Improving Software Quality using Automatic Invariant Discovery and Program Repair

ThanhVu (Vu) Nguyen





Software Bugs



World of Warcraft bug



Therac-25 machines X-rays overdose

T.

Ariane-5 rocket self-destructs



North America blackout



"Everyday, almost 300 bugs appear [..] far too many for only the Mozilla programmers to handle." Software bugs annually cost 0.6% of the U.S GDP and \$312 billion to the global economy

Average time to fix a security-critical error: 28 days



Automated program analysis techniques and tools can decrease debugging time by an average of 26% and \$41 billion annually



Check if a program meets a given specification

Fix a buggy program to satisfy a given specification

Invariant Generation

```
def intdiv(x, y):
  q = 0
  r = x
  while r \geq y:
    a = 1
    b = y
    while [??] r \geq 2b:
    a = 2a
    b = 2b
    r = r - b
    q = q + a
  [??]
    return q
```

- Discover invariant properties at certain program locations
- Answer the question *"what does this program do ?"*

Invariant Generation

```
def intdiv(x, y):
  q = 0
  r = x
  while r \geq y:
    a = 1
    b = y
  while [??] r \geq 2b:
    a = 2a
    b = 2b
  r = r - b
  q = q + a
  [??]
  return q
```

- Discover invariant properties at certain program locations
- Answer the question *"what does this program do ?"*

Automatic Program Repair

```
def intdiv(x, y):

q = 0

r = x

while r \ge y:

a = 1

3*y

b = y

while r \ge 2b:

a = 2a

b = 2b

r = r - b

q = q + a^{-2*a}
```

return q

- Localize errors and modify code to fix bugs
- A form of program synthesis

Core Research Areas



Invariant Discovery



Program Repair

steatorfg x +				Ĩ
🔄 🖉 Tintor aboutsconfig 👘 🔻 C		\$	ė »	=
Search				P
Preference Name	Satur	Тура	Value	1.5
brown Trees Link Arlanse Links	ditat	integer	4000	
brown download altroy plugins (in) list	default	bookan	true	
browser.dewnload.aseDewnloadDir	use set	bodean	false	
browser, error, al-enabled	detailt	boolean	twe	
browsex-enable_automatic_image_resizing	detailt	boolean	true	
browsenerable, click, image, residing	default	bookan	true	
browser Fersts handler	delait	sting	ek .	

Configurable System

Core Research Areas



Invariant Discovery



Program Repair

🔄 🖉 Tindox aboutsconfig 🛛 🕆 C		\$	ė »	=
Steph				ρ
Preference Name	25513	Тура	Value	1
trouver.com/coduces.com/arver.com/	default	integer	4000	
browner, download altrow plugier, jer, list	default	bookan	true	
browser.dewnload.aseDewnloatDir	user set	bodean	false	
browser, energia erabled	default	bookan	taue	
browse enable, substantic, image, residing	detault	bookan	true	
browser, enable, click, image, resizing	default	bookan	true	
brouse fresh handler	detail	sting	ek .	

Configurable System

New Research Directions

🖹 makefile 🕊

```
RM := rm -rf
TARGET := hello
OBJS := hello.o
SRCS := hello.c
```

all: \$(TARGET)

```
$(TARGET): $(0035) $(SRC5)
@echo`bluiding '$(TARGET)
@echo`bluiding '$(TARGET)
@echo`bluit Successfully'
%.o: %.c
@echo 'building $@ from $<'
@gcc -o $@ -c $<
Clean:
$(RM) $(0835) $(TARGET)
```

