C Timed Information Flow

June 8, 2016

Mikael Elkiær Christensen, michri11@student.aau.dk
Mikkel Sandø Larsen, milars11@student.aau.dk

Department of Computer Science
Aalborg University
Denmark

AALBORG UNIVERSITY
DENMARK

Introduction

Embedded devices / IoT

Estimated number of devices:

- 2006: 2 billion
- 2015: 15 billion
- 2020: 200 billion

Usage:

- 40.2 % Business/manufacturing
- 30.3 % Health care
- 8.3 % Retail
- 7.7 % Security

https://www-ssl.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html

Arduino and Raspberry

Estimated sold units:

- Arduino – 1.5 million (2013)[1]
- Raspberry Pi – 3 million (2014)[2]

Implications:

- More exposed devices
- More amateur implementations

[1] http://medea.mah.se/2013/04/arduino-faq/
[2] https://www.raspberrypi.org/blog/raspberry-pi-at-buckingham-palace-3-million-sold/

Programmer's perspective

Extension to C:

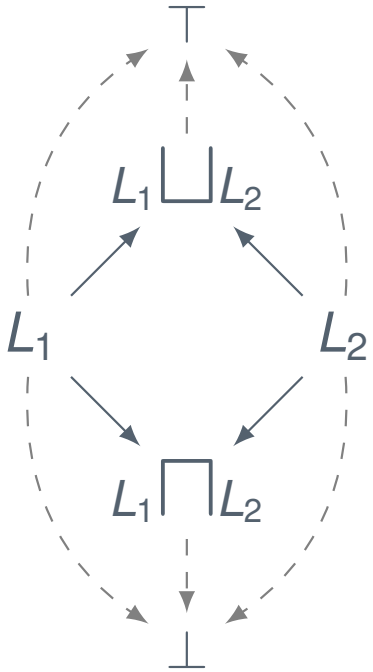
- Simple syntax
- Minimal intrusiveness
- Compiler errors
- (Visual aids)

