Mining Input Grammars

@AndreasZeller
Center for IT-Security, Privacy, and Accountability
Saarland University, Saarbrücken

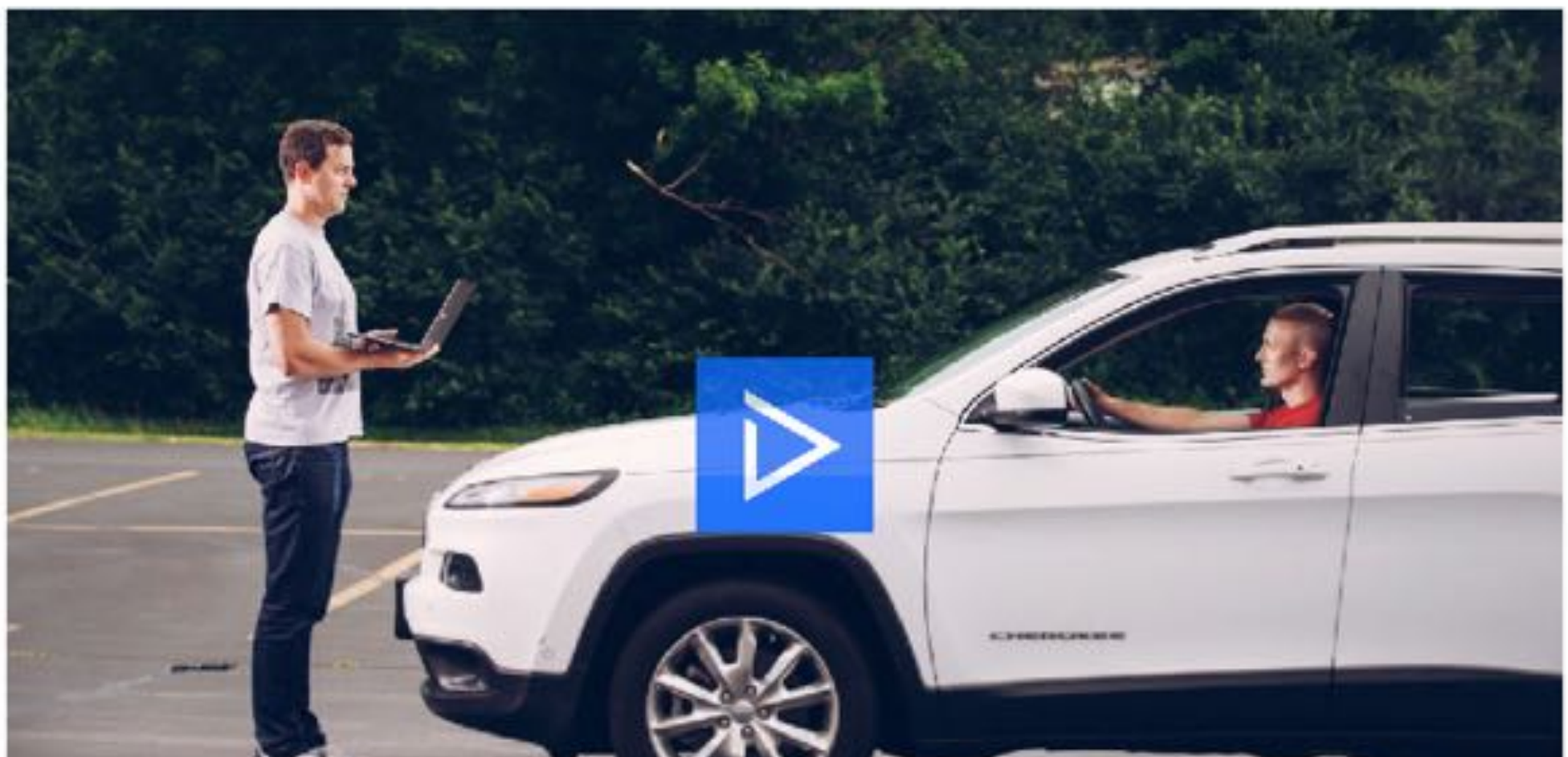
joint work with Nikolas Havrikov, Matthias Höschele, Alexander Kampmann, Konrad Jamrozik

https://www.st.cs.uni-saarland.de/

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7) / ERC Grant Agreement n. 290914 SPECMATE


