
西安交通大學
XI'AN JIAOTONG UNIVERSITY

ANTS
Advanced NeTworked System
NetVerify
Google

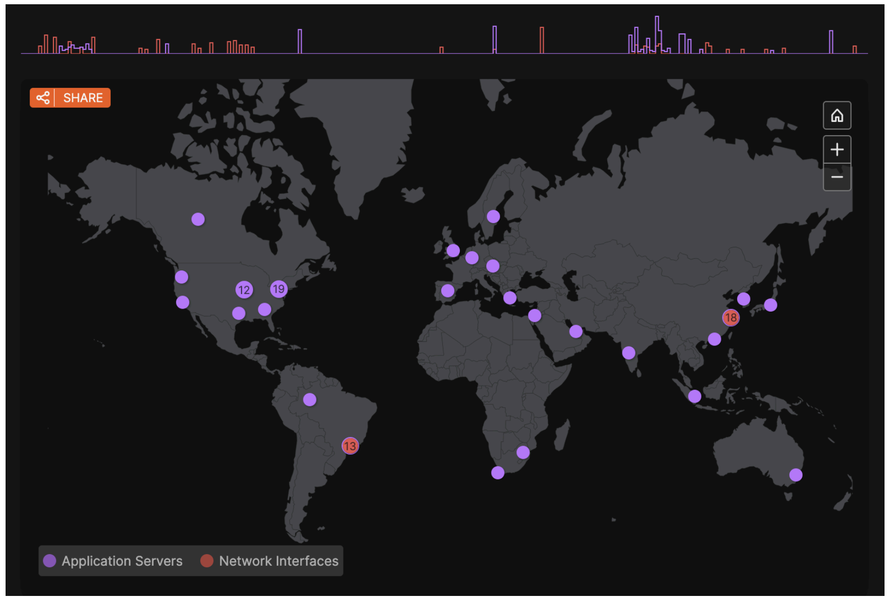
NDD: A Decision Diagram for Network Verification

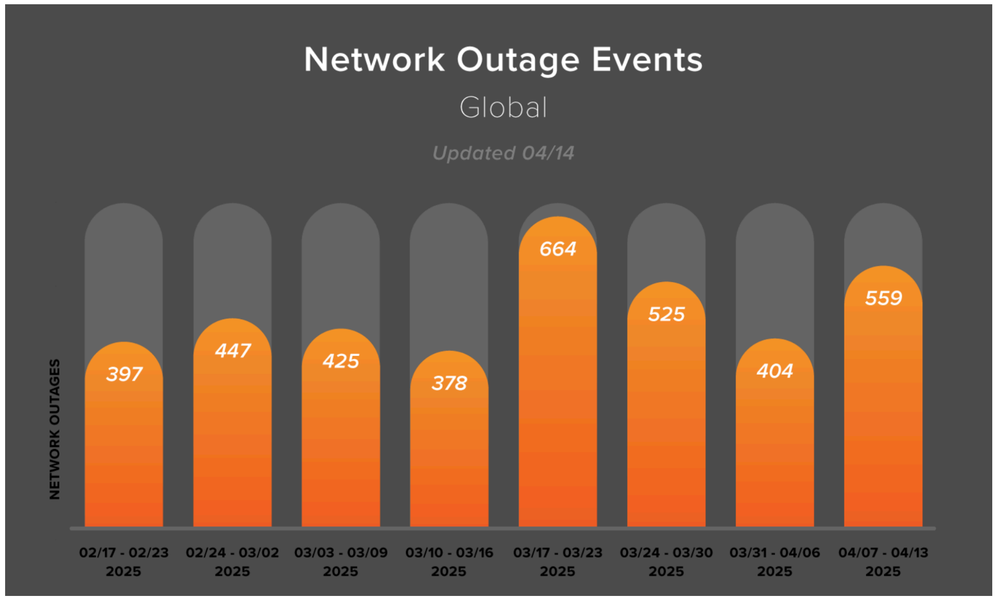
Zechun Li[1], Peng Zhang[1], Yichi Zhang[1], Hongkun Yang[2]

[1]Xi'an Jiaotong University, [2]Google

Network outages

- Network outages are common due to misconfigurations





www.thousandeyes.com/outages

Network verification

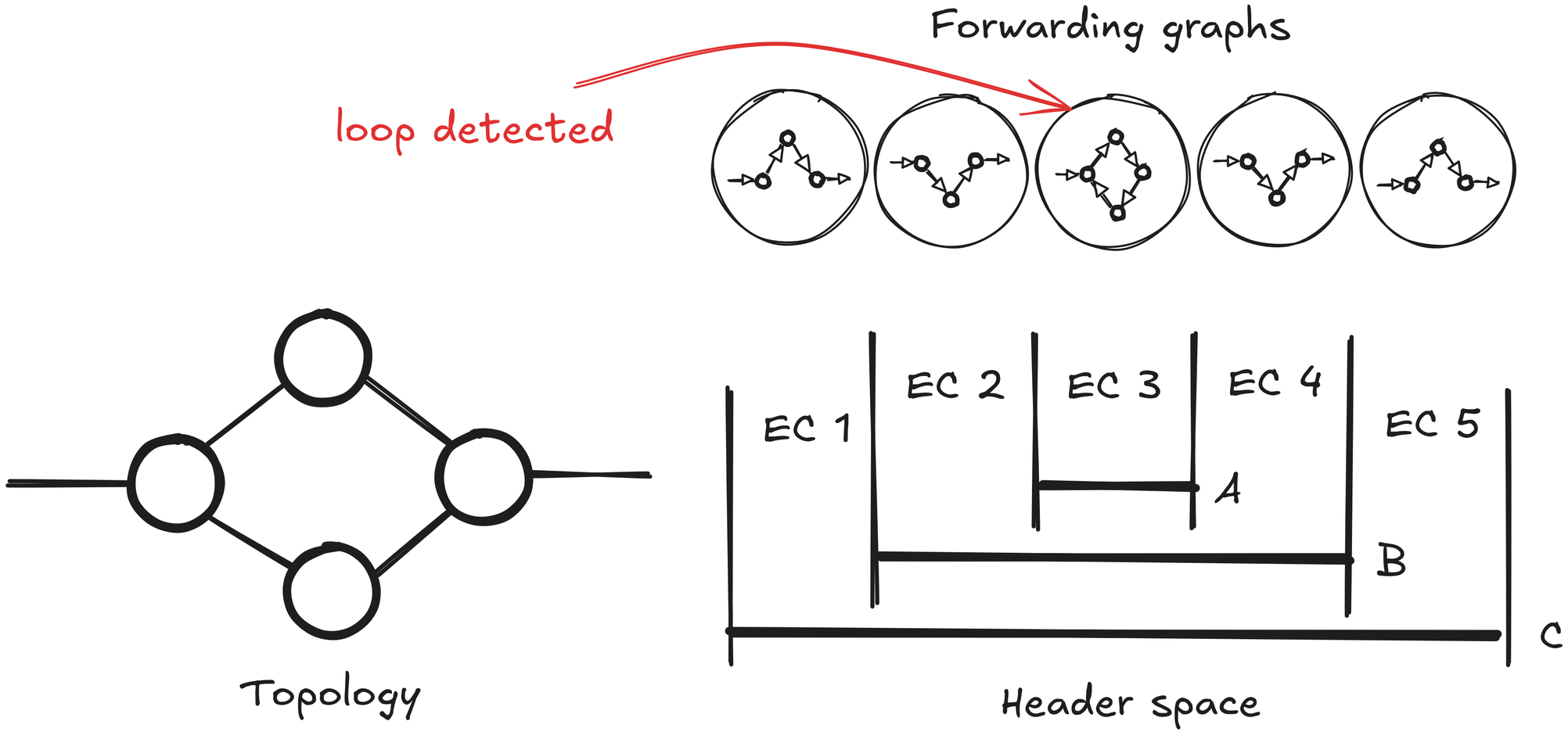
MNV CoNEXT'23
Katra NSDI'22
PPV TON'21
APKeep NSDI'20
Jinjing SIGCOMM'19
APT TON'17
Delta-net NSDI'17
AP Verifier TON'14
NetPlumber NSDI'13
ConfigChecker ICNP'09

Data plane

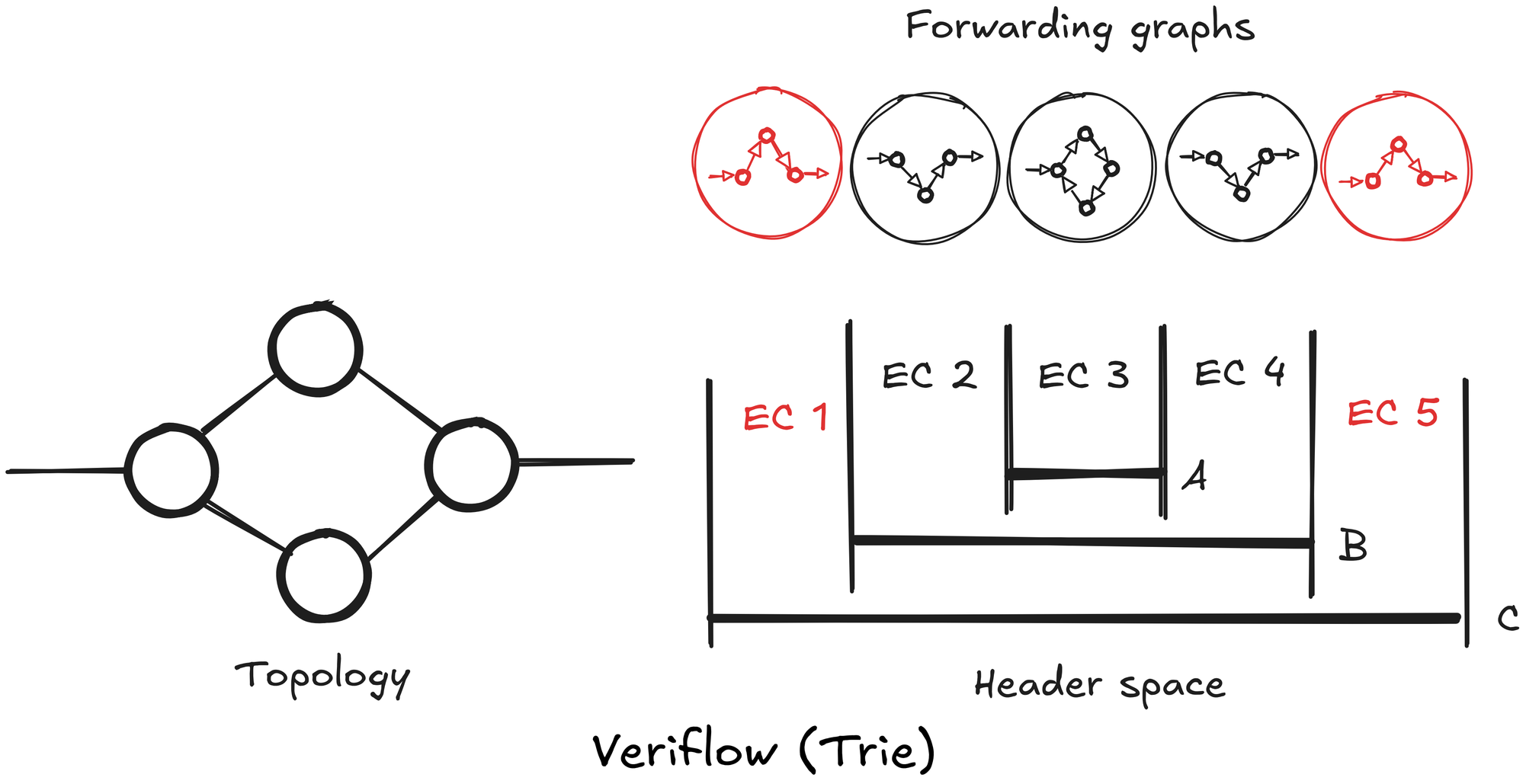
Expresso SIGCOMM'24
YU SIGCOMM'24
SRE SIGCOMM'22
DNA NSDI'22
ProbNV ICFP'21
Campion SIGCOMM'21
NV PLDI'20
ShapeShifter POPL'20
Tiramisu NSDI'20
Hoyan SIGCOMM'20
NetDice SIGCOMM'20
Bonsai SIGCOMM'18
Minesweeper SIGCOMM'17
ERA OSDI'16
ARC SIGCOMM'16
Batfish NSDI'15

Control plane

Equivalence classes (atoms)

