Artifacts Available acm V1.1

Artifacts Evaluated acm Reusable V1.1

# Computing Precise Control Interface Specifications

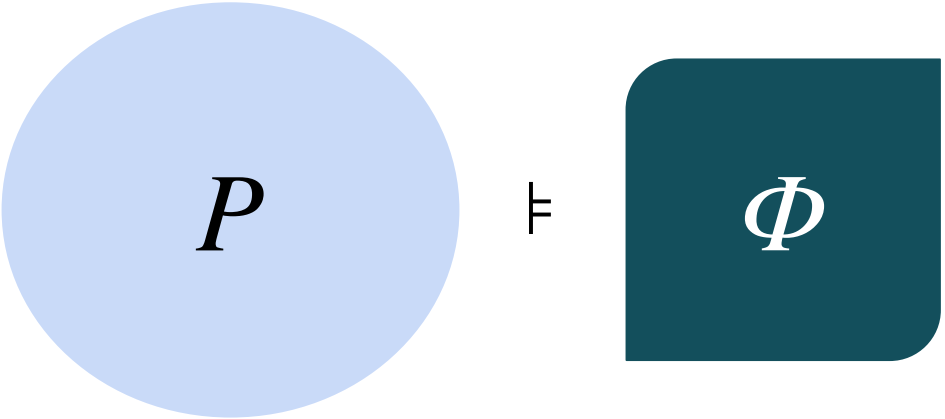
Eric Hayden Campbell

Hossein Hojjat    Nate Foster

# The Unknown

How do we verify programs with unknown code?

# A Verification Myth

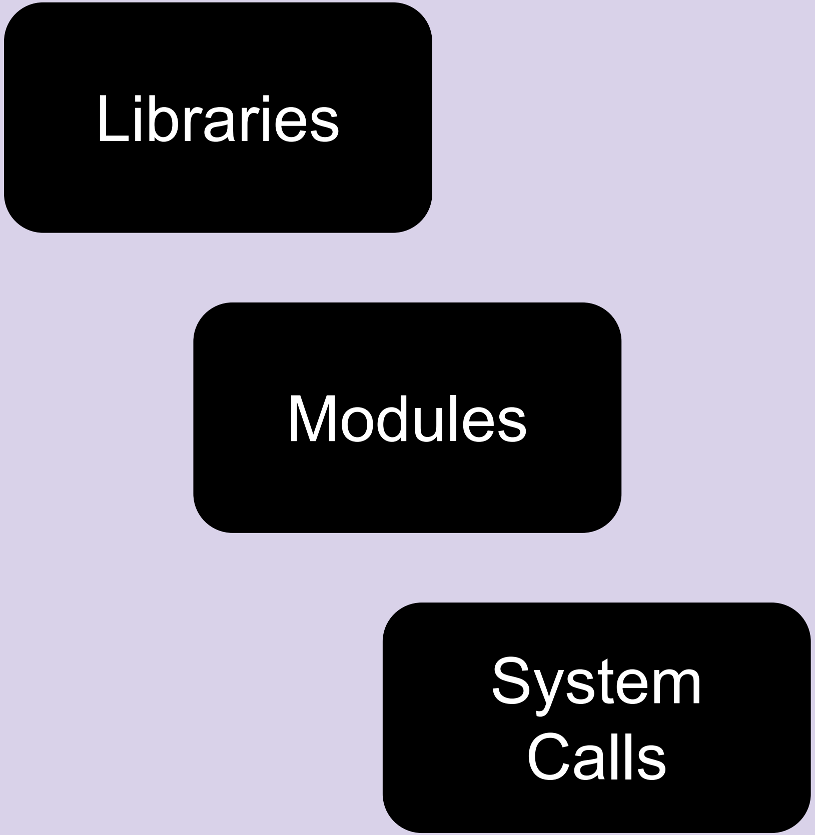


$P \models \Phi$

Program

Spec

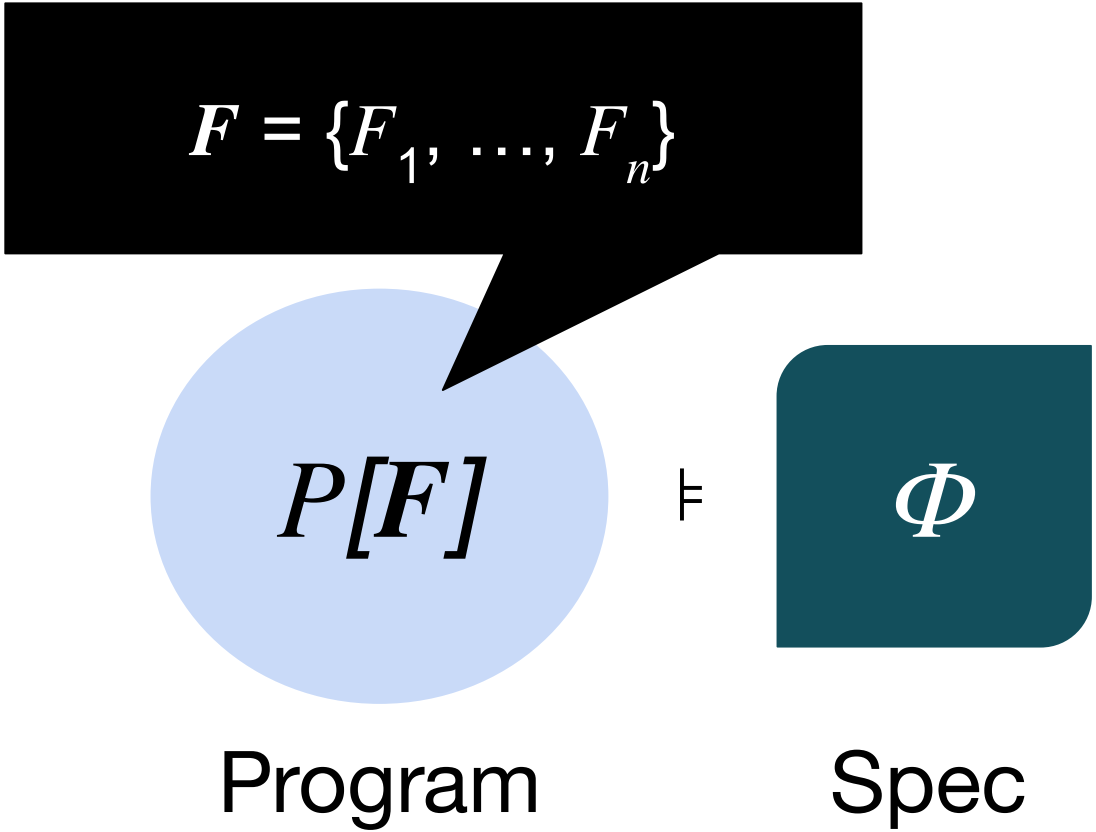
# Source code is incomplete!



Libraries

Modules

System Calls

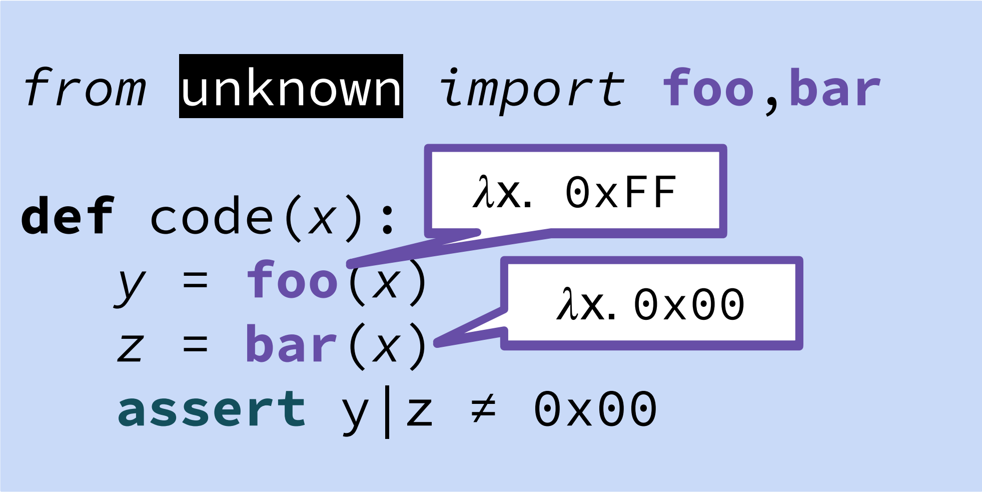


$F = \{F_1, \ldots, F_n\}$

$P[F] \models \Phi$

Program

Spec



```
from unknown import foo,bar

def code(x):
    y = foo(x)
    z = bar(x)
    assert y|z ≠ 0x00
```

λx. 0xFF

λx. 0x00

# **Which** implementations?

```
import unknown as foo,bar

def code(x):
    y = foo(x)
    z = bar(x)
    assert y|z ≠ 0x00
```

**Which** implementations satisfy the spec?

Independent Specs
[POPL ‘16, SIGCOMM ‘20]