Unix Build Systems



IoT systems



Al-generated Software

Outline

• SE/PL Research

Invariant Generation

Automatic Program Repair Highly-Configurable Systems

• Current/New Research Works

How We Analyze Programs



File Ed	lit Options Buffers Tools Help									
def	<pre>intdiv(x, y):</pre>	*	x x x	0, 1, 1,	y y y	1, 1, 5,	q q q	0, 1, 0,	r r r	0 0 1
	q = 0 r = x		x	1,	ý	10,	q	0,	r	1 0
	while r >= y: a = 1	=	x	3,	ý	4,	p	ō,	r	3
=	b = y		x	8,	ý	1,	q	8,	ŗ	0
	while r >= 2*b:		x	8,	y	9, 10	q	0,	ŗ	8
	a = 2 * a b = 2 * b		x	15,	y	1,	q	15,	r	0
	q = q + a		x	15,	y y	7,	q	2,	r	1
	print "x %d, y %d, q %d, r %d" %(x,y,q,r)		x	20, 20, 20,	y y	7,	q q	2,	r	6
0	return q,r		x	20, 100,	y y	10,	q q	2, 100,	r r	0
-			××	100, 100,	y y	5, 10,	q q	20, 10,	r r	0
-U:	intdiv.py All (18,0) (Python)	5:-l	J:	ir	ntdi	v.tra	ices	A11	(2	1,0)

How We Analyze Programs

	File Ed	it Options Buffer	rs Tools Help									
HERE>>>	def	intdiv(x, y)			^	x x	0, 1	/ 1, / 1.	q	0, 1.	r r	0
						x	1. 1	5,	q	0,	r	1
		q = 0 r = x				x	3, 1	10, 1,	q	0, 3,	r	0
		while r >= y	:			×	3,	4,	q	0,	ŗ	3
	=	a - 1 b - y				x	8,	1.	q	8,	r	0
		while r	= 7*h·			x	8, 1	, 2,	P	4,	r r	0
		a = :	2 * a			x	8,	10,	q	ŏ,	r	8
Drintf()		b = 1	2 * b			×	5,	1, 1,	P	15,	r	0
		q = q + a	a			x	5,	7,	q	2,	r.	1
		print "x %d.	v %d. a %d. r %d	" %(x.y.g.r)		x	20, j 20, j	, 2, , 7.	q	10, 2.	r	0 6
		return q,r	,, ,,			x	20,	10,	q	2,	r	0
"%0.2f: %64d, %c, \"%s\", %62.2f%6%".	U					x 10	00, j	, 1, , 5.	q	20,	r	0
'\${COMPANY_NAME}' << <end< td=""><td></td><td></td><td></td><td></td><td></td><td>x 10</td><td>)0, j</td><td>10,</td><td>ģ</td><td>10,</td><td>r</td><td>0</td></end<>						x 10)0, j	10,	ģ	10,	r	0
	-U:	intdiv.pv	All (18.0)	(Python)5	- U		int	div.tr	aces	A11	(2	1.0)

F	ile	Edi	t Option	is Buffer	s Tools	Python	Help
*	d	lef	intdiv assert	(x,y) y!= (:)		
			#с	ompute	result		
			assert assert	r >= (x >= () 1		
	C		return	q, r			
•			intdi	v.py	A1]	L (11,0)

- Software Testing course "GCC: 9000 assertions, LLVM: 13,000 assertions [..] 1 assertion per 110 loc" "program invariants are asserted properties, such as relations among variables, at certain locations in a program"



```
assert(x == 2*y);
assert(0 <= idx < |arr|);</pre>
```

"program invariants are asserted properties, such as relations among variables, at certain locations in a program"







```
assert(x == 2*y);
assert(0 <= idx < |arr|);</pre>
```

```
int getDateOfMonth(int m){
    /*pre: 1 <= m <= 12*/
    ...
    /*post: 0 <= result <= 31*/
}</pre>
```

"a loop invariant is a condition that is true on entry into a loop and is guaranteed to remain true on every iteration of the loop [..]"

