25 Years (more or less . . .)
of
Net Unfoldings
and
True-Concurrency Analysis Tools

Javier Esparza

Technische Universität München
Unfolding of a transition system
Nielsen, Plotkin, Winskel ’81: Petri nets can also be unfolded
Nielsen, Plotkin, Winskel ’81: Petri nets can also be unfolded

- Motivation: Denotational semantics of concurrent behaviour
  (extension of Scott’s domain of computable functions to concurrent computation)

- During the 80s, theory of unfoldings further developed by
  - Winskel (synchronization trees ’84, event structures ’86)
  - Engelfriet (branching processes ’91)
• A system composed of \( n \) independent components

\[
\text{\includegraphics[width=3cm]{diagram.png}}
\]

– has \( 2^n \) reachable states, but

– its unfolding is the system itself, and has size \( O(n) \)

• Question: Can we base verification on the unfolding?

• Obstacle: the unfolding is in most cases an infinite object!
• **Solution**: Construct a complete prefix of the unfolding containing all reachable states by identifying cut-off events.

![Diagram of a complete prefix of an unfolding containing all reachable states by identifying cut-off events.]

Cut-off events and complete prefixes
Cut-off events and complete prefixes

However, in the worst case McMillan’s complete prefix could be exponentially larger than the reachability graph!
Cut-off events and complete prefixes
E., Römer, Vogler ’96: Size-guarantee

- **Adequate orders**: orders on the events of the unfolding such that
  - if events added in this order, and
  - cut-offs identified as in McMillan’s approach
  then the prefix so constructed is complete.

- **Total adequate orders** guarantee that number of events never exceeds number of reachable markings.

- Problem of McMillan’s approach: His order was partial

- ERV ’96 found the first total adequate order; others followed (E., Römer ’99; Niebert, Qu ’06)
Extracting information from complete prefixes

- Complete prefixes are a “compact encoding” of the state space, but reachability information must be “extracted” from them.

- Heljanko and Khomenko (PhD theses, several papers): Reachability queries can be solved very efficiently using SAT / ILP.
• Further progress in SAT and SMT solving has turned the extraction problem into a non-issue.
From reachability to model-checking LTL

- Two unfolding-based algorithms to model-check arbitrary (next-free) LTL properties presented at ICALP ’00 (Couvreur, Grivet, Poitrenaud; E., Heljanko)

- The algorithm by E. and Heljanko is described in

  E., Heljanko: Unfoldings A Partial Order Approach to Model Checking
  Springer, 2008
2000-2010

Harry Potter and the Philosopher's Stone

Harry Potter and the Deathly Hallows: Part 2
Theory

- Parallel and distributed generation of the unfolding
  (Baldan, Haar, Heljanko, Khomenko, König, Koutny . . .)

- Even more compact representations: Merged processes
  (Khomenko, Koutny, Rodriguez, Schwoon, Vogler . . .)

- Extensions to more general models
  - Contextual nets (Baldan, Rodriguez, Schwoon, Vogler, Yakovlev . . .)
  - High-level nets (Khomenko, Koutny, Schöter . . .)
  - Timed models (Bouyer, Cassez, Chatain, Haddad, Jard . . .)
Tools

- PEP (Oldenburg, Best, Stehno, …)
- Mole (Schwoon)
- Unfolding Tools (Khomenko)
- unfsmodels, mcsmodels (Heljanko)
Applications

• Analysis of asynchronous circuits
  – Circuits specified as interpreted Petri nets
  – Concurrent Asynchronous Systems Group, University of Newcastle: tool-chain for verification and fault-fixing of STGs based on unfoldings (Khomenko, Koutny, Vogler, Yakovlev …)

• Monitoring and diagnosis
  – Distributed systems with alarms attached to some nodes
  – Problem: find cause of the alarms → true-concurrency approach
  – IRISA group in Rennes, MEXICO project at ENS Cachan: diagnosis tools (Benveniste, Chatain, Haar, Jard, Schwoon …)
Applications

- Verification of graph transformation systems
  - Unfolding used to overapproximate the set of reachable graphs
    (Baldan, Corradini, König, Kozioura . . .)

- AI Planning (Bonet, Haslum, Hickmott, Khomenko, Vogler, . . .)
Applications (2010-today)

- **Systems Biology**
  - Boolean networks used to model cellular regulatory processes
  - Unfoldings give compact representation of the reachable transitions
    (Pauleve, Chatain, Haar, Schwoon, ...)

- **Testing and verification of multithreaded programs**
  - Unfolding used to generate small set of test cases with high coverage (Heljanko, Kähkönen, Ponce de Leon, Saarikivi ...)
  - Unfolding used to guide partial-order reduction (Rodriguez, Sousa, Petrucci, Kröning ...)

- **Process discovery** (Carmona, Ponce de Leon, Rodriguez ...)

Conclusions

- Straight line from Petri’s nonsequential processes to concrete algorithms, tools, and application domains

- (Most?) successful spin-off of true-concurrency semantics

- Turning point: verification through algorithmic construction of semantic objects

- True-concurrency useful in two ways:
  - Compact representation of state spaces
  - Information about causality and independence

- Blockchain?