$\varphi(\mathbf{foo}) \land \psi(\mathbf{bar})$

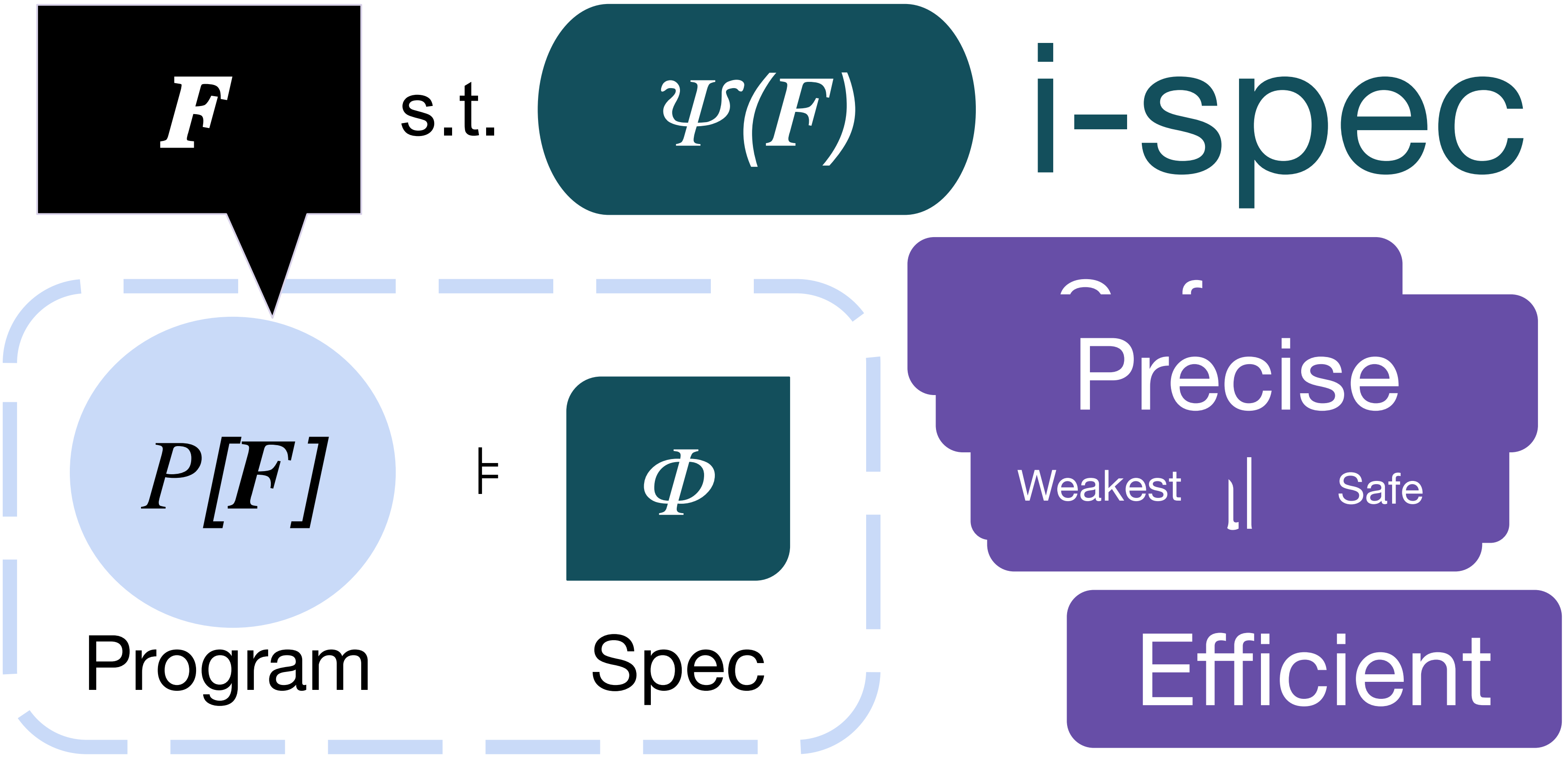
Necessary [SIGCOMM ‘20]

```
true
```

Not Safe!!!

Eliminates
no “good runs”
[VMCAI ‘13]

| | |
|---|---|
| Safe | Unsafe! |
| Overly Restrictive! | Permissive |
| [POPL ‘16] | [SIGCOMM ‘20] |



$F$ s.t. $\Psi(F)$ i-spec

$P[F] \models \Phi$

Program

Spec

Precise

Weakest

Safe

Efficient

```
import unknown as foo,bar

def code(x):
   y = foo(x)
   z = bar(x)
   assert y|z ≠ 0x00
```

**Goal:**
Compute
*precise* i-specs

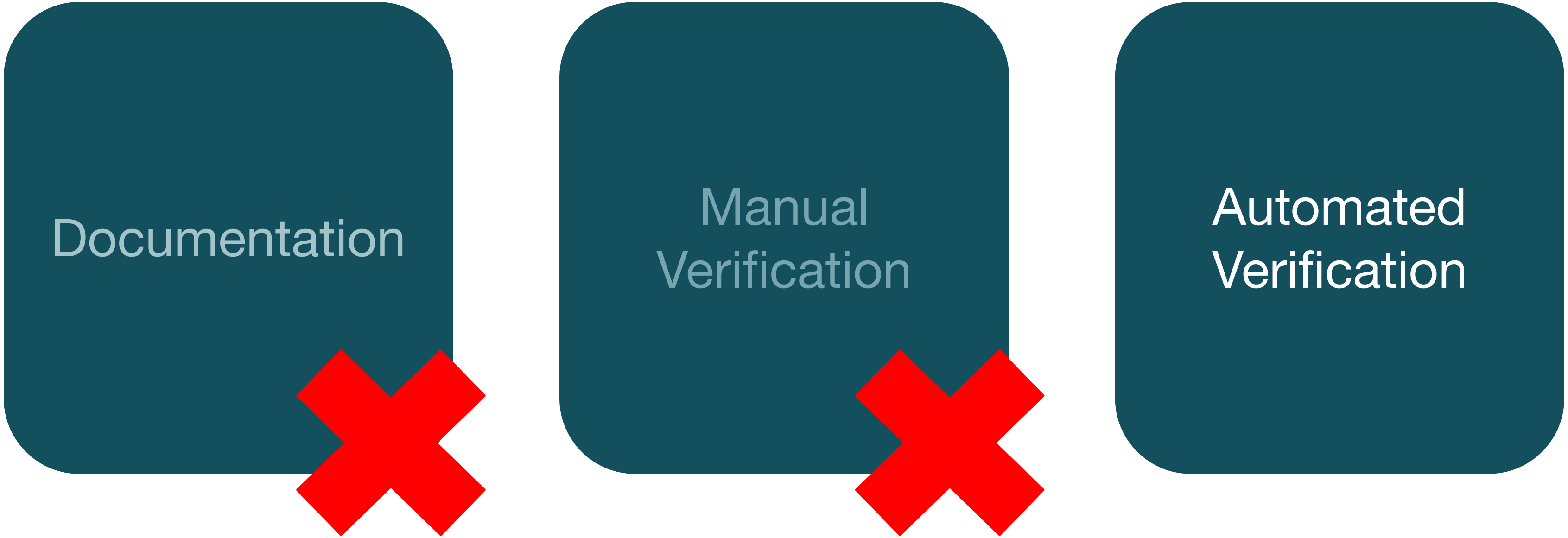
```
foo(x) | bar(x) ≠ 0x00
```

Safe

Weakest

Efficient

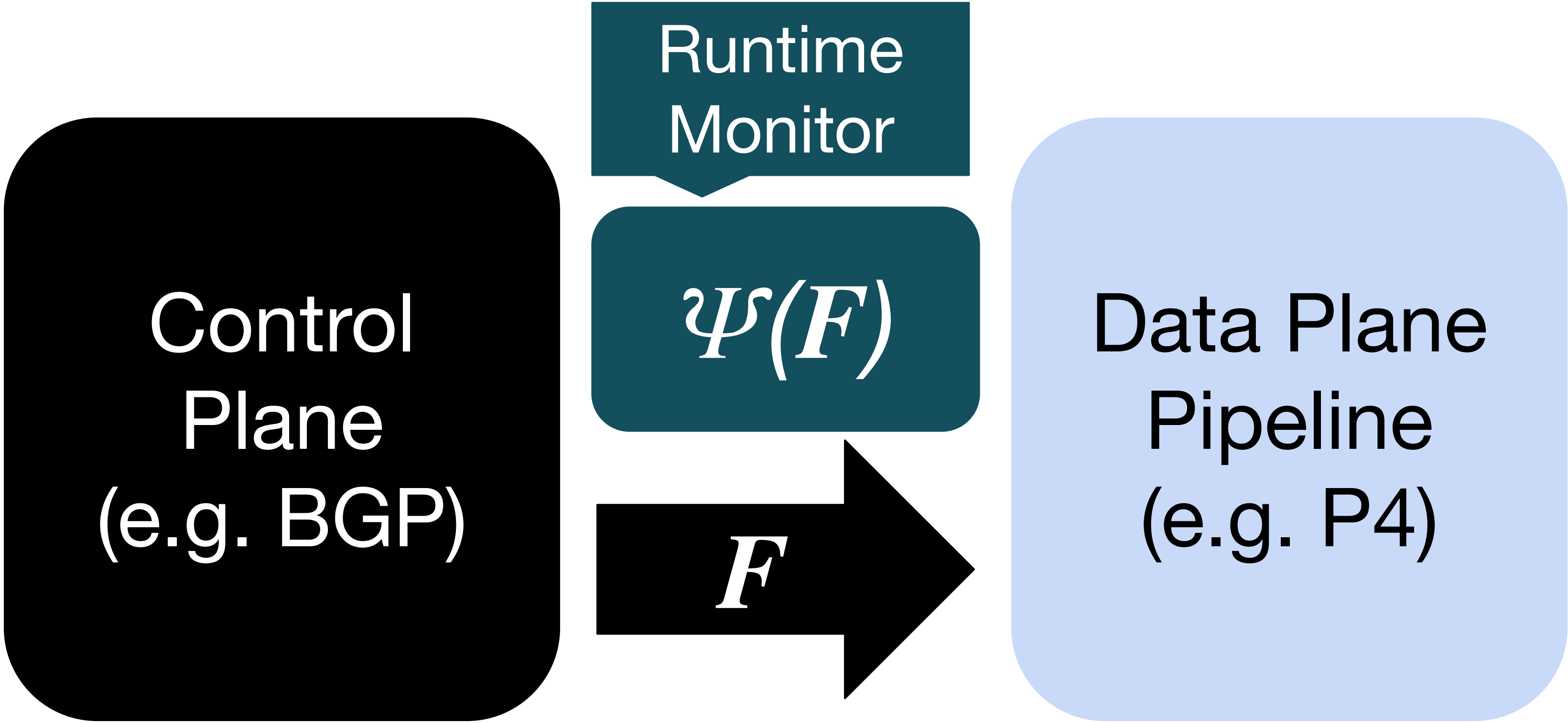
# How to use computed i-specs?



