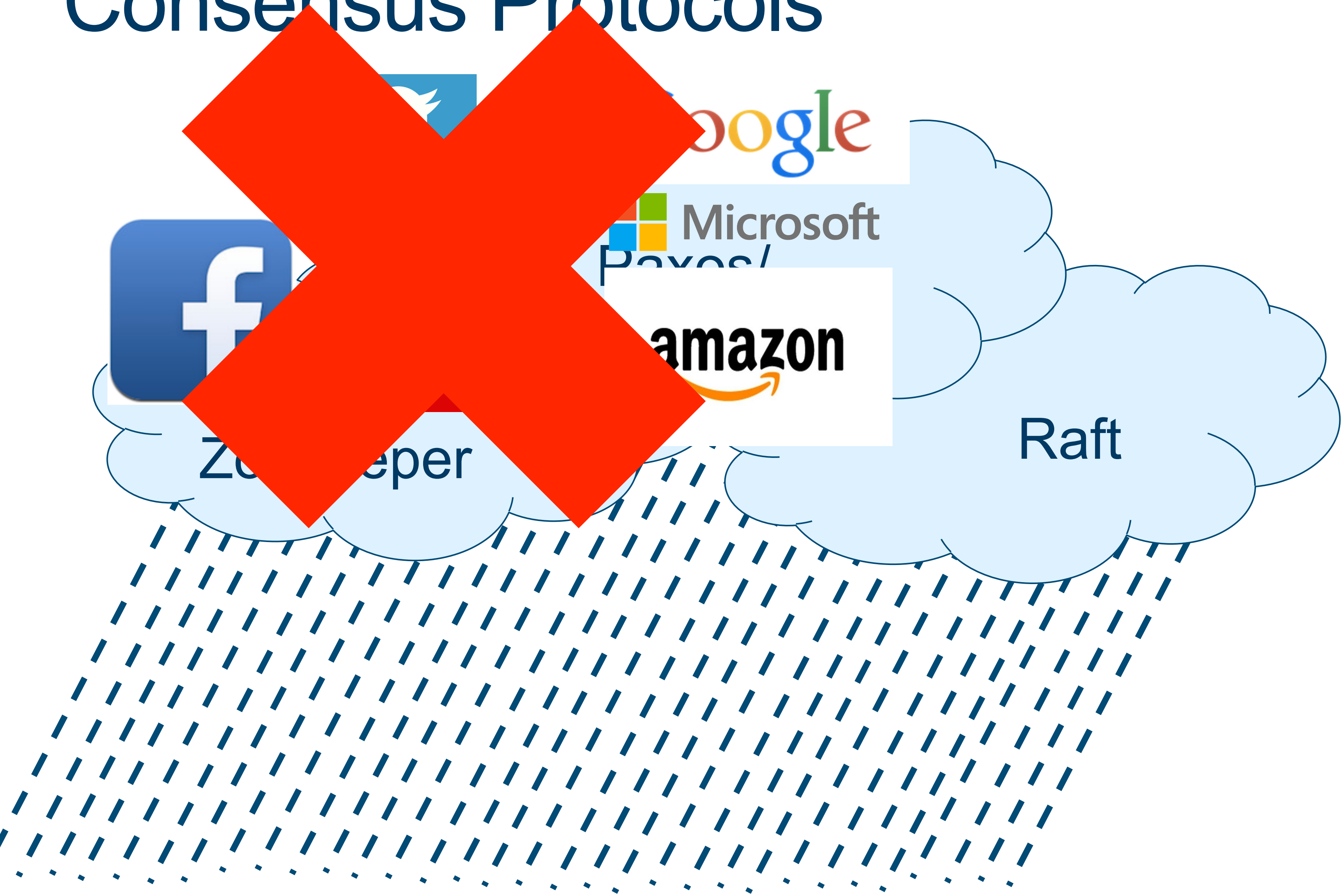
Synthesizing Cardinality Invariants for Parameterized Systems

