Towards Example-guided Network Synthesis

Haoxian Chen, University of Pennsylvania
Anduo Wang, Temple University
Boon Thau Loo, University of Pennsylvania

Network management is challenging

- Low-level, vendor-specific configurations
  - complex (~1000 lines in a Cisco router)
  - error-prone (AWS outage 2017)

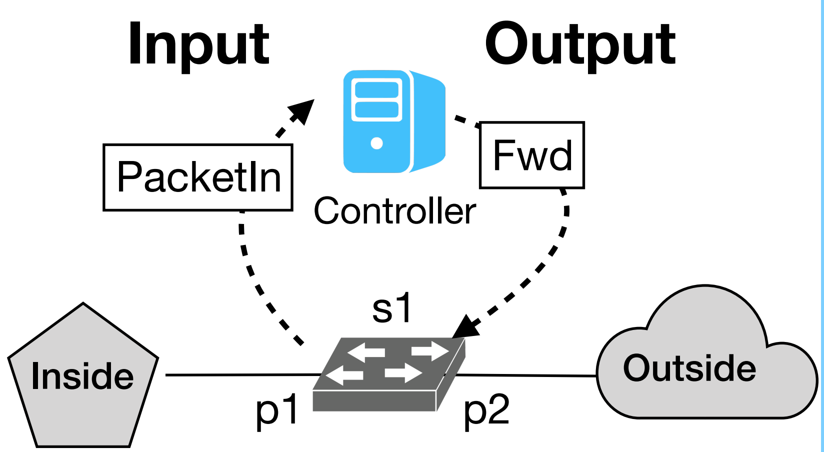
- Alternative: Software-defined networking (SDN)
  - mitigates distributed complexity by centralized view
  - but controller programs are still complicated to implement
  - high-level Domain-Specific Languages (DSL) reduce lines of codes, but have steep learning curve ([Frenetic], [Pyretic], [FlowLog])

Our solution: networking by input-output examples

1. Network operator provides some input-output (I/O) pairs
   - this work focus on I/O of the controller program in SDN
2. Computer automatically synthesizes a **program**

Example: stateful firewall

Synthesizer interface



The underlying network

Synthesize a program

- packet 10.0.0.1 -> 172.217.11.46 → allow
- packet 172.217.11.46 -> 10.0.0.2 → block
- packet 172.217.11.46 -> 10.0.0.3 → block

- packet 119.212.8.8 -> 10.0.0.2 — block
- packet 172.217.11.46 -> 10.0.0.1 — allow