Saarbrücken

HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT



Thermostats can now get infected with ransomware, because 2016

by MATTHEW HUGHES · 29 days ago in GADGETS

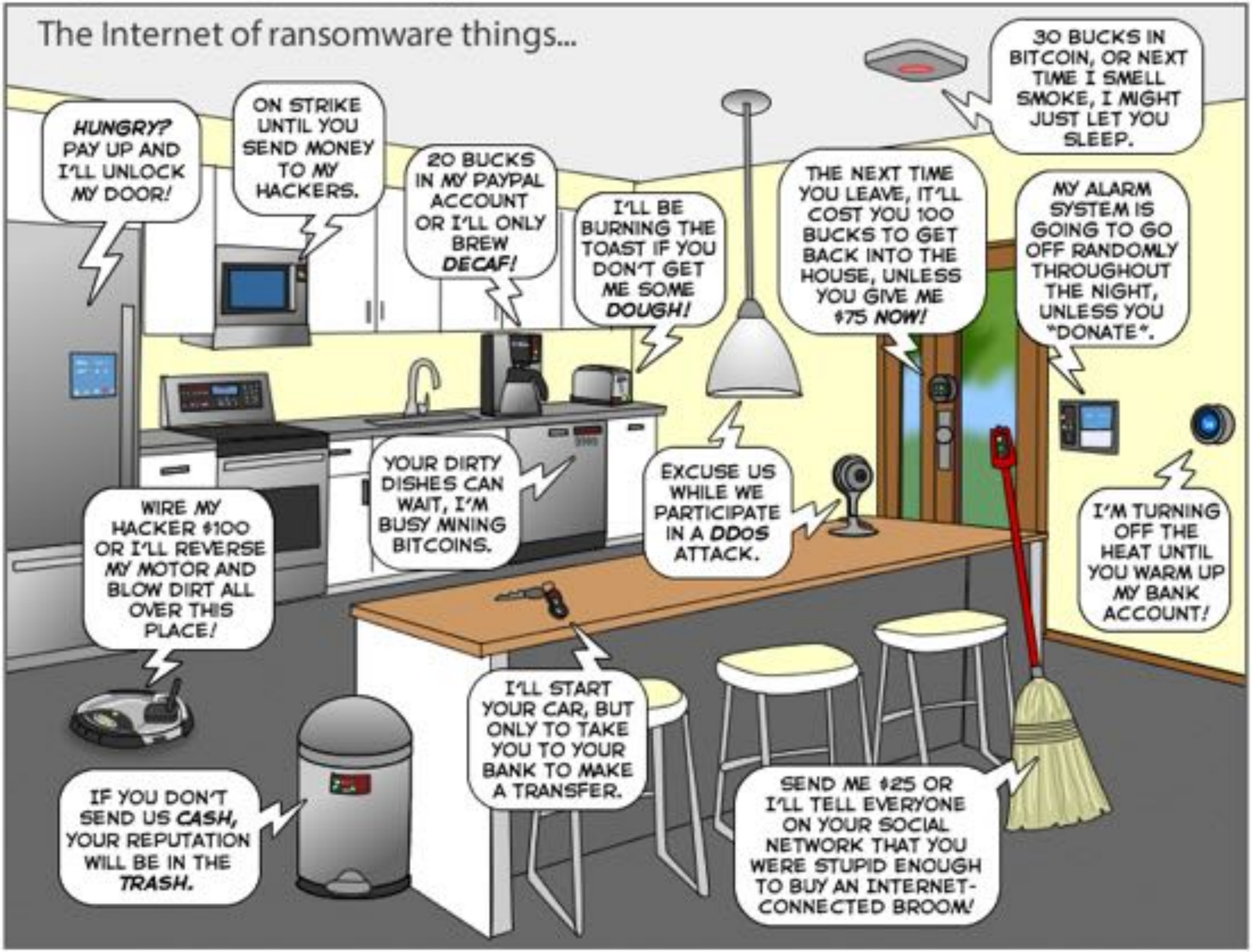
Recommended

5 reasons why wearables are still ruling our wrists (and everywhere else)

Most popular

1. Google Maps now has a 'Catching Pokémon' feature in Timeline
2. Facebook is testing a new Twitter-like feature to boost conversations
3. The world's first VR ballet experience is absolutely stunning
4. The best Apple Keynotes to watch before Wednesday's iPhone 7 Keynote
5. Warner Bros. shoots itself in the foot as it flags its own website for piracy