Microsoft	Visual C++ Debug Library 🛛 🔀							
	Debug Assertion Failed!							
~	Program: File: dbgheap.c Line: 1258							
	Expression: _BLOCK_TYPE_IS_VALID(pHead->nBlockUse)							
	For information on how your program can cause an assertion failure, see the Visual C++ documentation on asserts.							
	(Press Retry to debug the application)							
	Abort Retry Ignore							

Uses

- Understand and verify programs
- Formal proofs
- Debug (locate errors)
- Documentations

Approaches to Finding Invariants

```
int intdiv(int x, int y){
    int q=0; int r=x;
    while(r ≥ y){
        int a=1; int b=y;
        while[L](r ≥ 2*b){
            a = 2*a; b = 2*b;
        }
        r=r-b; q=q+a;
    }
    return q;
}
```

Static Analysis

- Analyze source code directly
- Pros: guaranteed results
- Cons: computationally intensive, infer simple invariants

Approaches to Finding Invariants

```
int intdiv(int x, int y){
    int q=0; int r=x;
    while(r ≥ y){
        int a=1; int b=y;
        while[L](r ≥ 2*b){
        a = 2*a; b = 2*b;
        }
        r=r-b; q=q+a;
    }
    return q;
}
```

Static Analysis

- Analyze source code directly
- Pros: guaranteed results
- Cons: computationally intensive, infer simple invariants

x	у	q	r
0	1	0	0
1	1	1	0
3	4	0	3
8	1	8	0
15	5	3	0
20	2	10	0
100	1	100	0
	÷	÷	

Dynamic Analysis

- Analyze program traces
- Pros: fast, source code not required
- Cons: results depend on traces, might not hold for all runs

Example

```
int intdiv(int x, int y){
 assert(x>0 && y>0);
 int q=0; int r=x;
 while (r \ge y) {
    int a=1;
   int b=y;
   while [L] (r \ge 2*b) {
     a = 2*a;
     b = 2*b;
   }
   r=r-b;
   q=q+a;
 }
 return q;
}
```

Example

```
int intdiv(int x, int y){
 assert(x>0 && y>0);
 int q=0; int r=x;
 while (r \ge y){
   int a=1;
   int b=y;
   while [L] (r \ge 2*b) {
     a = 2*a;
     b = 2*b;
   }
   r=r-b;
   q=q+a;
 }
 return q;
}
```

x	у	a	Ь	q	r
15	2	1	2	0	15
15	2	2	4	0	15
15	2	1	2	4	7
4	1	1	1	0	4
4	1	2	2	0	4

Invariants at L: b = ya, x = qy + r, $r \ge 2ya$

DIG discovers polynomial relations of the forms

Equalities
$$c_0 + c_1 x_1 + c_2 x_n + c_3 x_1 x_2 + \dots + c_m x_1^{d_1} \dots x_n^{d_n} = 0$$

Inequalities $c_0 + c_1 x_1 + c_2 x_n + c_3 x_1 x_2 + \dots + c_m x_1^{d_1} \dots x_n^{d_n} \ge 0, \quad c_i \in \mathbb{R}$

Examples

cubic
$$z - 6n = 6$$
, $\frac{1}{12}z^2 - y - \frac{1}{2}z = -1$
extended gcd $gcd(a, b) = ia + jb$
sqrt $x + \varepsilon \ge y^2 \ge x - \varepsilon$

DIG discovers polynomial relations of the forms

Equalities
$$c_0 + c_1 x_1 + c_2 x_n + c_3 x_1 x_2 + \dots + c_m x_1^{d_1} \dots x_n^{d_n} = 0$$

Inequalities $c_0 + c_1 x_1 + c_2 x_n + c_3 x_1 x_2 + \dots + c_m x_1^{d_1} \dots x_n^{d_n} \ge 0, \quad c_i \in \mathbb{R}$

Examples

cubic
$$z - 6n = 6$$
, $\frac{1}{12}z^2 - y - \frac{1}{2}z = -1$
extended gcd $gcd(a, b) = ia + jb$
sqrt $x + \varepsilon \ge y^2 \ge x - \varepsilon$