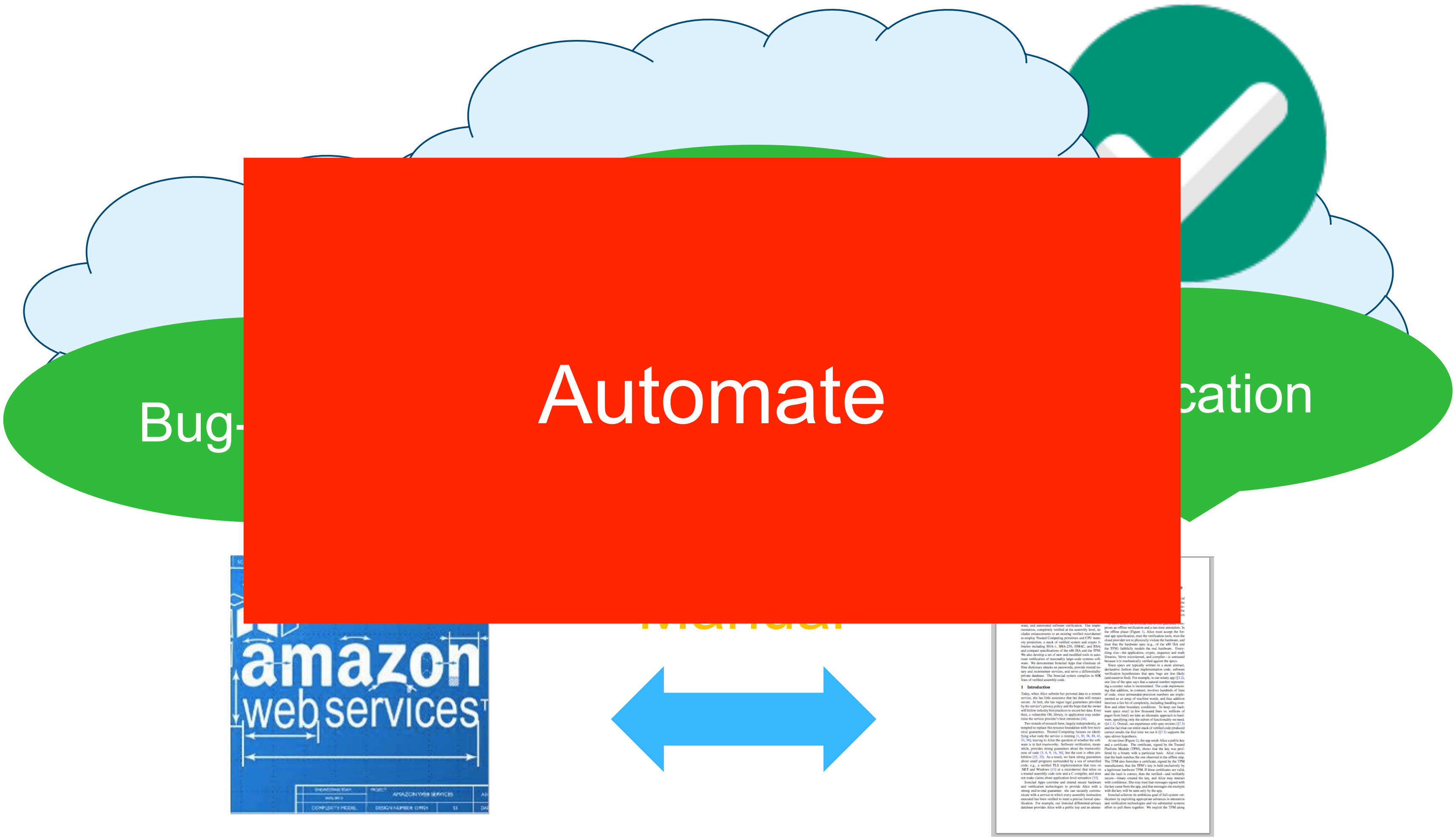
Klaus v. Gleissenthall, TU Munich

Nikolaj Bjørner and Andrey Rybalchenko,
Microsoft Research

Consensus Protocols



Verification efforts:



How AWS uses formal methods.
CACM’15

Fleet: Proving Practical Distributed Systems Correct.
SOSP'15

We need cardinalities

If the proposer receives the requested responses from a majority

Paxos
(ACM Sigact News '01)

Cardinalities in the description

if $(sCount_i \geq f+1)$ **then** $sFlag_i = true$
else $sFlag_i = false$

ASAP
(DISC'08)

26: T_p^r :
27: **if** $p = Coord(p, \phi)$ **and** number of $\langle ack \rangle$ received $> n/2$ **then**
28: $ready_p$:= true

Last Voting
(Distr. Computing'09)

We need cardinalities

Cardinalities in
the proof

Lemma 2.4.1 … $- j$ threads at
level j.

We need to reason about
cardinality.

A (very) simple example:

If there is a thread at location 2, then a>0

```
      global int a=0;
1:    a++;
2:
```

Example: in logic

Local variables as functions

primed = after transition

initial states:	$\forall t{:}\ pc(t)=1 \wedge a=0$
transition relation	$pc(me)=1 \wedge pc' := pc(me \leftarrow 2) \wedge a'=a+1$
safety	$pc(me)=2 \rightarrow a>0$

Example: constraints

$\exists$ inv

$\forall t\colon pc(t)=1 \wedge a=0 \quad \rightarrow \quad inv(a,pc)$

$inv(a,pc) \wedge \begin{array}{c} pc(me)=1 \wedge \\ pc' := pc(me \leftarrow 2) \wedge a'=a+1 \end{array} \rightarrow inv(a',pc')$

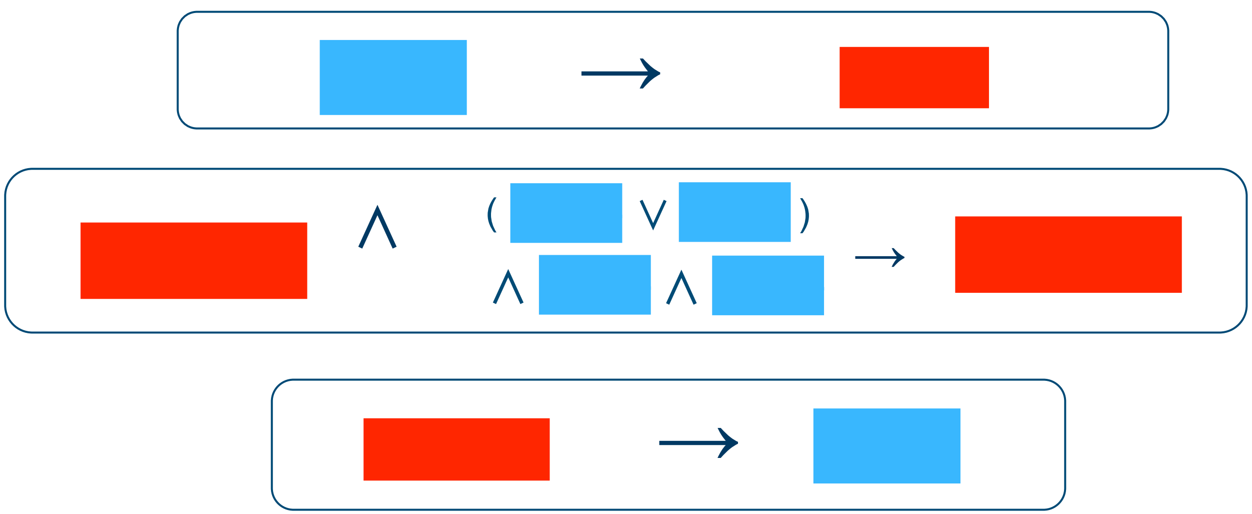
$inv(a,pc) \quad \rightarrow \quad pc(me)=2 \rightarrow a>0$

Restricting the search space:

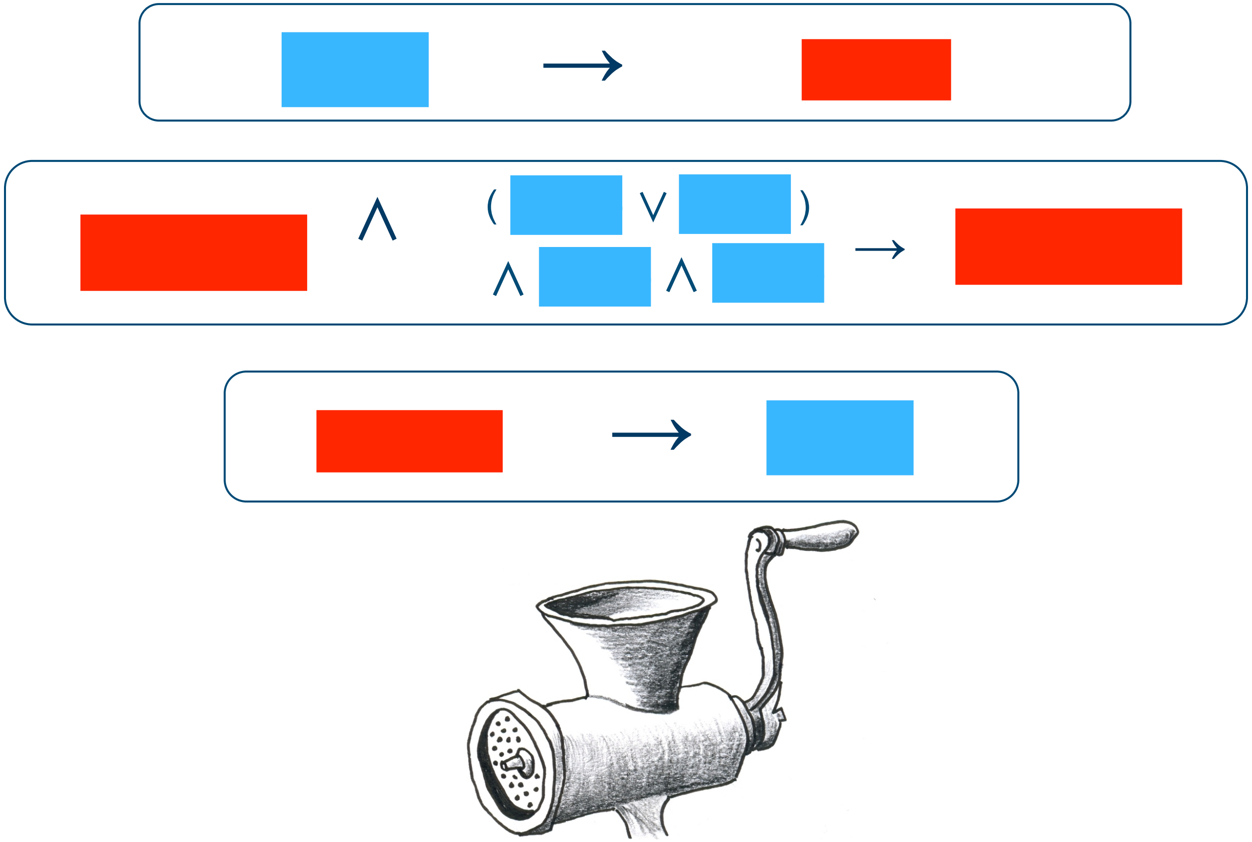
Predicate s describes set

cardinality free

∃ # inv ∧ inv(a,pc,k)



Solving:



$\#\{t \mid pc(t)>1\} \leq a$

Example: checking the solution

Set is empty

$\forall t: pc(t)=1 \wedge a=0 \;\rightarrow\; 0 \leq 0$

Set is non-empty

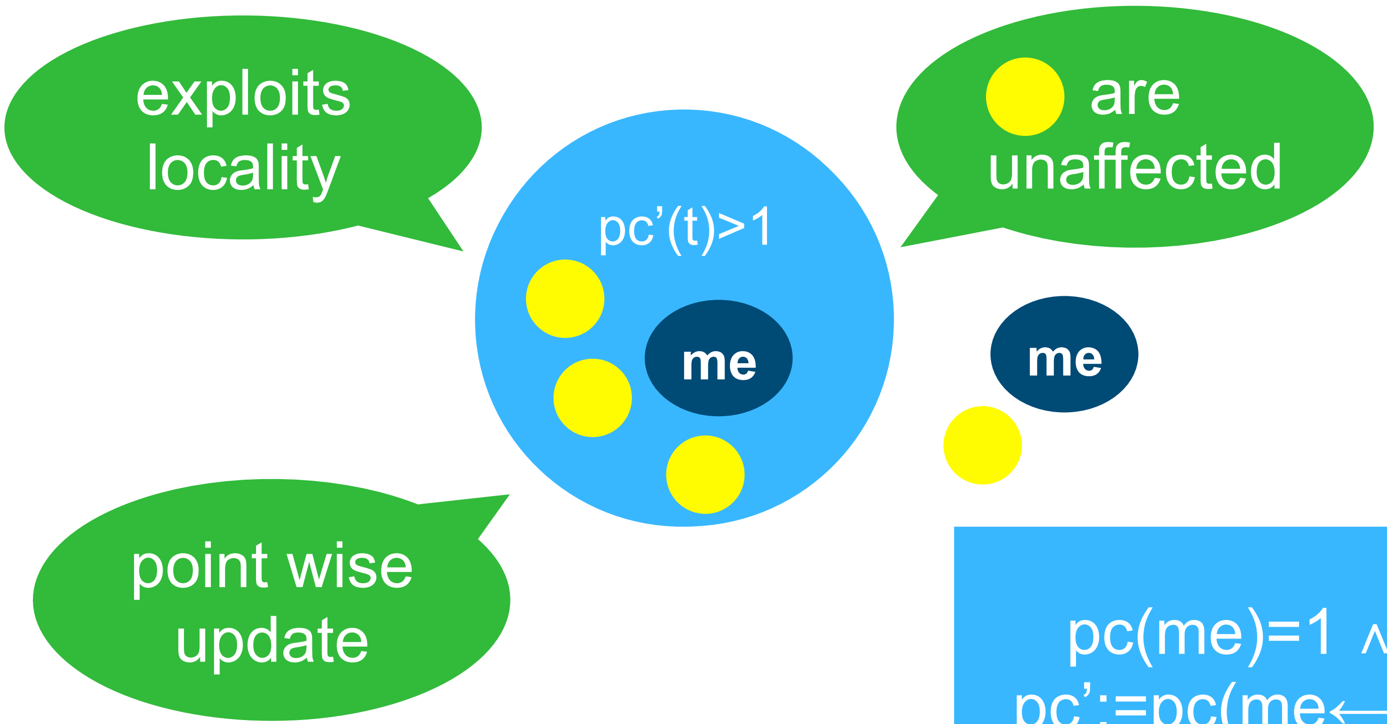
$1 \leq a \;\rightarrow\; pc(me)=2 \rightarrow a>0$