Time policies

- Same characteristics as above

DLM

Abstract lattice



Label policies

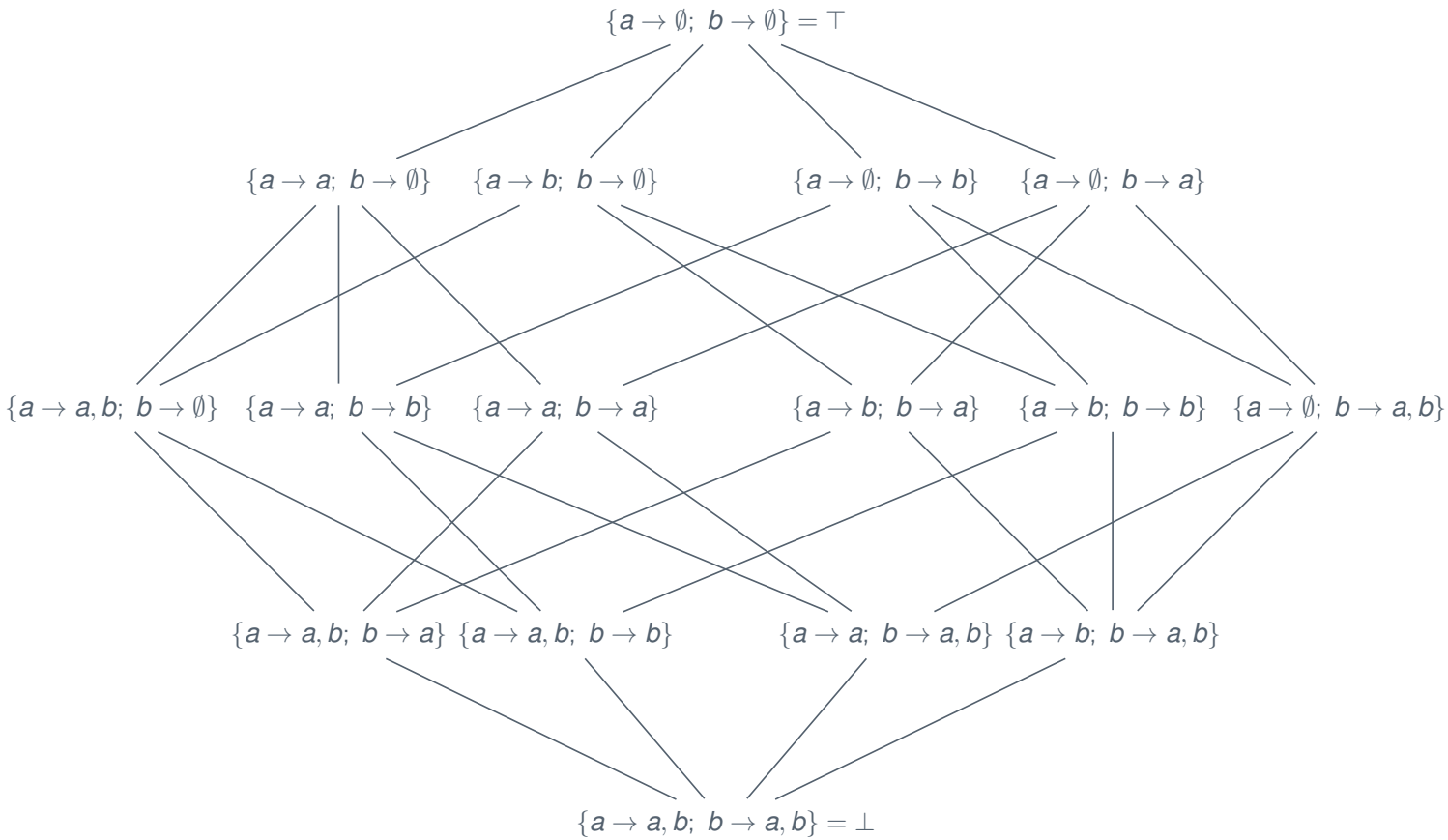
$$L_1 = \{o_1 \to r_1, r_2\} \quad L_2 = \{o_1 \to r_1;\ o_2 \to r_1\}$$

$$L_1 \sqcup L_2 = \{o_1 \to r_1;\ o_2 \to r_1\}$$

$$L_1 \sqcap L_2 = \{o_1 \to r_1, r_2\}$$

$$L_2 \sqsubseteq L_1 \wedge L_1 \not\sqsubseteq L_2$$

Example lattice



Scope

Run-time

- Principal hierarchy
- Label as first-class citizen
- Function call authority
- Run-time label checking

Integrity

- “Opposite” of privacy
- *CIF* and *CBIF* employ this

Inference

Constraint checking

Label types:

- Policy
- Variable
- Constant
- Composite labels (join/meet)
- Upper/lower bound

Output channel constraints:

- No simple check of reader sets
- Constraint for each argument, enabling inference

Algorithm


```
    Data: A set Q of constraints "L1 ⊑ L2"
 1  foreach "L1 ⊑ L2" ∈ Q do
 2  |   let Q' = Q \ {"L1 ⊑ L2"}
 3  |   Q := Q' ∪ unjoin("L1 ⊑ L2")
 4  foreach "L1 ⊑ L2" ∈ Q do
 5  |   if L1 is a variable label then
 6  |   |   cub(L1) := ⊤
 7  checked:= false
 8  while ¬checked do
 9  |   checked:= true
10  |   foreach "L1 ⊑ L2" ∈ Q do
11  |   |   if novar(L1) ⋢ novar(L2) then
12  |   |   |   if L1 is a variable label then
13  |   |   |   |   cub(L1) := cub(L1) ⊓ novar(L2)
14  |   |   |   |   checked:= false
15  |   |   |   else
16  |   |   |   |   ERROR
```