Method

- Equalities: solve equations
- Inequalities: construct polyhedra

Example

```
int intdiv(int x, int y){
 assert(x>0 && y>0);
 int q=0; int r=x;
 while (r \ge y){
   int a=1;
   int b=y;
   while[L] (r \ge 2*b){
     a = 2*a;
     b = 2*b;
   }
   r=r-b;
   q=q+a;
 }
 return q;
}
```

x	у	a	Ь	q	r
15	2	1	2	0	15
15	2	2	4	0	15
15	2	1	2	4	7
4	1	1	1	0	4
4	1	2	2	0	4

Invariants at L: b = ya, x = qy + r, $r \ge 2ya$

x	у	a	b	q	r
15	2	1	2	0	15
15	2	2	4	0	15
15	2	1	2	4	7
4	1	1	1	0	4
4	1	2	2	0	4

 $V = \{r, y, a\}; \text{ deg} = 2$ \downarrow $T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$

• Terms and degrees

X	y	a	b	q	r
15	2	1	2	0	15
15	2	2	4	0	15
15	2	1	2	4	7
4	1	1	1	0	4
4	1	2	2	0	4

 $V = \{r, y, a\}; \ \deg = 2$ \downarrow $T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$ $T = \{\dots, \log(r), a^y, \sin(y), \dots\}$

• Terms and degrees

X	y	a	b	q	r
15	2	1	2	0	15
15	2	2	4	0	15
15	2	1	2	4	7
4	1	1	1	0	4
4	1	2	2	0	4

 Terms and degrees 	X	y	a	b	q	r
$V = \{r, y, a\}; \ \mathtt{deg} = 2$ \downarrow	15	2	1	2	0	15
	15	2	2	4	0	15
	15	2	1	2	4	7
$T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$	4	1	1	1	0	4
	4	1	2	2	0	4

• Nonlinear equation template

 $c_1 + c_2r + c_3y + c_4a + c_5ry + c_6ra + c_7ya + c_8r^2 + c_9y^2 + c_{10}a^2 = 0$

 Terms and degrees 	X	у	a	b	q	r
$V = \{r, y, a\}; \ \mathtt{deg} = 2$	15	<mark>2</mark>	<mark>1</mark>	2	0	15
	15	2	2	4	0	15
	15	2	1	2	4	7
$T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$	4	1	1	1	0	4
	4	1	2	2	0	4

Nonlinear equation template

 $c_1 + c_2r + c_3y + c_4a + c_5ry + c_6ra + c_7ya + c_8r^2 + c_9y^2 + c_{10}a^2 = 0$

• System of linear equations

trace 1 : {r = 15, y = 2, a = 1} eq 1 : $c_1 + 15c_2 + 2c_3 + c_4 + 30c_5 + 15c_6 + 2c_7 + 225c_8 + 4c_9 + c_{10} = 0$:

 Terms and degrees 	x	y	а	Ь	q	r
$V = \{r, y, a\}; \ \mathtt{deg} = 2$ \downarrow	15	2	1	2	0	15
	15	2	2	4	0	15
	15	2	1	2	4	7
$T = \{1, r, y, a, ry, ra, ya, r^2, y^2, a^2\}$	4	1	1	1	0	4
	4	1	2	2	0	4

Nonlinear equation template

 $c_1 + c_2 r + c_3 y + c_4 a + c_5 ry + c_6 ra + c_7 ya + c_8 r^2 + c_9 y^2 + c_{10} a^2 = 0$

• System of linear equations

trace 1 : {r = 15, y = 2, a = 1} eq 1 : $c_1 + 15c_2 + 2c_3 + c_4 + 30c_5 + 15c_6 + 2c_7 + 225c_8 + 4c_9 + c_{10} = 0$

• Solve for coefficients *c_i*

$$V = \{x, y, a, b, q, r\}; deg = 2 \longrightarrow b = ya, x = qy+r$$

Geometric Invariant Inference



Geometric Invariant Inference



Iterative Invariant Generation



- Dynamic Analysis: learn candidate invariants from execution traces
- Static Analysis: use *theorem proving* and *constraint solving* to check invariants against program code and return *counterexample inputs*

```
void triple(int M, int N, int P){ Complexity of this program?
  assert (0 \le M);
                                        • Use t to count loop iterations
  assert (0 <= N);
  assert (0 \le P);
  int i = 0, j = 0, k = 0;
  int t = 0;
  while(i < N){</pre>
    j = 0; t++;
    while(j < M){</pre>
      j++; k = i; t++;
      while (k < P){
       k++; t++;
      }
     i = k;
    }
    i++:
  }
  [L]
}
```

