a high-level Petri nets library

Franck Pommereau

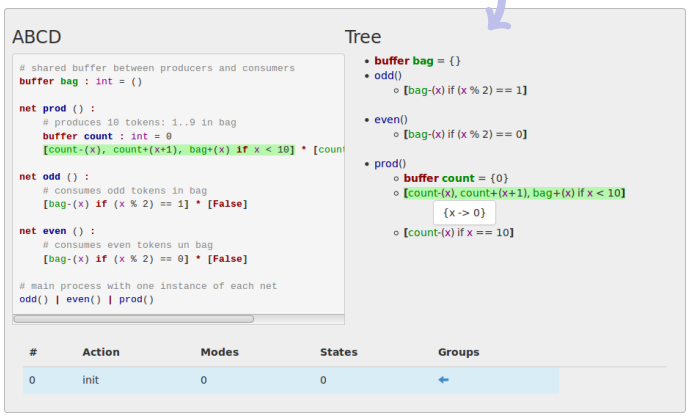
IBISC/COSMO, university of Évry/Paris-Saclay

franck.pommereau@ibisc.univ-evry.fr
snakes.ibisc.univ-evry.fr

- define and handle Petri nets
- very general definition
- numerous extensions
  read/inhibitor/whole-place arcs, ...
- annotations = Python expressions
- tokens = Python objects
- nets can be executed (fire transitions)

- extensible with plugins
- several plugins provided
- easy to add new ones
  time Petri nets $\leq$ 100 LoC
  nets-within-nets $\leq$ 30 LoC

- user-friendly modelling language
- algebra of coloured Petri nets
- Python-inspired syntax
- seamless integration of Python
- web-based interactive simulator



- pure Python library (works out of the box)
- free software (GNU LGPL)
- 81.5k lines of code
- maintained for 12+ years
- online doc $\Rightarrow$ 300+ unique visits / month

- utilities

core library

nets
Petri nets, places, transitions, arcs, markings, marking graphs, ...

simul
interactive simulation

plugins

ops
PBC/PNA & M-nets compositions

sync
transitions synchronisation

gv
drawing with GraphViz

pids
dynamic process spawn/kill

utilities

abcd
compiler/simulator for the Asynchronous Box Calculus with Data

python powered

included in SNAKES
not in SNAKES anymore

external tools

neco
net compiler, state-space computation & LTL model-checking

- Łukasz Fronc's companion tool
  github.com/Lvyn/neco-net-compiler
- compiles nets into fast native code
  per-net optimised marking structure
  per-transition optimised firing
  cannot optimise arbitrary Python code
- process-symmetries reductions (plugin pids)
- state space & LTL model-checking (using SPOT)
- awarded at the model-checking contest 2013

SNAKES out of Python

- write a binding in Cython

```
# this is Cython code = Python + C types
cdef public int foo (...):
    # Python code using SNAKES goes here
```

- Cython compiles to C/C++ with a `.h` file:

```
// this is C/C++ code
extern int foo(...);
```

- use this API in your project

SNAKES' future

- development name: ZINC
- Python 3 only (drop Python 2 support)
- net compilation at its heart *à la* Neco
- "any language"-coloured Petri nets
  currently: Python, Go, CoffeeScript/JS
- even more general Petri nets definition
- cleaner, lighter, more modern