Documentation ✗

Manual Verification ✗

Automated Verification



Runtime Monitor

$\Psi(F)$

Control Plane (e.g. BGP)

$F$

Data Plane Pipeline (e.g. P4)

Computer Network

# Capisce

computes control interface specs
(ci-specs)

Precise

Safe

Weakest

Efficient

# **Step 1:** Model Pipeline Programs

# GCL(**F**)

?

```
c ::= assume φ
    | x := e
    | c ; c
    | c [] c
```

**F** : $2^{w} \to 2^{l}$

e ::= … | **F**(e)

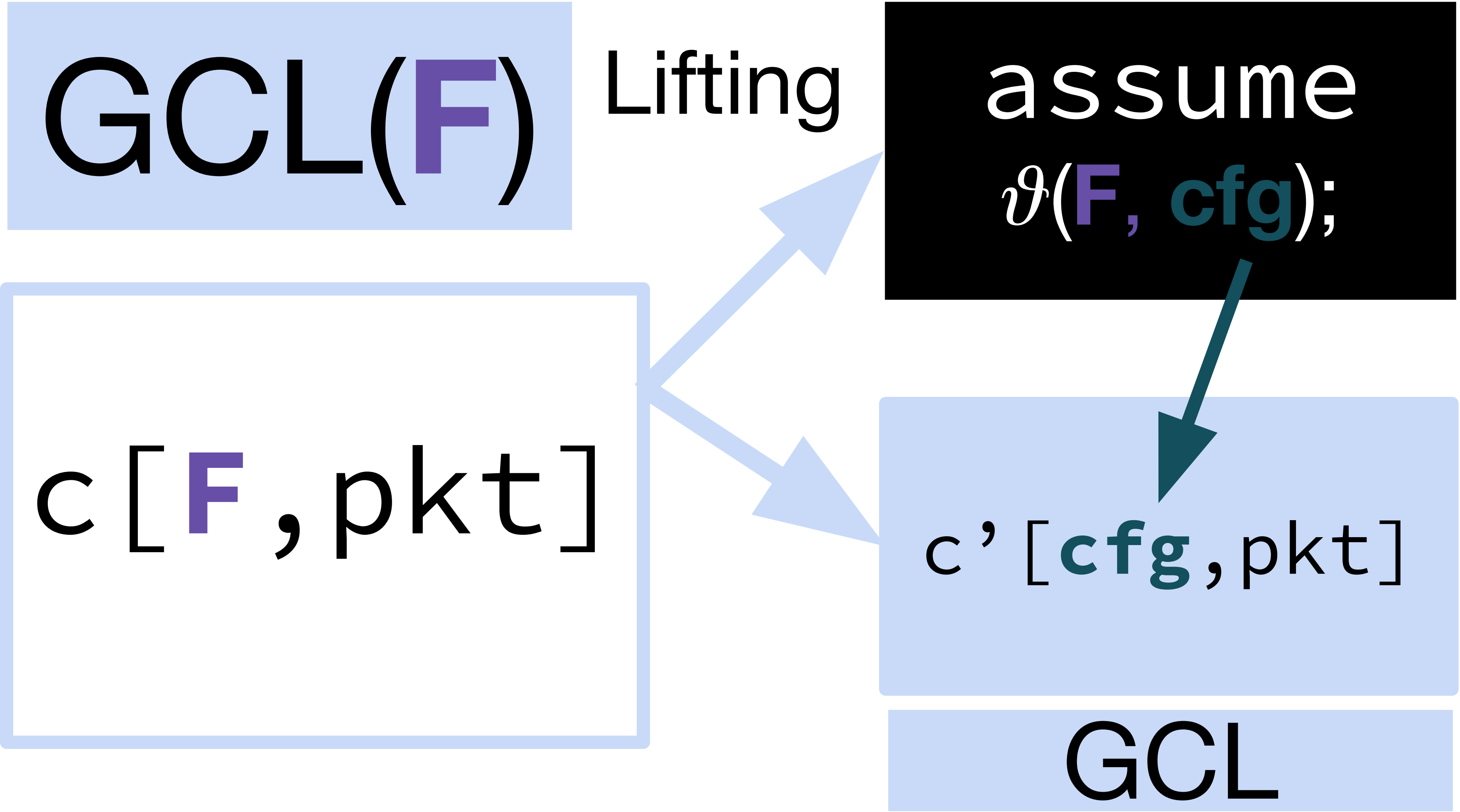
$\varphi \in$ QFBV

# Control Flow in GCL(**F**)

$$\texttt{if}\ \varphi\ \texttt{c}_1\ \texttt{c}_2$$

$$\begin{array}{c}\texttt{assume}\ \ \varphi;\texttt{c}_1\\ \texttt{[]}\\ \texttt{assume}\ \neg\varphi;\texttt{c}_2\end{array}$$

# **Step 2**: Symbolic Compilation



GCL(**F**)

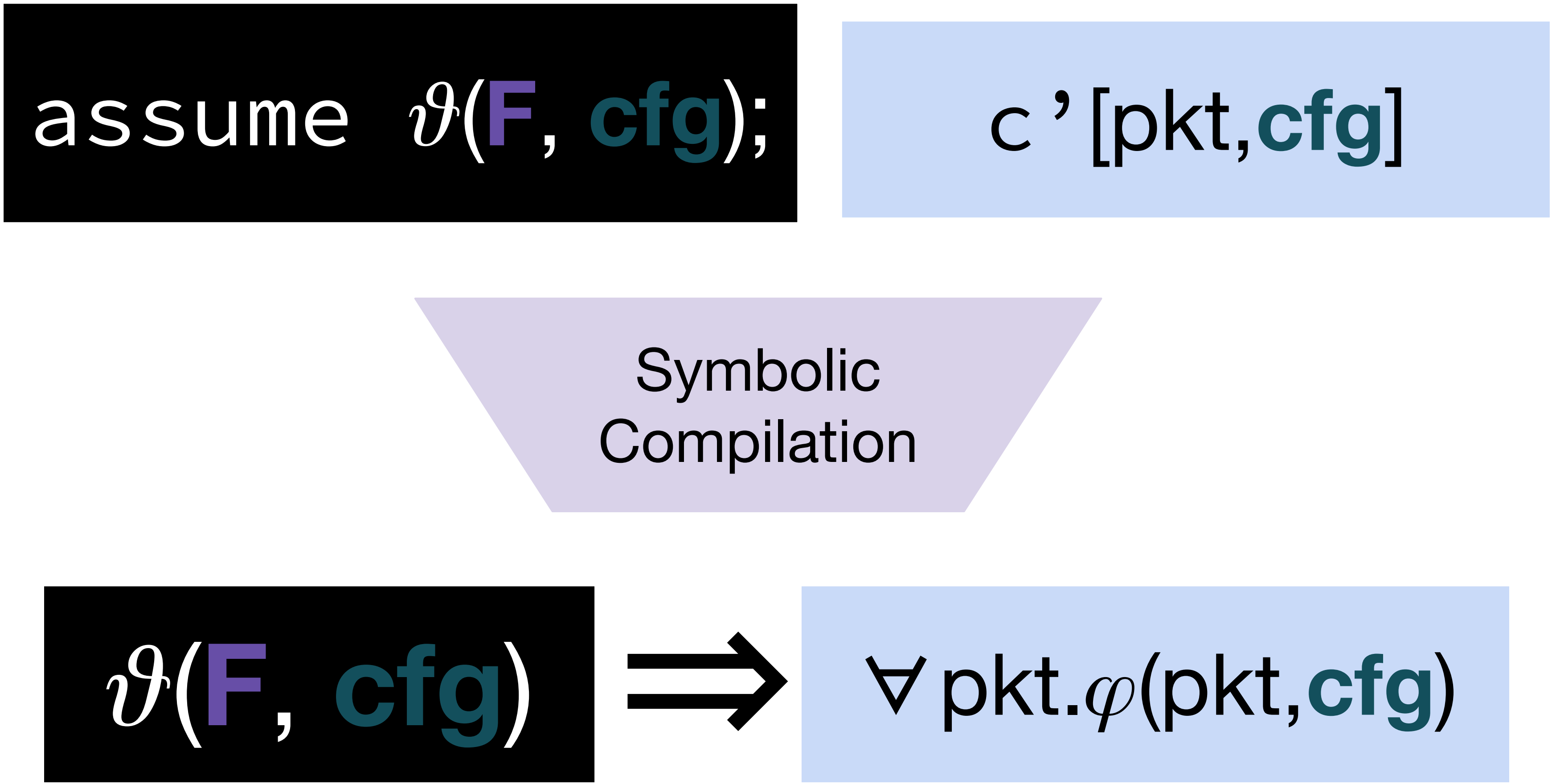
Lifting

assume $\vartheta$(**F**, **cfg**);

c[**F**,pkt]

c'[**cfg**,pkt]

GCL



`assume` $\vartheta(\mathbf{F}, \mathbf{cfg})$;

c’[pkt,**cfg**]

Symbolic Compilation

$\vartheta(\mathbf{F}, \mathbf{cfg}) \Rightarrow \forall \text{pkt}.\varphi(\text{pkt}, \mathbf{cfg})$

# ci-spec

$$\vartheta(F, \mathrm{cfg}) \Rightarrow \forall \mathrm{pkt}.\varphi(\mathrm{pkt}, \mathrm{cfg})$$

Precise

pkt is big

~1500 bits

¬Efficient to monitor

# **Step 3:** Quantifier Elimination

$$\begin{tikzcd}
\vartheta(\mathrm{F}, \mathrm{cfg}) \arrow[r, Rightarrow] & \forall\, \mathrm{pkt}.\varphi(\mathrm{pkt},\mathrm{cfg}) \arrow[d, "\mathrm{QE}"] \\
\vartheta(\mathrm{F}, \mathrm{cfg}) \arrow[r, Rightarrow] & \psi(\mathrm{cfg})
\end{tikzcd}$$