Design space

Synthesis target: controller programs v.s. data plane configurations

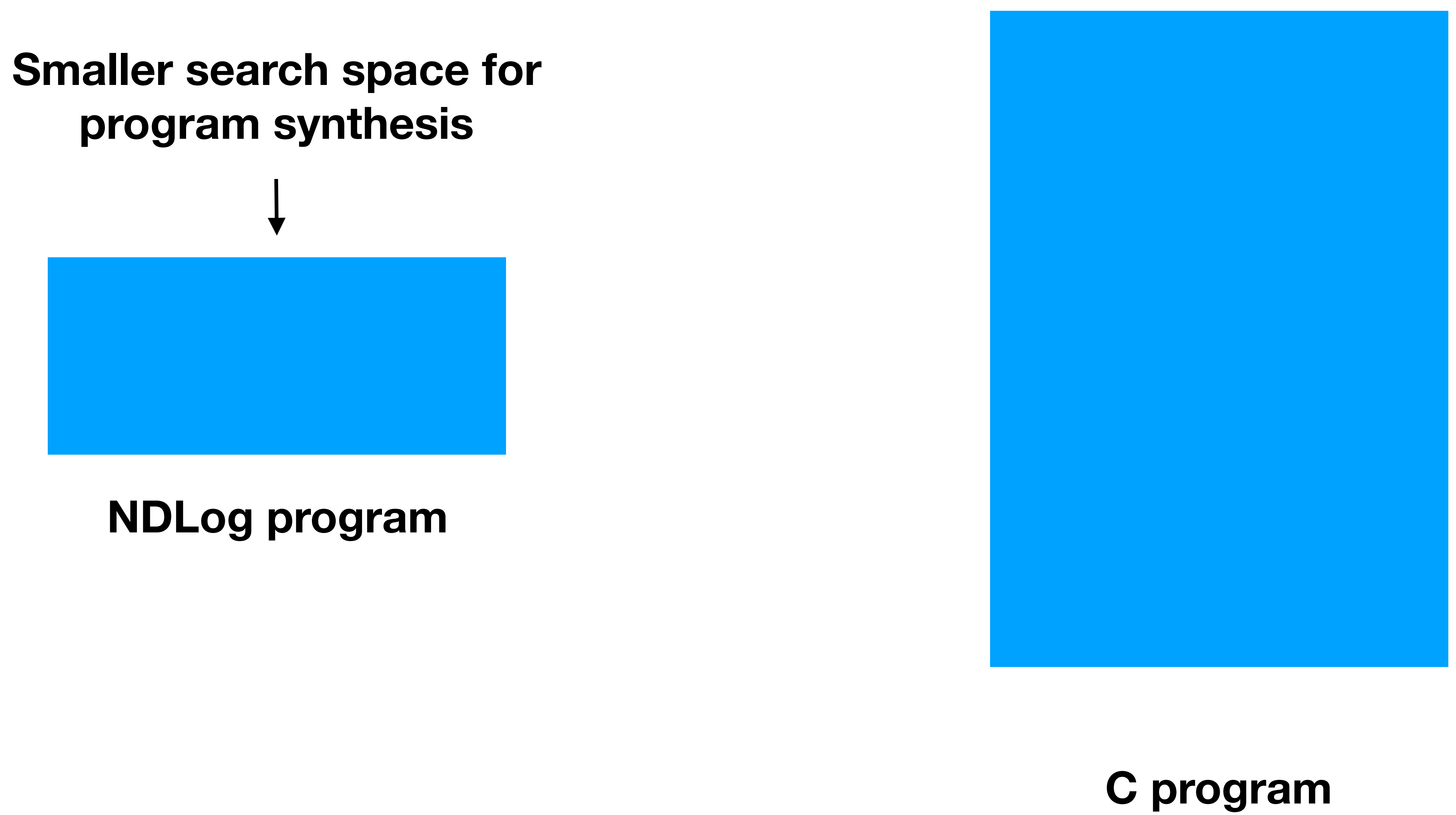
Design space

Synthesis target: controller programs

- Understandable to human
- Verifiable
- Compose with other programs to form complex features [Frenetic]
- Reuse in other settings

Synthesize NDLog program

Leverage the compactness of NDLog programs



Synthesize NDLog program

NDLog evaluates each rule independently

so that we can synthesize one rule at a time

Background: NDLog

- One of the Logic-programming family.
- Inputs and Outputs are organized as structured tables.
- Program consists of a set of rules.
- Rules tranform input to output

Input: packetIn

SrcIP	DstIP	InPort
10.0.0.1	10.0.0.2	1
10.0.0.3	10.0.0.2	2
10.0.0.4	10.0.0.5	1

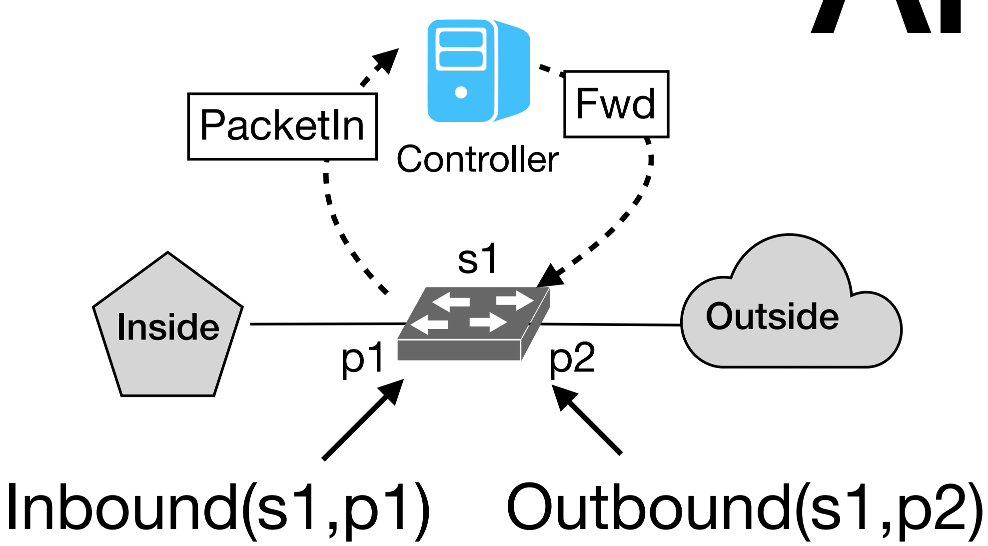
fwd(IP, Port) :-
packetIn(SrcIP, DstIP, InPort),
IP=DstIP, InPort=Port.