Example: point wise update

How to deal with upd?

$$\#\{t \mid pc(t)>1\} \le a \;\wedge\; \begin{array}{c} pc(me)=1 \\ \wedge\, pc' := pc(me \leftarrow 2) \;\wedge \\ a' = a+1 \end{array} \;\rightarrow\; \#\{t \mid pc'(t)>1\} \le a'$$

Example: point wise update



$pc(me)=1 \wedge pc' := pc(me \leftarrow 2)$

Example: finding the solution

Example: finding the solution

$\#\{t \mid s(t)\}=k$

Cardinality -free

$\forall t: pc(t)=1 \wedge a=0$ $\wedge$ $\rightarrow$ $k \leq a$

k is a fresh variable

$(\forall t: \neg s(t)) \rightarrow k \leq 0$

Complete instantiation

Ship to standard solver

Axioms: inequality

Equality comparison

$$\#\{t \mid s(t)\}=k$$

$$\#\{t \mid p(t)\}=l$$

$$(\forall t{:}\, s(t) \longrightarrow p(t)) \longrightarrow k \leq l$$

Axioms: emptiness (derived)

$$\#\{t \mid s(t)\}=k$$

$$\#\{t \mid \text{false}\}=0$$

$$(\forall t{:}\neg s(t)) \rightarrow k \leq 0$$

Axioms: strict inequality

$$\frac{\#\{t \mid s\}=k \qquad \#\{t \mid p\}=l}{(\forall t{:}s(t)\rightarrow p(t)) \wedge (\exists t{:}\neg s(t) \wedge p(t)) \rightarrow k<l}$$

Additional witness

Axioms: update

Relatively complete wrt. Difference bound constrains.