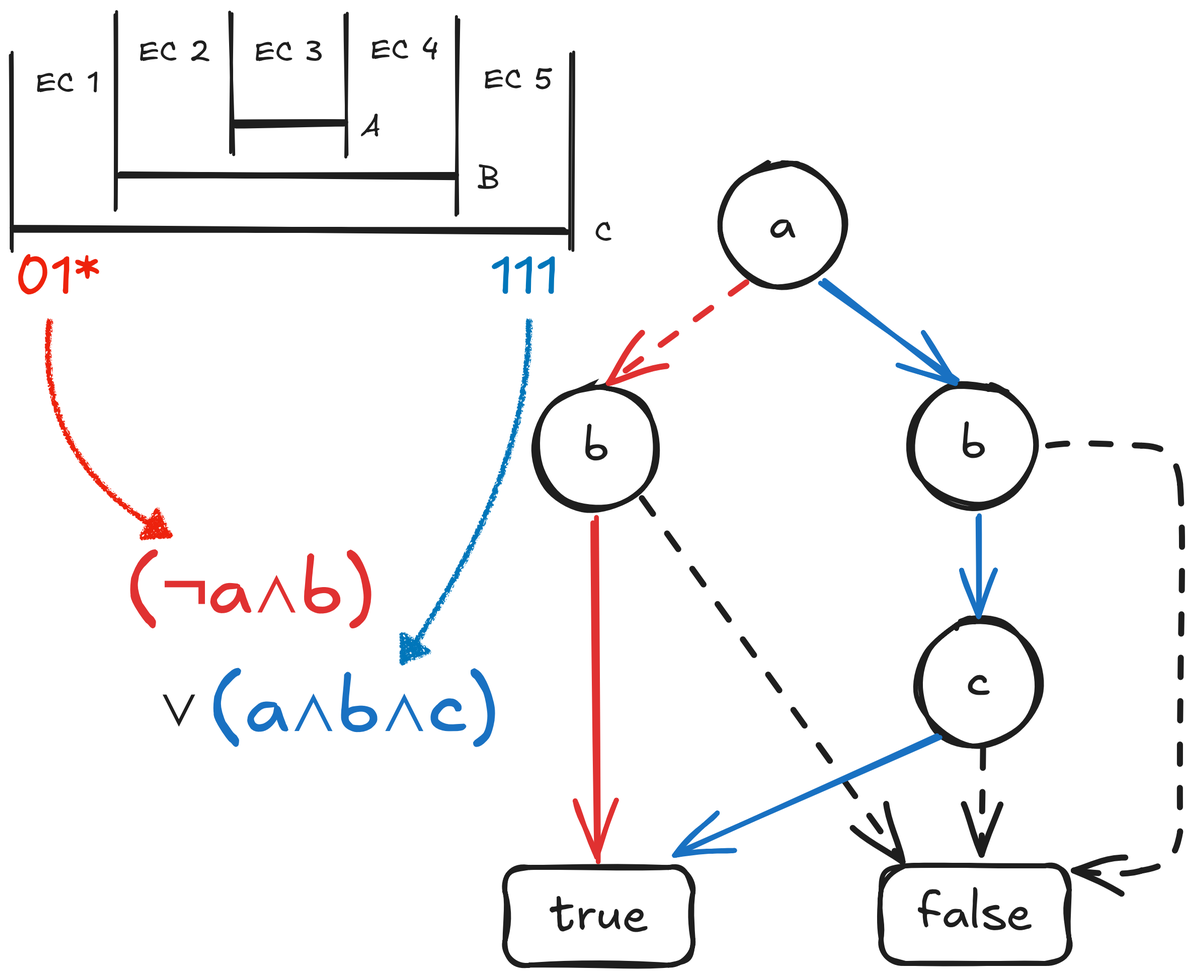


Equivalence classes (atoms)

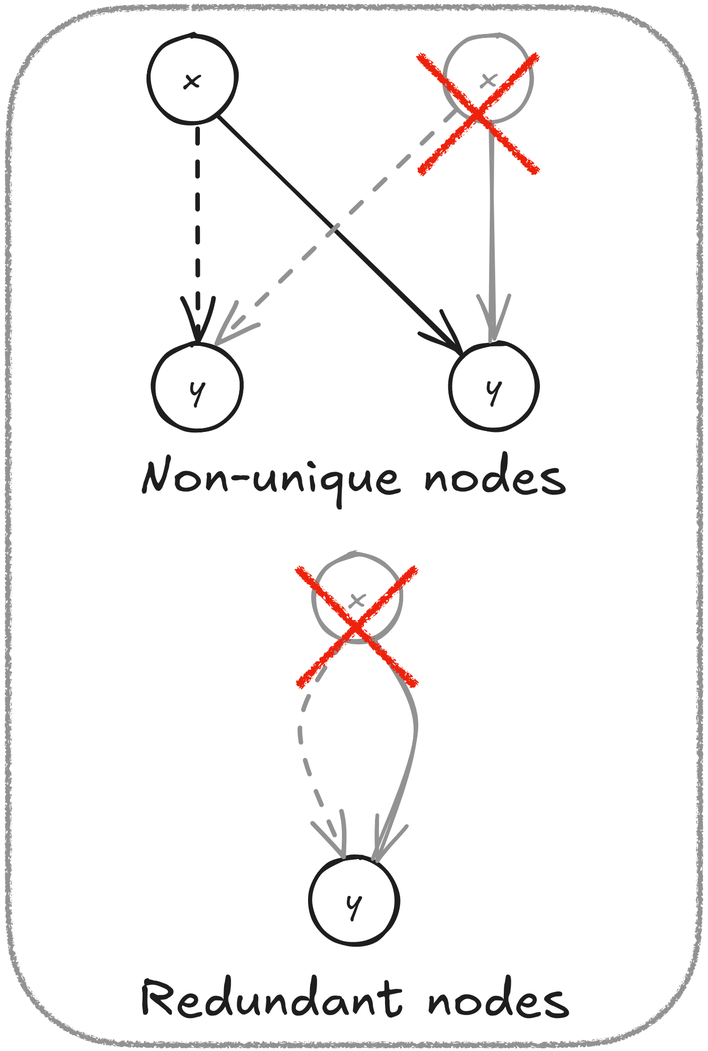


Veriflow (Trie)

Binary Decision Diagram



Reduction rules of Reduced Ordered BDD



Network verification

MNV
CoNEXT'23

Katra
NSDI'22

PPV
TON'21

APKeep
NSDI'20

APT
TON'17

AP Verifier
TON'14

ConfigChecker
ICNP'09

Data plane

Expresso
SIGCOMM'24

YU
SIGCOMM'24

SRE
SIGCOMM'22

DNA
NSDI'22

ProbNV
ICFP'21

Campion
SIGCOMM'21

NV
PLDI'20

ShapeShifter
POPL'20

Bonsai
SIGCOMM'18

Minesweeper
SIGCOMM'17

ERA
OSDI'16

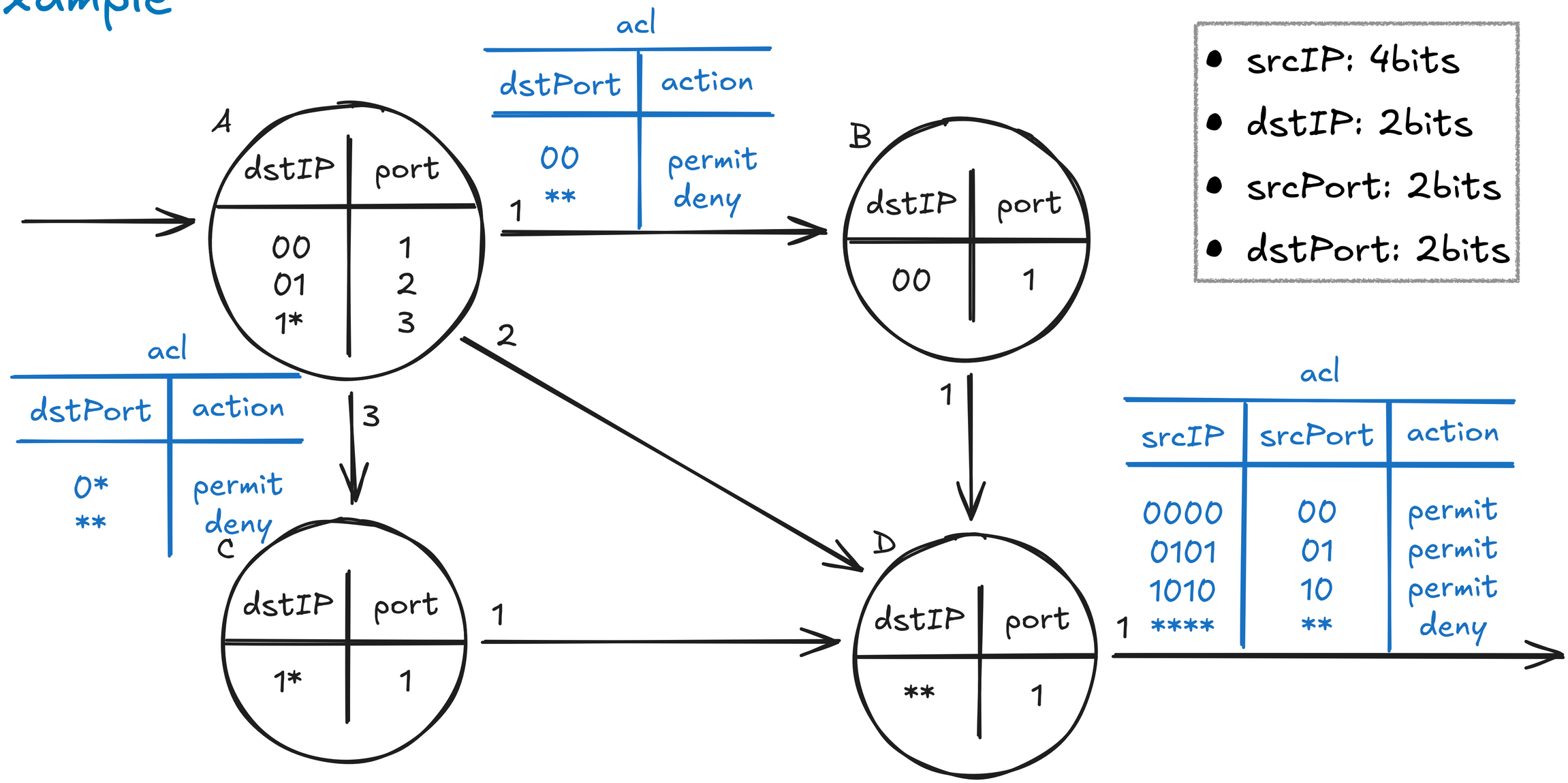
Batfish
NSDI'15

Control plane

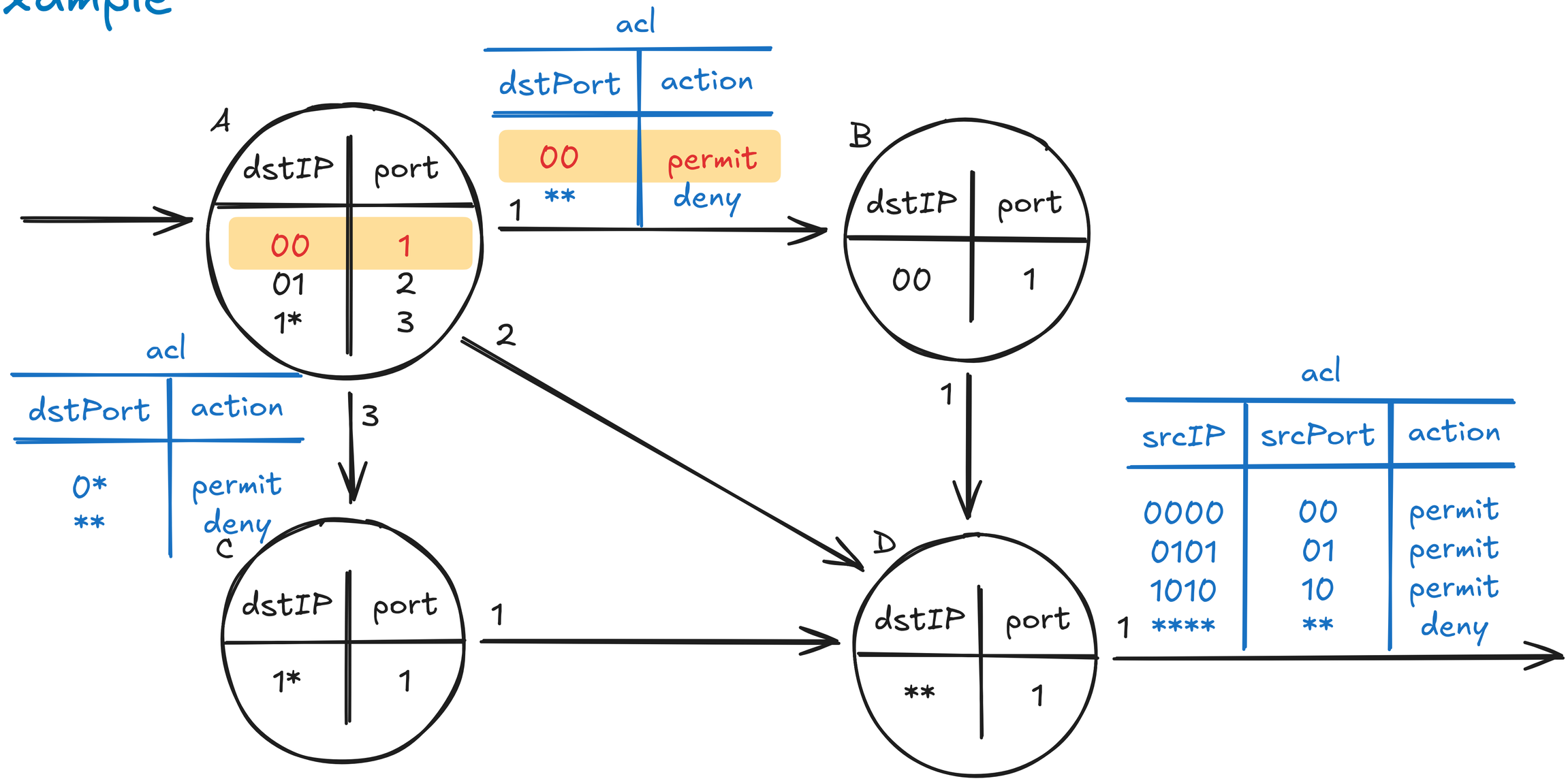
Limitations of BDD in network verification