Algorithm 1: Label inference from constraint set

Polyvariant function evaluation

Example where inference fails

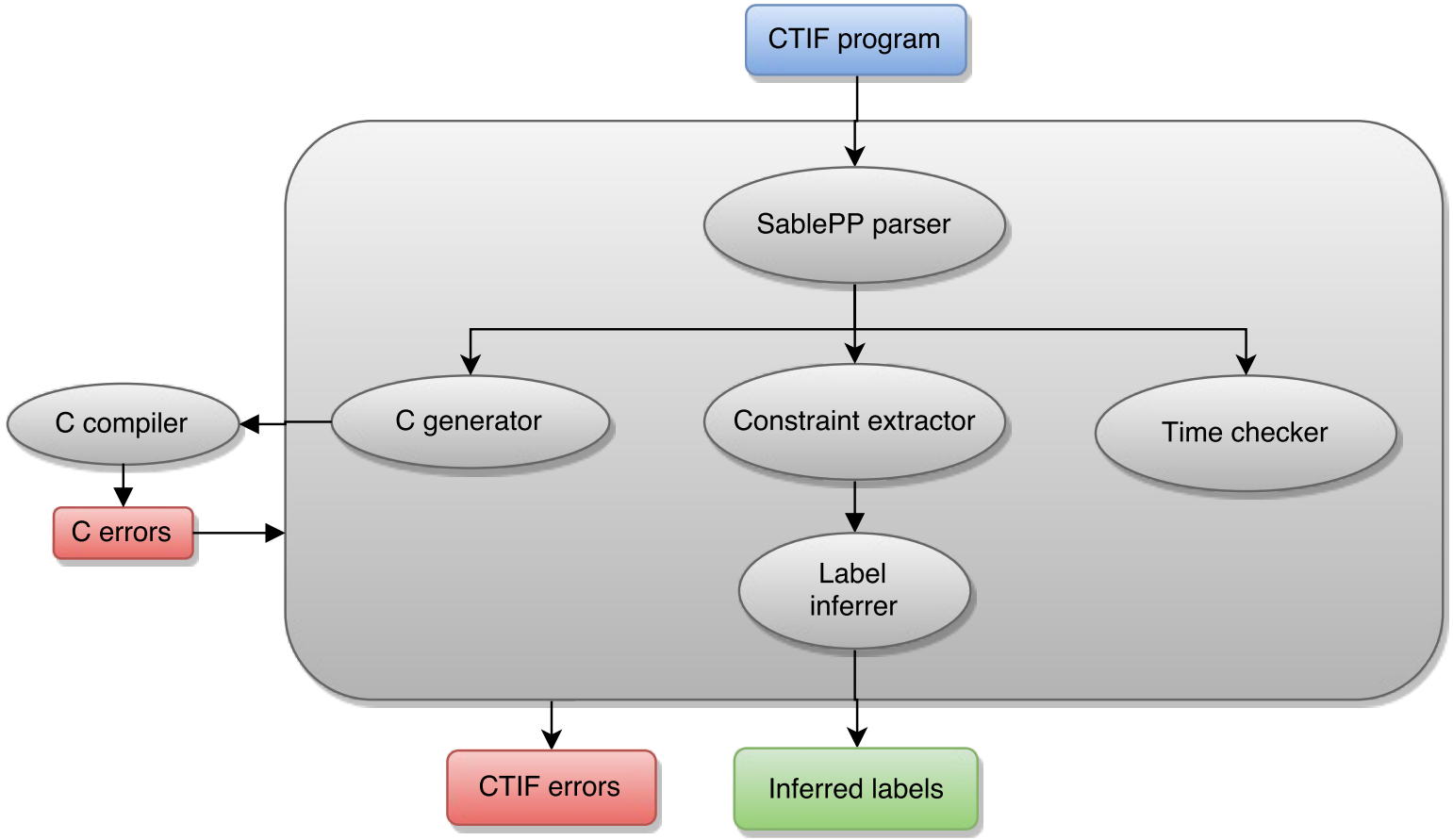
```
int foo(int {{p->p}} x);
p <- int foo(int x);

int bar(int y) { return foo(y); }
```