$\#\{t \mid s(t)\}=k$

$p=s[g/f]$

$\#\{t \mid p(t)\}=l$

$g = f[\, me \leftarrow \_ \,]$

$l = k - s(me) + p(me)$

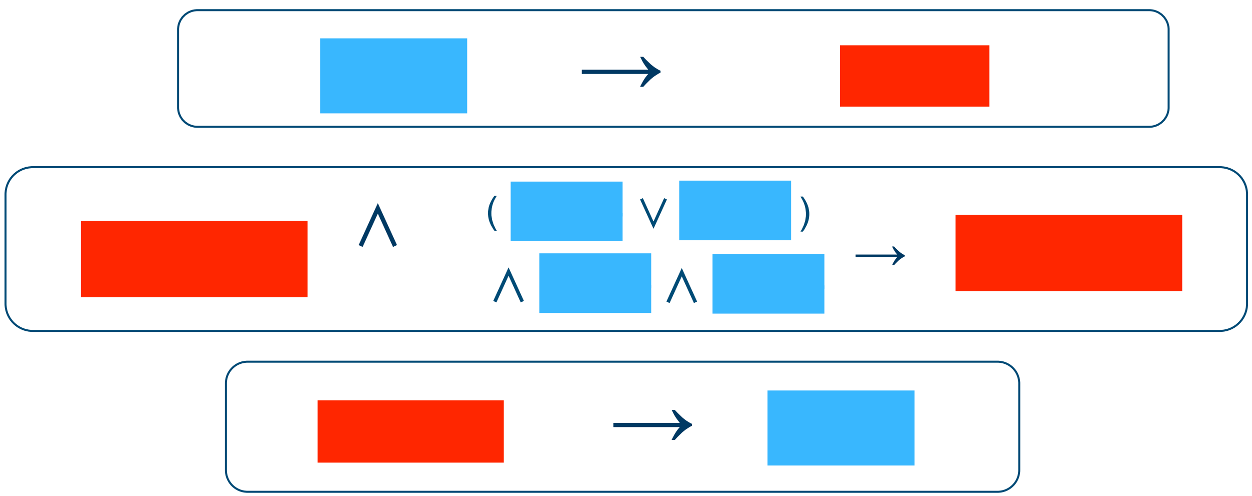
We also do Venn-decomposition

Assume they evaluate to 0 or 1

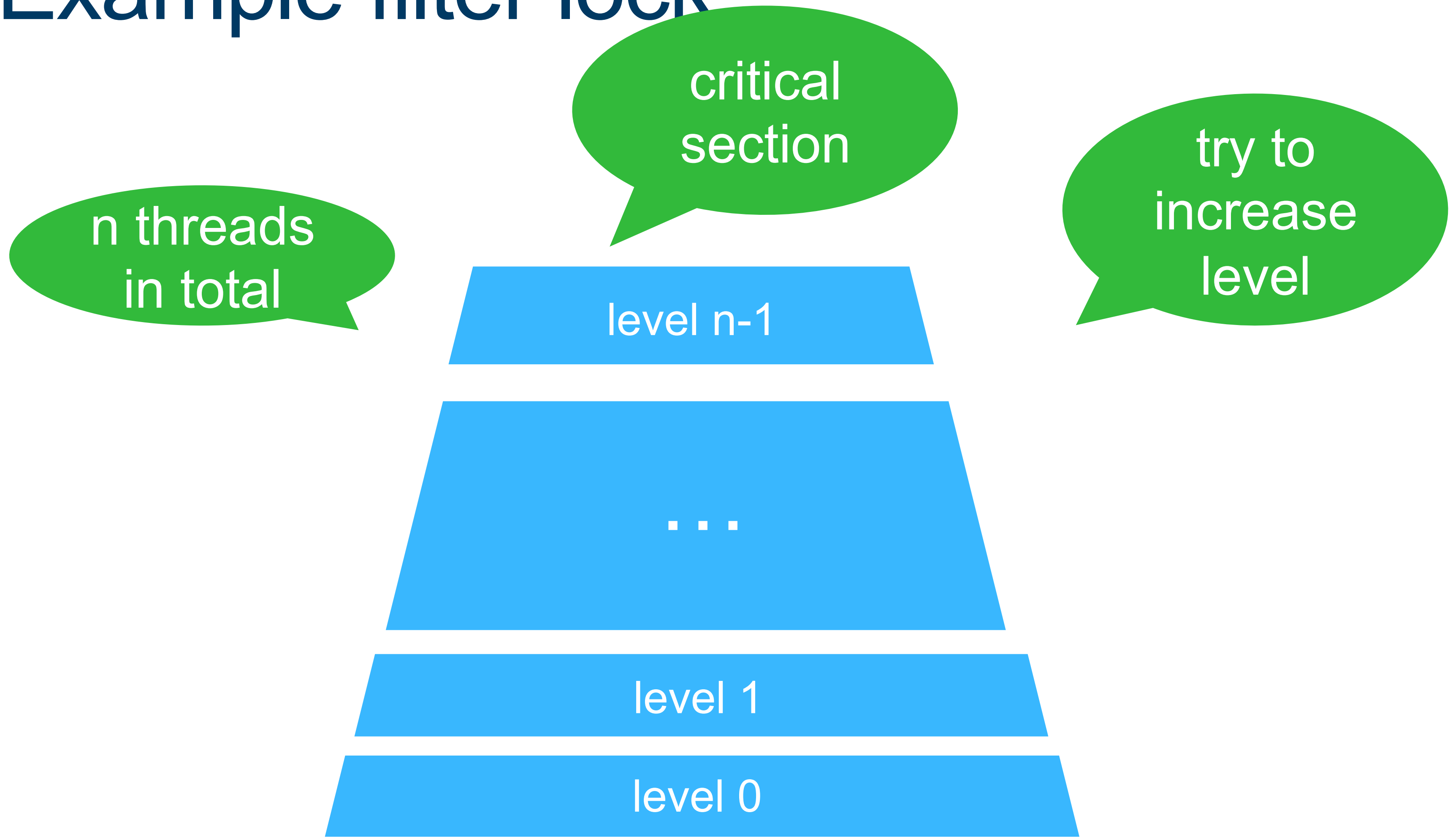
Adding quantifiers:

forall-quantifier

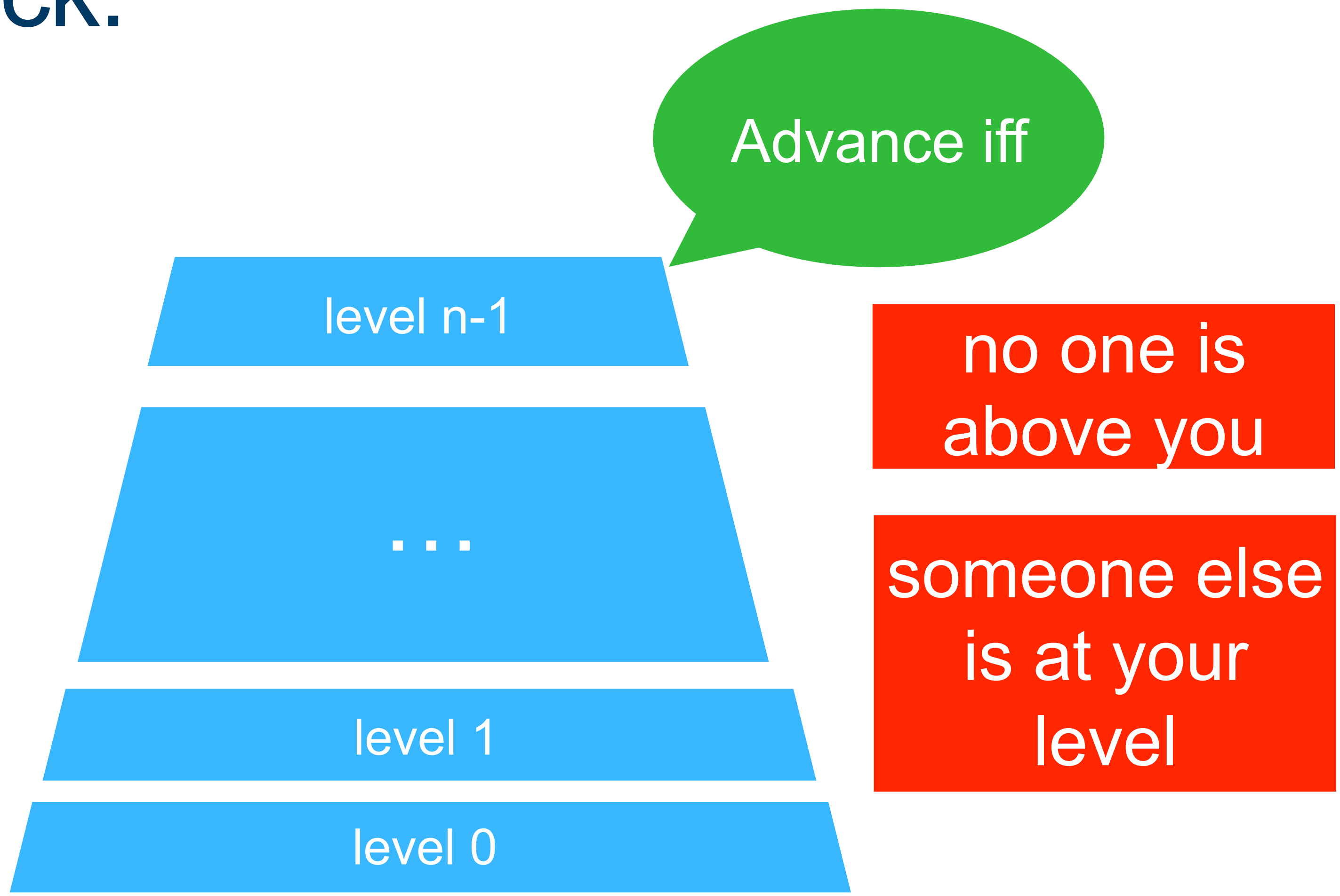
$\exists$ $\forall q$: $\#\{t \mid p\} = k \wedge inv(a, pc, q, k)$



Example filter lock:



Filter lock:



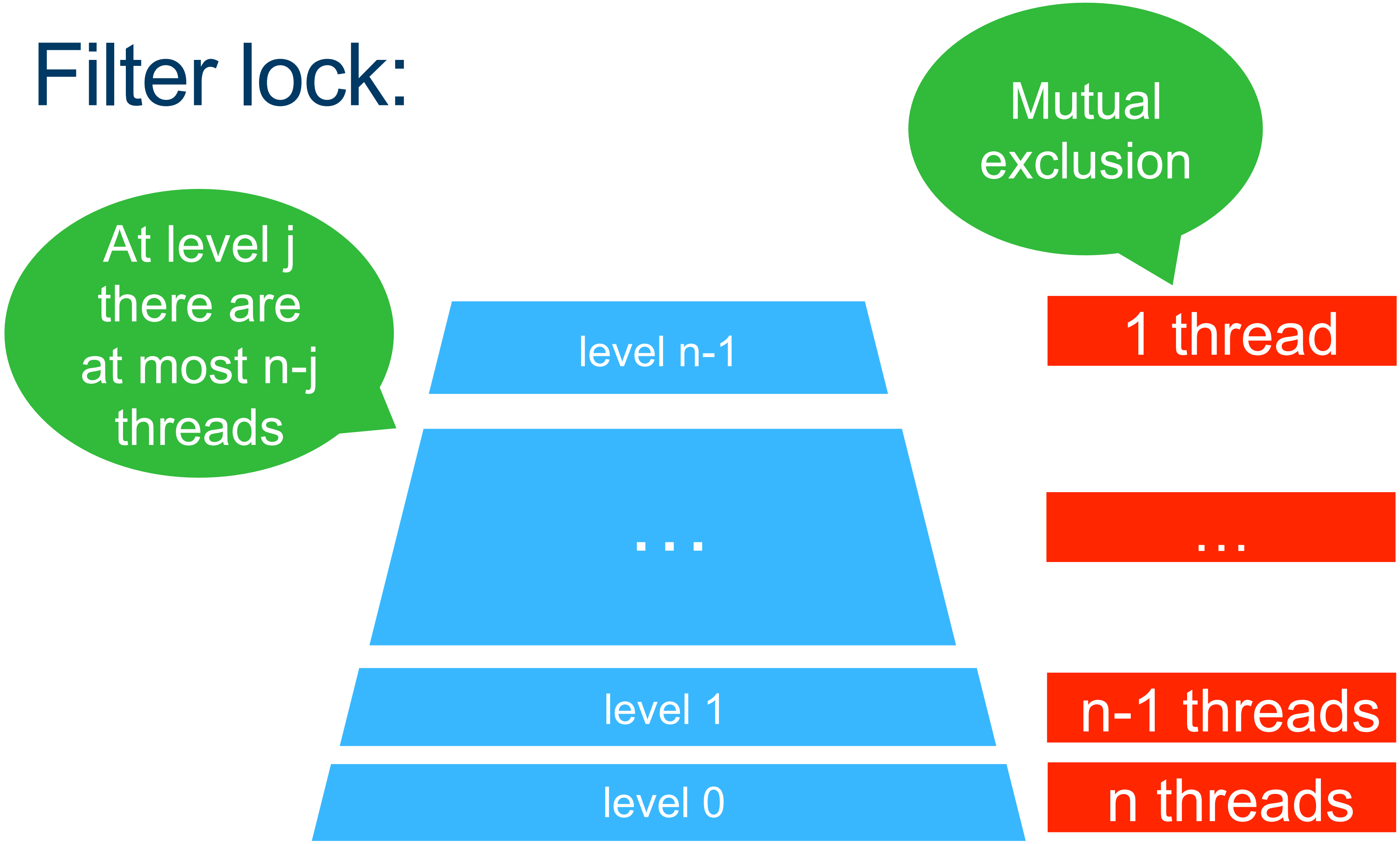
Advance iff
level n-1
...
level 1
level 0
no one is above you
someone else is at your level

Filter lock:

no one is above you

someone else is at your level

Filter lock:



Mutual exclusion
At level j there are at most n-j threads
level n-1
...
level 1
level 0
1 thread
...
n-1 threads
n threads

Filter lock in logic:

State modelled as function

no one is above you

someone else is at your level

$\#\{\ t \mid lv(t) > lv(me)\ \} = 0$

$\#\{\ t \mid lv(t) = lv(me)\ \} > 1$

Filter lock in logic:

$$\big(\ \#\{\ t \mid lv(t) > lv(me)\ \}=0\ \ \vee\ \ \#\{\ t \mid lv(t) = lv(me)\ \}>1\ \big)$$

$$\wedge\ \ lv' := lv[me \leftarrow lv(me)+1]\ \ \wedge\ \ \ldots$$

Constraints on invariant:

∃ inv(v) :

$\forall t: lv(t)=0 \wedge n \geq 2 \rightarrow$ inv(v)

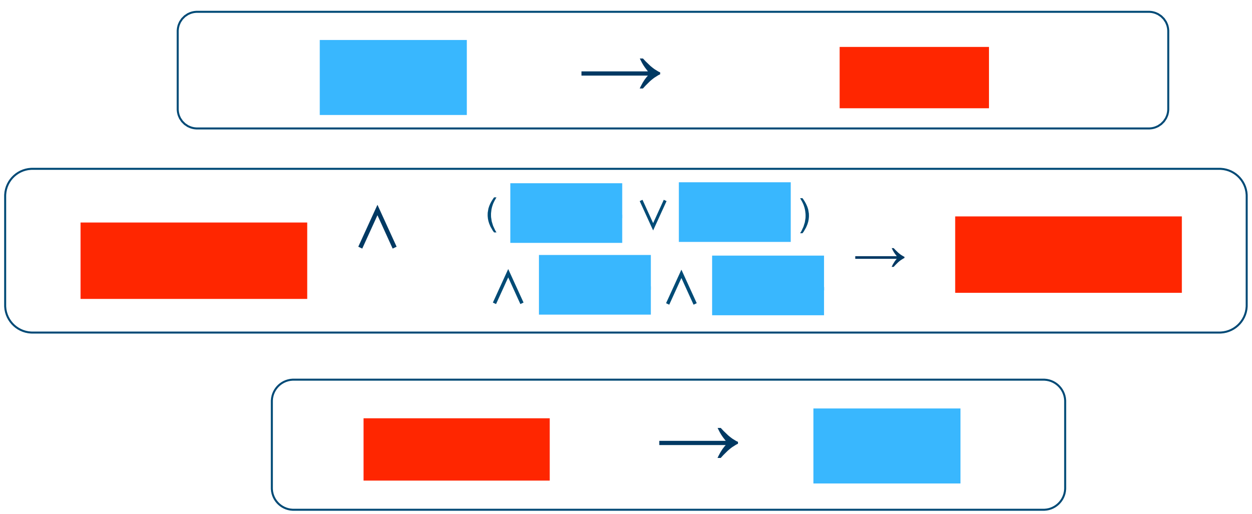
inv(v) $\wedge$ ($\square$ $\vee$ $\square$) $\wedge$ $\square$ $\wedge$ $\square$ $\rightarrow$ inv(v')