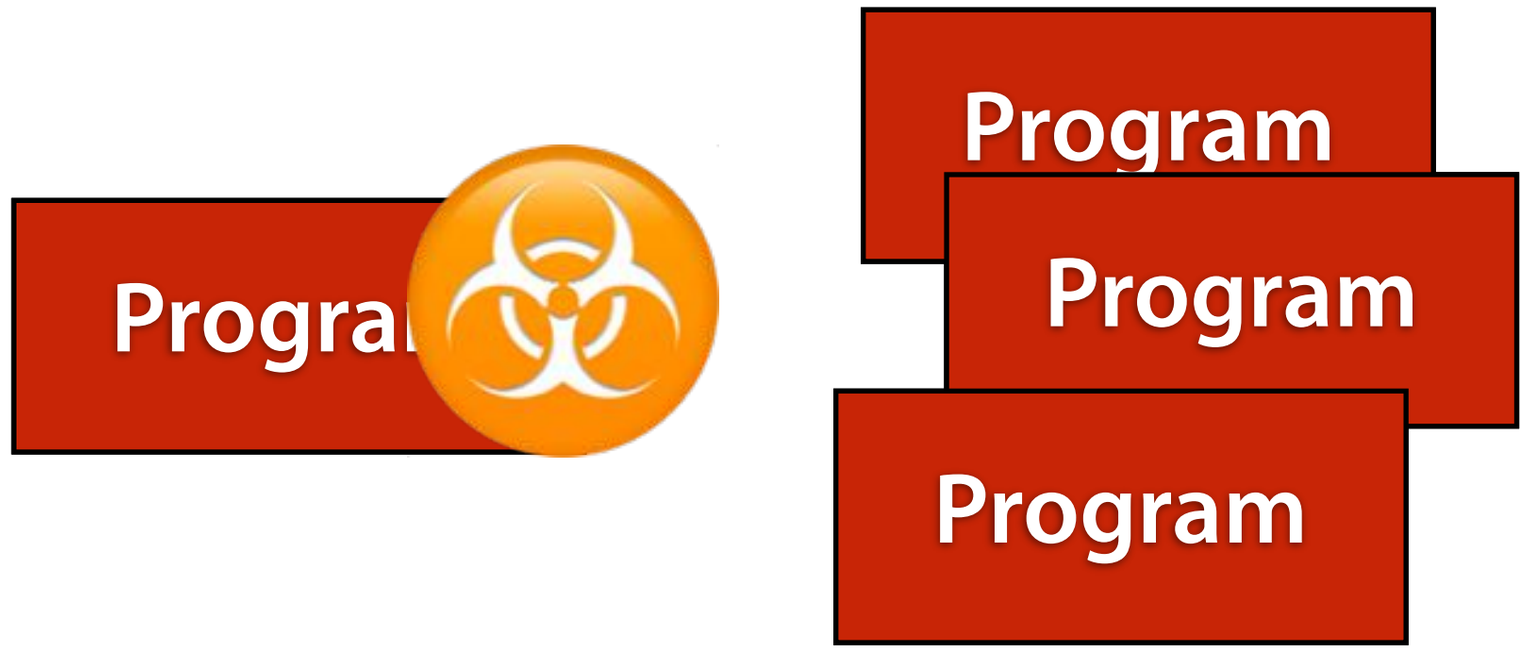
The Joy of Tech™ by Nitrozac & Snaggy



External Attacks

- At the heart of each attack is a *change in program behavior*

Latent Malware

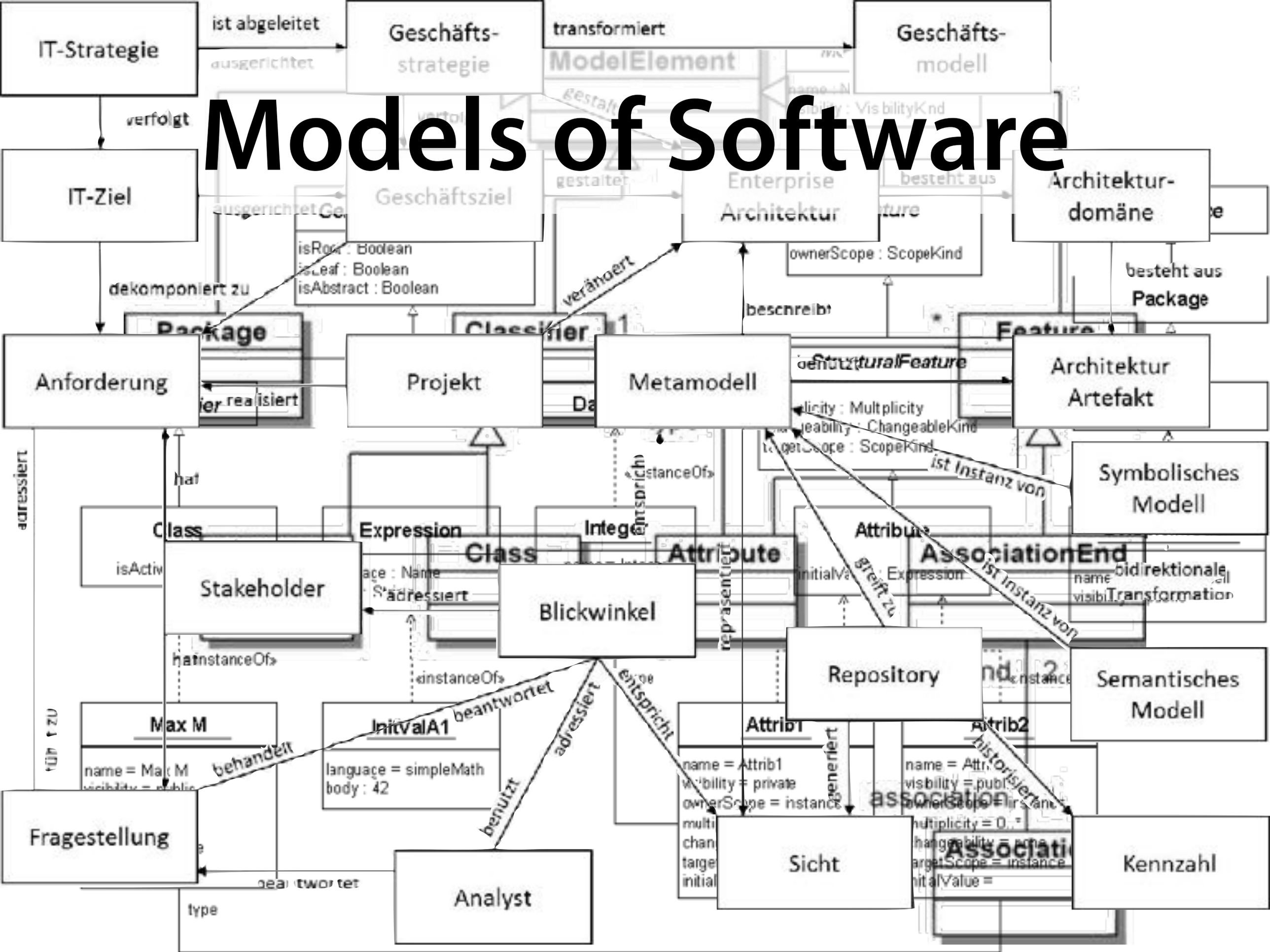


- At the heart of each attack is a *change in program behavior*

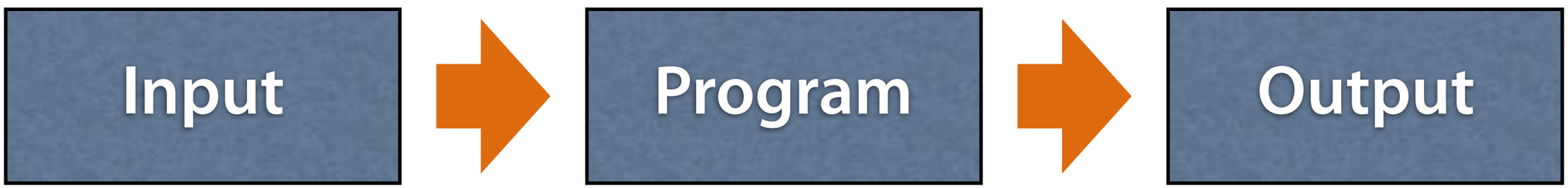
Behavior Changes

- At the heart of each attack is a *change in program behavior*
- How can we *characterize* and *constrain* program behavior?

Models of Software



Program Behavior



- Which inputs does the program accept?
- Which outputs can the program produce?

Language Models

- A *language* denotes a set of strings
- Modeled as regular expressions, grammars, …

```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp' | ...
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= /[0-9]+/
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= /[a-z0-9=&]+/
REF ::= /[a-z]+/
```

Modeling Behavior


```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp' | ...
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= /[0-9]+/
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= /[a-z0-9=&]+/
REF ::= /[a-z]+/
```

```
REPLY ::= 'HTTP/1.1 ' CODE '\n' \
          HEADER+ '\n\n' DATA
CODE ::= '200 OK' | '404 Not Found'
HEADER ::= ...
DATA ::= ...
```