- Redundant nodes
- Deep recursions
- Explosion of atoms

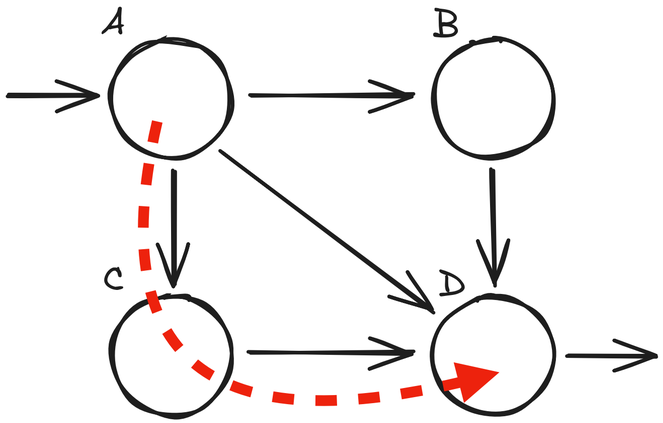
Example


acl
dstPort
action
00
permit
**
deny
A
dstIP
port
00
1
01
2
1*
3
1
B
dstIP
port
00
1
2
3
acl
dstPort
action
0*
permit
**
deny
C
dstIP
port
1*
1
1
D
dstIP
port
**
1
1
acl
srcIP
srcPort
action
0000
00
permit
0101
01
permit
1010
10
permit
****
**
deny
1
srcIP: 4bits
dstIP: 2bits
srcPort: 2bits
dstPort: 2bits

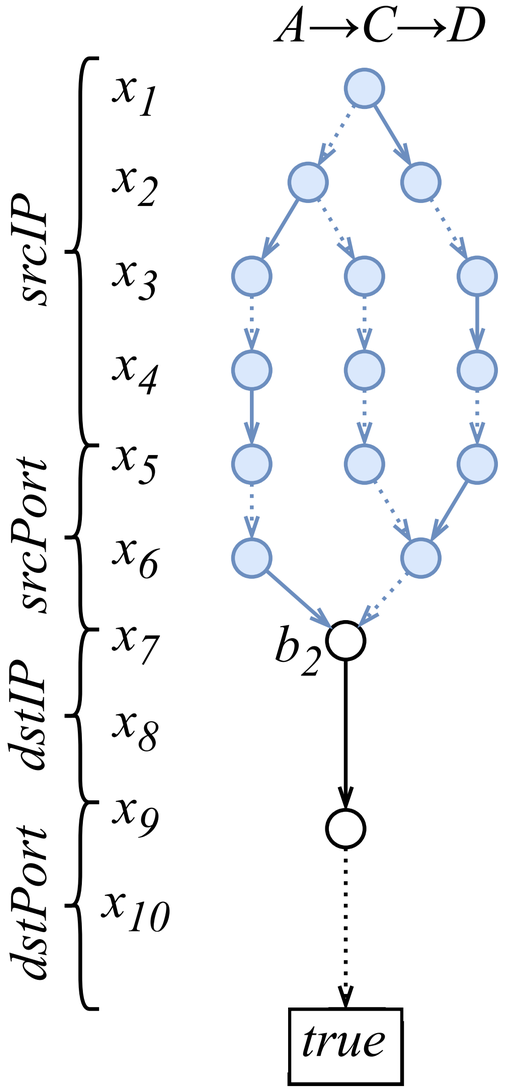
Example


acl
dstPort
action
00
permit
**
deny
1
A
dstIP
port
00
1
01
2
1*
3
B
dstIP
port
00
1
2
3
1
acl
dstPort
action
0*
permit
**
deny
C
dstIP
port
1*
1
1
D
dstIP
port
**
1
acl
srcIP
srcPort
action
0000
00
permit
0101
01
permit
1010
10
permit
****
**
deny
1

#1 Redundant nodes

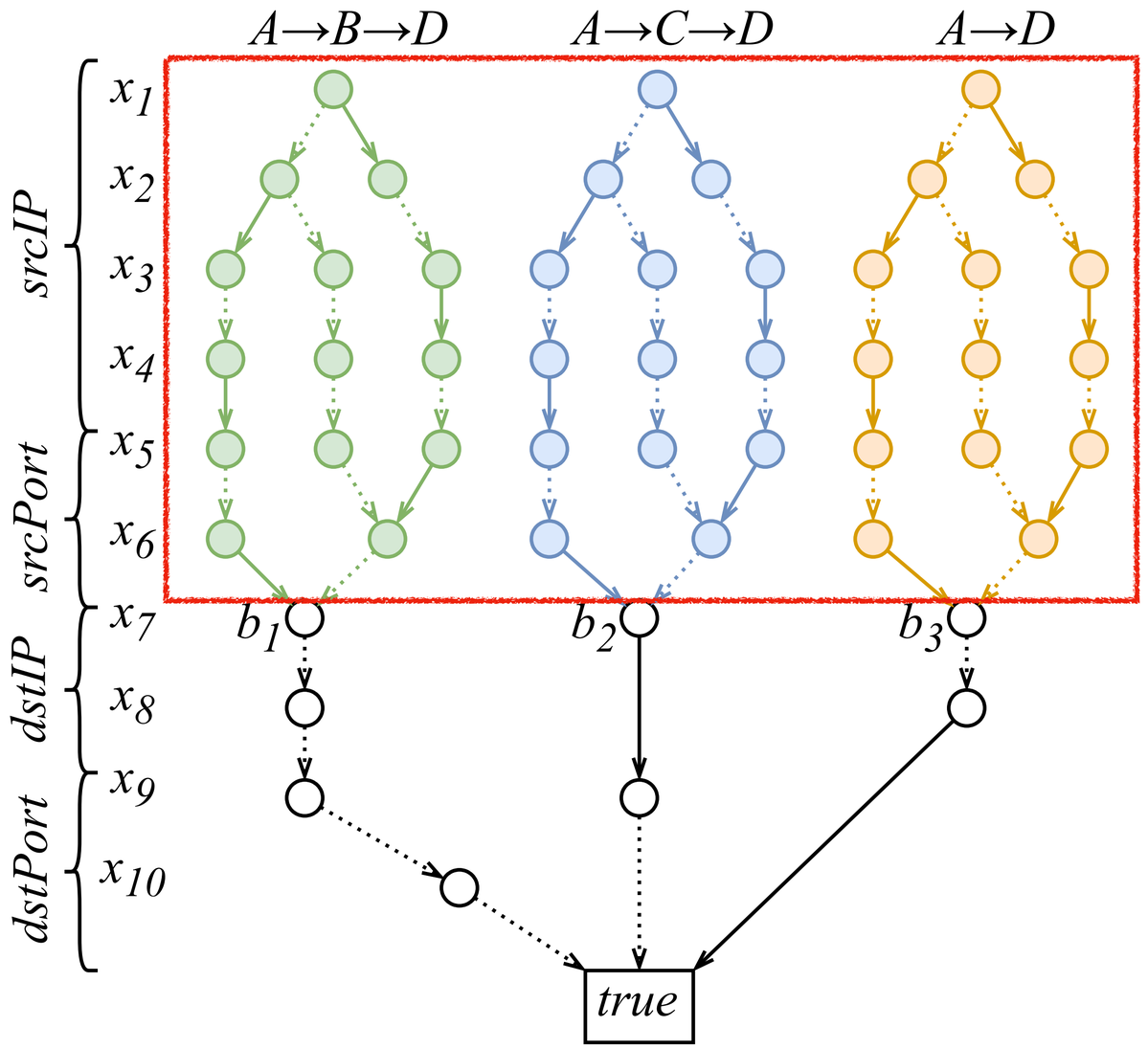


Path	srcIP	srcPort	dstIP	dstPort
	0101	01	1*	0*
A-C-D	0000	00	1*	0*
	1010	10	1*	0*



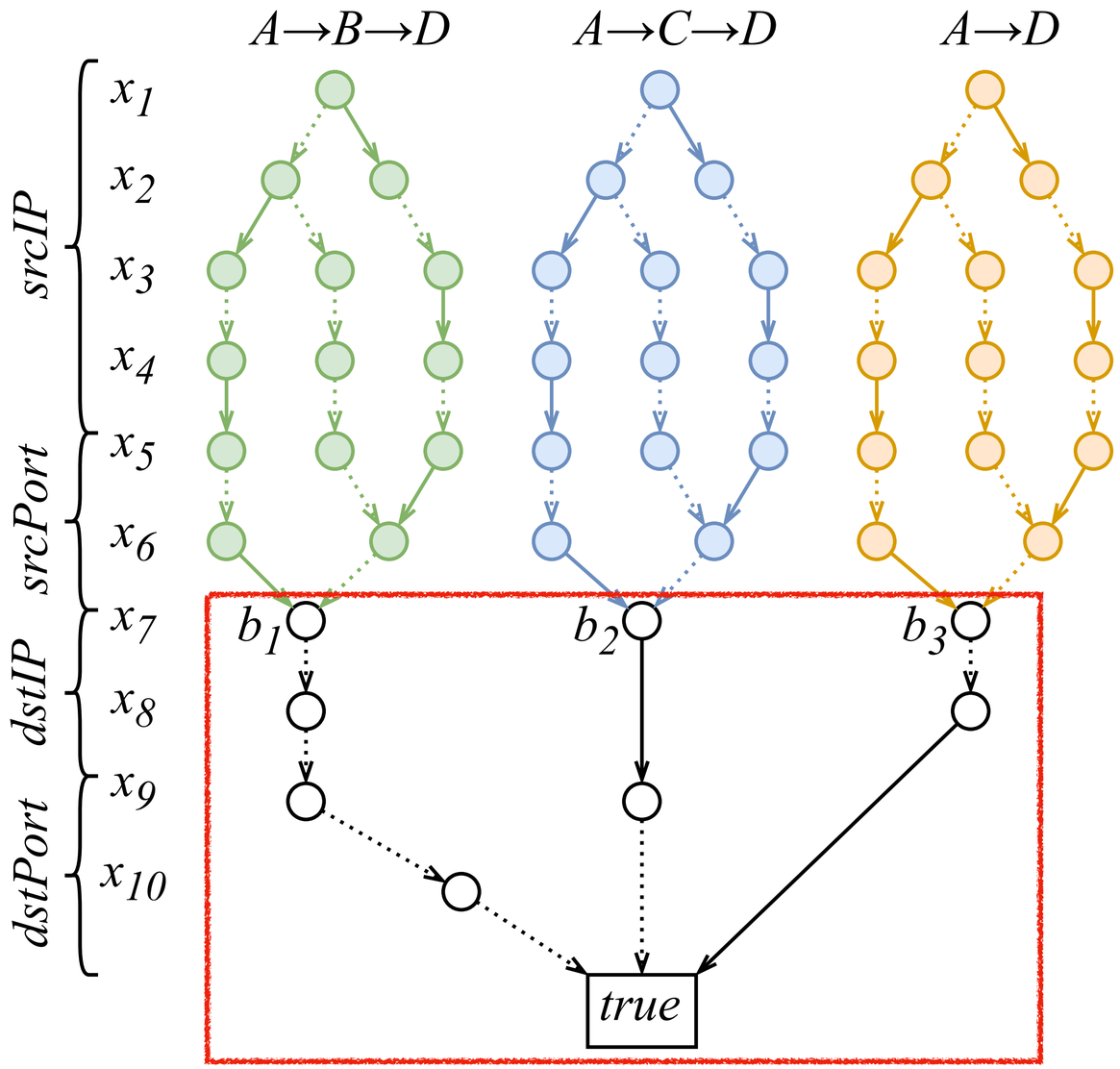
#1 Redundant nodes

Path	srcIP	srcPort	dstIP	dstPort
A-C-D	0101	01	1*	0*
	0000	00	1*	0*
	1010	10	1*	0*
A-B-D	0101	01	00	00
	0000	00	00	00
	1010	10	00	00
A-D	0101	01	01	**
	0000	00	01	**
	1010	10	01	**

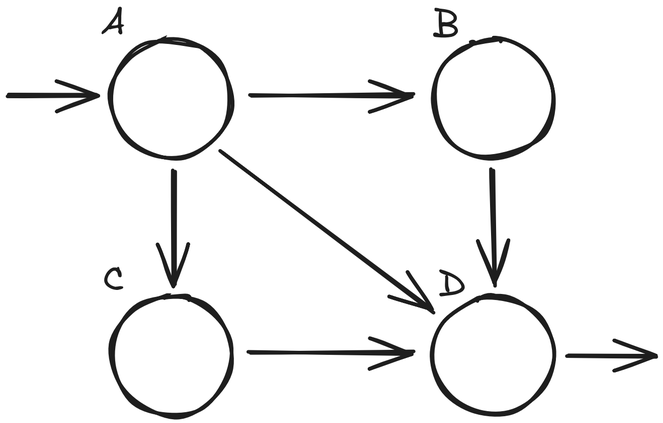


#1 Redundant nodes

Path	srcIP	srcPort	dstIP	dstPort
A-C-D	0101	01	1*	0*
	0000	00	1*	0*
	1010	10	1*	0*
A-B-D	0101	01	00	00
	0000	00	00	00
	1010	10	00	00
A-D	0101	01	01	**
	0000	00	01	**
	1010	10	01	**