}

```
void triple(int M, int N, int P){ Complexity of this program?
  assert (0 \le M);
                                        • Use t to count loop iterations
  assert (0 \le N);
  assert (0 \le P);
                                        • At first glance: t = O(MNP)
  int i = 0, j = 0, k = 0;
  int t = 0;
  while(i < N){</pre>
    j = 0; t++;
    while(j < M){</pre>
      j++; k = i; t++;
      while (k < P){
       k++; t++;
      }
     i = k;
    }
   i++:
  }
  [L]
```

```
void triple(int M, int N, int P){ Complexity of this program?
  assert (0 \le M);
                                            Use t to count loop iterations
  assert (0 \le N):
  assert (0 \le P);
                                            • At first glance: t = O(MNP)
  int i = 0, j = 0, k = 0;

    Dlg found an interesting (and

  int t = 0;
                                               unexpected) nonlinear invariant at L:
  while(i < N){</pre>
    j = 0; t++;
                                                P^2Mt + PM^2t - PMNt - M^2Nt -
    while(j < M){</pre>
                                                PMt^{2} + MNt^{2} + PMt - PNt - 2MNt +
                                                Pt^{2} + Mt^{2} + Nt^{2} - t^{3} - Nt + t^{2} = 0
      j++; k = i; t++;
      while (k < P){
        k++; t++;
      }
      i = k;
    }
    i++;
  }
  [L]
```

```
void triple(int M, int N, int P){ Complexity of this program?
  assert (0 \le M):
                                          • Use t to count loop iterations
  assert (0 \le N):
  assert (0 \le P);
                                          • At first glance: t = O(MNP)
  int i = 0, j = 0, k = 0;

    Dlg found an interesting (and

  int t = 0;
                                            unexpected) nonlinear invariant at L:
  while(i < N){</pre>
    j = 0; t++;
                                              P^2Mt + PM^2t - PMNt - M^2Nt -
    while(j < M){</pre>
                                              PMt^2 + MNt^2 + PMt - PNt - 2MNt +
                                              Pt^{2} + Mt^{2} + Nt^{2} - t^{3} - Nt + t^{2} = 0
      j++; k = i; t++;
      while (k < P){
                                          • Solve for t yields the most precise,
        k++; t++:
                                            unpublished bound:
      }
                                              t = 0
                                                    when N = 0,
      i = k;
                                              t = P + M + 1 when N \leq P,
    }
                                              t = N - M(P - N) when N > P
    i++:
  }
  [L]
```

Applications

Security

- complexity and side-channel attacks (FSE'17, ASE'17, SEAD Workshop '20)
- AES analysis (ICSE'12, TOSEM'13)

Applications

Security

- complexity and side-channel attacks (FSE'17, ASE'17, SEAD Workshop '20)
- AES analysis (ICSE'12, TOSEM'13)

Others

- termination/liveness (OOPSLA'20)
- heap/pointer (PLDI'19)
- program (API) synthesis (OOPSLA'19)
- disjunctive/geometric invs (ICSE'14, J. Automated Reasoning'13)

Applications

Security

- complexity and side-channel attacks (FSE'17, ASE'17, SEAD Workshop '20)
- AES analysis (ICSE'12, TOSEM'13)

Others

- termination/liveness (OOPSLA'20)
- heap/pointer (PLDI'19)
- program (API) synthesis (OOPSLA'19)
- disjunctive/geometric invs (ICSE'14, J. Automated Reasoning'13)

Highly-Configurable Systems: iGen (FSE '16), GenTree (ICSE '21)