Mining Input Grammars

Learning
Program
Behavior

Testing
Program
Behavior

Checking
Program
Behavior

fully automatic • scalable • practical

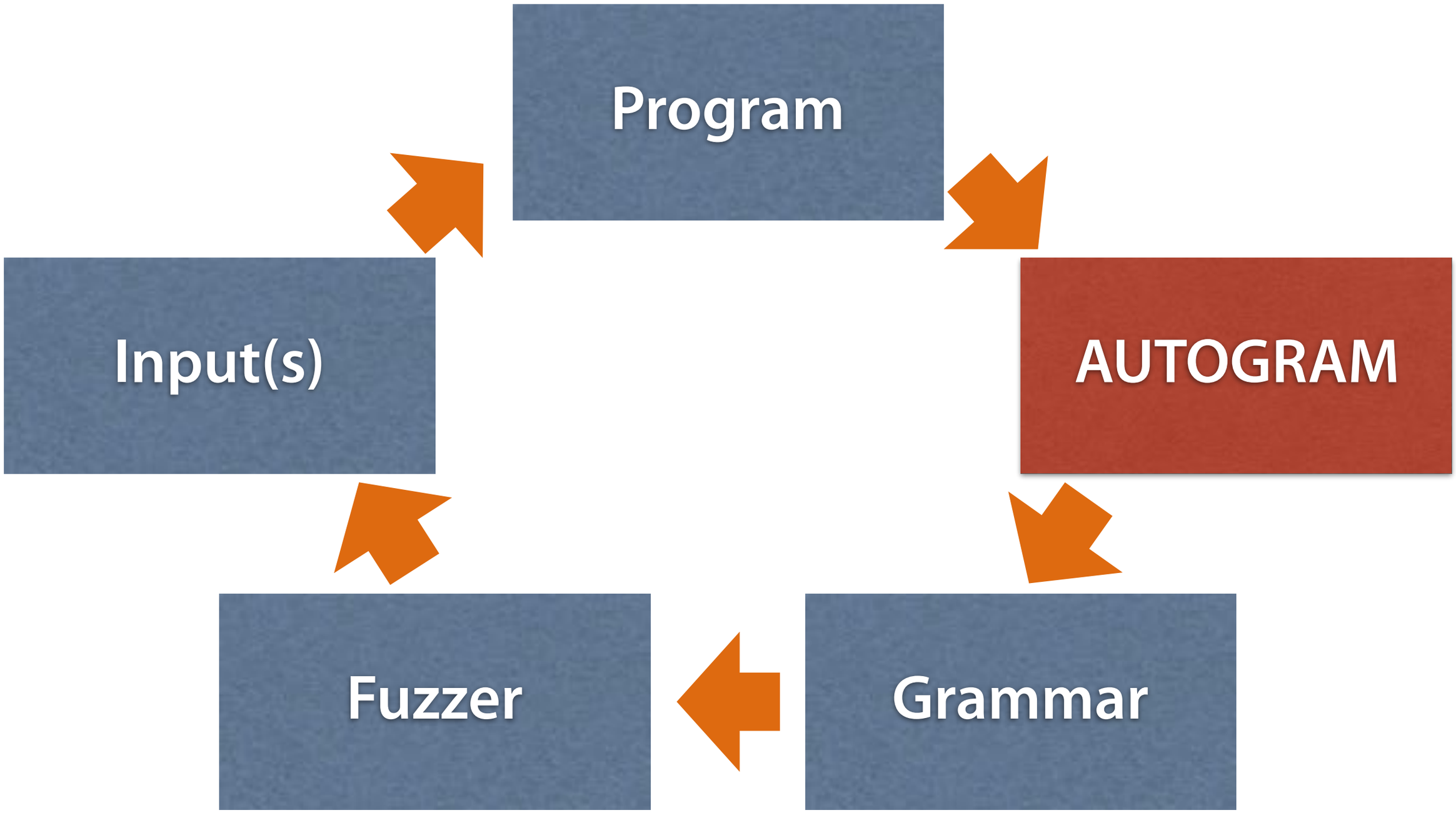
Mining Input Grammars

Learning
Program
Behavior

Creating Grammars

```
URL ::= PR                                  Y] ['#' REF]
AUTHORITY
PROTOCOL :
USERINFO :
HOST ::= /
PORT ::= /
PATH ::= /
QUERY ::=
REF ::= /
```

Learning Grammars



Höschele, Zeller "Mining Input Grammars from Dynamic Taints", ASE 2016

Learning Grammars

http://user:pass@www.google.com:80/path

Program

Learning Grammars

`http://user:pass@www.google.com:80/path`

`http` – protocol

Learning Grammars

`http://user:pass@www.google.com:80/path`

`http` – protocol

`www.google.com` – host name

Learning Grammars

`http://user:pass@www.google.com:80/path`

`http`	– protocol
`www.google.com`	– host name
`80`	– port

Learning Grammars

`http://user:pass@www.google.com:80/path`

`http`	– protocol
`www.google.com`	– host name
`80`	– port
`user pass`	– login

Learning Grammars

`http://user:pass@www.google.com:80/path`

`http`	– protocol
`www.google.com`	– host name
`80`	– port
`user pass`	– login
`path`	– page request

Learning Grammars

`http://user:pass@www.google.com:80/path`

`http`	– protocol
`www.google.com`	– host name
`80`	– port
`user pass`	– login
`path`	– page request
`:// : @ : /`	– terminals

Learning Grammars

`http://user:pass@www.google.com:80/path`

`http`	– protocol
`www.google.com`	– host name
`80`	– port
`user pass`	– login
`path`	– page request
`://  :  @  :  /`	– terminals

processed in *different functions*

stored in *different variables*

http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    |
  param: protocol
    |
  param: host
    |
  param: port
    |
  param: authority
    |
  param: userinfo
    |
  param: path
    |
  param: query
    |
  param: ref
    |
```

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    |
  param: protocol
    | http
  param: host
    |
  param: port
    |
  param: authority
    |
  param: userinfo
    |
  param: path
    |
  param: query
    |
  param: ref
    |
```

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    |
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |
  param: authority
    |
  param: userinfo
    |
  param: path
    |
  param: query
    |
  param: ref
    |
```

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    |
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |
  param: authority
    |
  param: userinfo
    |        user:password
  param: path
    |
  param: query
    |
  param: ref
    |
```

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    |
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |                                   80
  param: authority
    |
  param: userinfo
    |        user:password
  param: path
    |
  param: query
    |
  param: ref
    |
```

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    |
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |                                  80
  param: authority
    |
  param: userinfo
    |        user:password
  param: path
    |                                     /command
  param: query
    |
  param: ref
    |
```

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    |
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |                                   80
  param: authority
    |
  param: userinfo
    |        user:password
  param: path
    |                                     /command
  param: query
    |                                               foo=bar&lorem=ipsum
  param: ref
    |
```

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    |
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |                                  80
  param: authority
    |
  param: userinfo
    |       user:password
  param: path
    |                                     /command
  param: query
    |                                              foo=bar&lorem=ipsum
  param: ref
    |                                                                  fragment
```

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    |
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |                                  80
  param: authority
    |        user:password@www.google.com:80
  param: userinfo
    |        user:password
  param: path
    |                                  /command
  param: query
    |                                  foo=bar&lorem=ipsum
  param: ref
    |                                                      fragment
```

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    | http   user:password@www.google.com:80/command foo=bar&lorem=ipsum fragment
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |                                  80
  param: authority
    |        user:password@www.google.com:80
  param: userinfo
    |        user:password
  param: path
    |                                  /command
  param: query
    |                                  foo=bar&lorem=ipsum
  param: ref
    |                                                      fragment
```