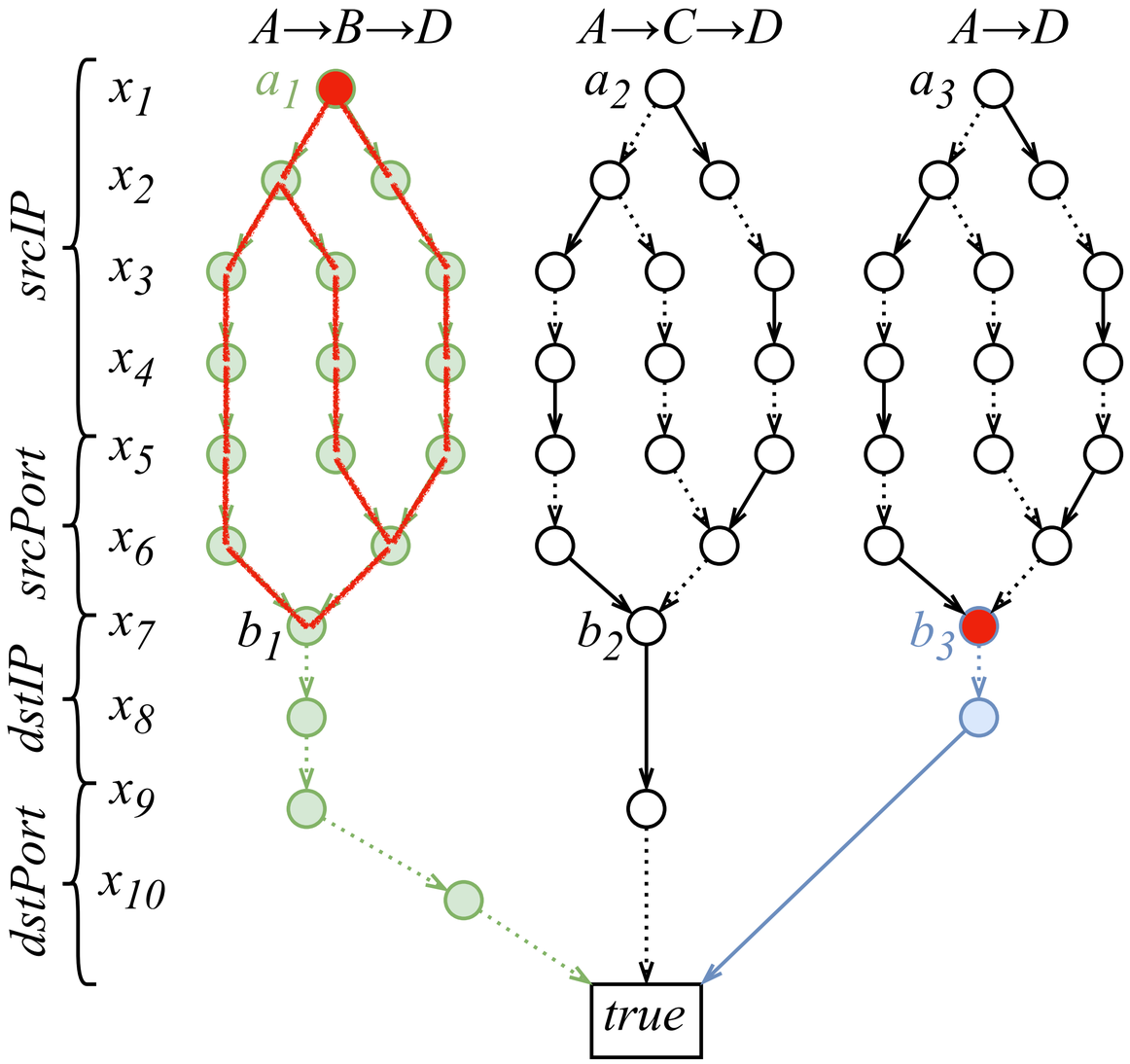


#2 Deep recursions

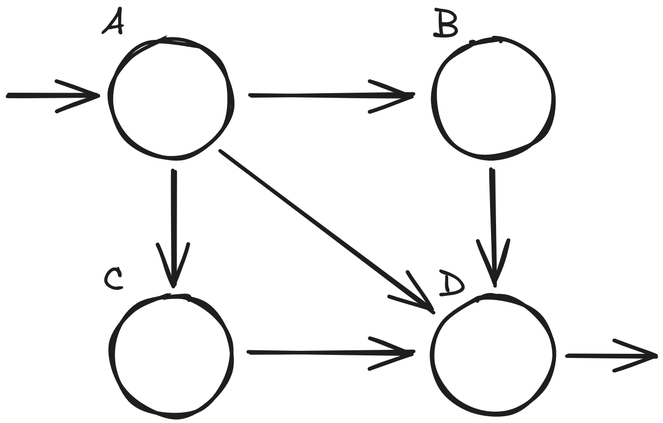


a1 ∧ b3

17 recursions needed just to align variable

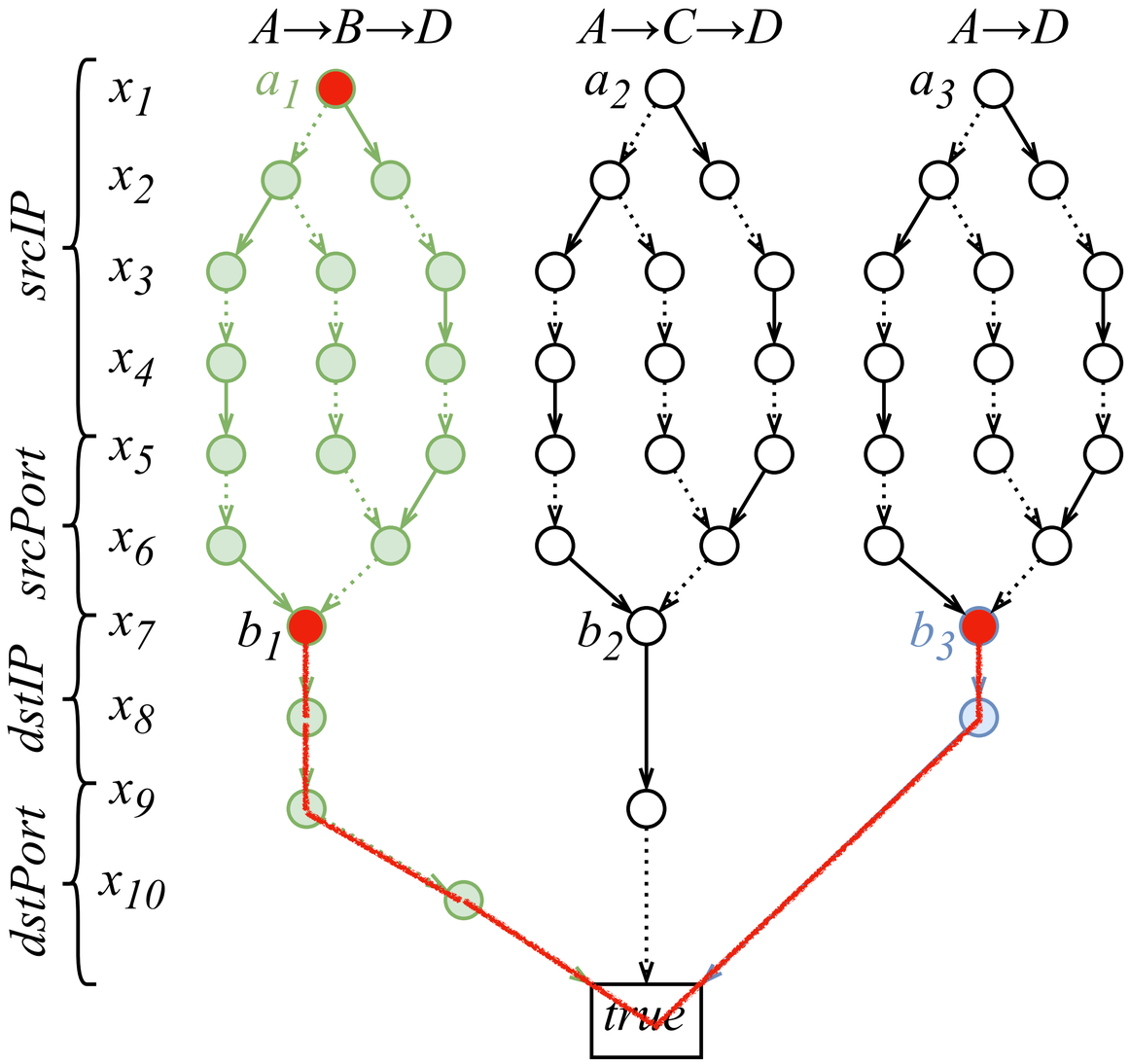


#2 Deep recursions

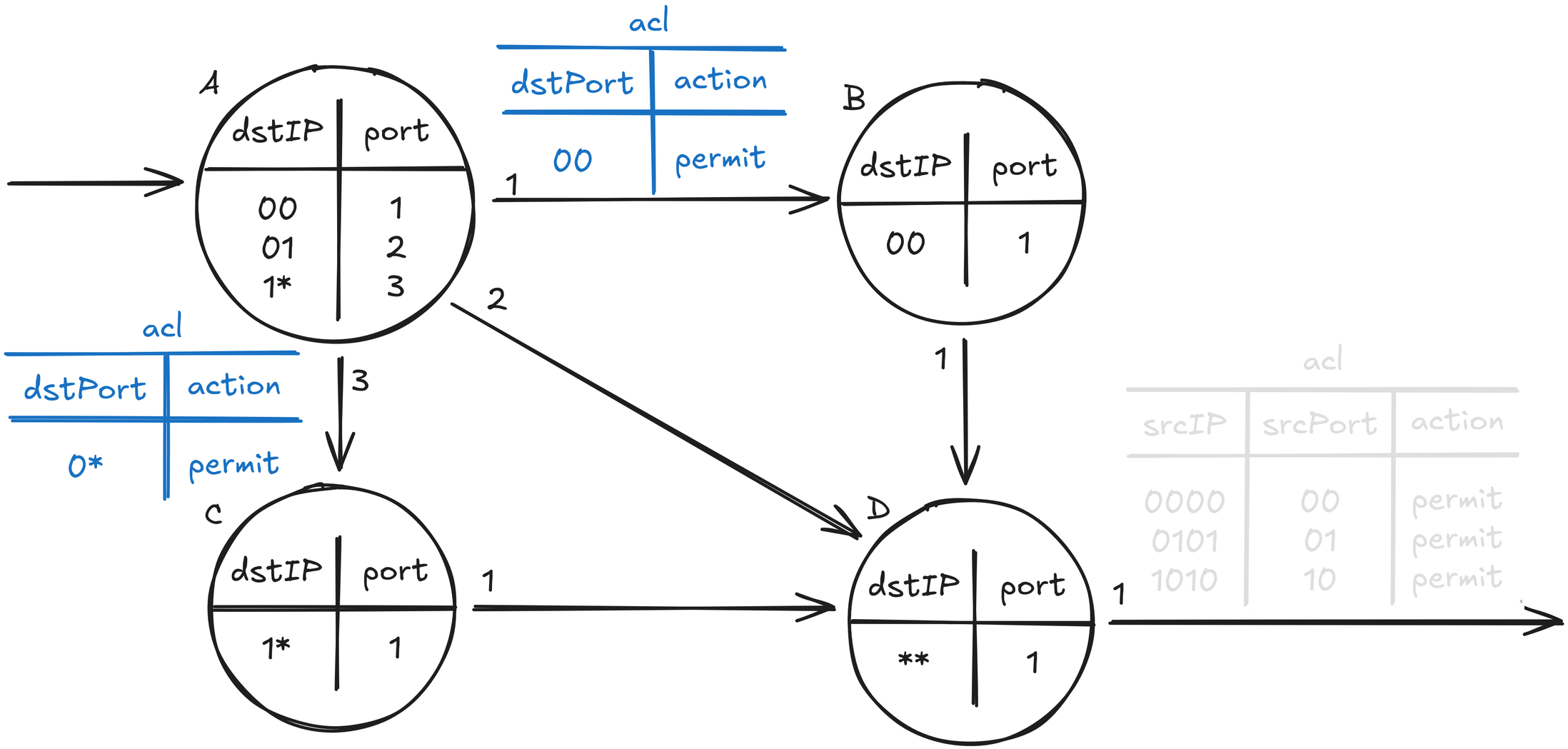


a1 ∧ b3

17 recursions needed just to align variable



#3 Explosion of atoms



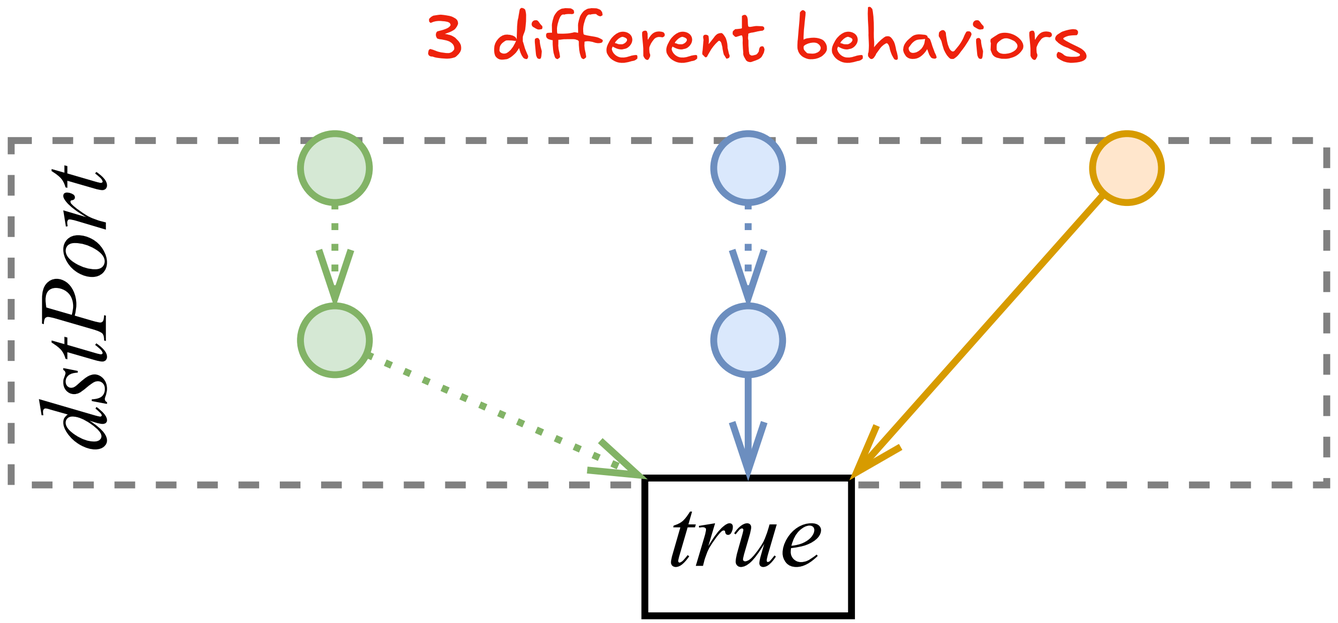
Behaviors in dstPort field

acl 1

dstPort	action
0*	permit

acl 2

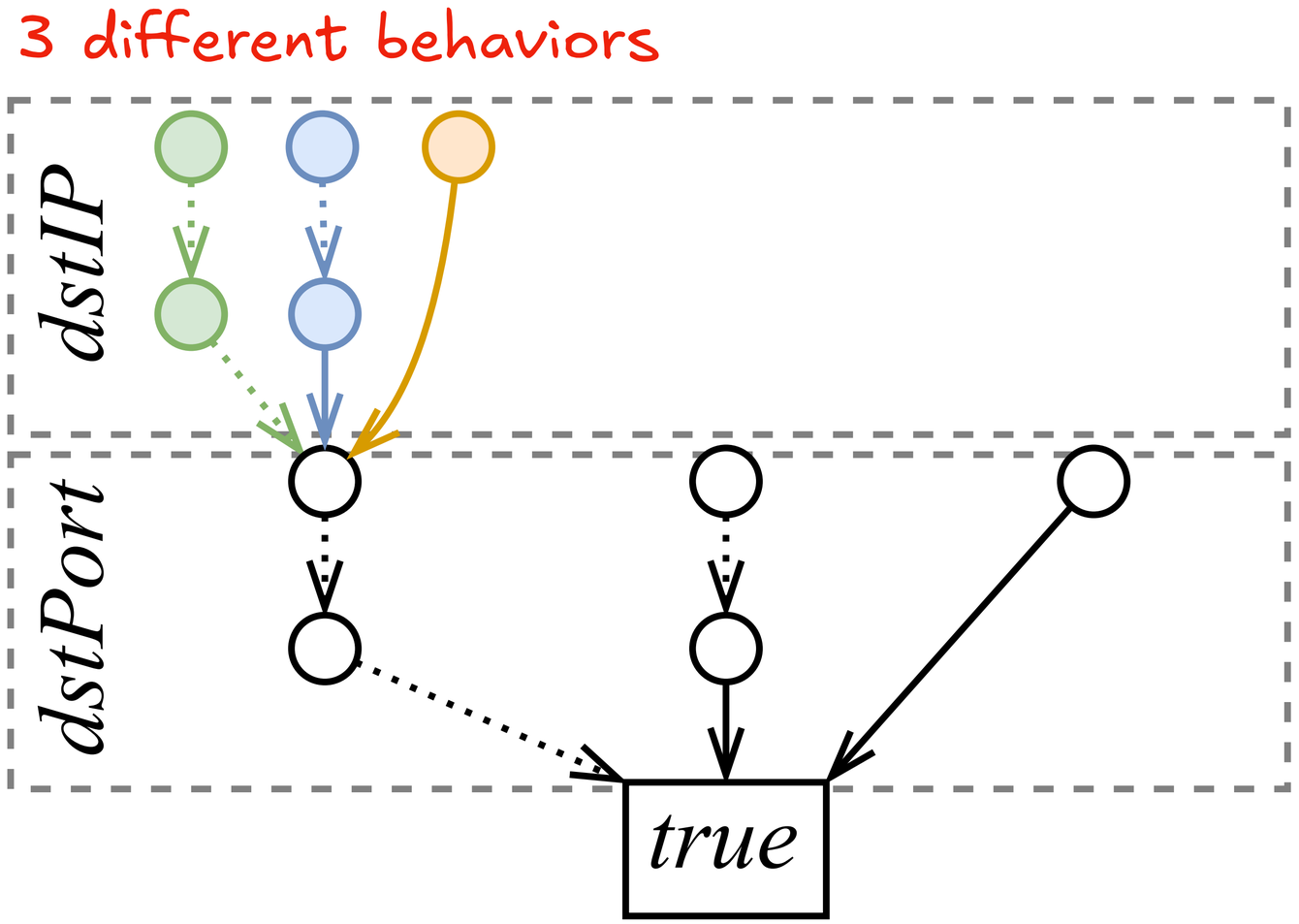
dstPort	action
00	permit



Behaviors in dstIP field

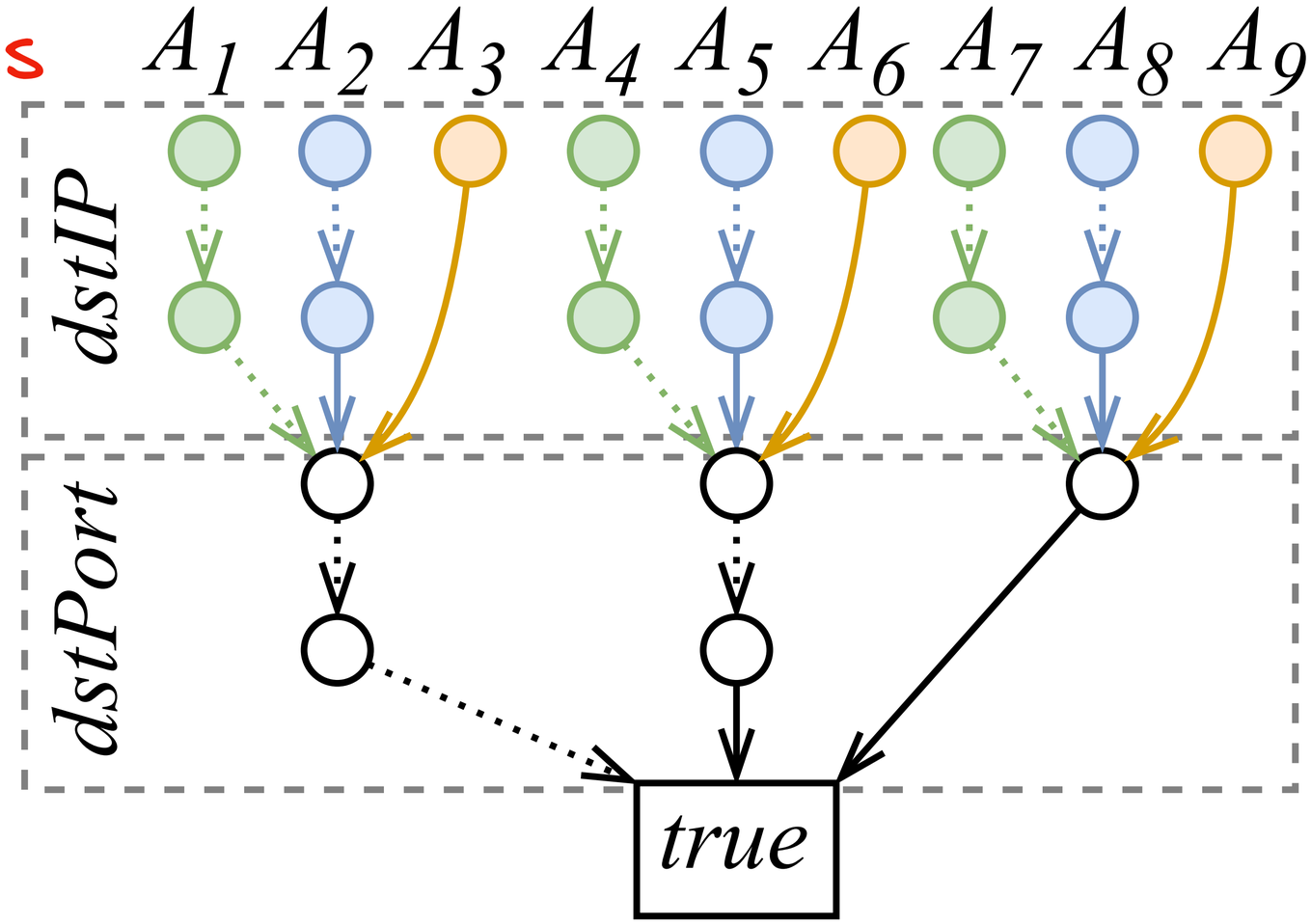
fib

dstIP	path
00	A-B-D
01	A-D
1*	A-C-D



