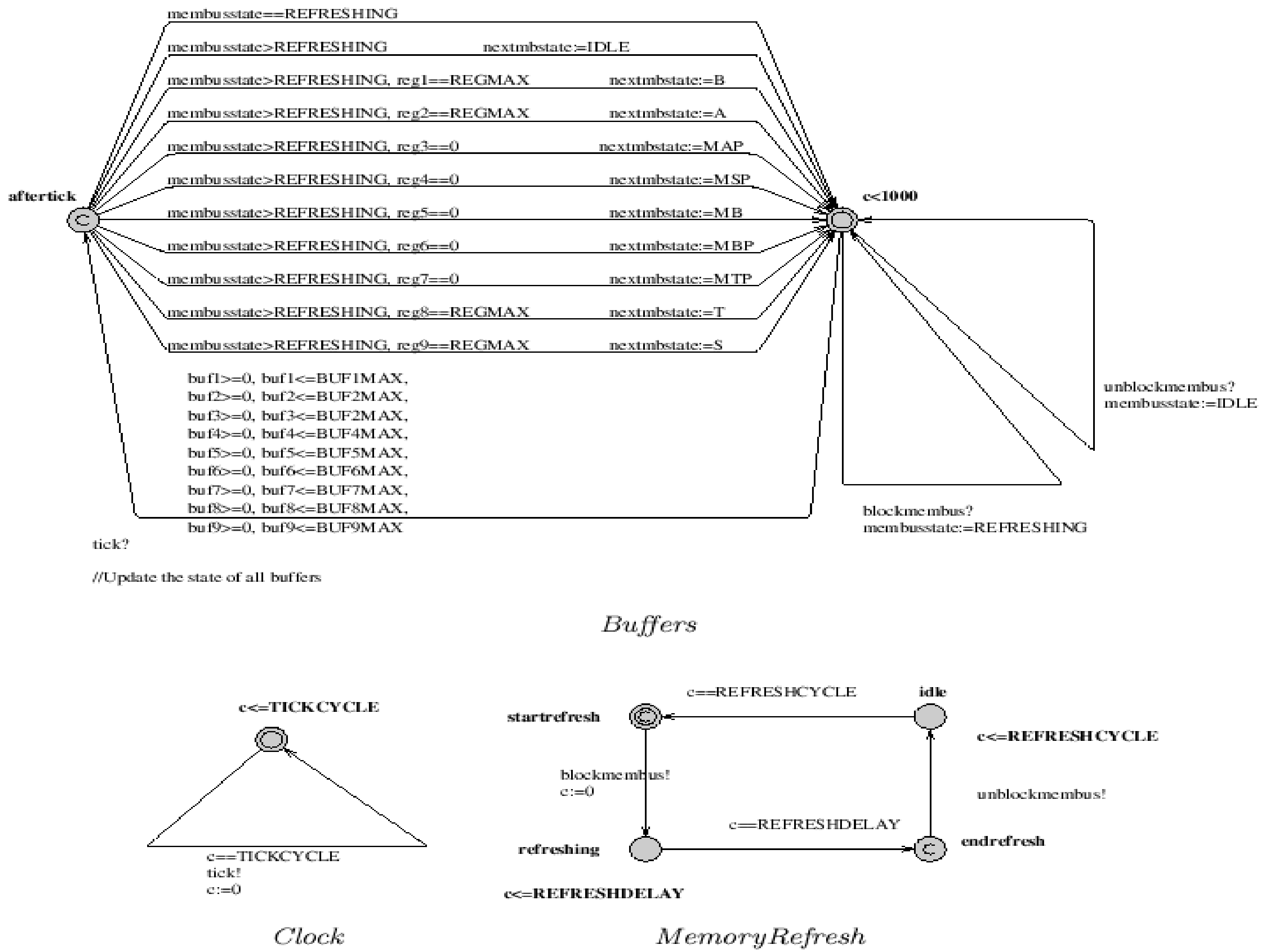
Bit-state hashing on steroids
or
Speeding up model checking by hash table size sweep

Juhan Ernits
TSEM 01.12.2005

What do we want to do?