Alternative to function evaluation

- Evaluation dependent on each call
- Set of constraints based on parameters
- Downside: Evaluation of library functions

The CTIF Tool

Process



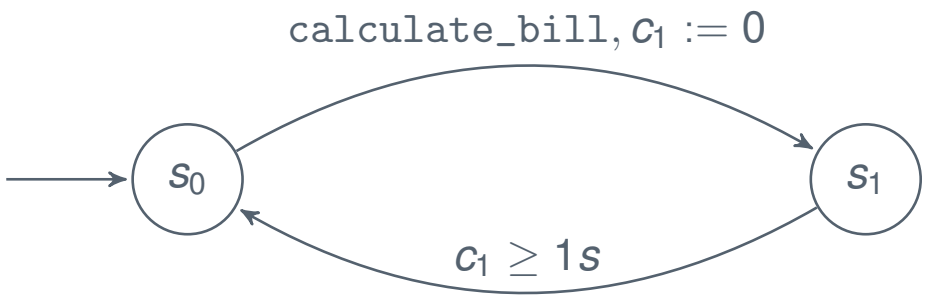
Demonstration

Time Policies

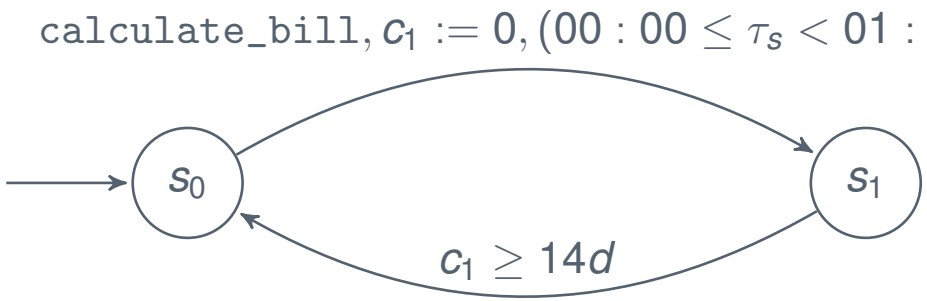
Bill calculator

```
{{u->u, ec @ u: 1s; 00:00-01:00 14d}}
```

