The State Explosion Problem

Lecture #5a of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling and Verification

E-mail: `katoen@cs.rwth-aachen.de`

November 4, 2008

The state explosion problem

- Time-complexity of model-checking algorithms
  - depends on the property to be checked
  - and on the size of the transition system
  - . . . . . . that models the system to be checked

- Size of a transition system
  - $|TS| \;=\; |S| + |\rightarrow|$

- The size of transition systems underlying
  - program graphs is exponential in number of program variables
  - concurrent systems is exponential in number of components
  - channel systems is exponential in number of channels

Sequential programs

- The $\#$ states of a program graph is:

$$\Big| \#\text{program locations} \Big| \cdot \prod_{variable\ x} |\, dom(x)\, |$$

$\Rightarrow$ number of states grows *exponentially* in the number of program variables

- N variables with k possible values each yields k^N states
- this is called *the state explosion problem*

- A program with 10 locations, 3 bools, 5 integers (in range $0 \ldots 9$):

$$10 \cdot 2^3 \cdot 10^5 = 800,000 \text{ states}$$

- Adding a single 50-positions bit-array yields $800,000 \cdot 2^{50}$ states

Concurrent programs

- The $\#$ states of $P \equiv P_1 \parallel \ \ldots \parallel P_n$ is maximally:

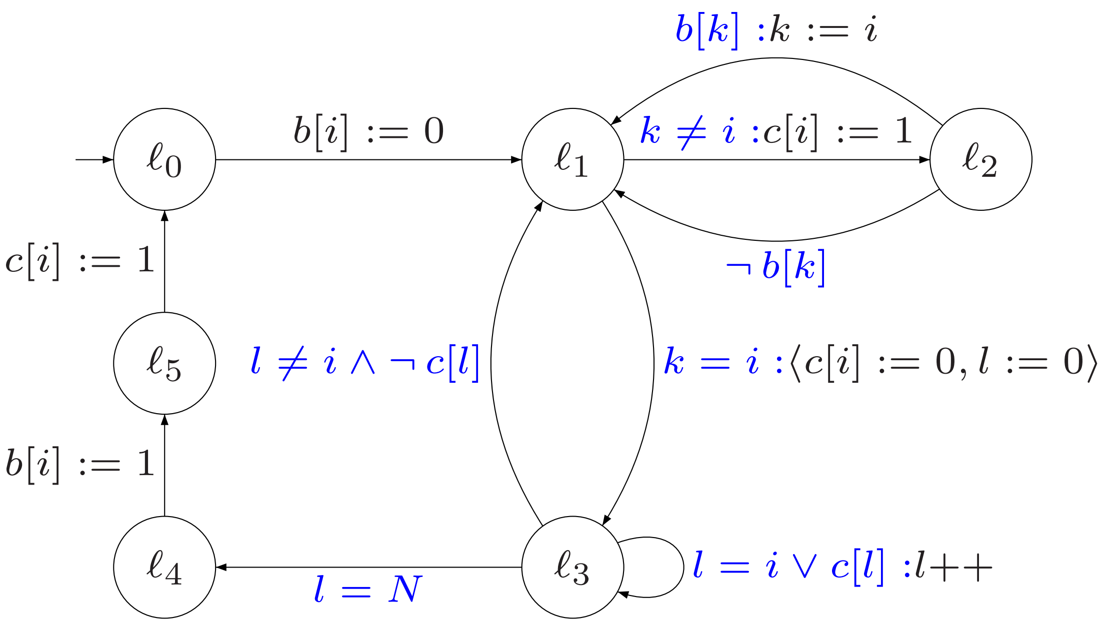
$$\#\text{states of } P_1 \times \ldots \times \#\text{states of } P_n$$

$\Rightarrow$ $\#$ states grows *exponentially* with the number of components

- The composition of N components of size k each yields k^N states

- This is called *the state-space explosion problem*

Dijkstra's mutual exclusion program



- two bit-arrays of size N
- global variable k
  - with value in $1, \ldots, N$
- local variable l
  - with value in $1, \ldots, N$
- 6 program locations per process