# ci-spec

$$\vartheta(F, y) \Rightarrow \psi(y)$$

**Theorem.** *Precise*
=> *safe*
=> *weakest*

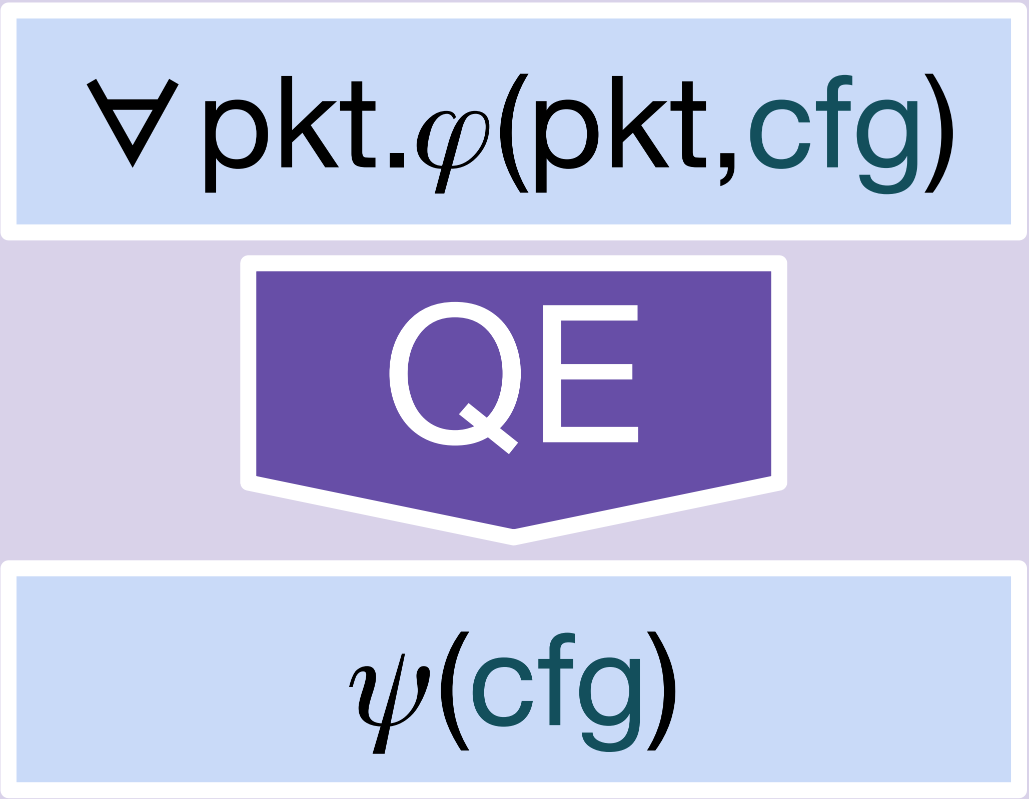
**Theorem.** *Terminates*

**Theorem.** *Efficiently monitorable*

Efficiently Control-Monitorable Sentences

… have polynomial expression complexity

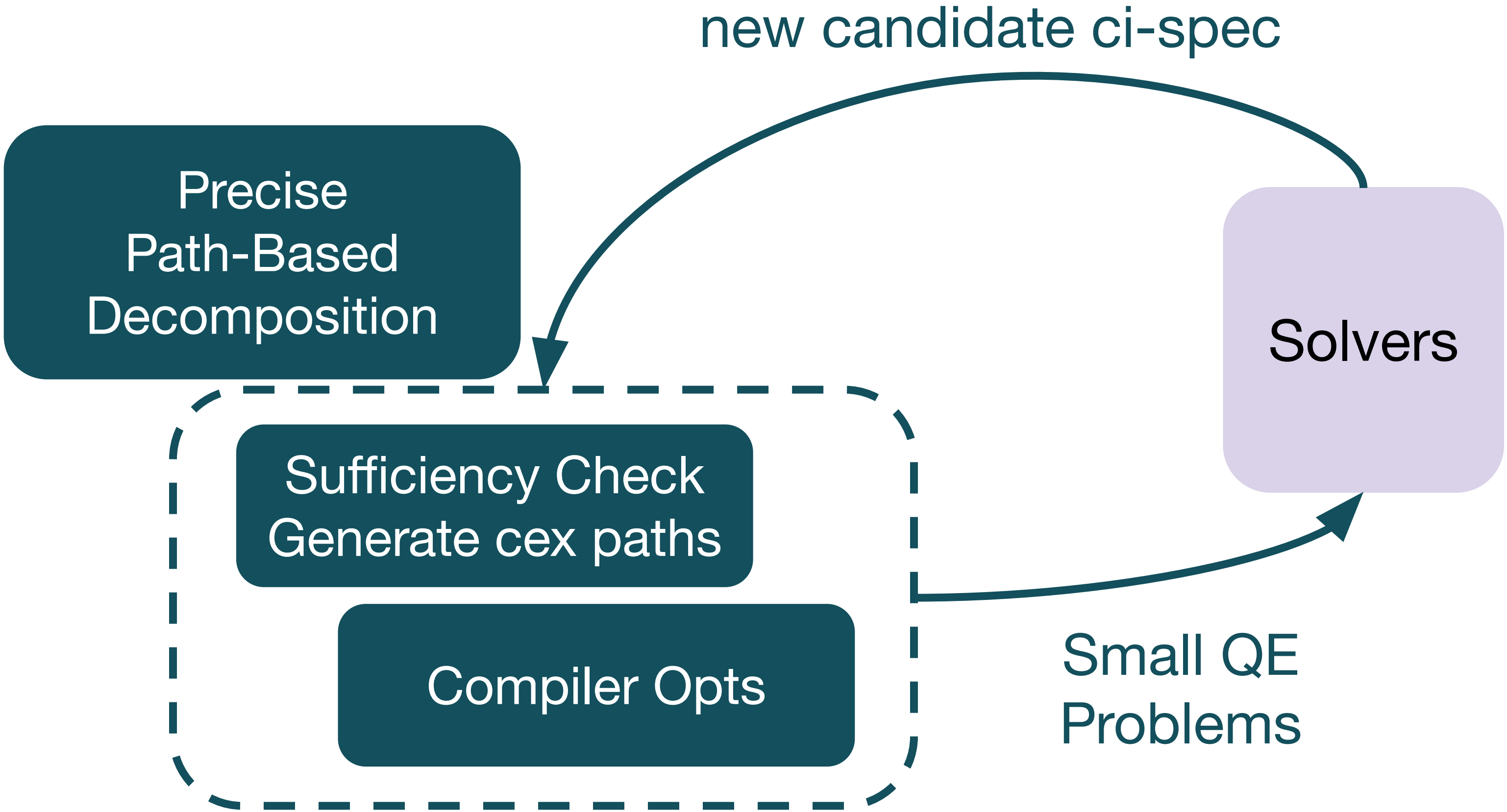
# Quantifier Elimination is *Intractable*