u

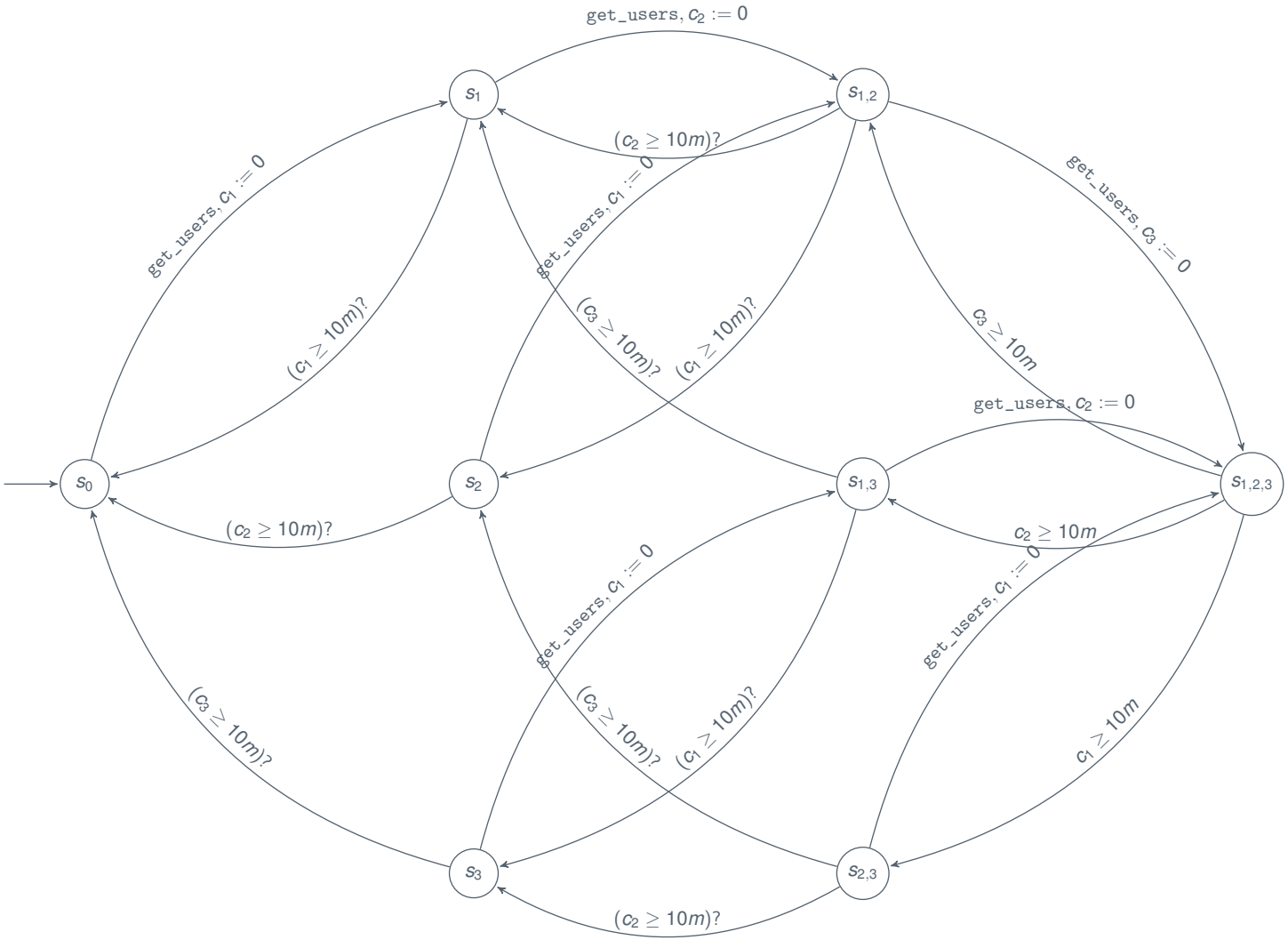


All



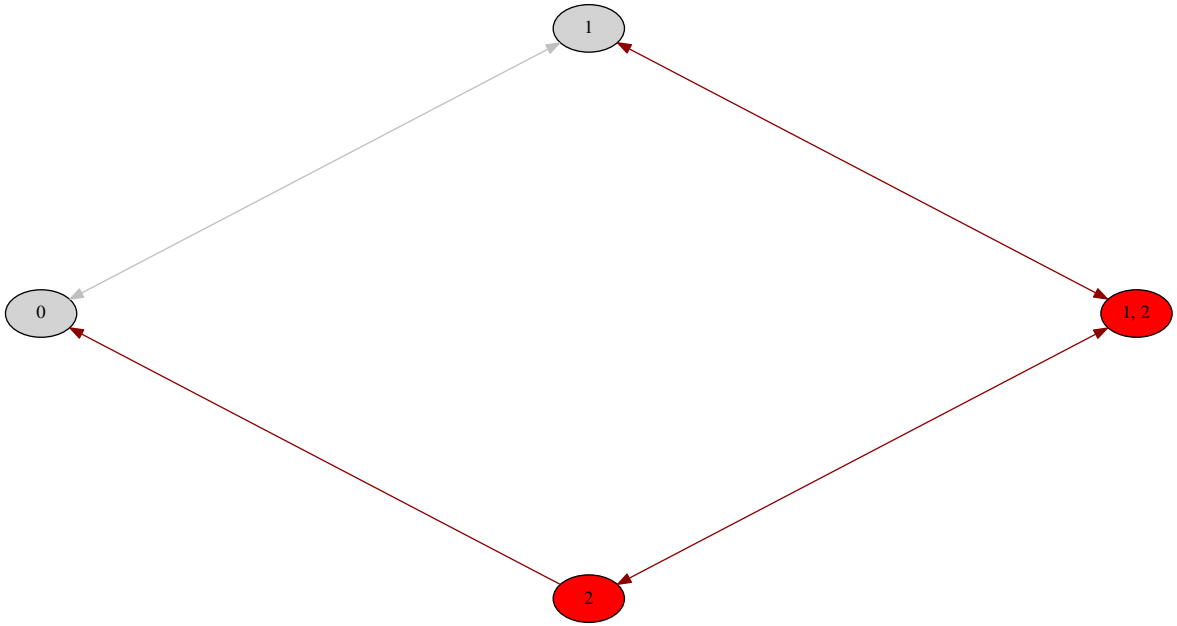
Password checker


```
{{pc-> @ 10m * 3}}
```