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    | http    user:password@www.google.com:80/command foo=bar&lorem=ipsum fragment
  param: protocol
    | http
  param: host
    |                      www.google.com
  param: port
    |                                    80
  param: authority
    |          user:password@www.google.com:80
  param: userinfo
    |          user:password
  param: path
    |                                      /command
  param: query
    |                                                foo=bar&lorem=ipsum
  param: ref
    |                                                                    fragment
```

URL ::= PROTOCOL '://' AUTHORITY

AUTHORITY ::= USERINFO '@' HOST

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    | http   user:password@www.google.com:80/command foo=bar&lorem=ipsum fragment
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |                                80
  param: authority
    |        user:password@www.google.com:80
  param: userinfo
    |        user:password
  param: path
    |                                /command
  param: query
    |                                         foo=bar&lorem=ipsum
  param: ref
    |                                                              fragment
```

```
URL ::= PROTOCOL '://' AUTHORITY PATH '?' QUERY '#' REF
AUTHORITY ::= USERINFO '@' HOST ':' PORT
PROTOCOL ::= 'http'
USERINFO ::= 'user:password'
HOST ::= 'www.google.com'
PORT ::= '80'
PATH ::= '/command'
QUERY ::= 'foo=bar&lorem=ipsum'
REF ::= 'fragment'
```

URLs

```
http://user:password@www.google.com:80/command?foo=bar&lorem=ipsum#fragment
http://www.guardian.co.uk/sports/worldcup#results
ftp://bob:12345@ftp.example.com/oss/debian7.iso
```

```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp'
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= '80'
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= 'foo=bar&lorem=ipsum'
REF ::= /[a-z]+/
```

INI Files

```
[Application]
Version = 0.5
WorkingDir = /tmp/mydir/
[User]
User = Bob
Password = 12345
```

```
INI ::= LINE+
LINE ::= SECTION_LINE '\r'
       | OPTION_LINE  ['\r']
SECTION_LINE ::= '[' KEY ']'
OPTION_LINE ::= KEY ' = ' VALUE
KEY ::= /[a-zA-Z]*/
VALUE ::= /[a-zA-Z0-9\/]/
```

JSON Input

```
{
 "v": true,
 "x": 25,
 "y": -36,
 …
}
```

```
JSON ::= VALUE
VALUE ::= JSONOBJECT | ARRAY | STRINGVALUE |
          TRUE | FALSE | NULL | NUMBER
TRUE ::= 'true'
FALSE ::= 'false'
NULL ::= 'null'
NUMBER ::= ['-'] /[0-9]+/
STRINGVALUE ::= '"' INTERNALSTRING '"'
INTERNALSTRING ::= /[a-zA-Z0-9 ]+/
ARRAY ::= '['
 [VALUE [',' VALUE]+]
']'
JSONOBJECT ::= '{'
 [STRINGVALUE ':' VALUE
  [',' STRINGVALUE ':' VALUE]
  +]
'}'
```

AUTOGRAM Grammars

- give insights into the *structure of inputs*
  → reverse engineering
  → writing tests
  → writing parsers
- first technique to mine input grammars from programs

  fully automatic • scalable • practical

Mining Input Grammars

Learning
Program Behavior

fully automatic • scalable • practical

Mining Input Grammars

Learning
Program
Behavior

Testing
Program
Behavior

Checking
Program
Behavior

fully automatic • scalable • practical

Mining Input Grammars

Testing
Program
Behavior

Fuzz Testing

```
[;x1-GPZ+wcckc];,N9J+?#6^6\e?]9lu2_%'4GX"0VUB[E/r
~fApu6b8<{%siq8Zh.6{V,hr?;{Ti.r3PIxMMMv6{xS^+'Hq!
AxB"YXRS@!Kd6;wtAMefFWM(`|J_<1~o}z3K(CCzRH
JIIvHz>_*.\>JrlU32~eGP?lR=bF3+;y$3lodQ<B89!
5"W2fK*vE7v{')KC-i,c{<[~m!]o;{.'}Gj\(X}
EtYetrpbY@aGZ1{P!AZU7x#4(Rtn!q4nCwqol^y6}0|
Ko=*JK~;zMKV=9Nai:wxu{J&UV#HaU)*BiC<),`+t*gka<W=Z.
%T5WGHZpI30D<Pq>&]BS6R&j?#tP7iaV}-}`\?[_[Z^LBMPG-
FKj'\xwuZ1=Q`^`5,$N$Q@[!CuRzJ2D|vBy!^zkhdf3C5PAkR?
V hn|
3='i2Qx]D$qs4O`1@fevnG'2\11Vf3piU37@55ap\zIyl"'f,
$ee,J4Gw:cgNKLie3nx9(`efSlg6#[K"@WjhZ}
r[Scun&sBCS,T[/vY'pduwgzDlVNy7'rnzxNwI)(ynBa>%|
b`;`9fG]P_0hdG~$@6 3]KAeEnQ7lU)3Pn,0)G/6N-wyzj/
MTd#A;r
```