$\forall\,\mathrm{pkt}.\varphi(\mathrm{pkt},\mathrm{cfg})$

QE

$\psi(\mathrm{cfg})$



new candidate ci-spec

Precise Path-Based Decomposition

Sufficiency Check
Generate cex paths

Compiler Opts

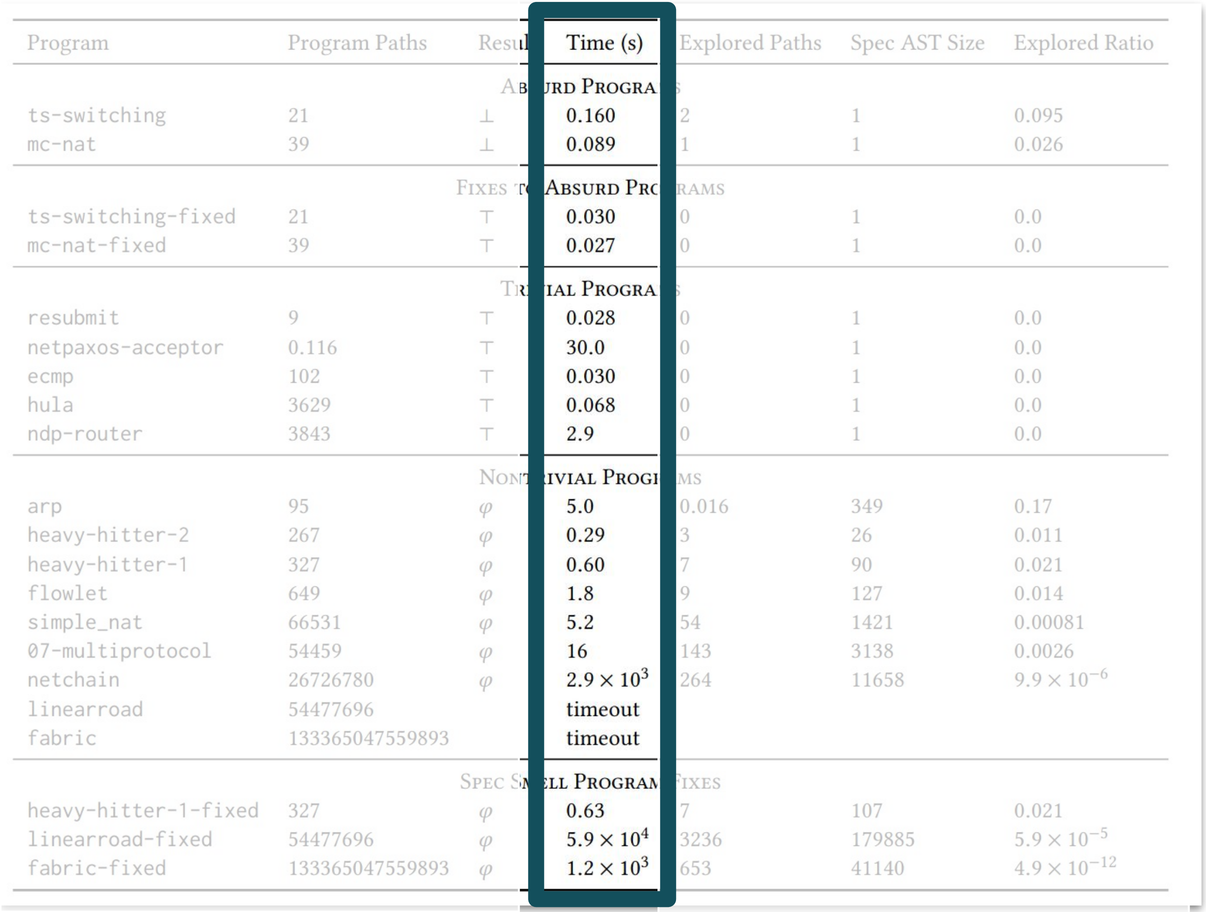
Solvers

Small QE Problems

# Evaluation

# Survey of Industrial and Academic P4 Programs

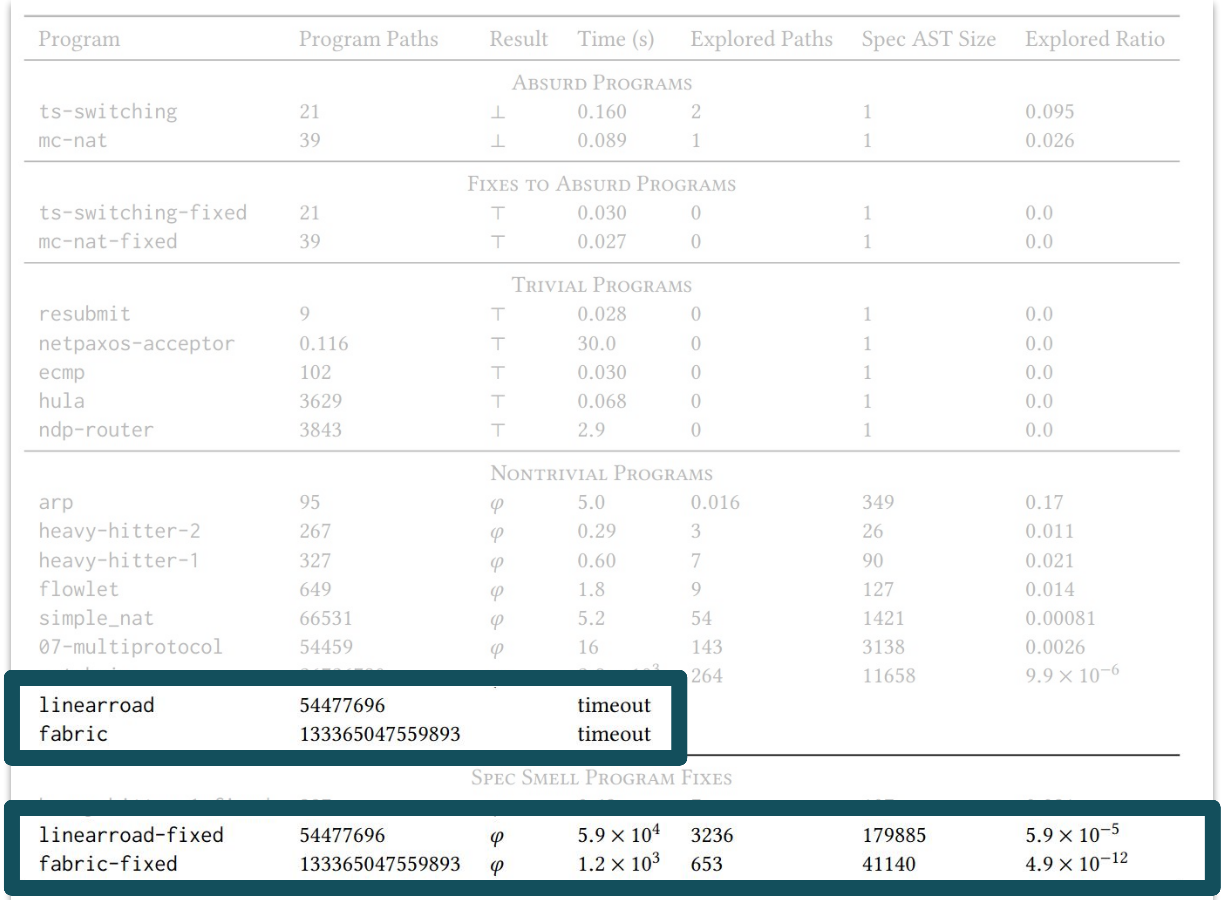
Ensure Invalid Data is not Read



| Program | Program Paths | Result | Time (s) | Explored Paths | Spec AST Size | Explored Ratio |
|---|---|---|---|---|---|---|
| | | | Absurd Programs | | | |
| ts-switching | 21 | ⊥ | 0.160 | 2 | 1 | 0.095 |
| mc-nat | 39 | ⊥ | 0.089 | 1 | 1 | 0.026 |
| | | | Fixes to Absurd Programs | | | |
| ts-switching-fixed | 21 | ⊤ | 0.030 | 0 | 1 | 0.0 |
| mc-nat-fixed | 39 | ⊤ | 0.027 | 0 | 1 | 0.0 |
| | | | Trivial Programs | | | |
| resubmit | 9 | ⊤ | 0.028 | 0 | 1 | 0.0 |
| netpaxos-acceptor | 0.116 | ⊤ | 30.0 | 0 | 1 | 0.0 |
| ecmp | 102 | ⊤ | 0.030 | 0 | 1 | 0.0 |
| hula | 3629 | ⊤ | 0.068 | 0 | 1 | 0.0 |
| ndp-router | 3843 | ⊤ | 2.9 | 0 | 1 | 0.0 |
| | | | Nontrivial Programs | | | |
| arp | 95 | $\varphi$ | 5.0 | 0.016 | 349 | 0.17 |
| heavy-hitter-2 | 267 | $\varphi$ | 0.29 | 3 | 26 | 0.011 |
| heavy-hitter-1 | 327 | $\varphi$ | 0.60 | 7 | 90 | 0.021 |
| flowlet | 649 | $\varphi$ | 1.8 | 9 | 127 | 0.014 |
| simple_nat | 66531 | $\varphi$ | 5.2 | 54 | 1421 | 0.00081 |
| 07-multiprotocol | 54459 | $\varphi$ | 16 | 143 | 3138 | 0.0026 |
| netchain | 26726780 | $\varphi$ | $2.9 \times 10^3$ | 264 | 11658 | $9.9 \times 10^{-6}$ |
| linearroad | 54477696 | | timeout | | | |
| fabric | 133365047559893 | | timeout | | | |
| | | | Spec Smell Program Fixes | | | |
| heavy-hitter-1-fixed | 327 | $\varphi$ | 0.63 | 7 | 107 | 0.021 |
| linearroad-fixed | 54477696 | $\varphi$ | $5.9 \times 10^4$ | 3236 | 179885 | $5.9 \times 10^{-5}$ |
| fabric-fixed | 133365047559893 | $\varphi$ | $1.2 \times 10^3$ | 653 | 41140 | $4.9 \times 10^{-12}$ |

