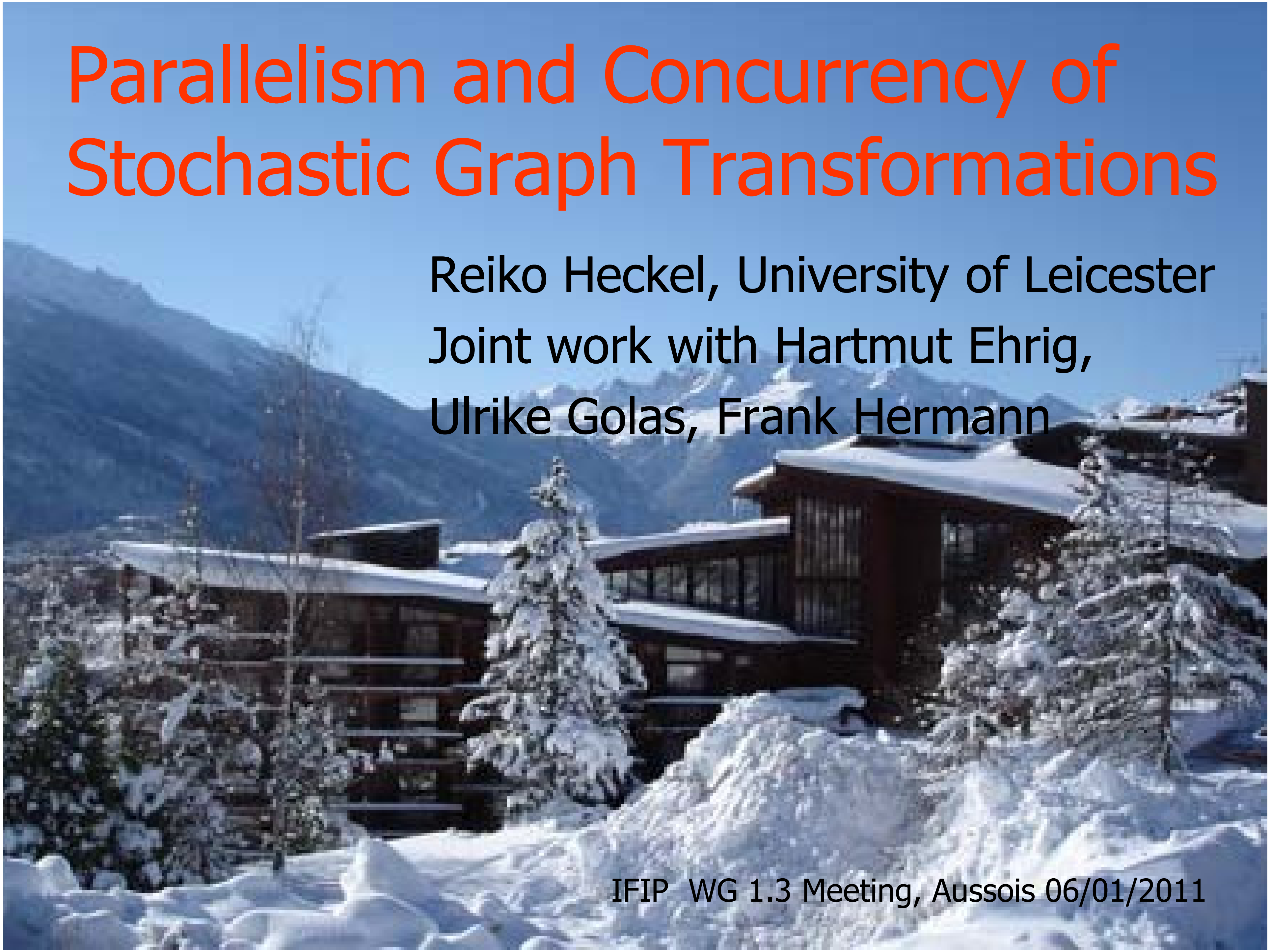
Parallelism and Concurrency of Stochastic Graph Transformations

Reiko Heckel, University of Leicester

Joint work with Hartmut Ehrig,

Ulrike Golas, Frank Hermann



IFIP WG 1.3 Meeting, Aussois 06/01/2011

Motivation

- Quantitative analysis of processes with dynamic reconfigurations, modelled as stochastic graph transformations

- Analysis through
  - Model checking: limited in scale
  - Stochastic simulation: statistical results only
  - *Symbolic calculation*

Question: What is the *distribution of completion times of a stochastic process* (deterministic, concurrent), given the *distribution of delays of its constituent steps*.