inv(v) $\rightarrow$ $\#\{ t \mid lv(t) = n-1 \} \leq 1$

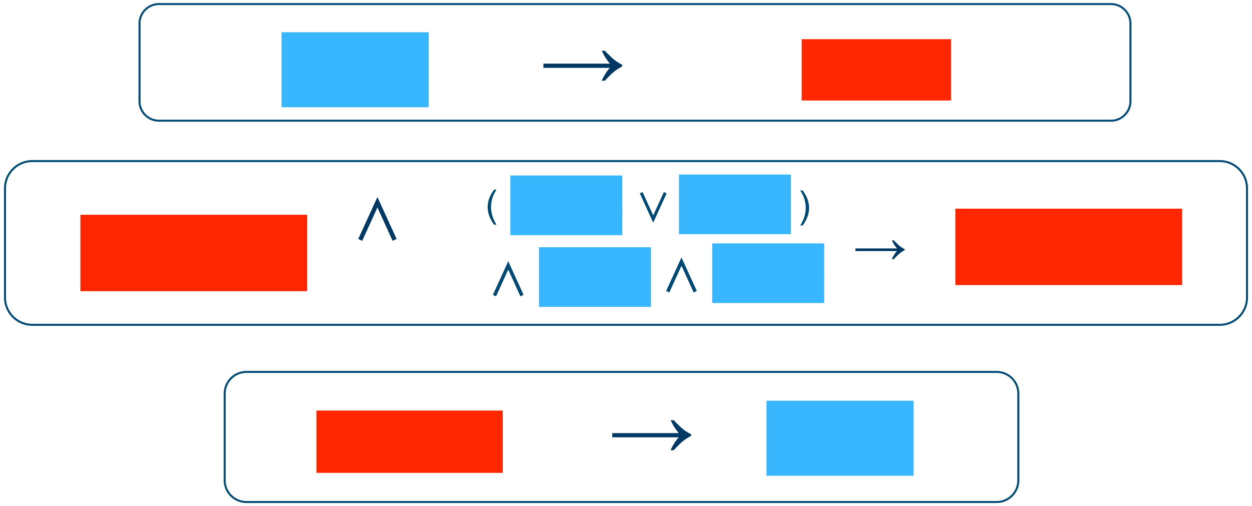
Restricting the search space:

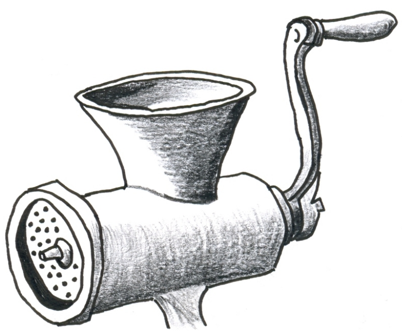
$\exists$ $\forall q$: $\#\{t \mid p\}=k \wedge inv(v,q,k)$:

One forall-
quantifier,
one set



Solving:





Beyond scalar counters

$\forall q: 0 \leq q \leq n-1 \rightarrow \#\{t \mid lv(t) \geq q\} \leq n-q$

Evaluation:

Quantifiers + cardinalities

First to automatically verify

Program	Card	Property	Inferred cardinalities	Time
intro [21]	✓	$(\exists t : pc(t) = 2) \rightarrow b < a$	$\#\{t \mid pc(t) = 2\}$	1.2s
bluetooth [21]	✓	$(\exists t : pc(t) = 2) \rightarrow st = 0$	$\#\{t \mid pc(t) = 2\}$	1.6s
tree traverse [21]	×	$leaves = nodes + 1$	–	4.2s
cache [59]	✓	$\#\{t \mid pc(t) = 3\} \leq 1$	$\#\{t \mid pc(t) \geq 3\}$	0.7s
garbage collection	✓	$\#\{t \mid 2 \leq pc(t) \leq 4\} \leq 1 \wedge m = 1$	$\#\{t \mid 2 \leq pc(t) \leq 4\}$	10.1s

Program	Property	[illegible]	[illegible]
ticket lock [21]	$\#\{t \mid pc(t) = 3\} \leq 1$	[illegible]	[illegible]
filter lock [31]	$\#\{t \mid lv(t) = n - 1\} \leq 1$	$\#\{t \mid lv(t) \geq q\}$	27.5s
[illegible]	see Section 2	$\#\{t \mid x(t) = x(q)\}$	0.8s

