# Automatic Decomposition of Petri Nets into Automata Networks – A Synthetic Account

**Pierre Bouvier**[1] **Hubert Garavel**[1]

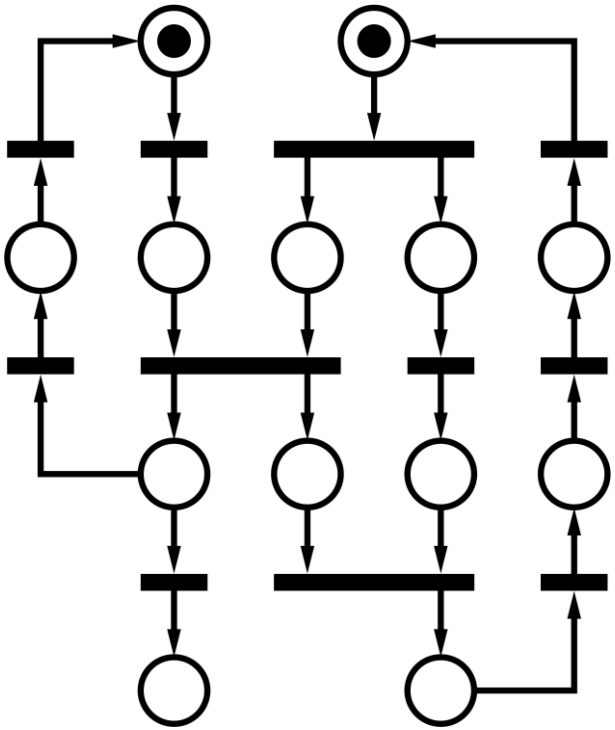
**Hernán Ponce de León**[2]

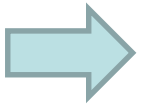
[1]Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, France

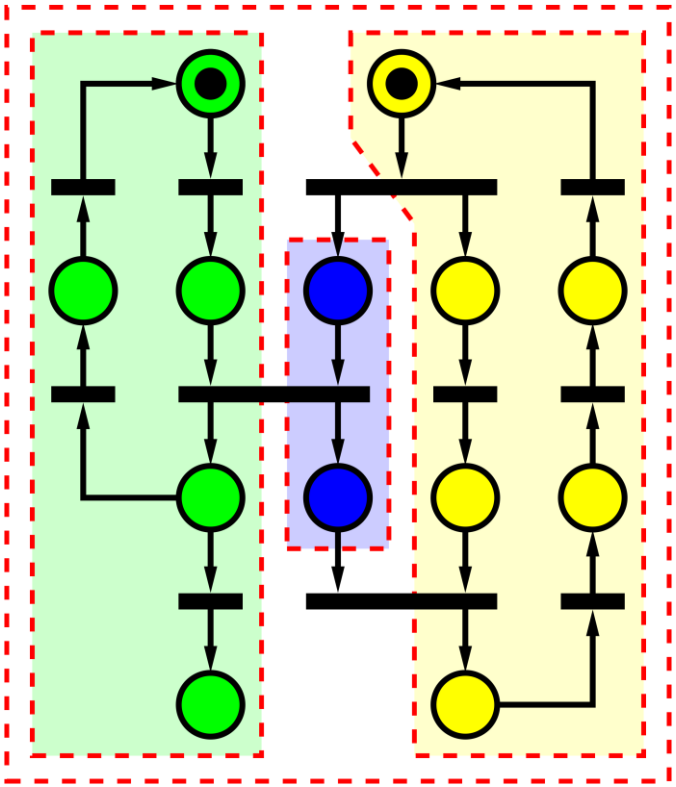
[2]Research Institute CODE, Bundeswehr University, Munich, Germany