Basic Notions of Probability

Delays and durations are real-valued random variables, known to fall into certain intervals with probabilities described by distributions.

- $F : \mathbb{R}_+ \longrightarrow [0, 1]$ is distribution of real-valued random variable v if $F(x)$ is the probability for $v \leq x$.
- A value chosen randomly based on F is denoted $v = RN_F$.
- Sum and maximum of independent random variables correspond to operations on distributions, such as convolution

$$F_1 * F_2(u) = \int_{-\infty}^{\infty} F_1(x) F_2(u - x) dx$$

and product

$$F_1 \cdot F_2(x) = F_1(x) \cdot F_2(x)$$

Stochastic Graph Transformation Systems

SG = (TG, P, π, F)

- **TG** – type graph
- **P** – set of rule names
- **π(p): L → R** – rules typed over **TG**
- **F: P → ($\mathbb{R}$ → [0,1])** – distribution functions for delay

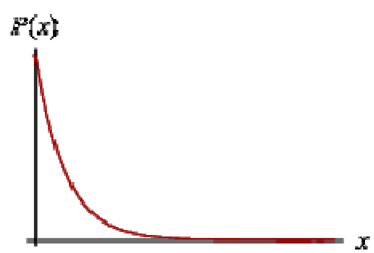
Exponential distribution

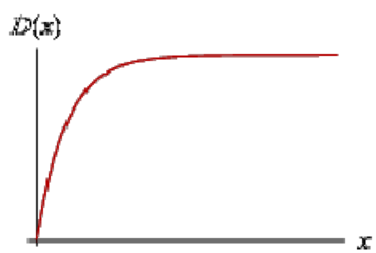
- given by rate λ

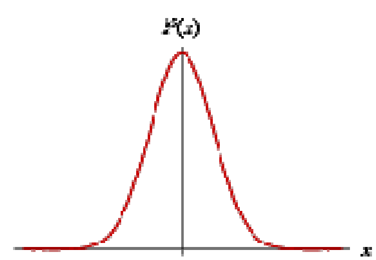
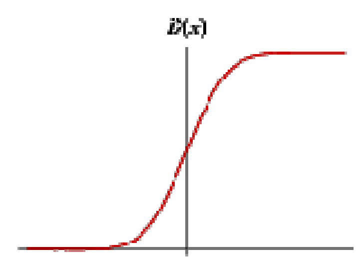
→ 1/λ avg. delay

Normal distribution

- given by mean and deviation





Generalised Semi-Markov Processes

- Transitions do not depend on *history prior to the current state*

$$
\begin{aligned}
\mathcal{P} \quad = \langle \quad & S : State\ set \\
& E : Event\ set \\
& \Gamma : State \longrightarrow \wp(Event\ set) \\
& \Sigma : State \times Event \longrightarrow State \\
& \Delta : Event \rightarrow (\mathbb{R} \rightarrow [1,0]) \\
& s_0 : State \qquad \rangle
\end{aligned}
$$

Timed Runs and Simulation

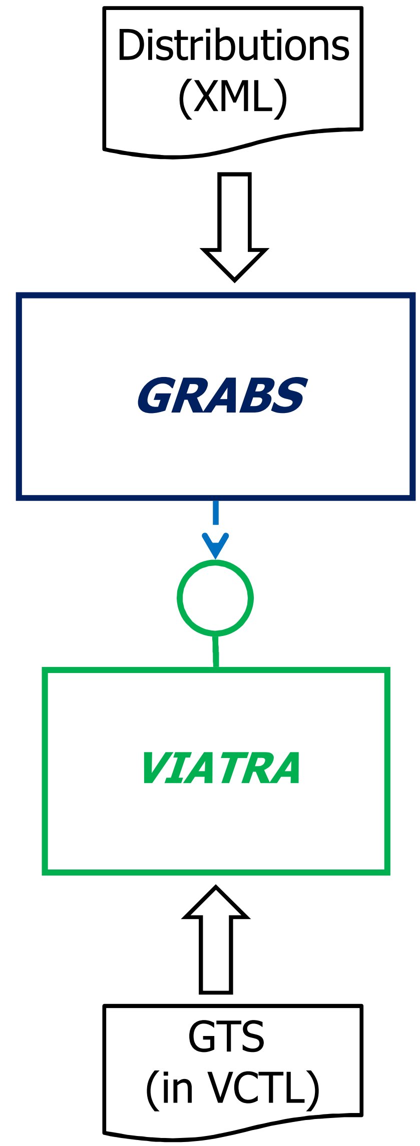
- **SG*** - sequences of transformations labelled by time stamps

  $$G_0 \rightarrow_{p1,t1} G_1 \rightarrow_{p2,t2} \ldots \rightarrow_{pn,tn} G_n$$