Dragoi et al. VMCAI'14

Can do cardinalities, if required

Evaluation

Quantifiers w/o cardinalities

Abdulla et al. CAV'07

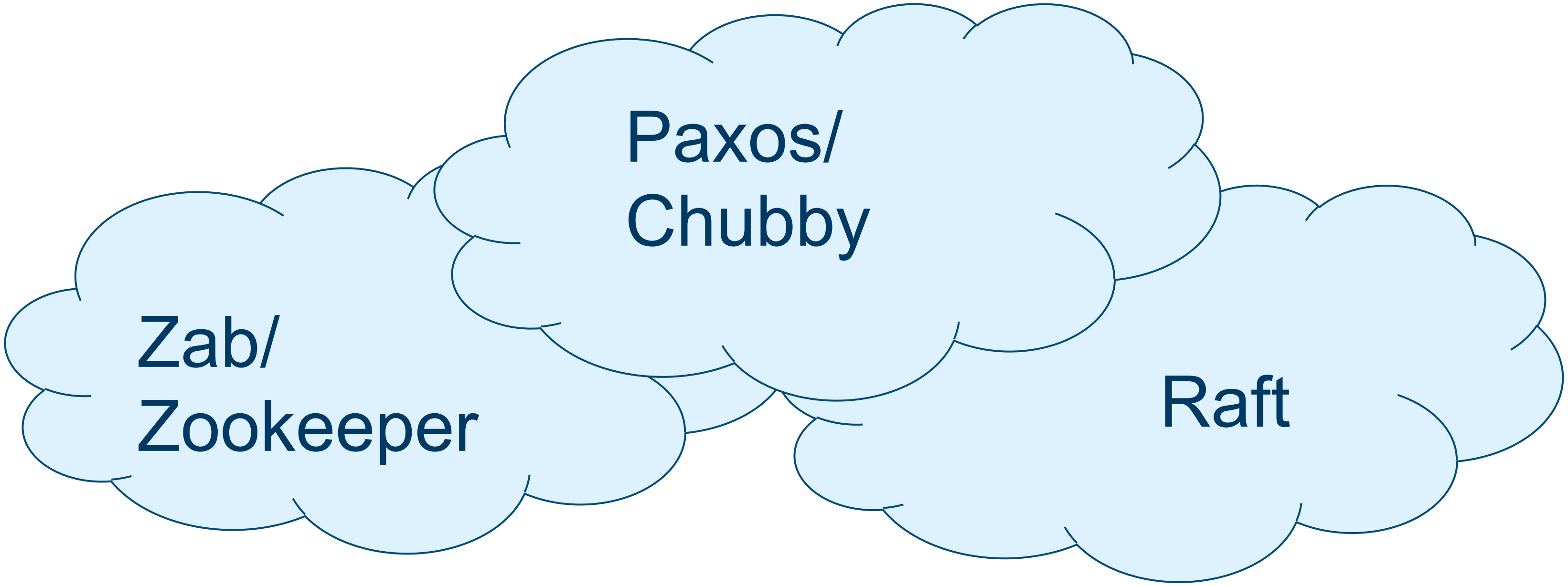
Competitive when no cardinalities are needed

Sanchez et al. SAS'12

Program	Quantifiers	Time			
		#Π	[illegible]	[illegible]	[illegible]
Simplified Bakery [1]	2	0.4s	[illegible]	[illegible]	[illegible]
Lamport's Bakery [1]	2	0.5[illegible]	[illegible]	[illegible]	[illegible]
Bogus Bakery [1]	2	0.6s	[illegible]	[illegible]	[illegible]
Ticket Mutex [1]	2	0.5s	[illegible]	[illegible]	[illegible]

Program	Quantifiers	Time			
		#Π	I[54]	P[54]	O[54]
barrier [54]	1	0.4s	0.1s	0.1s	0.1s
central barrier [54]	1	0.4s	0.1s	1.1s	6.2s
work stealing [22, 54]	1	0.5s	0.1s	0.1s	6.2s
dining philosophers [54]	0	8.2s	0.1s	6.3s	20s
robot 2x2 [54]	2	2.8s	0.2s	5.8s	1m45s
robot 2x3 [54]	2	16.1s	0.5s	16s	5m20s
robot 3x3 [54]	2	34.0s	0.9s	52s	19m28s
robot 4x4 [54]	2	TO	3.2s	5m3s	TO

What's with:



What's with:

#{…}

Encoding of Paxos

Algorithm 9 The *OneThirdRule* algorithm

Initialization:
$x_p := v_p$ … value of p }

Round r**:**
S_p^r :
send $\langle x_p$ …

T_p^r :
if $|HO(p,r)|$ …
$x_p :=$ the sm… …d value
if more than $2n/$… …are equal to $\overline{x}$ **then**
DECIDE($\overline{x}$)

Algorithm 8 The LastVoting algorithm

Initialization:
$x_p \in V$, initially v_p {v_p *is the initial value of* p}
$vote_p \in V \cup \{?\}$, initially ?
$commit_p$ a Boolean, initially `false`
$ready_p$ a Boolean, initially `false`
$ts_p \in \mathbb{N}$, initially 0

Round $r = 4\phi - 3$ **:**
S_p^r :
send $\langle x_p\ ,\ ts_p \rangle$ to $Coord(p, \phi)$

T_p^r :
if $p = Coord(p, \phi)$ **and**
number of $\langle \nu\ ,\ \theta \rangle$ received $> n/2$ **then**
let $\overline{\theta}$ be the largest θ from $\langle \nu\ ,\ \theta \rangle$ received
$vote_p :=$ one ν such that $\langle \nu\ ,\ \overline{\theta} \rangle$ is received
$commit_p :=$ `true`

Round $r = 4\phi - 2$ **:**
S_p^r :
if $p = Coord(p, \phi)$ **and** $commit_p$ **then**
send $\langle vote_p \rangle$ to all processes

T_p^r :
if received $\langle v \rangle$ from $Coord(p, \phi)$ **then**
$x_p := v$; $ts_p := \phi$

Round $r = 4\phi - 1$ **:**
S_p^r :
if $ts_p = \phi$ **then**
send $\langle ack \rangle$ to $Coord(p, \phi)$

T_p^r :
if $p = Coord(p, \phi)$ **and**
number of $\langle ack \rangle$ received $> n/2$ **then**
$ready_p :=$ `true`

Round $r = 4\phi$ **:**
S_p^r :
if $p = Coord(p, \phi)$ **and** $ready_p$ **then**
send $\langle vote_p \rangle$ to all processes

T_p^r :
if received $\langle v \rangle$ from $Coord(p, \phi)$ **then**
DECIDE(v)
if $p = Coord(p, \phi)$ **then**
$ready_p :=$ `false`
$commit_p :=$ `false`

Questions?

Invariant: one third rule

$$\forall p: dec(p) \geq 0 \longrightarrow$$
$$(\#\{ t \mid x(t)=x(p) \} > 2n/3 \wedge x(p) = dec(p))$$