Concurrent Logic Programming

Met and Unmet Promises

Iliano Cervesato and Edmund S.L Lam

We are grateful to Paul Fodor for agreeing to give this talk in our absence

Logic Programming …

… the quintessential declarative paradigm

- Declarative? Promises to
  - promote human-friendly description of a problem
    - » as opposed to hardware-oriented encoding of solution
    - » aka *abstraction*
  - simplify reasoning
    - » Strengthens assurance, security, performance

After 50 years …

- Making it easier to write programs?
  - Definitely, in some domains
    - » Datalog renaissance of the last 5 years
    - » CLP, tabling, …
  - By and large, (really) hard to write large programs
    - » Extra-logical constructs, flat name space, …
- Simplifying reasoning?
  - Largely unmet promise

… logic programming is a fringe paradigm

A new challenge

Concurrent and distributed applications

- Popping up everywhere
  - Mobile apps, Internet-of-things, cloud applications
- Really hard to get right
  - Communication challenges
    - Consistent messaging, available sender/receiver, …
    - … across multiple communicating programs
  - Synchronization challenges
    - Deadlock, live locks, unwanted race conditions, …
- APIs available to novice programmers …

… a great *opportunity* for LP

- No competition from traditional paradigms
  - (yet)
- Simple, abstract logical specifications of communication and synchronization
  - Forward-chaining AKA logic-based rewriting
- Nascent reasoning and assurance support
  - Straight from proof theory

Writing a distributed application …

Node-centric way

- 1 program for each device
- Peephole view of messaging
- No support to handle messaging/synchronization
  - Programmer on his/her own
- Error-prone and costly

How most distributed software is written

System-centric way

- A single program
- Bird eye's view of messaging
- Centralized analysis
- Automatically compiled to code that runs on each device
- Simple, fast, abstract

…

(opportunity knocking)

Comingle

A language for mobile distributed applications

- Not your usual multiset rewriting language …
  - Implements a fragment of first-order linear logic
    - » with locations, strong typing, multiset comprehensions
    - » interfaces with local computation (Android SDK)
  - Forward-chaining semantics (high-level)
  - Distributed stack-based machines (low-level)
- Implementation for Android devices and i386
- A dozen applications
  - A day each to implement, some by undergrads

Opportunities

- Other logic-based forays
  - Meld: language for programming shared-memory multicore systems
  - Netlog: language for P2P applications
  - Yedalog: Datalog for the cloud
- What is missing
  - Scale
  - Assurance

Reasoning about concurrent apps

Several promising logic-based techniques

- Session types
  - Statically catch messaging errors and deadlocks
  - Logical foundation of process algebra
- Coinductive methods
  - Concurrent programs don't return a result
    - » Progress through interaction
    - » Termination is unimportant
  - Bridge to process algebraic methods
    - » (Bi)-simulation, refinement, equivalence

Take-home message

- Mobile, concurrent, distributed applications are in need of a good programming model
- Logic programming can be that model
  - System-centric programming
  - Untapped reasoning potential
- First initial attempts are promising