$\Rightarrow$ totally $2^{2N} \cdot N \cdot (6N)^N$ states

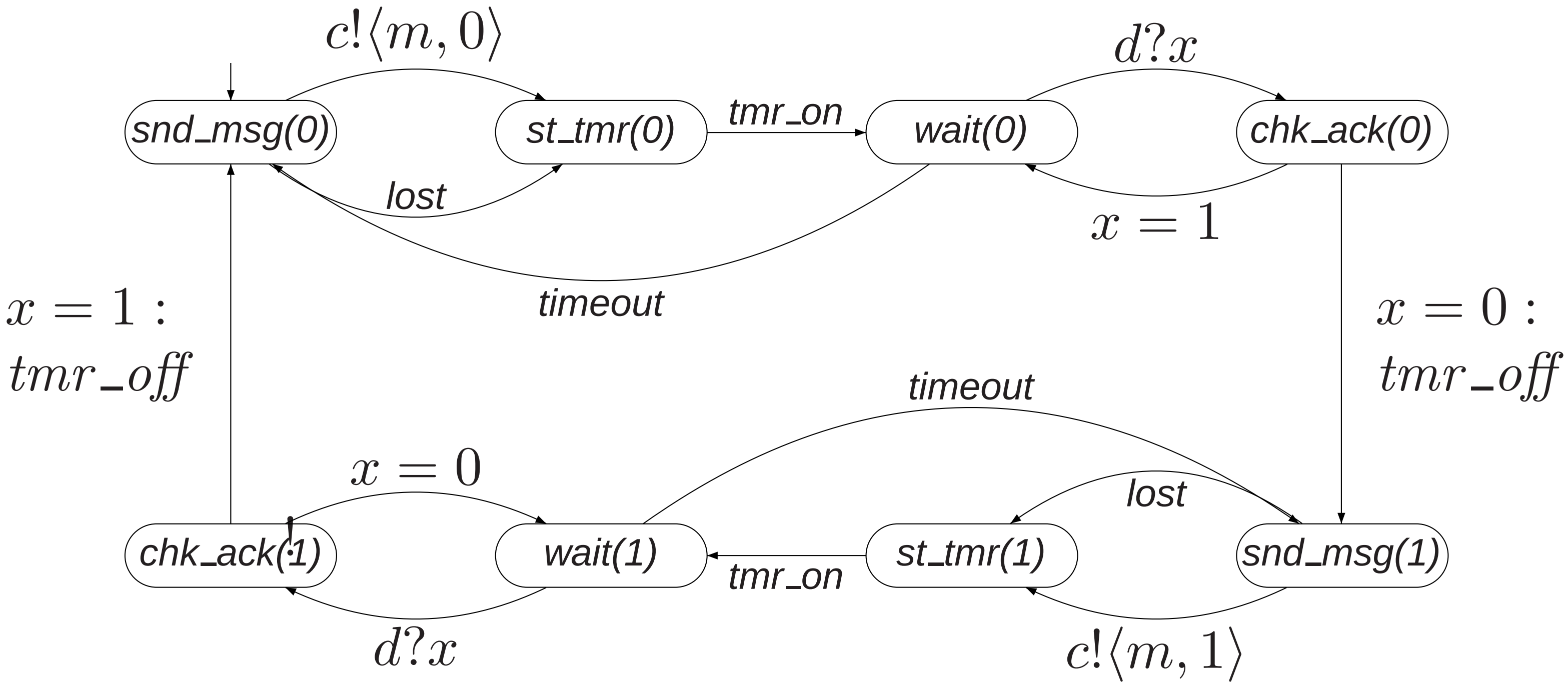
Channel systems

- Asynchronous communication of processes via *channels*
  - each channel c has a bounded capacity $cap(c)$
  - if a channel has capacity 0, we obtain handshaking
- $\#$ states of system with N components and K channels is maximally:

$$\prod_{i=1}^{N} \left(|\#\text{program locations}| \prod_{variable\ x} |\, dom(x)| \right) \cdot \prod_{j=1}^{K} |dom(c_j)|^{cap(c_j)}$$

this is the underlying structure of Promela

The alternating bit protocol



channel capacity 10, and datums are bits, yields $2{\cdot}8{\cdot}6{\cdot}4^{10}{\cdot}2^{10} = 3{\cdot}2^{35} \approx 10^{11}$ states

Summary of Chapter 2

- Transition systems
  - are a fundamental model for modeling software and hardware systems
- Executions
  - are alternating sequences of states and actions that cannot be prolonged
- Interleaving
  - execution of independent concurrent processes by nondeterminism
- Shared variables
  - parallel composition on transition systems is not adequate
  - instead, parallel composition of program graphs is used

Summary of Chapter 2

- Handshaking on a set H of actions
  - execute actions in H simultaneously and those not in H autonomously
- Channel systems = program graphs + FIFO communication channels
  - handshaking (cap = 0) or asynchronous communication (cap ¿ 0)
  - semantical model of `nanoPromela` modeling language
- State explosion problem
  - size of transition system is exponential in number of variables, concurrent components, and channels