Behaviors in dstIP + dstPort fields

3 * 3 = 9 atoms

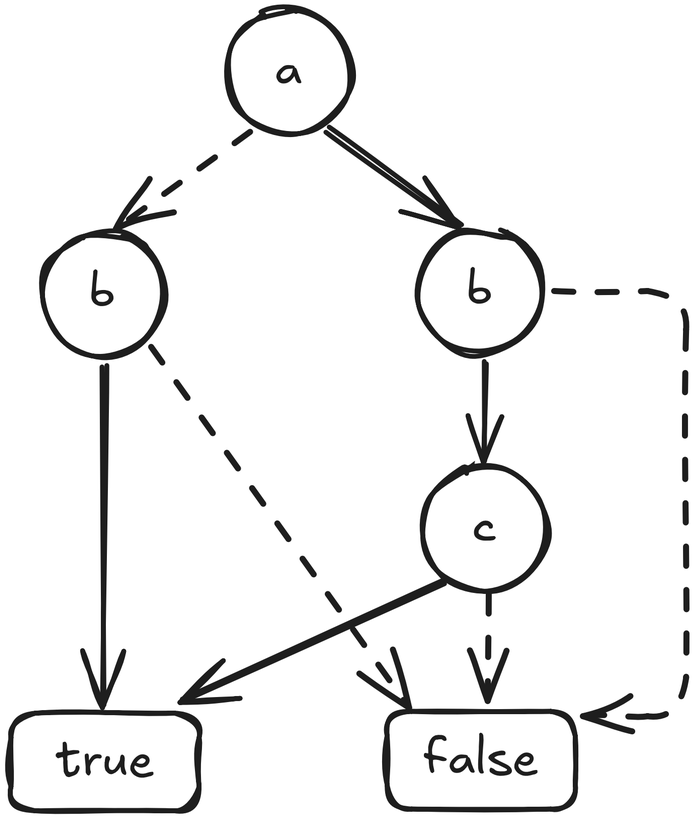


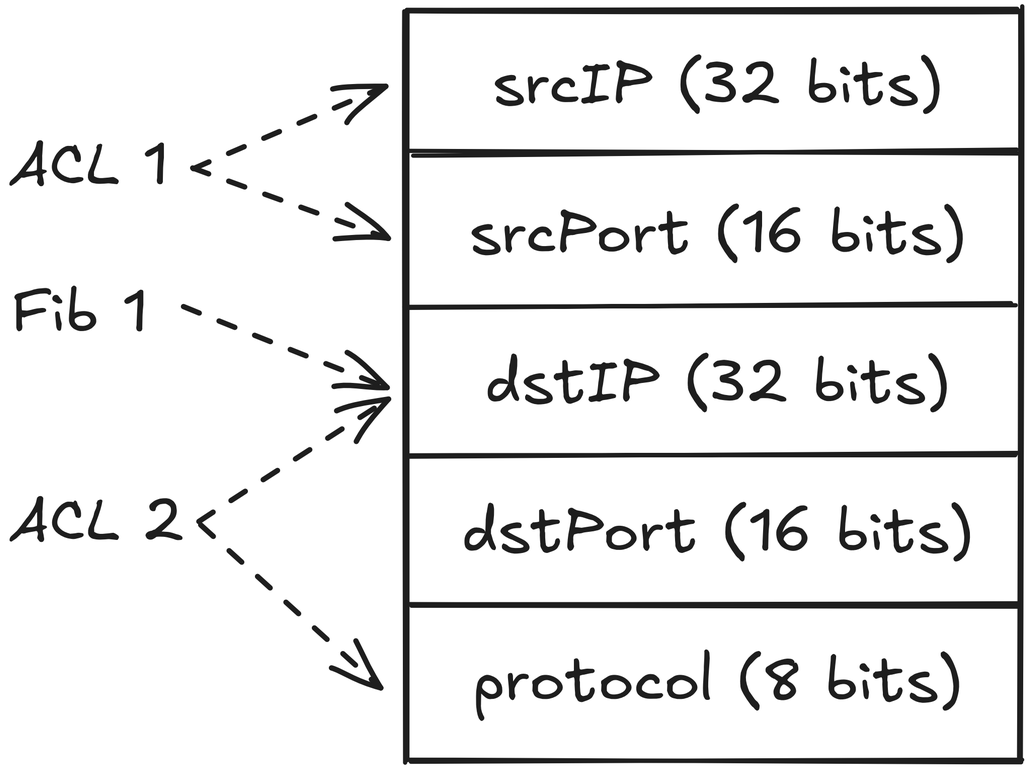
Reason for the limitations above

bit-level decisions

mismatch

field-level decisions





Network Decision Diagram

A decision diagram for network verification

#1 Decoupling fields

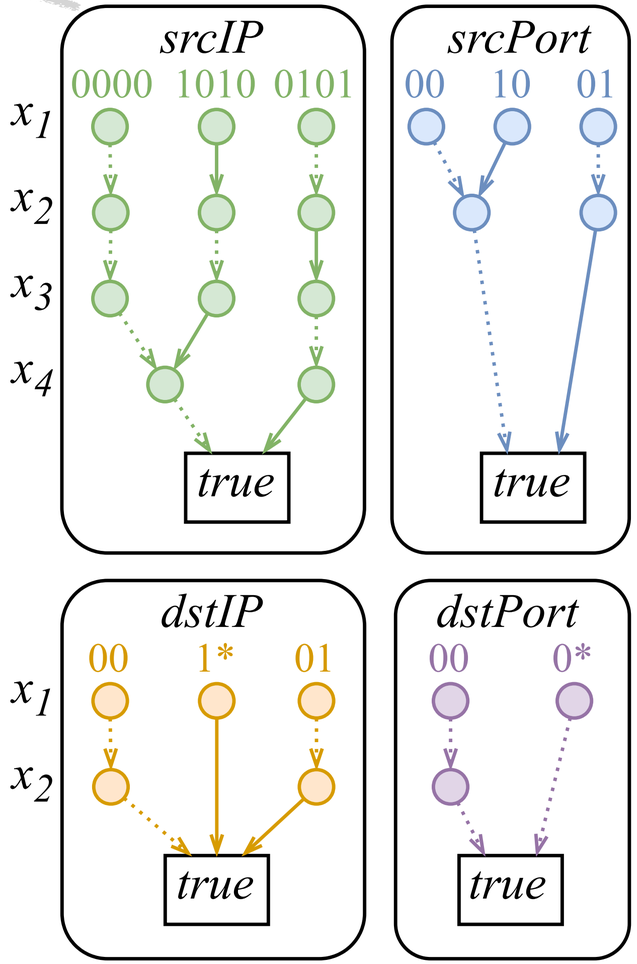
no redundant nodes

acl

srcIP	srcPort	action
0000	0000	permit
0101	0101	permit
1010	1010	permit

fib

dstIP	path
00	A-B-D
01	A-C-D
1*	A-D



4 namespaces

acl1

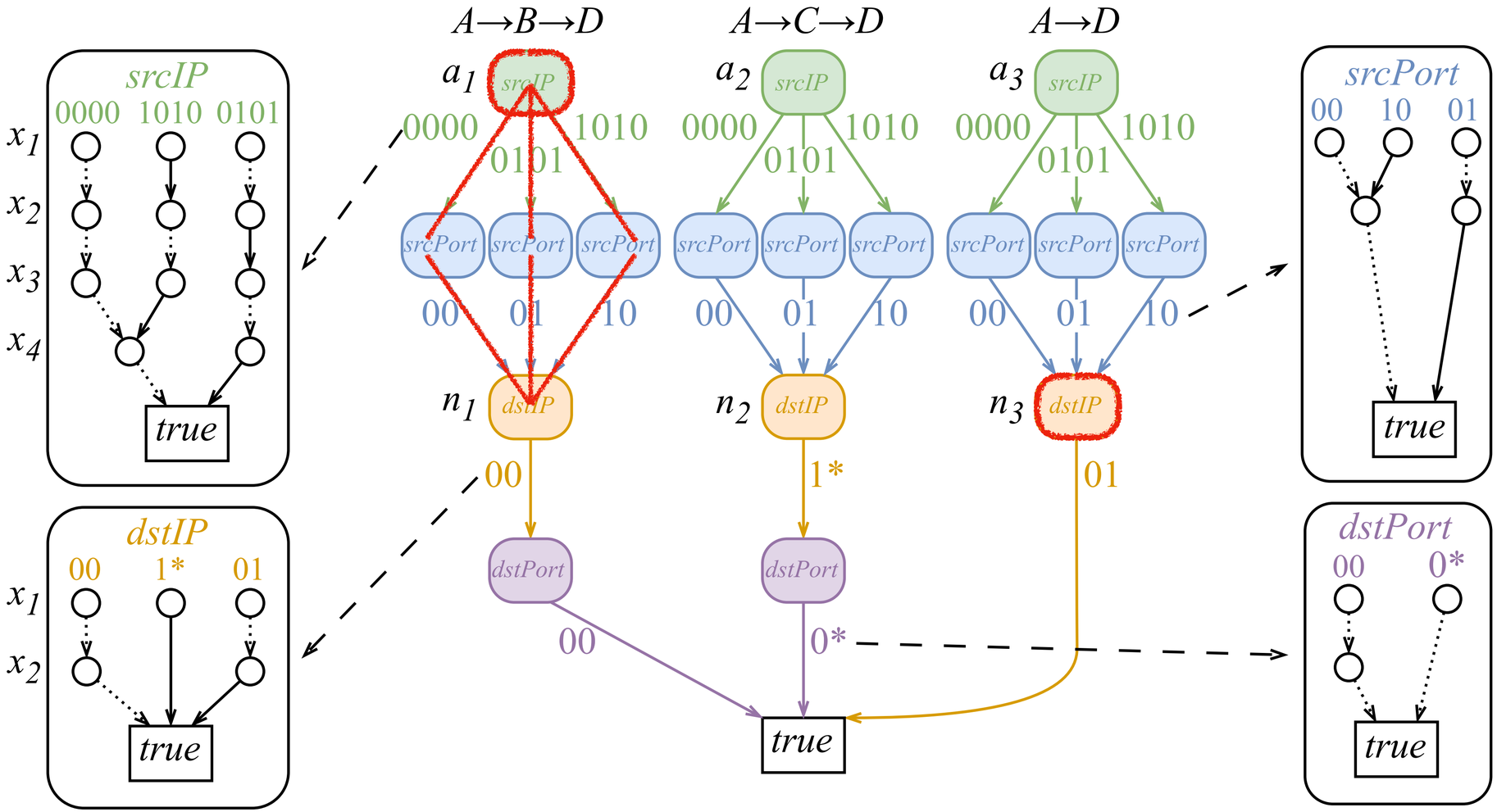
dstPort	action
0*	permit

acl2

dstPort	action
00	permit

#2 Assembling fields

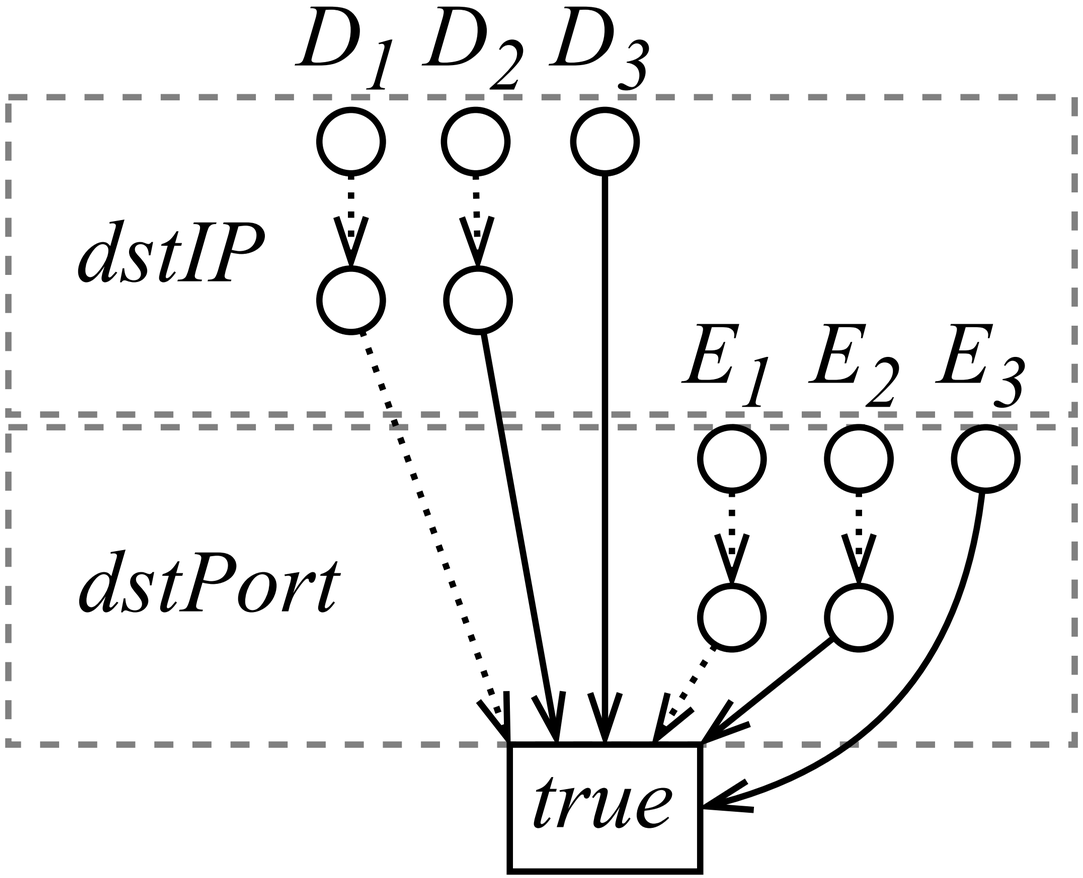