# Problem statement

Decomposition of a Petri net into an automata network





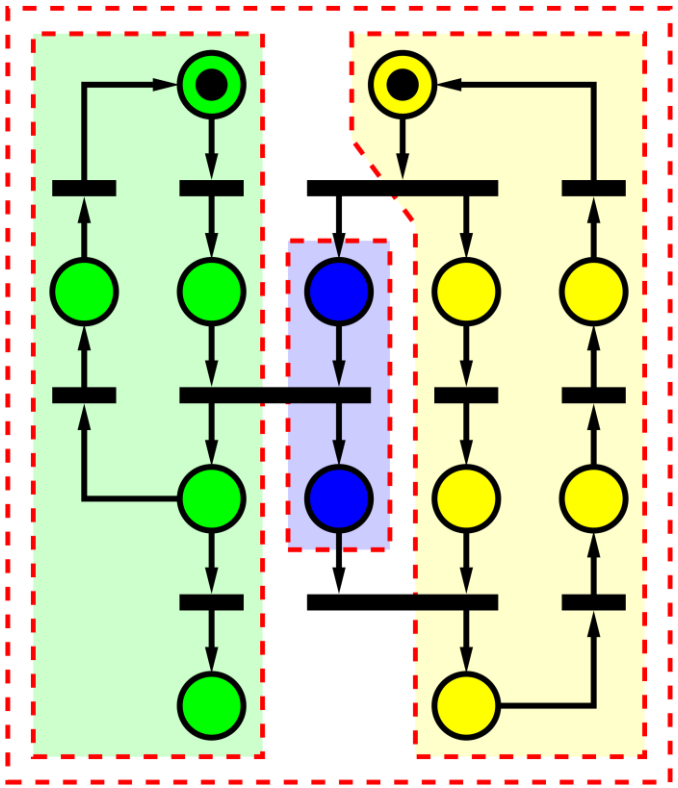


Input: ordinary, safe Petri net

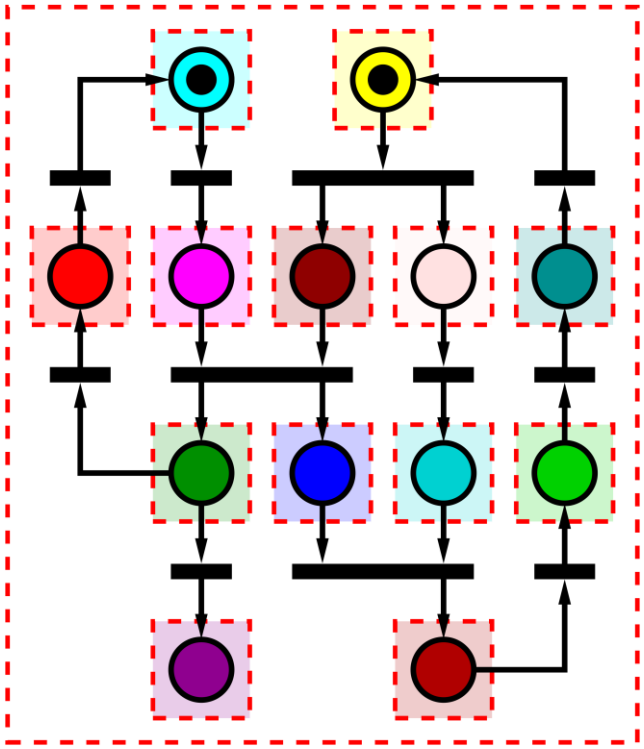
Output: automata network (flat, unit-safe NUPN)

# Output Model

- automata network = NUPN of height 1 (i.e., no hierarchy)
- each automaton is a NUPN unit
- the input net is kept unchanged: no insertion or deletion of places, transitions, or arcs
- Petri-net markings and firing rules are kept unchanged
- no shared place between units
- each unit has at most one token (it can gain or loose its token)



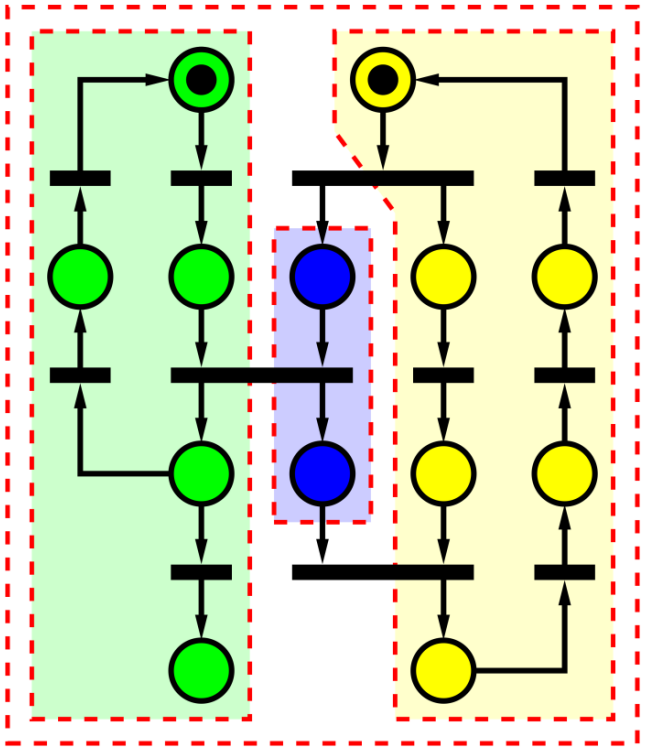
# Existence / multiplicity of solutions



- Always at least one solution:
  - one unit per place (“trivial” NUPN )
  - not so useful in practice
- In general, many solutions