# Program Survey

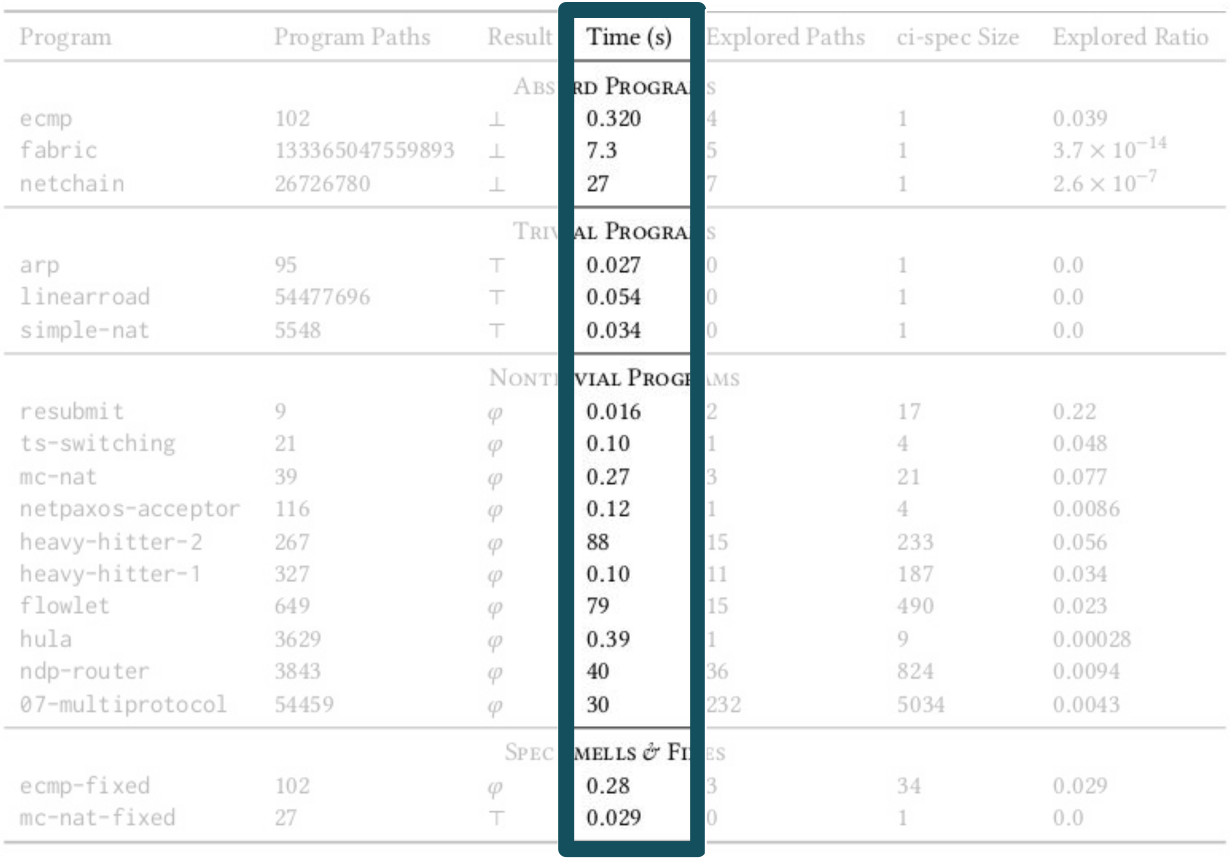
Ensure Invalid Data is not Read



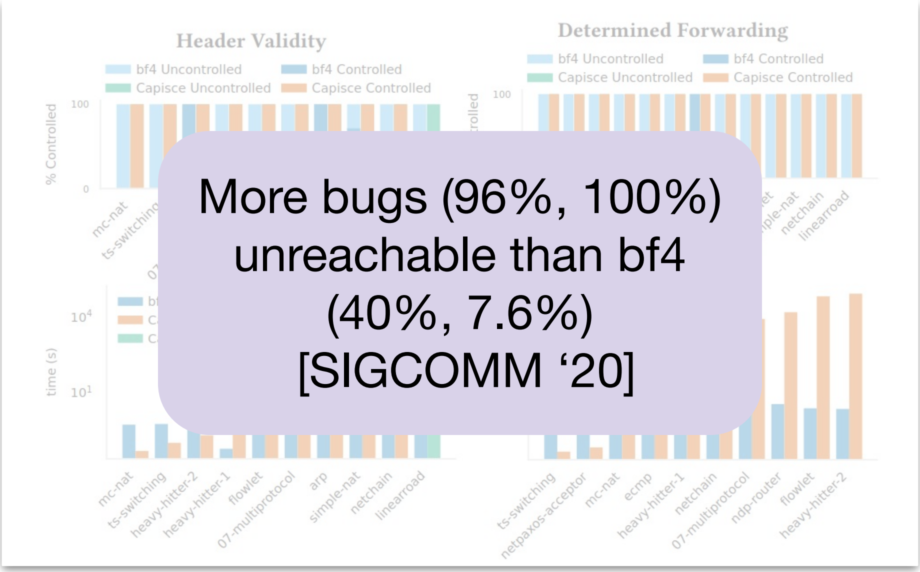
| Program | Program Paths | Result | Time (s) | Explored Paths | Spec AST Size | Explored Ratio |
|---|---|---|---|---|---|---|
| | | | Absurd Programs | | | |
| ts-switching | 21 | ⊥ | 0.160 | 2 | 1 | 0.095 |
| mc-nat | 39 | ⊥ | 0.089 | 1 | 1 | 0.026 |
| | | | Fixes to Absurd Programs | | | |
| ts-switching-fixed | 21 | ⊤ | 0.030 | 0 | 1 | 0.0 |
| mc-nat-fixed | 39 | ⊤ | 0.027 | 0 | 1 | 0.0 |
| | | | Trivial Programs | | | |
| resubmit | 9 | ⊤ | 0.028 | 0 | 1 | 0.0 |
| netpaxos-acceptor | 0.116 | ⊤ | 30.0 | 0 | 1 | 0.0 |
| ecmp | 102 | ⊤ | 0.030 | 0 | 1 | 0.0 |
| hula | 3629 | ⊤ | 0.068 | 0 | 1 | 0.0 |
| ndp-router | 3843 | ⊤ | 2.9 | 0 | 1 | 0.0 |
| | | | Nontrivial Programs | | | |
| arp | 95 | $\varphi$ | 5.0 | 0.016 | 349 | 0.17 |
| heavy-hitter-2 | 267 | $\varphi$ | 0.29 | 3 | 26 | 0.011 |
| heavy-hitter-1 | 327 | $\varphi$ | 0.60 | 7 | 90 | 0.021 |
| flowlet | 649 | $\varphi$ | 1.8 | 9 | 127 | 0.014 |
| simple_nat | 66531 | $\varphi$ | 5.2 | 54 | 1421 | 0.00081 |
| 07-multiprotocol | 54459 | $\varphi$ | 16 | 143 | 3138 | 0.0026 |
| [illegible] | [illegible] | [illegible] | [illegible] | 264 | 11658 | $9.9 \times 10^{-6}$ |
| linearroad | 54477696 | | timeout | | | |
| fabric | 133365047559893 | | timeout | | | |
| | | | Spec Smell Program Fixes | | | |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| linearroad-fixed | 54477696 | $\varphi$ | $5.9 \times 10^4$ | 3236 | 179885 | $5.9 \times 10^{-5}$ |
| fabric-fixed | 133365047559893 | $\varphi$ | $1.2 \times 10^3$ | 653 | 41140 | $4.9 \times 10^{-12}$ |

# Program Survey

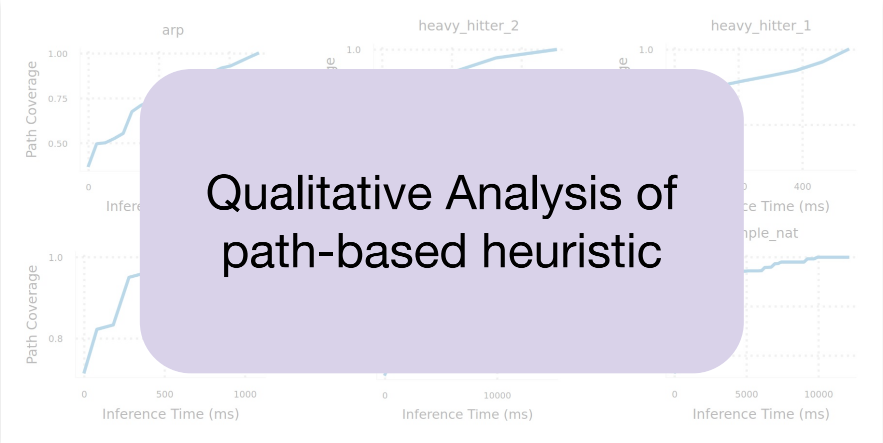
Defined Forwarding



| Program | Program Paths | Result | Time (s) | Explored Paths | ci-spec Size | Explored Ratio |
|---|---|---|---|---|---|---|
| | | Abs | rd Progra | s | | |
| ecmp | 102 | ⊥ | 0.320 | 4 | 1 | 0.039 |
| fabric | 133365047559893 | ⊥ | 7.3 | 5 | 1 | $3.7 \times 10^{-14}$ |
| netchain | 26726780 | ⊥ | 27 | 7 | 1 | $2.6 \times 10^{-7}$ |
| | | Triv | al Progra | s | | |
| arp | 95 | ⊤ | 0.027 | 0 | 1 | 0.0 |
| linearroad | 54477696 | ⊤ | 0.054 | 0 | 1 | 0.0 |
| simple-nat | 5548 | ⊤ | 0.034 | 0 | 1 | 0.0 |
| | | Nontr | vial Progr | ms | | |
| resubmit | 9 | $\varphi$ | 0.016 | 2 | 17 | 0.22 |
| ts-switching | 21 | $\varphi$ | 0.10 | 1 | 4 | 0.048 |
| mc-nat | 39 | $\varphi$ | 0.27 | 3 | 21 | 0.077 |
| netpaxos-acceptor | 116 | $\varphi$ | 0.12 | 1 | 4 | 0.0086 |
| heavy-hitter-2 | 267 | $\varphi$ | 88 | 15 | 233 | 0.056 |
| heavy-hitter-1 | 327 | $\varphi$ | 0.10 | 11 | 187 | 0.034 |
| flowlet | 649 | $\varphi$ | 79 | 15 | 490 | 0.023 |
| hula | 3629 | $\varphi$ | 0.39 | 1 | 9 | 0.00028 |
| ndp-router | 3843 | $\varphi$ | 40 | 36 | 824 | 0.0094 |
| 07-multiprotocol | 54459 | $\varphi$ | 30 | 232 | 5034 | 0.0043 |
| | | Spec | mells & Fi | es | | |
| ecmp-fixed | 102 | $\varphi$ | 0.28 | 3 | 34 | 0.029 |
| mc-nat-fixed | 27 | ⊤ | 0.029 | 0 | 1 | 0.0 |

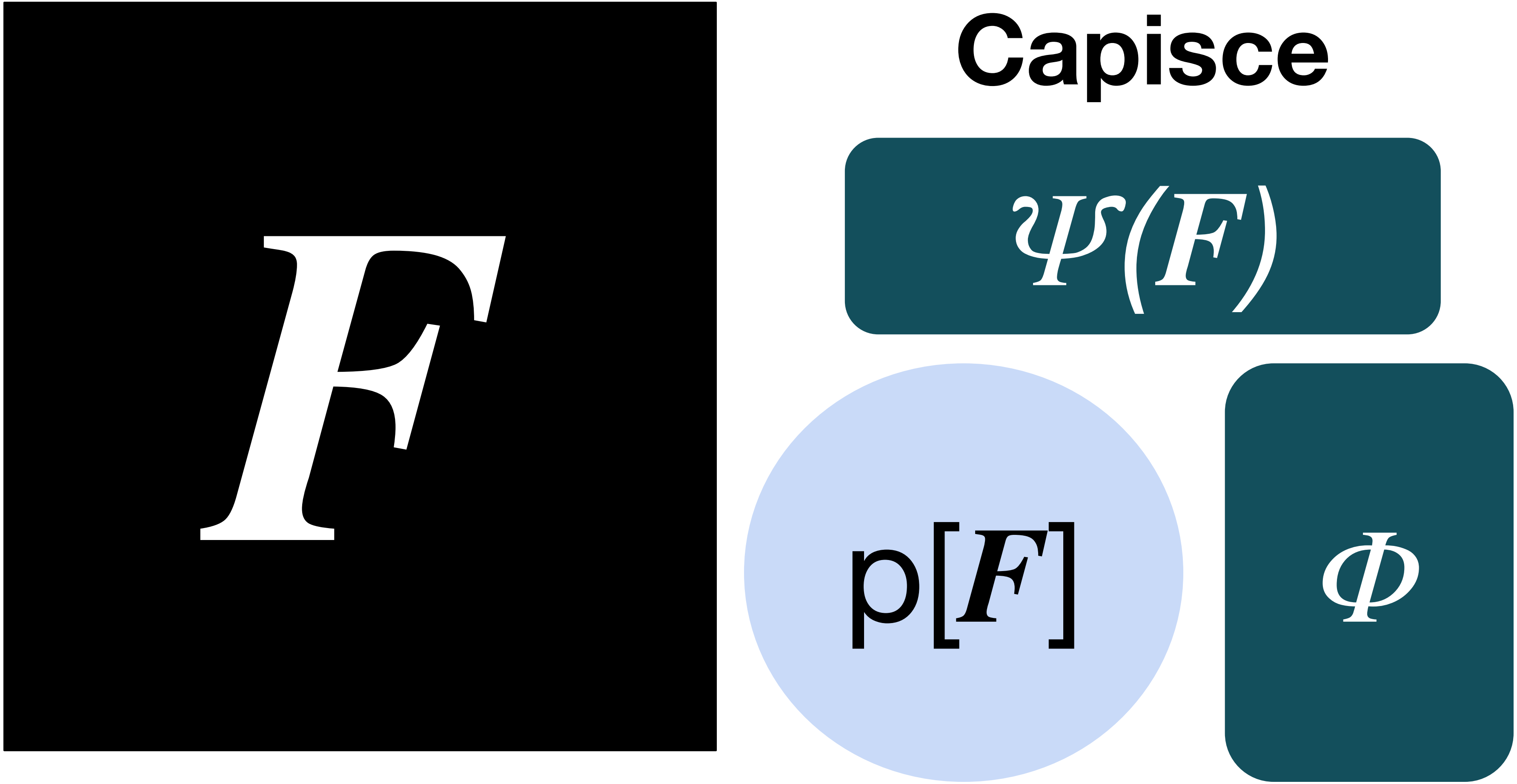


More bugs (96%, 100%) unreachable than bf4 (40%, 7.6%) [SIGCOMM ‘20]