Outline

• SE/PL Research

Invariant Generation Automatic Program Repair Highly-Configurable Systems

• Current/New Research Works





Wed morning, Dec 31, 2008: Microsoft Zune music players mysteriously froze



Wed morning, Dec 31, 2008: Microsoft Zune music players mysteriously froze

— Matt Akers (Microsoft Zune spokesman)

"By [Thursday] you should allow the battery to fully run out of power before the unit can restart successfully, then simply ensure that your device is recharged, then turn it back on"

```
int zunebug(int days) {
 int year = 1980;
 while (days > 365) {
   if (isLeapYear(year)){
     if (days > 366) {
       days -= 366;
       year += 1;
     }
   }
   else {
     days -= 365;
     year += 1;
   }
  }
 return year;
}
```

```
int zunebug(int days) {
 int year = 1980;
 while (days > 365) {
   if (isLeapYear(year)){
     if (days > 366) {
       days -= 366;
       year += 1;
     }
   }
   else {
     days -= 365;
     year += 1;
   }
 }
 return year;
}
```

```
int zunebug_repair(int days) {
 int year = 1980;
 while (days > 365) {
   if (isLeapYear(year)){
     if (days > 366) {
       // days -= 366; // repair deletes
       year += 1;
     }
     days -= 366; // repair inserts
   } else {
     days -= 365;
     year += 1;
   }
 }
 return year;
}
```

GenProg: Program Repair using Genetic Algorithm



- 1 Isolate faults
- 2 Mutate program statements and reuse existing code
- **3** Check repair candidates

GenProg: Program Repair using Genetic Algorithm



Results

- demonstrated on bugs in real-world software (repair 16 programs over 1.25 MLocs, 2 mins avg)
- 10-year Most Influential Paper award (ICSE '19) and 10-year Most Impact Paper award (GECCO '19)

APR Techniques

- Evolutionary computing (ICSE'09, GECCO'09, ICST'09, CACM'10, TSE'13)
- Theory and Formal Analysis: equivalence between program repair and reachability, apply input generation techniques to repair programs (TACAS'17)

APR Techniques

- Evolutionary computing (ICSE'09, GECCO'09, ICST'09, CACM'10, TSE'13)
- Theory and Formal Analysis: equivalence between program repair and reachability, apply input generation techniques to repair programs (TACAS'17)

Non-traditional Repairs

- Corrupted *data structures* (Google Summer of Code'18, FSE JPF Workshop'18)
- Fault localization in declarative models (ICSE '21)
- Repair declarative programs (ICSE '21)

Outline

• SE/PL Research

Invariant Generation Automatic Program Repair Highly-Configurable Systems

• Current/New Research Works

Analyzing Configurable Software

Modern software are highly-configurable

• Allowing for customization and flexiblity

Analyzing Configurable Software

Modern software are highly-configurable

. .

- Allowing for customization and flexiblity
- But can have misconfigurations (rank 6th on 2020 OWASP list of most critical security risks)



```
# a. ~/.htaccess
<Limit
PUT DELLETE TRACE
...
</Limit>
# b. /etc/apache/httpd.conf
RewriteCond TRACE
```

c. load mod_rewrite
a2enmod mod_rewrite

d. /etc/apache/httpd.conf LoadModule rewrite_module "mod_rewrite.so"

• • •

• Program with 7 options: $s, t, u, v, x, y, z \in \{0 \dots 4\}$

- Program with 7 options: $s, t, u, v, x, y, z \in \{0 \dots 4\}$
- Interactions discovery

interaction	behavior
$x \wedge y$	B0 B1
$s \lor t$	B1 B2
$(\neg s \land \neg t) \lor (\neg u \lor \neg v)$	B3
· ·	

- Program with 7 options: $s, t, u, v, x, y, z \in \{0 \dots 4\}$
- Interactions discovery

$x \wedge y$ B0	r
$ \begin{array}{ll} x \land y \land (z \in \{0, 3, 4\}) & B1 \\ s \lor t & B2 \\ (\neg s \land \neg t) \lor (\neg u \lor \neg v) & B3 \\ \vdots & \end{array} $	