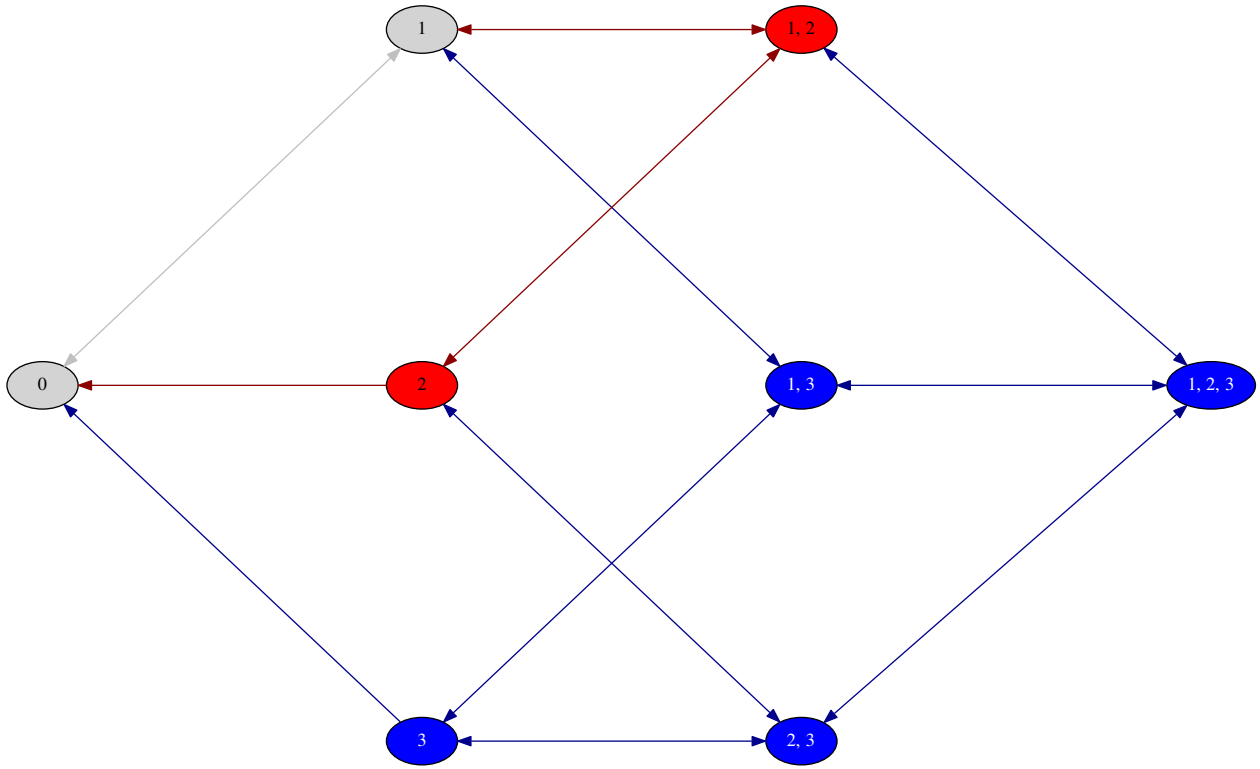


1 count

2 count

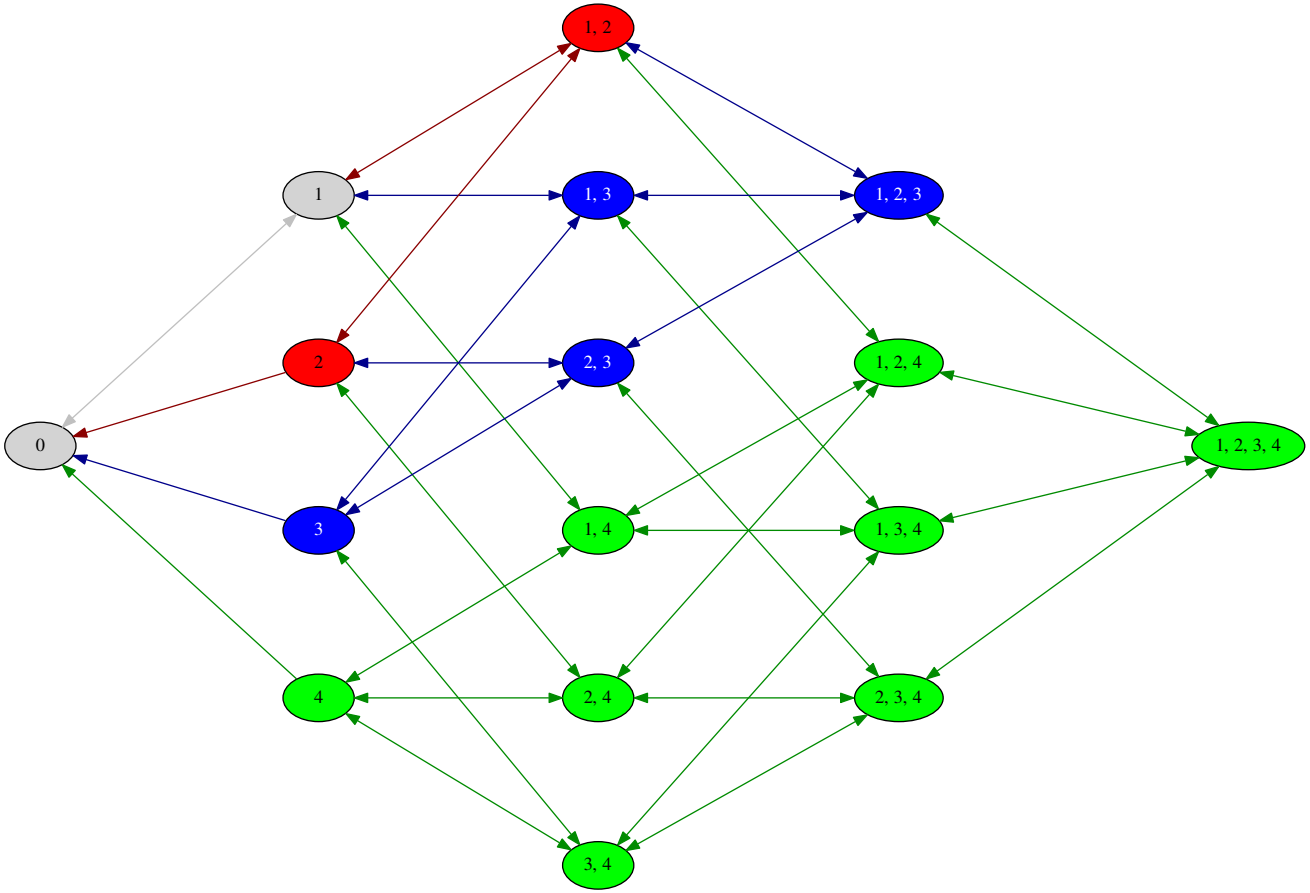


3 count




BREAKING NEW GROUND
AALBORG UNIVERSITY

4 count


1, 2
1
1, 3
1, 2, 3
2
2, 3
1, 2, 4
0
1, 2, 3, 4
3
1, 4
1, 3, 4
4
2, 4
2, 3, 4
3, 4

Questions?

AALBORG UNIVERSITY
DENMARK