- We want to check for reachability on a structure representing a constraint system.
- (this is equivalent to) We want to check if the behaviour of the model is included in the behaviours of the specification

Example


membusstate==REFRESHING
membusstate>REFRESHING
nextmbstate:=IDLE
membusstate>REFRESHING, reg1==REGMAX
nextmbstate:=B
membusstate>REFRESHING, reg2==REGMAX
nextmbstate:=A
membusstate>REFRESHING, reg3==0
nextmbstate:=MAP
membusstate>REFRESHING, reg4==0
nextmbstate:=MSP
membusstate>REFRESHING, reg5==0
nextmbstate:=MB
membusstate>REFRESHING, reg6==0
nextmbstate:=MBP
membusstate>REFRESHING, reg7==0
nextmbstate:=MTP
membusstate>REFRESHING, reg8==REGMAX
nextmbstate:=T
membusstate>REFRESHING, reg9==REGMAX
nextmbstate:=S
aftertick
c<1000
buf1>=0, buf1<=BUF1MAX,
buf2>=0, buf2<=BUF2MAX,
buf3>=0, buf3<=BUF2MAX,
buf4>=0, buf4<=BUF4MAX,
buf5>=0, buf5<=BUF5MAX,
buf6>=0, buf6<=BUF6MAX,
buf7>=0, buf7<=BUF7MAX,
buf8>=0, buf8<=BUF8MAX,
buf9>=0, buf9<=BUF9MAX
tick?
//Update the state of all buffers
unblockmembus?
membusstate:=IDLE
blockmembus?
membusstate:=REFRESHING
Buffers
c<=TICKCYCLE
c==TICKCYCLE
tick!
c:=0
Clock
c==REFRESHCYCLE
idle
startrefresh
c<=REFRESHCYCLE
blockmembus!
c:=0
unblockmembus!
c==REFRESHDELAY
refreshing
endrefresh
c<=REFRESHDELAY
MemoryRefresh

Explicit state model checking

- We consider explicit state model checking.
  - all control states and data states are represented explicitly.
- As opposed to symbolic model checking
  - where the states are represented by some symbolic construct, for example BDD-s.

Ways of reducing memory

- Partial order reduction
- Lossless state compression
  - Collapse compression
  - Minimized automaton representation
- Lossy state compression
  - bit-state hashing
  - hash compaction

Collapse compression

- The state explosion is due to small changes in many places
- Store different parts of the state space in separate descriptors and represent the actual state as an index to relevant state descriptors

Minimized automaton representation

- Build a recognizer automaton for states. All states that have been seen lead to an accepting state.
- The recognizer automaton is interrogated on each step of the model checker.
- The recognizer automaton is modified each time a new state is seen.

What is hash compaction

- A method where each state is represented by a hash (for example 128 bits). This is stored in a regular hash table.
- Used in Spin, Zing, Bogor, ...
- Can achieve very good coverage.

Bit-state hashing

- Let us look at how a hash table works.
- Instead of a state, store one bit.

Hash functions

- mod sucks!
- Look at Jenkins' hash funcion:

```
// Most hashes can be modeled
// like this:

  initialize(internal state)
  for (each text block)
  {
    combine(internal state, text block);
    mix(internal state);
  }
  return postprocess(internal state);
```