• Use dynamic analysis

$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
- 0 1 1 1 1 0 3 B2

GenTree: Dynamic Interaction Discovery (FSE'16, ICSE'21)





GenTree: Dynamic Interaction Discovery (FSE'16, ICSE'21)



• C5_i: new classification algorithm

GenTree: Dynamic Interaction Discovery (FSE'16, ICSE'21)





• C5_i: new classification algorithm

Iterative approach



Outline

• SE/PL Research

Invariant Generation Automatic Program Repair Highly-Configurable Systems

• Current/New Research Works

IoT Interaction Analysis and Repair (UNL Faculty grant'21)



Build Systems (SPLC Workshop'20, ICSME NIER'20, NSF CRII'20)

	Network device support
[*]	Network core driver support
<m></m>	Bonding driver support
<m></m>	Dummy net driver support
<m></m>	EQL (serial line load balancing) support
[]	Fibre Channel driver support
<m></m>	Intermediate Functional Block support
<m></m>	Ethernet team driver support>
<*>	MAC-VLAN support
<m></m>	MAC-VLAN based tap driver
< >	IP-VLAN support
< >	Virtual eXtensible Local Area Network (VXLAN)
<m></m>	Generic Network Virtualization Encapsulation
	(CDDC) T 21 D 1 2 1 1 (CTD 11)
<m></m>	GPRS Tunneling Protocol datapath (GTP-U)
<m></m>	GPRS funneling Protocol datapath (GPP-U) EEE 802.1AE MAC-level encryption (MACsec)
M> <p< td=""><td>GPRS lunneling Protocol datapath (GIP-U) EEE 802.1AE MAC-level encryption (MACsec) Network console logging support</td></p<>	GPRS lunneling Protocol datapath (GIP-U) EEE 802.1AE MAC-level encryption (MACsec) Network console logging support
<m> <m> <m> [*]</m></m></m>	GPRS lunneling Protocol datapath (GIP-U) EEE 802.1AE MAC-level encryption (MACsec) Network console logging support Dynamic reconfiguration of logging targets
A <p< td=""><td>BPKS luneling Protocol datapath (GIP-U) EEE 802.1EE MAC-level encryption (MACsec) Network console logging support Dynamic reconfiguration of logging targets Universal TUN/TAP device driver support</td></p<>	BPKS luneling Protocol datapath (GIP-U) EEE 802.1EE MAC-level encryption (MACsec) Network console logging support Dynamic reconfiguration of logging targets Universal TUN/TAP device driver support
₩ ₩ ₩ [*] []	GPKS luneling Protocol datapath (GIP-U) EEE 802.1AE MAC-level encryption (MACsec) Network console logging support Dynamic reconfiguration of logging targets Universal TUN/TAP device driver support Support for cross-endian vnet headers on littl
M <p< td=""><td>BPRS Tunneling Protocol datapath (GIP-U) EEE 802.14E MAC-level encryption (MACSec) Network console logging support Dynamic reconfiguration of logging targets Universal TUN/TAP device driver support Support for cross-endian vnet headers on littl Virtual ethernet pair device</td></p<>	BPRS Tunneling Protocol datapath (GIP-U) EEE 802.14E MAC-level encryption (MACSec) Network console logging support Dynamic reconfiguration of logging targets Universal TUN/TAP device driver support Support for cross-endian vnet headers on littl Virtual ethernet pair device
\$ \$ 2 2 3 3 2 3 3 4 3 4 3 4 3 4 3 4 3 4 3 4	<pre>PKS luneling Protocol datapath (GIP-U) FEE 802.1&E MAC-level encryption (MACsec) Network console logging support Dynamic reconfiguration of logging targets Universal TUN/TAP device driver support Support for cross-endian vnet headers on littl Virtual ethernet pair device Virtio network driver</pre>
\$ \$ \$] \$ * \$ \$ \$	BPRS Tunneling Protocol datapath (GIP-U) EEE 802.14E MAC-level encryption (MACSec) Network console logging support Dynamic reconfuguration of logging targets Universal TUN/TAP device driver support Support for cross-endian vnet headers on littl Virtual ethermet pair device Virtual netlink monitoring device
\$ \$ \$ \$] \$] \$ \$ \$ \$ \$ \$	BPRS luneling Protocol datapath (GIP-U) EEE 802.1EE MAC-level encryption (MACsec) Network console logging support Dynamic reconfiguration of logging targets Universal TUN/TAP device driver support Support for cross-endian vnet headers on littl Virtual ethernet pair device Virtio network driver Virtual netlink monitoring device Virtual Routing and Forwarding (Lite)
\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	CPRS lumeting Protocol datapath (GIP-U) EEE 802.14E KAC-level encryption (MACSec) Network console logging support Dynamic reconfuguration of logging targets Universal TUN/TAP device driver support Support for cross-endian vneu headers on littl Virtual ethernet pair device Virtual ethernet device Virtual netlink monitoring device Virtual Nosck monitoring device Virtual vosck monitoring device
\$ \$ \$ \$ \$ \$ <u></u> \$ \$ \$ \$ \$ \$ \$ <u></u> \$ <u></u> \$ \$ \$ \$ \$ \$ \$ \$ 	<pre>BPRS luneLing Protocol datapath (GIP-U) EEE 802.1&E MAC-level encryption (MACsec) Network console logging support Dynamic reconfuguration of logging targets Universal TUN/TAP device driver support Support for cross-endian vnet headers on littl Virtual ethernet pair device Virtual atentink monitoring device Virtual Nouting and Forwarding (Lite) Virtual Routing and Forwarding (Lite) Virtual support></pre>
$\begin{array}{c} \begin{array}{c} \\ \\ \\ \end{array}\end{array}$	<pre>EPRS lumeting Protocol datapath (GIP-U) EEE 802.14E KAC-level encryption (MACsec) Network console logging support Dynamic recordiguration of logging targets Universal TUN/TAP device driver support Support for cross-endian vneu headers on littl Virtual ethernet pair device Virtual network driver Virtual network driver Virtual network driver Virtual Routing and Forwarding (Lite) Virtual vsock monitoring device ARCnet support></pre>