✅ avoid deep recursions



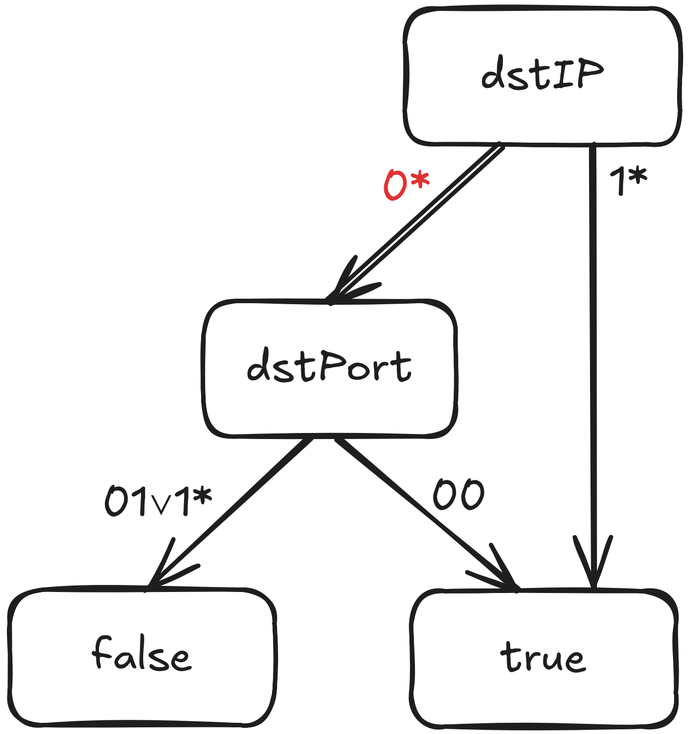
#3 Atomizing fields

no atom explosions

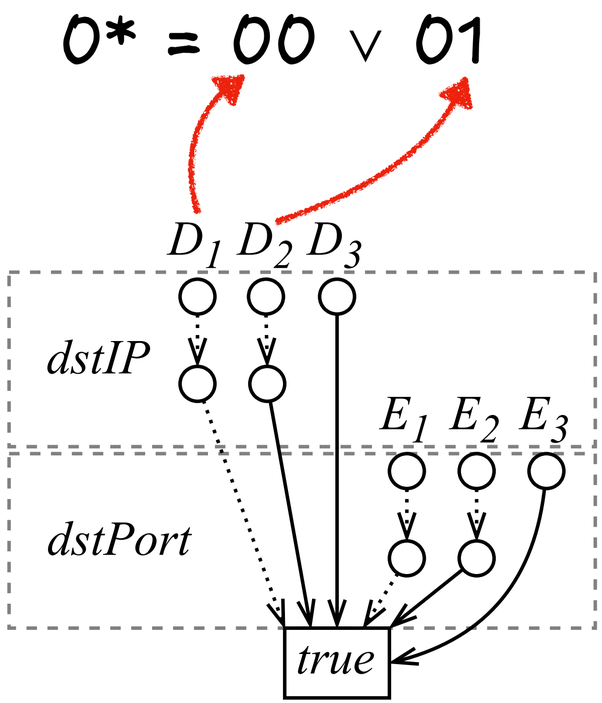
3 + 3 = 6 atoms



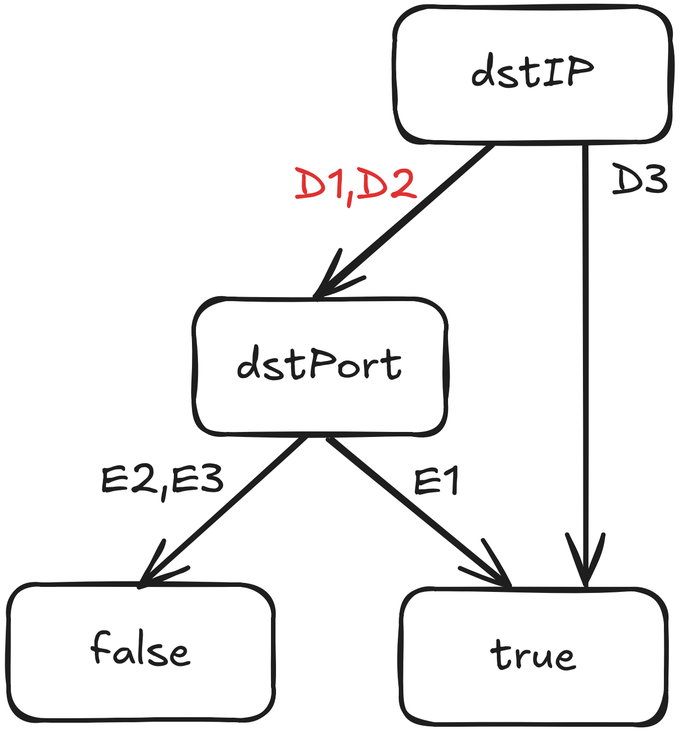
#3 Atomizing fields



NDD



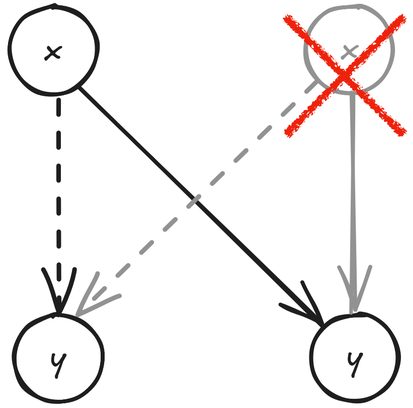
Per-field atoms



Atomized NDD

Reduced Ordered NDD

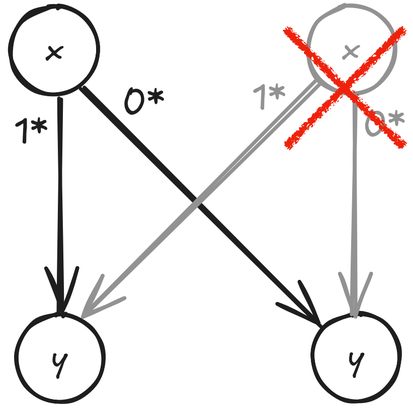
BDD



Non-unique nodes

Redundant nodes

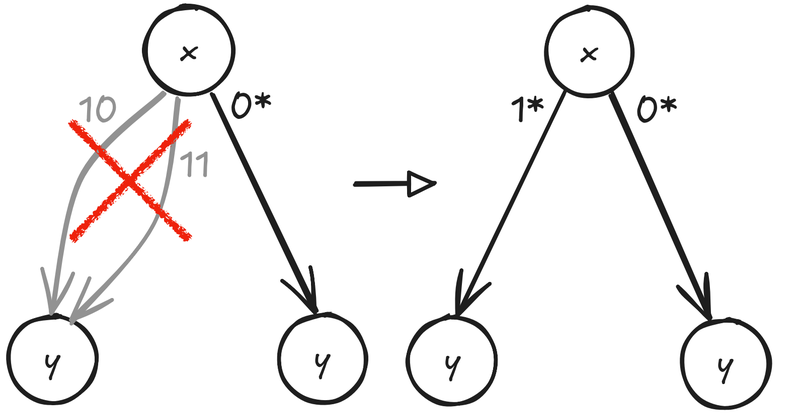
NDD



Non-unique nodes

Redundant nodes

New



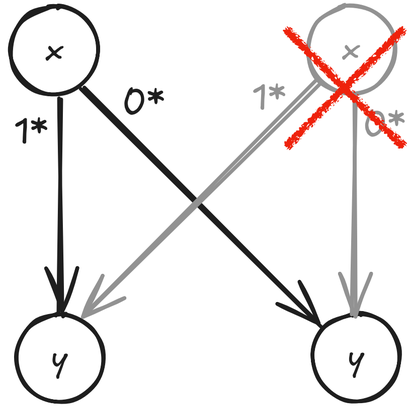
Redundant edges

Reduced Ordered NDD

Theorem: Canonicity of NDD