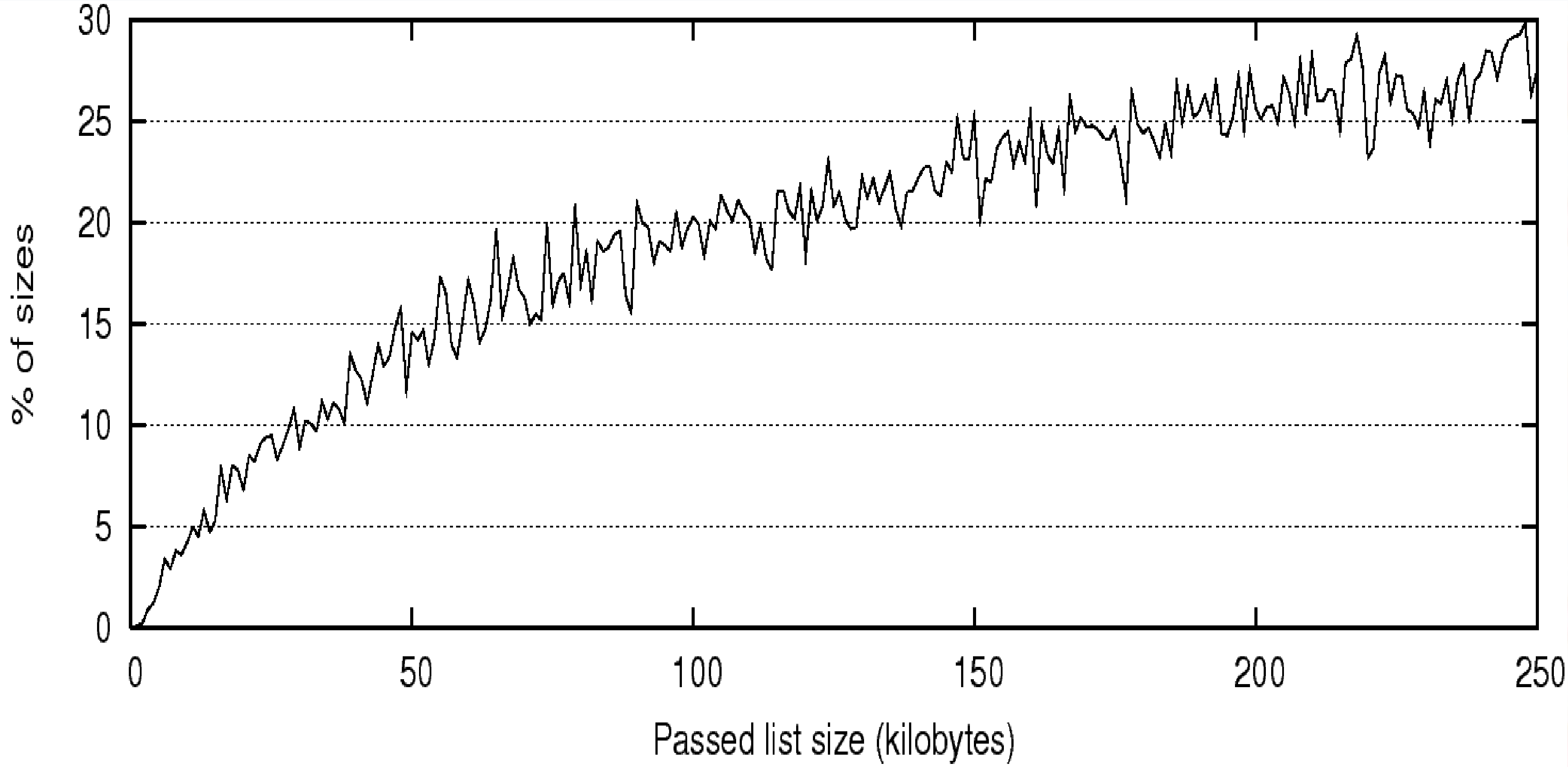
Hash functions 2

- Hash functions are well researched to be as pseudorandom as possible.
- Can we do better?
- Can we encode some relevant simple abstraction function into the hash function?

Hash table size sweep

- Start with a really small hash table size and modify the size of the table while keeping the hash function constant.
- Works well for synthesis tasks
  - task failed with exceeding 3 GB of mem in the explicit case;
  - worked with 100 MB of memory with bit state hashing enabled,
  - but

Hash table size sweep



- Percentage of queries yielding a trace to the desired state (not “may be”).

Hardware vs software checking

- Hardware in general has a lot of control states and relatively few data variables
- Software has loooots of data and weird constructs like threads, dynamic creation of objects, garbage collection ...
- One has to be really careful when attemting to use bit-state hashing for software.

Ideas

- By modifying the size of the hash table we got an answer to the query in seconds and by using a few kilobytes for the hash table.
- The cache memory of modern processors is 1-2 MB. This should make such sweep really fast.

Help needed!!!

- To write an extension to Bogor (remember John Hatcliff?)
- Experiment with hash table size sweep on BIR examples.
- Put it all into a paper and produce a (preferably ISISISI) publication.