Linux/Unix Build Systems

Build Systems (SPLC Workshop'20, ICSME NIER'20, NSF CRII'20)

--- Network device support [*] Network core driver support <M> Bonding driver support <M> Dummy net driver support <M> EQL (serial line load balancing) support []] Fibre Channel driver support <M> Intermediate Functional Block support <M> Ethernet team driver support ---> <*> MAC-VLAN support <M> MAC-VLAN based tap driver < > IP-VLAN support < > Virtual eXtensible Local Area Network (VXLAN) Generic Network Virtualization Encapsulation <M> <M> GPRS Tunneling Protocol datapath (GTP-U) < > TEEE 802.1AE MAC-level encryption (MACsec) <M> Network console logging support [*] Dynamic reconfiguration of logging targets <M> Universal TUN/TAP device driver support [] Support for cross-endian vnet headers on littl <M> Virtual ethernet pair device <M> Virtio network driver <M> Virtual netlink monitoring device Virtual Routing and Forwarding (Lite) <M> <M> Virtual vsock monitoring device <M> ARCnet support ---> v(+) <Select> < Exit > < Help > < Save >

Linux/Unix Build Systems

```
ifeq ($(abs_otree),$(CURDIR))
MAKEFLAGS += --no-print-dir
else
need-sub-make := 1
endif
```

```
abs_stree := $(rpath $(dir $
    (this-makefile)))
ifneq ($(words $(subst :,
    ,$(abs_stree))), 1)
$(error src dir cannot contain
    spaces or colons)
endif
ifneq ($(abs_stree),$(abs_otree))
MAKEFLAGS +=
    --include-dir=$(abs_stree)
```

Make/CMake

Deep Neural Networks



- Invariants (activation patterns) discovery
- Symbolic testing

IDE Integration (https://grammatech.gitlab.io/Mnemosyne/docs/muses/)



Thank you!