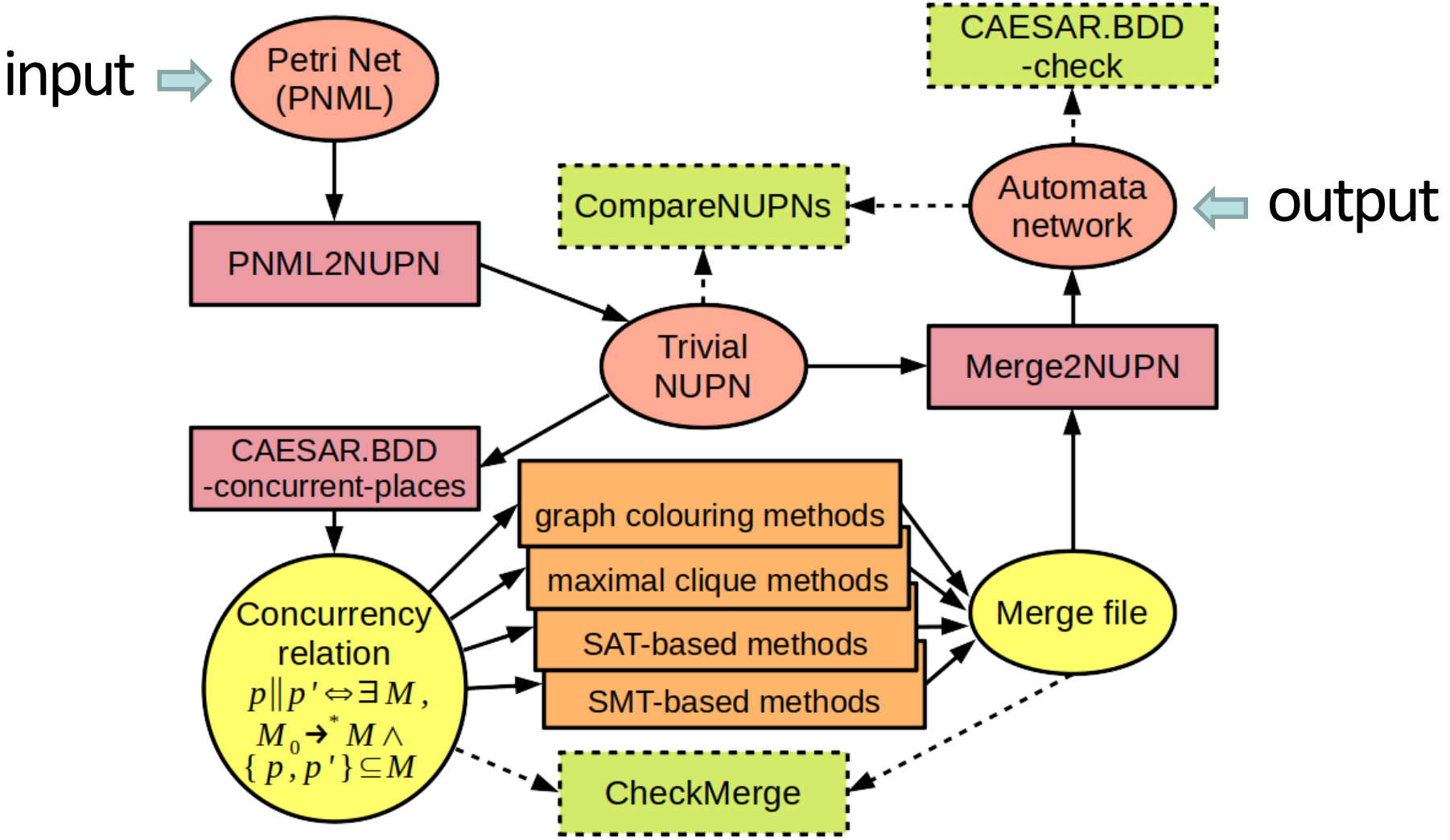
13 units
13 bits/marking

3 units
8 bits/marking



- Criterion: reduce the number of bits to encode reachable markings
  - larger units ⇒ fewer bits

# A complete tool chain




22 methods (12,000 lines of Awk, C, Python, shell)

# Experimental results

- A diverse collection of 12,728 Petri nets:
  - academic, industrial, competition (MCC and RERS)
  - in average: 220 places, 9,400 transitions and 74,600 arcs
- Our 22 methods:
  - can handle 88% … 99% of this collection
  - reduce by 34% … 42% the bits to encode reachable markings
- Assessments on the 223 Hippo benchmarks (Zielona Góra)
  - Hippo can process 92.5% of models in 25 minutes
  - our tool chain can process 100% in 16 seconds and provides better reductions (39.4%)