→

Output: fwd

IP	Port
10.0.0.2	1
10.0.0.2	2
10.0.0.5	1

Example-guided synthesis: An overview



background knowledge

Input-output	
PacketIn 10.0.0.1 -> 172.217.11.46	Fwd 10.0.0.1, port 2
PacketIn 10.0.0.1 -> 172.217.11.46	Fwd 172.217.11.46, port 1

examples

Facon the synthesizer

An NDLog program consists of a set of symbolic rules

```
Fwd(swi, dstIP, srcIP, prt) :- PacketIn(swi, srcIP, dstIP, prt),
    InBound(swi, prt).
Fwd(swi, srcIP, dstIP, prt) :- PacketIn(swi, srcIP, dstIP, prt2),
    InBound(swi, prt2), Outbound(swi, prt).
```

Symbolic Rules

Synthesis algorithm

1. Divide-and-conquer principle: one rule at a time, combine them into the final program
   - because NDLog evaluates each rule independently

2. Prune search space
   - Only search within the syntax-correct rule space

Synthesis algorithm

Find the set of rules cover all examples

Inbound	
switch 1	port 1

Outbound	
switch 1	port 2

background knowledge

Fwd(Switch, Dst, Src, Port) :-
  PacketIn(Switch, Src, Dst, Port),
  InBound(Switch, Port).

cover

Input-output examples	
PacketIn	Fwd
switch 1, 10.0.0.1 -> 172.217.11.46, port 1	switch 1, 172.217.11.46, 10.0.0.1, port 1
switch 1, 10.0.0.1 -> 172.217.11.46, port 1	switch 1, 10.0.0.1, 172.217.11.46, port 2

Synthesize individual rule

Inbound	
switch 1	port 1

Outbound	
switch 1	port 2

background knowledge

Input-output examples	
PacketIn	Fwd
switch 1, 10.0.0.1 -> 172.217.11.46, port 1	switch 1, 172.217.11.46, 10.0.0.1, port 1
switch 1, 10.0.0.1 -> 172.217.11.46, port 1	switch 1, 10.0.0.1, 172.217.11.46, port 2

relation name

variable names

? (?,?) :- ?(?,?), ?(?,?), …

Skeleton of an NDLog rule

4 possible Relation Names:
PacketIn,
Fwd,
Inbound,
Outbound

Fwd(?,?) :- PacketIn(?,?,?,?),
Inbound(?,?), Outbound(?,?).

(Order of relations within the rule body does not matter)

Synthesize individual rule

Inbound	
switch 1	port 1

Outbound	
switch 1	port 2

background knowledge

Fwd(?,?) :- PacketIn(?,?,?,?), Inbound(?,?), Outbound(?,?).

Enumerate on all possible variable instantiation, until we find a rule that covers some examples

Input-output examples	
PacketIn	Fwd
switch 1, 10.0.0.1 -> 172.217.11.46, port 1	switch 1, 172.217.11.46, 10.0.0.1, port 1
switch 1, 10.0.0.1 -> 172.217.11.46, port 1	switch 1, 10.0.0.1, 172.217.11.46, port 2

Preliminary results

Synthesis programs:

- Reachability
  - Query if any pair of nodes can reach each other in the network
- MAC learning switch
- Stateful firewall
- App-based forwarding
  - Look up forward destination by application

Preliminary results

These reductions come from two insights:
(1) factor program into individual rules
(2) type information

Program (# possible programs)	# rules tried	Time (s)
reachability (10^5)	226	0.4
MAC learning (10^6)	11	0.02
stateful firewall (10^11)	13497	72
APP-based forwarding (10^14)	28829	149

- The major bottleneck of synthesis efficiency comes from the enumerative nature
- Examples were carefully hand-crafted, in order to synthesize correct programs.

Ongoing work

- Speed up synthesis
  - model it as reinforcement problem, use heuristic to direct searching

- Automatic example generation
  - collect from network program execution traces

- Richer DSL support

Conclusion

- Propose new approach: synthesize declarative controller program using input-output examples
- Synthesis algorithm: leverage both syntactic restrictions and semantic features of declarative programs
- Proof-of-concept prototype: synthesize declarative programs with fewer than 4 relations, within 2 minutes.