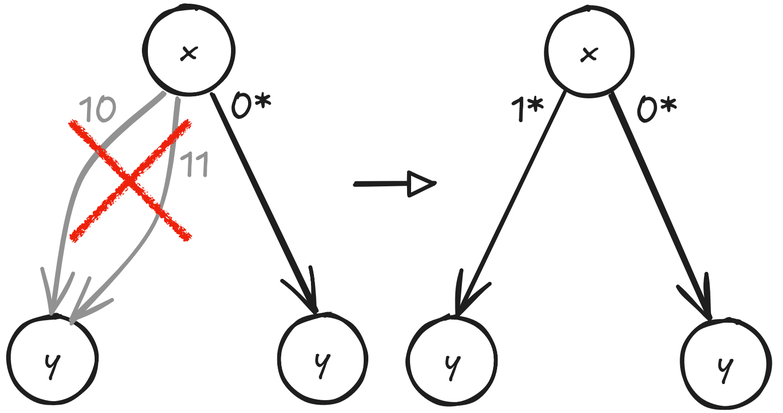
For any boolean function, if the ordering of variables is fixed, there is exactly one NDD node that can represent the function.

NDD



Non-unique nodes

Redundant nodes



Redundant edges

Technical challenges

- Memory cost due to BDD labels

  -> remove edges to the `false` node

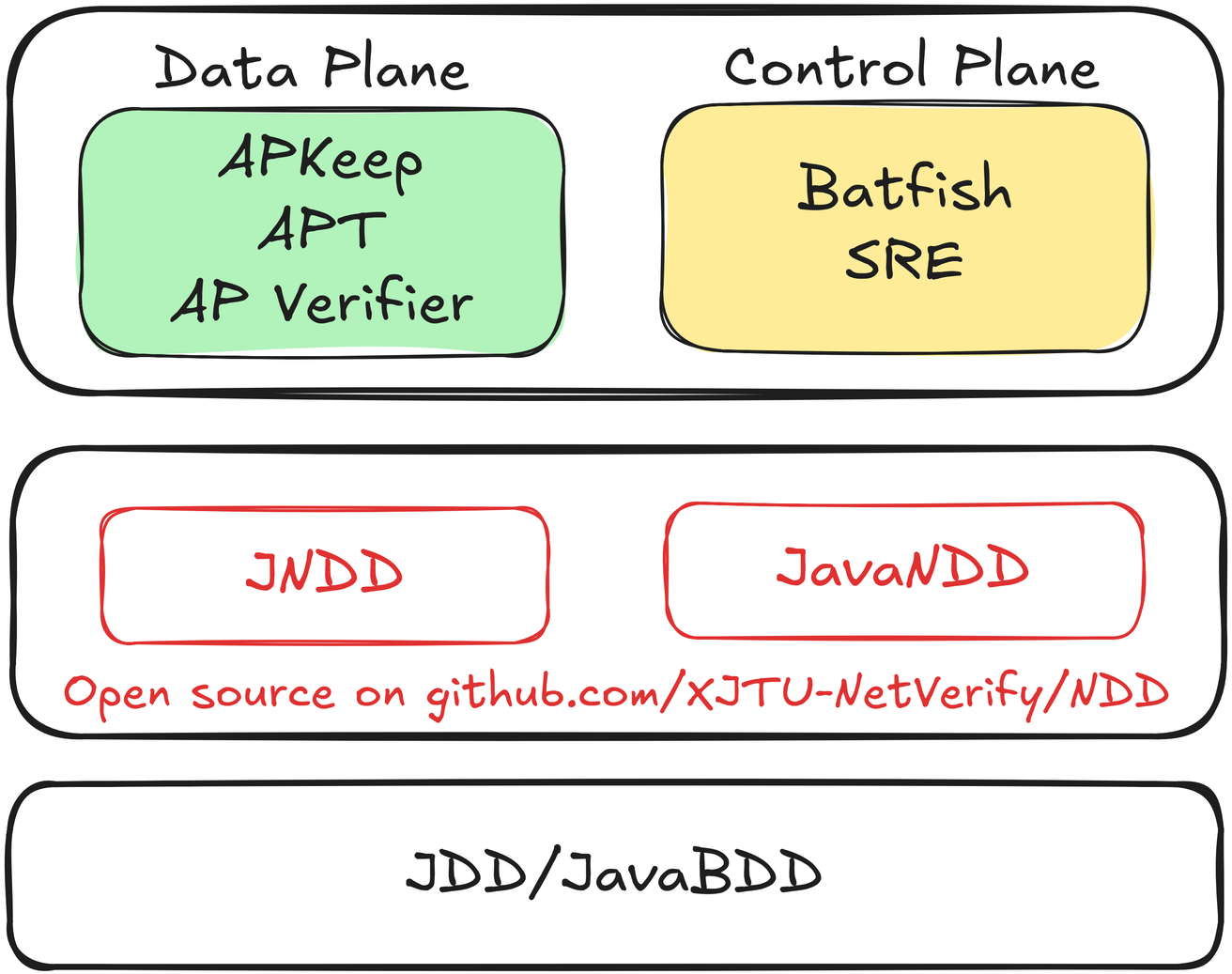
- Memory cost due to large number of edges

  -> find and merge redundant edges

- Time overhead during configuration updates

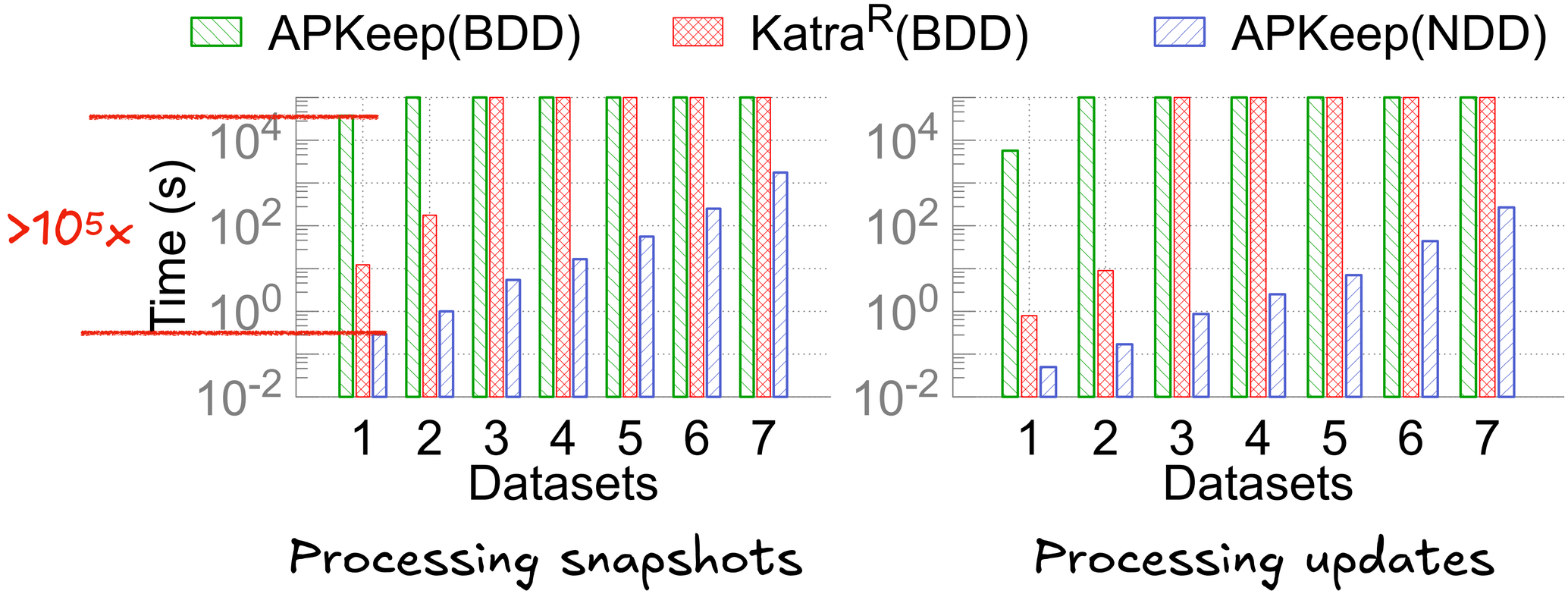
  -> update atoms incrementally

Implementation



Data plane verification - Time

VXLAN based virtualized Data Center Networks[1][2]