Qualitative Analysis of path-based heuristic

# Capisce



$F$

$\Psi(F)$

$\mathrm{p}[\boldsymbol{F}]$

$\Phi$

# Guarded Pipeline Language (GPL)

?

$p ::= \mathbf{t}(e)$ *table*

$\mid c$ $\in$ *GCL*

$e ::= \ldots$ $\varphi ::= \ldots$

$\mathbf{t} : 2^{w} \rightarrow \{a_1, \ldots, a_n\}$

$a ::= \lambda x.\ c$ $c \in$ *GCL*

## Declaration

```
fwd : $2^{32}$ → {
  (λp:$2^{9}$. port := p),
  (λ_:$2^{0}$. drop := 1)
}
```

## Program

```
fwd(ipv4_dst)
```

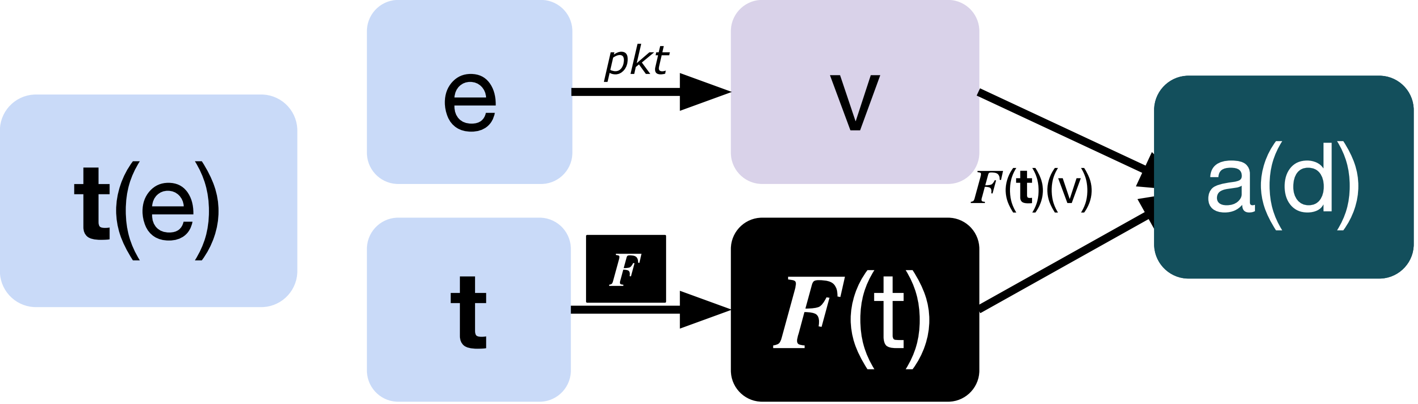
Unknown Implementation

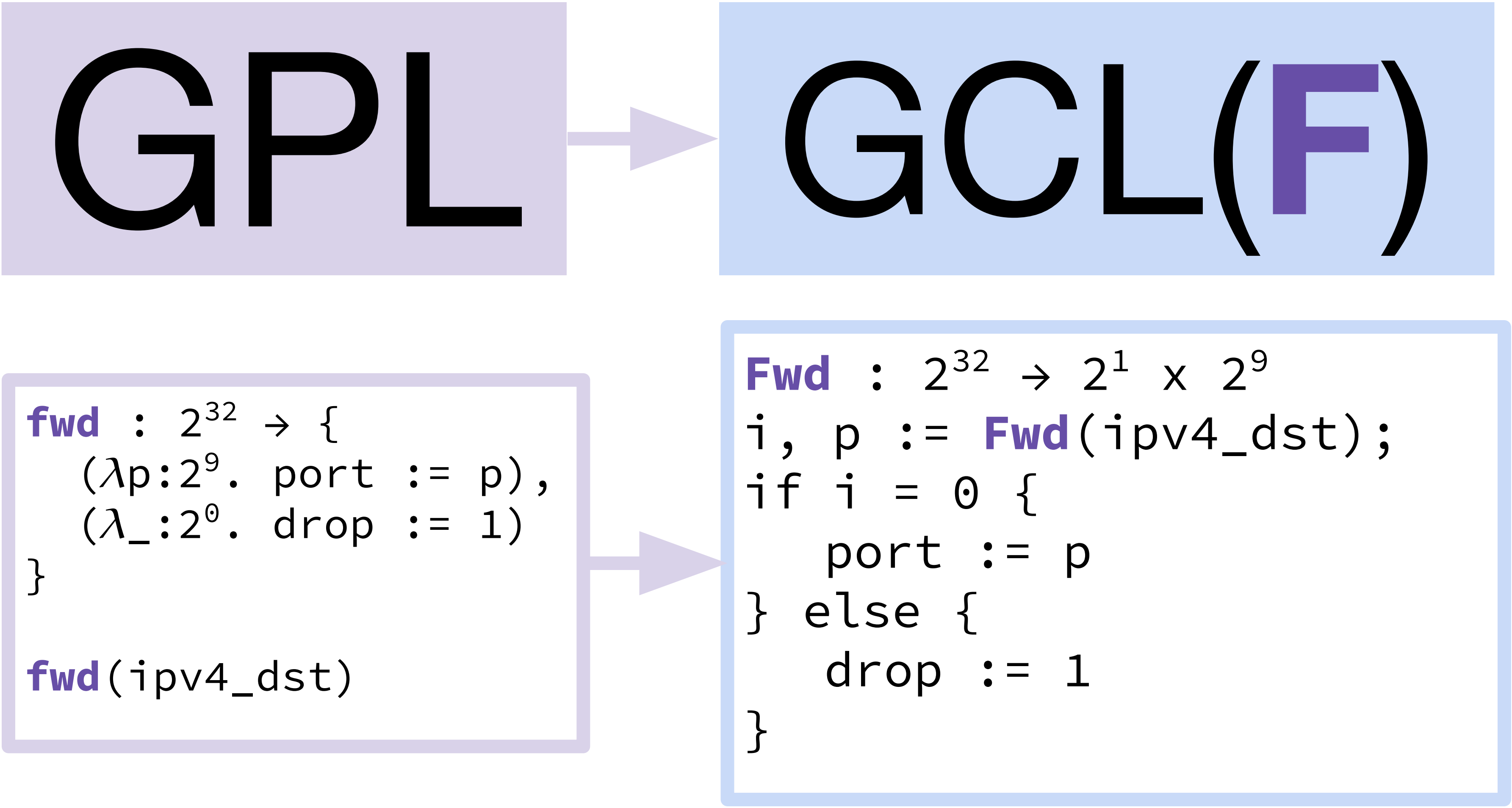
$\boxed{F} : \text{Tbl} \rightarrow \text{BV} \rightarrow \text{Act x BV}$

Variable Store

$pkt : \text{Var} \rightarrow \text{BV}$

Table Semantics





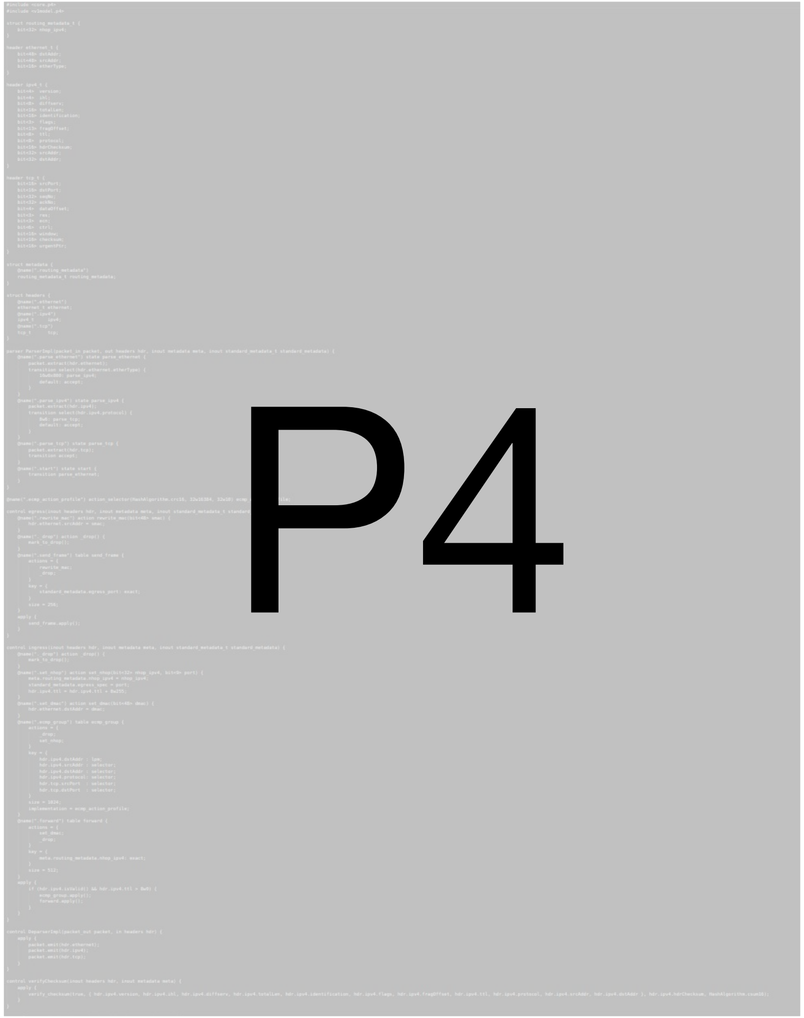
GPL

GCL(F)

```
fwd : 2^32 → {
   (λp:2^9. port := p),
   (λ_:2^0. drop := 1)
}

fwd(ipv4_dst)
```