- Sampled by *stochastic simulation*

  Initialisation
  - compute all enabled events (rule, match)
  - for each event determine randomly (based on distributions) their delay

  Iteration
  - select next event and apply (rule, match)
  - update enabled events and remaining delays



Properties of Runs

For $r = (G_0 \overset{p_1,m_1,t_1}{\Longrightarrow} \cdots \overset{p_n,m_n,t_n}{\Longrightarrow} G_n)$, $ct(r) = t_n$ is *completion time.*

- $G_{i-1} \overset{p_i,m_i}{\Longrightarrow} G_i$ has application time $at(i) = t_i$, enabling time $et(i)$
- If m_i exists at the start of the run, $et(i) = 0$.
- Otherwise, if enabled by $G_{i-j} \overset{p_j,m_j}{\Longrightarrow} G_j$ with $j < i$, $et(i) = at(j)$
- $delay(i) = at(i) - et(i)$ is random variable with distribution $F(p_i)$

Run r *follows* sequence s if both contain the same steps (rules and matches) in the same order.

Completion time distribution of (runs following) sequence s assigns conditional probability for a run to complete within time $ctd(s)$ if the run follows s.

$$ctd(s)(x) = Prob\{ct(r) \leq x \mid \text{run } r \text{ follows } s\}$$

Completion Time Distribution

Proposition 1 (completion time distribution). *Assume a sequence* $s = (G_0 \overset{p_1,m_1}{\Longrightarrow} \cdots \overset{p_n,m_n}{\Longrightarrow} G_n)$ *in* $\mathcal{SG}$. *The set of critical steps* $CS(s) \subseteq \underline{n} = \{1, \ldots, n\}$ *of* s *is the smallest subset of indices of steps such that*

- $n \in CS(s)$
- *For all* $k \in CS(s), j \leq k$ *with* $j \rightsquigarrow k$, *such that for all* $j' \leq k$ *with* $j' \rightsquigarrow k$ *implies* $j \geq j'$, *also* $j \in CS(s)$

The completion time distribution of runs following s *is given by*

$$ctd(s) = *_{i \in CS(s)} F(p_i)$$

Lifting the Basic Theory of Graph Transformation

- Complex transformations via composed rules
  - Parallel transformation
  - Concurrent transformations
- General pattern
  - Define composition operation **#**
  - Define relation of sequences **(G#)*** using composed rules to basic sequences **G***
- Stochastic GTS
  - Lift definition of ctd from **G*** to **(G#)***
  - Fine delay distribution **F(c)** for composed rules **c** such that **F(c) = ctd(G** $\rightarrow_c$ **H)**

Series-parallel Productions

- Using + for disjoint and ; for dependent concurrent productions, series-parallel rule expressions are

$$c ::= p \mid c ;_{e_1, e_2} c \mid c + c$$

- Assignments π of productions and **F** of distribution functions are extended inductively

- Exact s-p productions, recursively
  - in **c ; d** all components in **d** depend on all components in **c**
  - in **c + d**, both **c** and **d** are s-p

Series-parallel Transformations

- If **d** is a step using an exact s-p production **c**, then **ctd(d) = F(c)**

- If **c** is not exact, then **F(c)** is an upper bound, i.e., for each deadline **t, F(c)** is less likely to meet it than **ctd(t)**

$$\mathbf{F(c)(t)} \leq \mathbf{ctd(d)(t)}$$

Proof: Induction on the structure of rule expressions, using parallelism and concurrency theorems as induction steps.

Conclusion

- ✓ Basic theory of algebraic GTS lifted
- ✓ Symbolic computation of completion time distribution for series-parallel processes

- ✖ Connections to be explored
  - Scheduling theory: *makespan* of a partially ordered network of tasks with stochastic durations
  - Concurrent semantics: deterministic graph processes, non-deterministic unfoldings

- ✖ Potential applications
  - Local optimisation of processes
  - Refinement of stochastic system