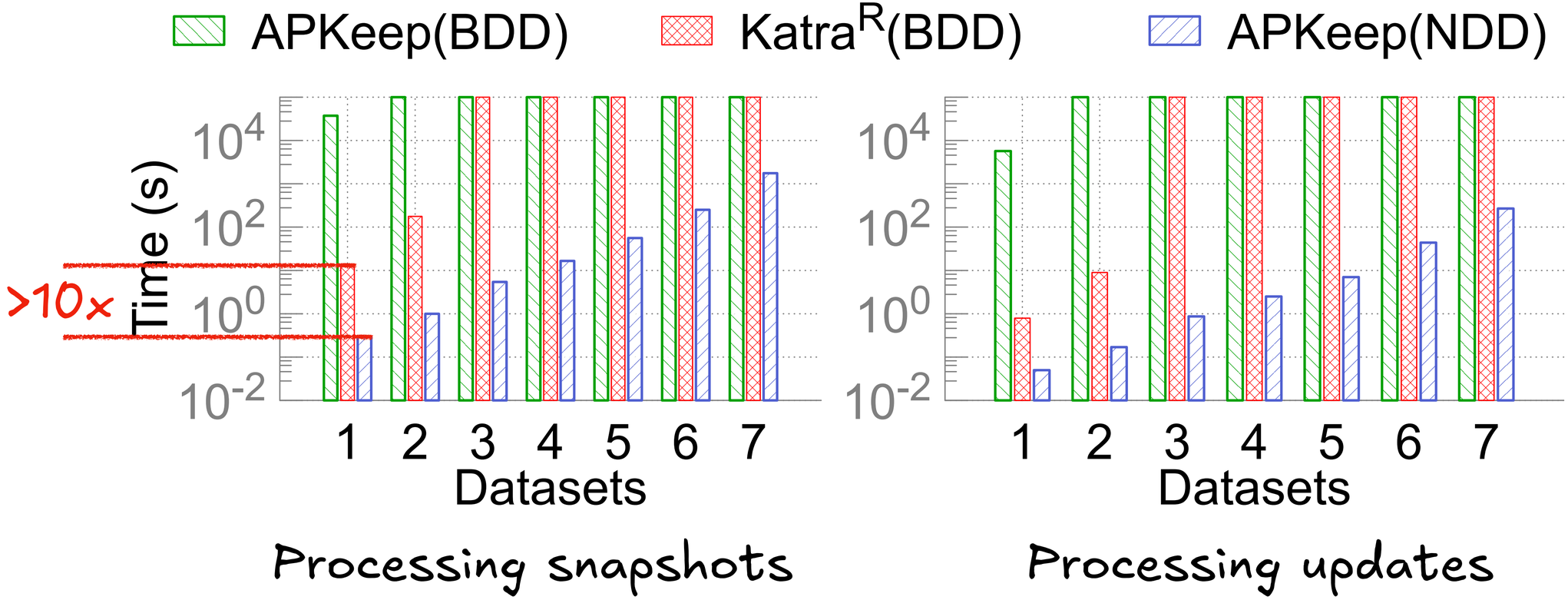


[1] Modular data plane verification for compositional networks. CoNEXT'23

[2] Fast configuration change impact analysis for network overlay data center networks. TON'21

Data plane verification - Time

VXLAN based virtualized Data Center Networks[1][2]



[1] Modular data plane verification for compositional networks. CoNEXT'23

[2] Fast configuration change impact analysis for network overlay data center networks. TON'21

Data plane verification - Atoms

VXLAN based virtualized Data Center Networks[1][2]

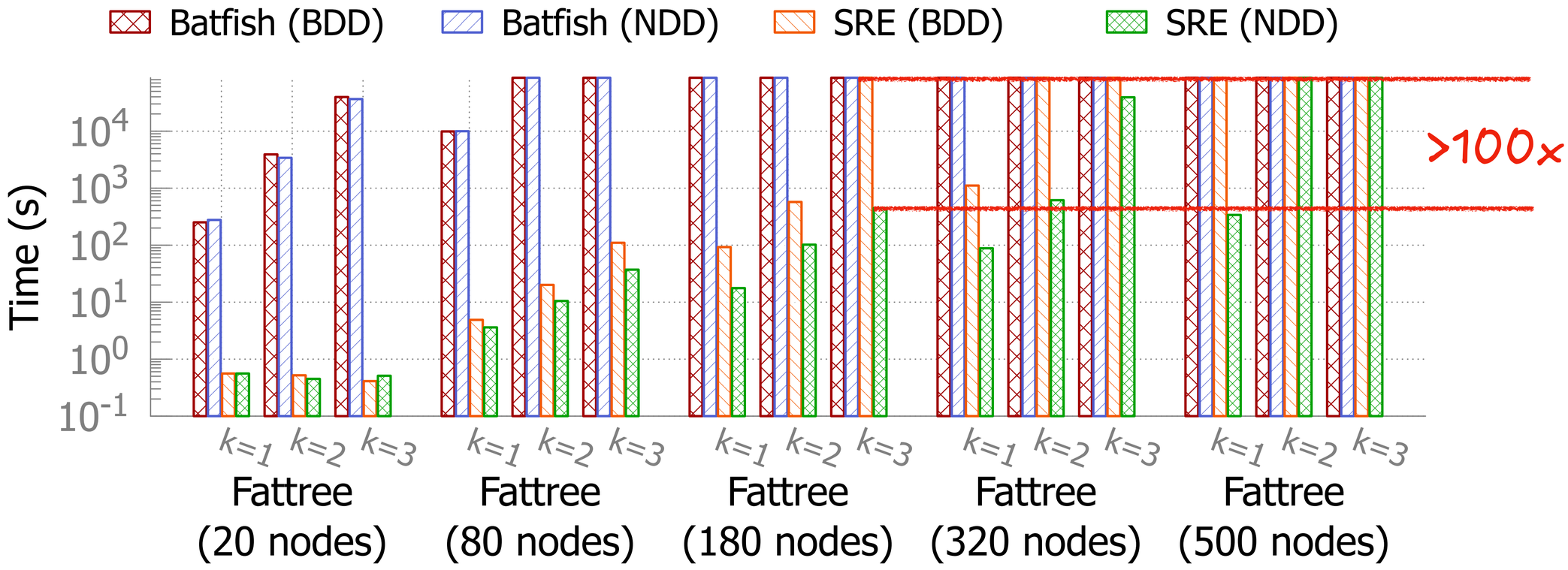
	#Atoms		
#Leaf nodes	APKeep (BDD)	Katra (BDD)	APKeep (NDD)
6	28077	5467	112
10	TO	25542	195
20	TO	TO	483
50	TO	TO	1173
100	TO	TO	2321
200	TO	TO	4623
500	TO	TO	11544

[1] Modular data plane verification for compositional networks. CoNEXT'23

[2] Fast configuration change impact analysis for network overlay data center networks. TON'21

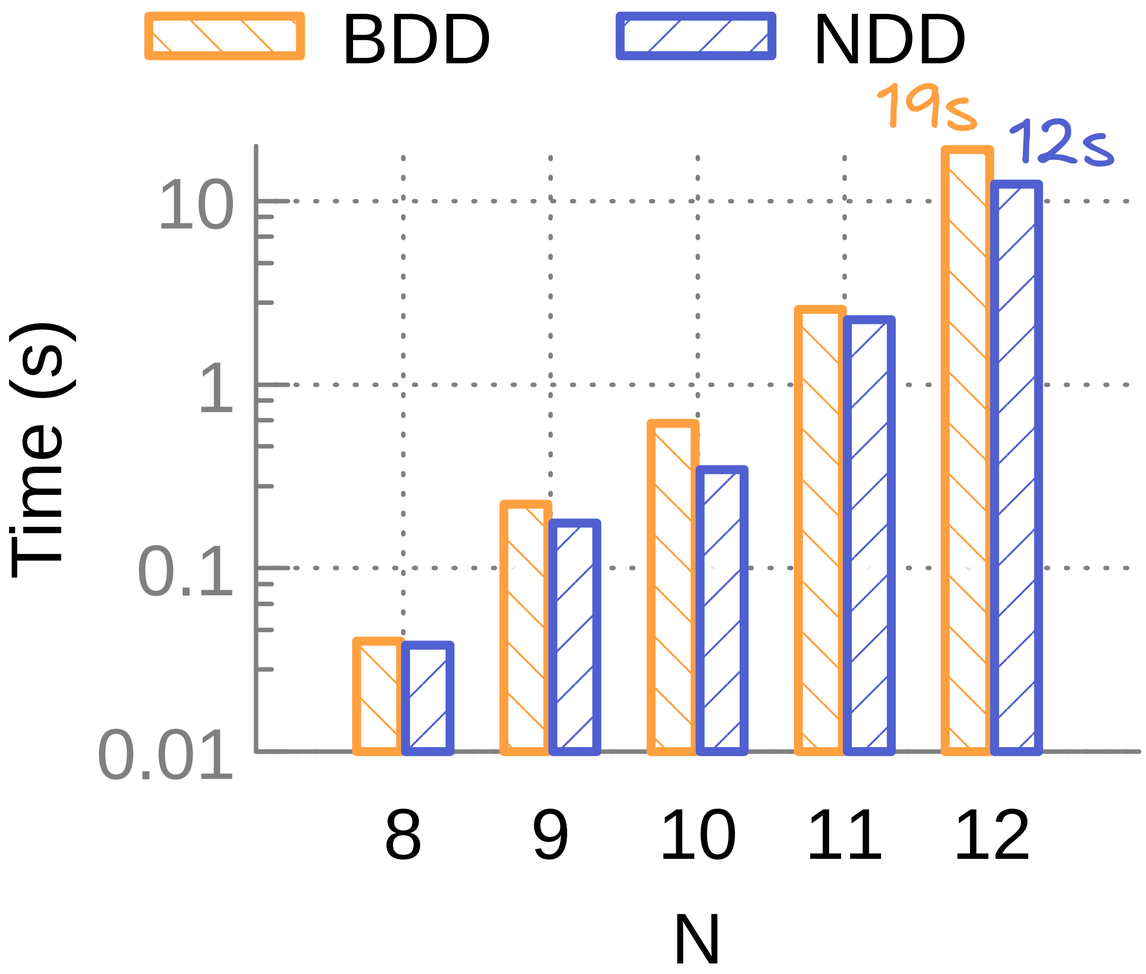
Control plane verification

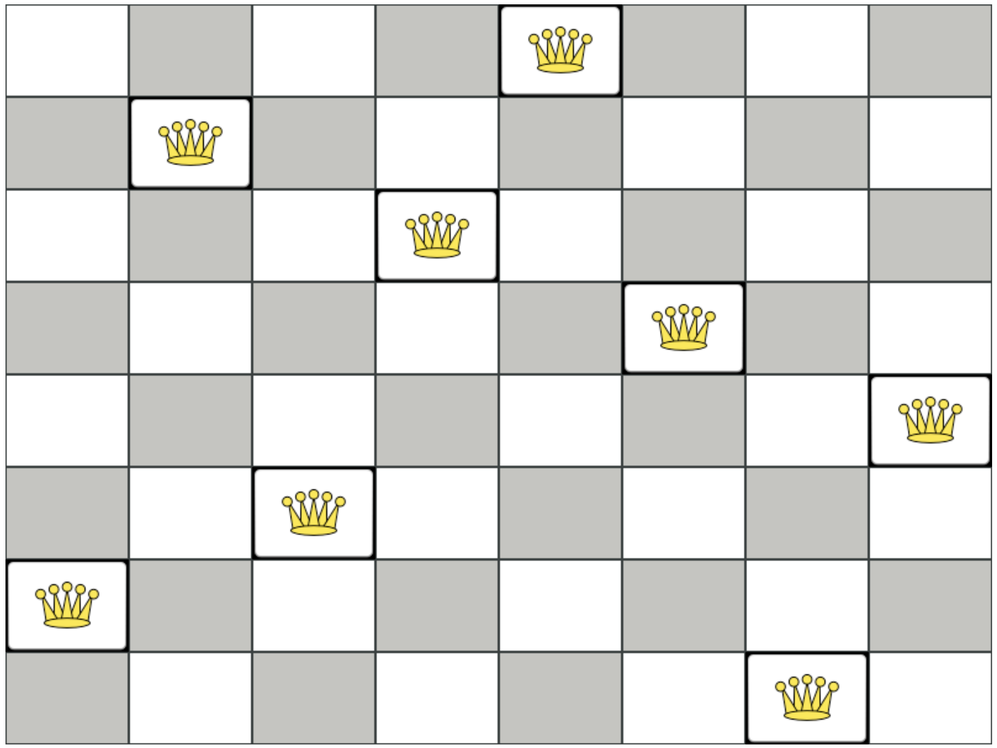
Fattree networks running BGP[3]



[3] Symbolic router execution. SIGCOMM'22

Beyond network verification





N-Queens

Conclusion

- Limitations of BDD in network verification
  - redundant nodes
  - deep recursions
  - explosion of atoms
- We proposed the Network Decision Diagram which is customized for network verification.
- NDD performs better in data/control plane verification.

Questions & Answers

NDD: A Decision Diagram for Network Verification

Zechun Li[1], Peng Zhang[1], Yichi Zhang[1], Hongkun Yang[2]
[1]Xi'an Jiaotong University, [2]Google

https://github.com/XJTU-NetVerify/NDD

nsdi'25