

- input GTS and export to file
- define rates in separate file
- generate CTMC + interpretation of atomic propositions for stochastic model checker PRISM





Results

$S_{>?}$ (*disconnected*) for LTS generated from

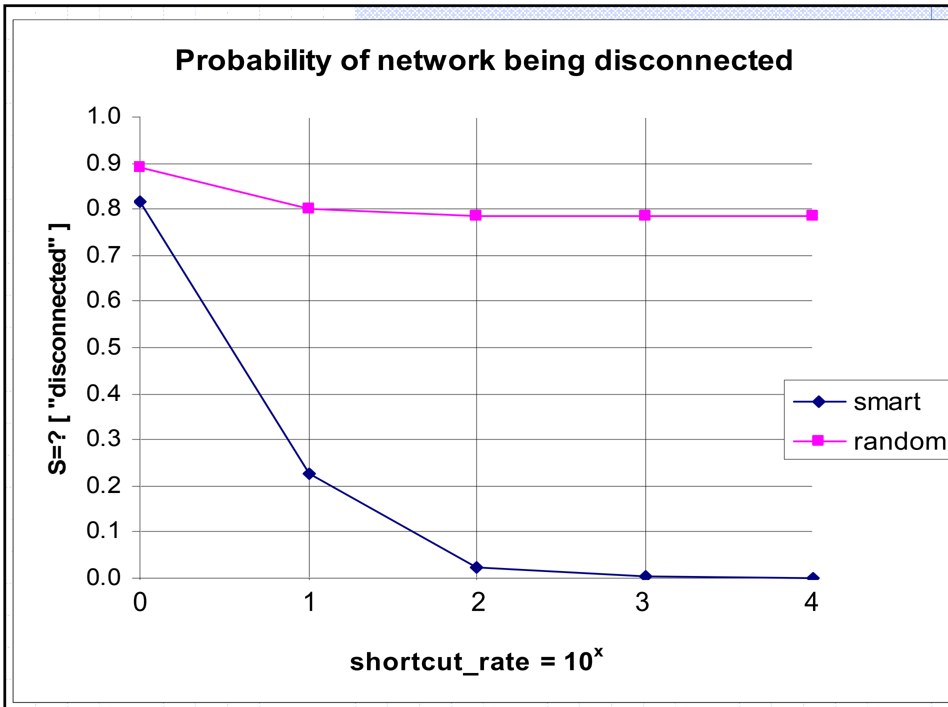
$SG_{random,\,x}$ $SG_{smart,\,x}$

for x = 1, ..., 10^4,

bounded to 7 peers and one registry.

➔ 798 states, 16293 transitions

➔ 487 states, 9593 transitions



Conclusion

Graph transformation

- intuitive diagrammatic formalism
- mathematical theory

Integration of GTS and stochastic time

- simple interface (LTS) supports / enforces separation of structural and stochastic aspect
- experimental tool support

Issues / future work:

- modularity of specification and analysis
- limitations of model checking: stochastic simulation
- other types of distributions: mapping to probabilistic RL