Fuzz Testing

```
[;x1-GPZ+wcckc];,N9J+?#6^6\e?]9lu2_%'4GX"0VUB[E/r
~fApu6b8<{%siq8Zh.6{V,hr?;{Ti.r3PIxMMMv6{xS^+'Hq!
AxB"YXRS@!Kd6;wtAMefFWM(`|J_<1~o}z3K(CCzRH
JIIvHz>_*.\>JrlU32~eGP?lR=bF3+;y$3lodQ<B89!
5"W2fK*vE7v{')KC-i,c{<[~m!]o;{.'}Gj\(X}
EtYetrpbY@aGZ1{P!AZU7x#4(Rtn!q4nCwqol^y6}0|
Ko=*JK~;zMKV=9Nai:wxu{J&UV#HaU)*BiC<),`+t*gka<W=Z.
%T5WGHZpI30D<Pq>&]BS6R&j?#tP7iaV}-}`\?[_[Z^LBMPG-
FKj'\xwuZ1=Q`^`5,$N$Q@[!CuRzJ2D|vBy!^zkhdf3C5PAkR?
V hn|
3='i2Qx]D$qs4O`1@fevnG'2\11Vf3piU37@55ap\zIyl"'f,
$ee,J4Gw:cgNKLie3nx9(`efSlg6#[K"@WjhZ}
r[Scun&sBCS,T[/vY'pduwgzDlVNy7'rnzxNwI)(ynBa>%|
b`;`9fG]P_0hdG~$@6 3]KAeEnQ7lU)3Pn,0)G/6N-wyzj/
MTd#A;r
```

Syntax Error

An Input Grammar

If Statement

IfStatement^{full} ⇒
  if *ParenthesizedExpression* *Statement*^{full}
| **if** *ParenthesizedExpression* *Statement*^{noShortIf} **else** *Statement*^{full}
IfStatement^{noShortIf} ⇒ **if** *ParenthesizedExpression* *Statement*^{noShortIf} **else** *Statement*^{noShortIf}

Switch Statement

SwitchStatement ⇒
  switch *ParenthesizedExpression* **{ }**
| **switch** *ParenthesizedExpression* **{** *CaseGroups* *LastCaseGroup* **}**
CaseGroups ⇒
  «empty»
| *CaseGroups* *CaseGroup*
CaseGroup ⇒ *CaseGuards* *BlockStatementsPrefix*
LastCaseGroup ⇒ *CaseGuards* *BlockStatements*
CaseGuards ⇒
  CaseGuard
| *CaseGuards* *CaseGuard*
CaseGuard ⇒

Grammar-Based Fuzzing

```
var haystack = "foo";
var re_text  = "^foo";
haystack += "x";
re_text  += "(x)";
var re = new RegExp(re_text);
re.test(haystack);
Reg
prin
```