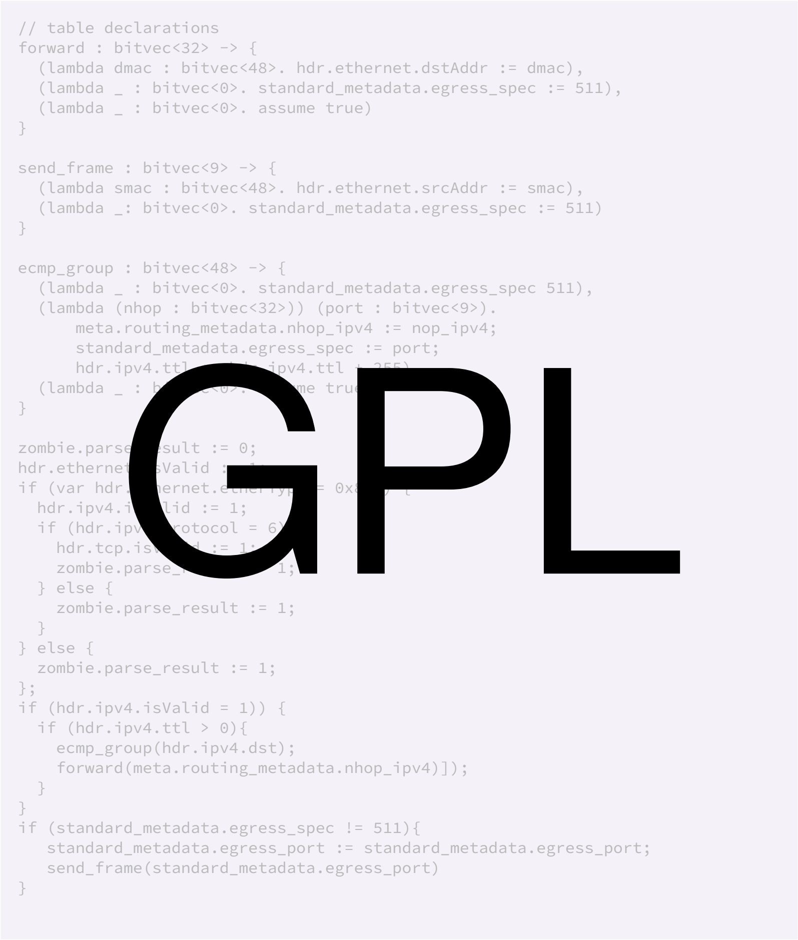
```
Fwd : 2^32 → 2^1 x 2^9
i, p := Fwd(ipv4_dst);
if i = 0 {
   port := p
} else {
   drop := 1
}
```

# P4



```
#include <core.p4>
#include <v1model.p4>

struct routing_metadata_t {
    bit<32> nhop_ipv4;
}

header ethernet_t {
    bit<48> dstAddr;
    bit<48> srcAddr;
    bit<16> etherType;
}

header ipv4_t {
    bit<4>  version;
    bit<4>  ihl;
    bit<8>  diffserv;
    bit<16> totalLen;
    bit<16> identification;
    bit<3>  flags;
    bit<13> fragOffset;
    bit<8>  ttl;
    bit<8>  protocol;
    bit<16> hdrChecksum;
    bit<32> srcAddr;
    bit<32> dstAddr;
}

header tcp_t {
    bit<16> srcPort;
    bit<16> dstPort;
    bit<32> seqNo;
    bit<32> ackNo;
    bit<4>  dataOffset;
    bit<3>  res;
    bit<3>  ecn;
    bit<6>  ctrl;
    bit<16> window;
    bit<16> checksum;
    bit<16> urgentPtr;
}

struct metadata {
    @name(".routing_metadata")
    routing_metadata_t routing_metadata;
}

struct headers {
    @name(".ethernet")
    ethernet_t ethernet;
    @name(".ipv4")
    ipv4_t     ipv4;
    @name(".tcp")
    tcp_t      tcp;
}

parser ParserImpl(packet_in packet, out headers hdr, inout metadata meta, inout standard_metadata_t standard_metadata) {
    @name(".parse_ethernet") state parse_ethernet {
        packet.extract(hdr.ethernet);
        transition select(hdr.ethernet.etherType) {
            16w0x800: parse_ipv4;
            default: accept;
        }
    }
    @name(".parse_ipv4") state parse_ipv4 {
        packet.extract(hdr.ipv4);
        transition select(hdr.ipv4.protocol) {
            8w6: parse_tcp;
            default: accept;
        }
    }
    @name(".parse_tcp") state parse_tcp {
        packet.extract(hdr.tcp);
        transition accept;
    }
    @name(".start") state start {
        transition parse_ethernet;
    }
}

@name(".ecmp_action_profile") action_selector(HashAlgorithm.crc16, 32w16384, 32w10) ecmp_          file;

control egress(inout headers hdr, inout metadata meta, inout standard_metadata_t standard
    @name(".rewrite_mac") action rewrite_mac(bit<48> smac) {
        hdr.ethernet.srcAddr = smac;
    }
    @name("._drop") action _drop() {
        mark_to_drop();
    }
    @name(".send_frame") table send_frame {
        actions = {
            rewrite_mac;
            _drop;
        }
        key = {
            standard_metadata.egress_port: exact;
        }
        size = 256;
    }
    apply {
        send_frame.apply();
    }
}

control ingress(inout headers hdr, inout metadata meta, inout standard_metadata_t standard_metadata) {
    @name("._drop") action _drop() {
        mark_to_drop();
    }
    @name(".set_nhop") action set_nhop(bit<32> nhop_ipv4, bit<9> port) {
        meta.routing_metadata.nhop_ipv4 = nhop_ipv4;
        standard_metadata.egress_spec = port;
        hdr.ipv4.ttl = hdr.ipv4.ttl + 8w255;
    }
    @name(".set_dmac") action set_dmac(bit<48> dmac) {
        hdr.ethernet.dstAddr = dmac;
    }
    @name(".ecmp_group") table ecmp_group {
        actions = {
            _drop;
            set_nhop;
        }
        key = {
            hdr.ipv4.dstAddr : lpm;
            hdr.ipv4.srcAddr : selector;
            hdr.ipv4.dstAddr : selector;
            hdr.ipv4.protocol: selector;
            hdr.tcp.srcPort  : selector;
            hdr.tcp.dstPort  : selector;
        }
        size = 1024;
        implementation = ecmp_action_profile;
    }
    @name(".forward") table forward {
        actions = {
            set_dmac;
            _drop;
        }
        key = {
            meta.routing_metadata.nhop_ipv4: exact;
        }
        size = 512;
    }
    apply {
        if (hdr.ipv4.isValid() && hdr.ipv4.ttl > 8w0) {
            ecmp_group.apply();
            forward.apply();
        }
    }
}

control DeparserImpl(packet_out packet, in headers hdr) {
    apply {
        packet.emit(hdr.ethernet);
        packet.emit(hdr.ipv4);
        packet.emit(hdr.tcp);
    }
}

control verifyChecksum(inout headers hdr, inout metadata meta) {
    apply {
        verify_checksum(true, { hdr.ipv4.version, hdr.ipv4.ihl, hdr.ipv4.diffserv, hdr.ipv4.totalLen, hdr.ipv4.identification, hdr.ipv4.flags, hdr.ipv4.fragOffset, hdr.ipv4.ttl, hdr.ipv4.protocol, hdr.ipv4.srcAddr, hdr.ipv4.dstAddr }, hdr.ipv4.hdrChecksum, HashAlgorithm.csum16);
    }
}
```

# GPL



```
// table declarations
forward : bitvec<32> -> {
  (lambda dmac : bitvec<48>. hdr.ethernet.dstAddr := dmac),
  (lambda _ : bitvec<0>. standard_metadata.egress_spec := 511),
  (lambda _ : bitvec<0>. assume true)
}

send_frame : bitvec<9> -> {
  (lambda smac : bitvec<48>. hdr.ethernet.srcAddr := smac),
  (lambda _: bitvec<0>. standard_metadata.egress_spec := 511)
}

ecmp_group : bitvec<48> -> {
  (lambda _ : bitvec<0>. standard_metadata.egress_spec 511),
  (lambda (nhop : bitvec<32>)) (port : bitvec<9>).
      meta.routing_metadata.nhop_ipv4 := nop_ipv4;
      standard_metadata.egress_spec := port;
      hdr.ipv4.ttl       ipv4.ttl + 255)
  (lambda _ : b      c<0>.      me true
}

zombie.parse  esult := 0;
hdr.ethernet  sValid :
if (var hdr.  hernet.e       yp   = 0x8     ) {
  hdr.ipv4.i  lid := 1;
  if (hdr.ipv  rotocol = 6)
    hdr.tcp.is     d := 1;
    zombie.parse_        1;
  } else {
    zombie.parse_result := 1;
  }
} else {
  zombie.parse_result := 1;
};
if (hdr.ipv4.isValid = 1)) {
  if (hdr.ipv4.ttl > 0){
    ecmp_group(hdr.ipv4.dst);
    forward(meta.routing_metadata.nhop_ipv4)]);
  }
}
if (standard_metadata.egress_spec != 511){
   standard_metadata.egress_port := standard_metadata.egress_port;
   send_frame(standard_metadata.egress_port)
}
```