30 Chromium + Mozilla Security Rewards
53,000 US$ in Bug Bounties

C. Holler

Holler, Herzig, Zeller: "Fuzzing with Code Fragments", USENIX 2012

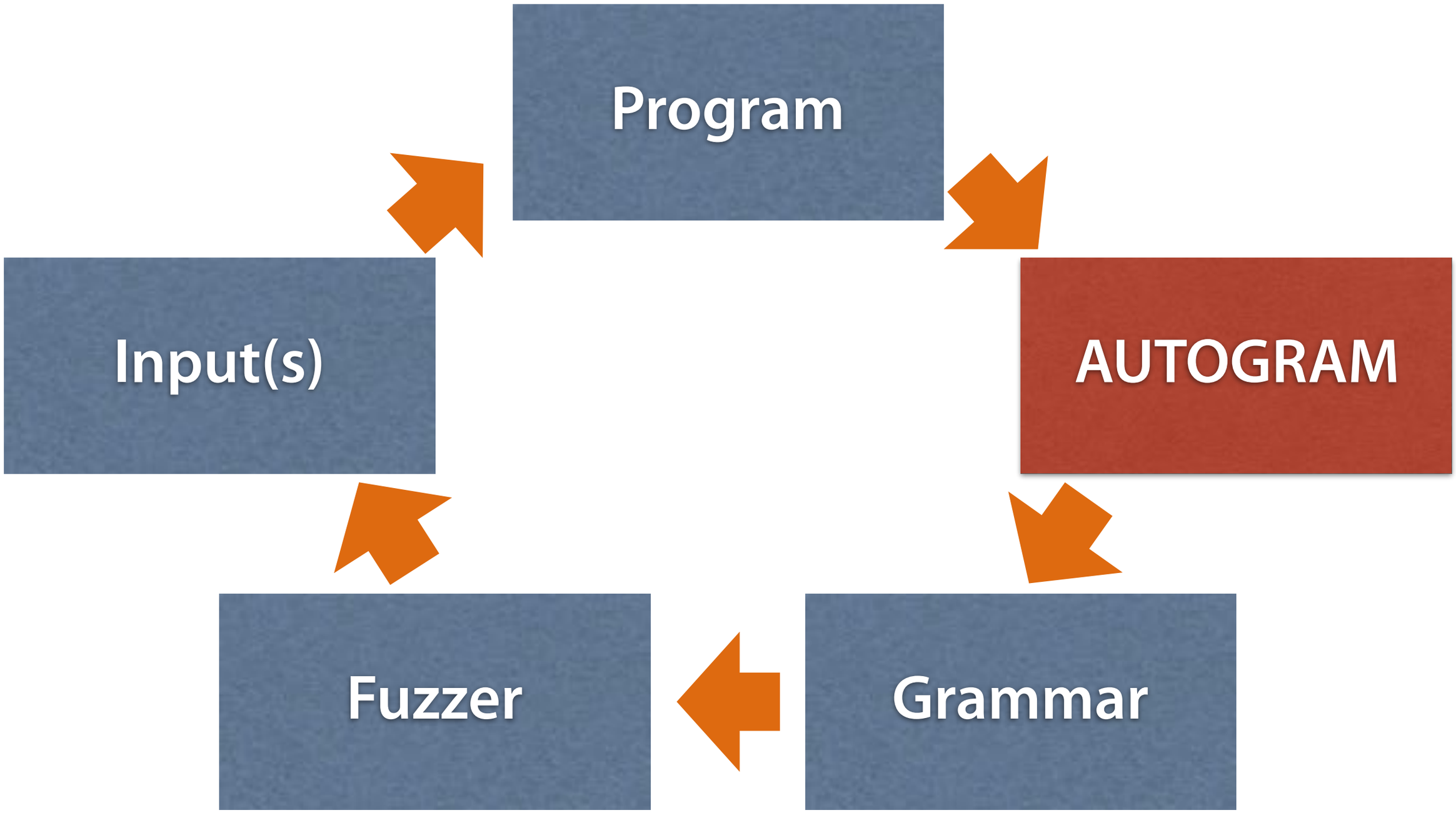
URLs

```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp'
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= '80'
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= 'foo=bar&lorem=ipsum'
REF ::= /[a-z]+/
```

```
http://6F35:PkT5v@2.5/,,
http://.g:8
http://C.Ta.2./p.,//1.#14cq5
http://.37...g:776/,.,
http://.:07//,.8B,#eUN027
http://87.:2117//?&=&&38#207
http://S1t26c:7223i@.1..:16207
ftp://wb428:lr@00.8y.#5W7V9U2
ftp://012304:xt9Ut@k:285?250===K
```

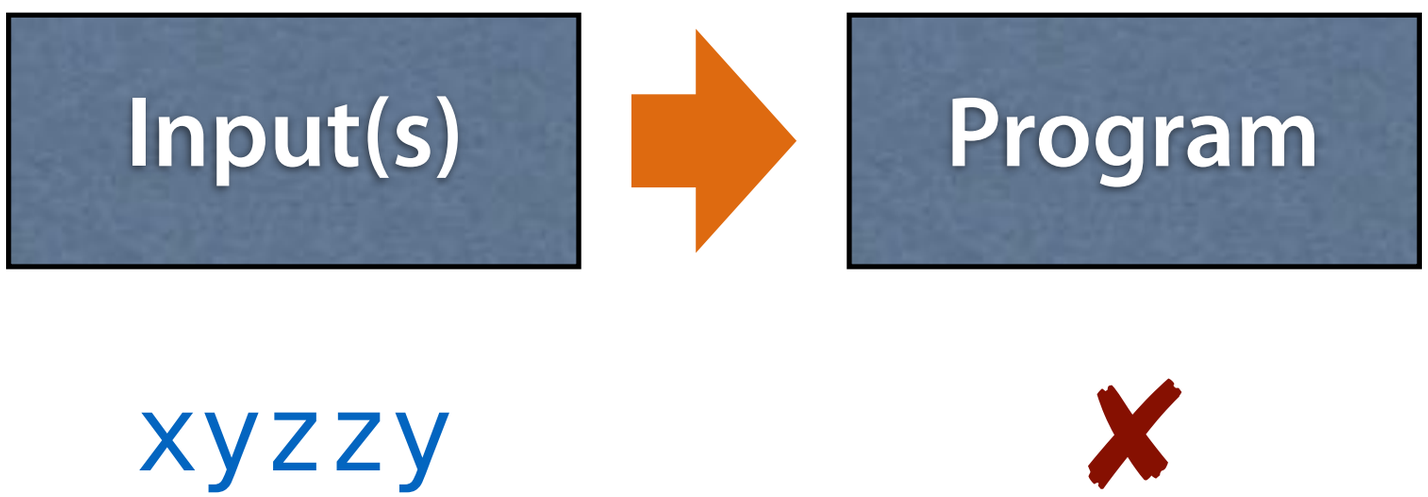
```
http://mE:26Ciu@.8..:1528/8,,2.,,?====&r&
ftp://rW:L@0H....:8111/7.,,g/,
http://D...C
http://2.6j0:032277
http://x1fO..:332334?&==2==&
http://3u8Wabn:tN@m:3592#36
ftp://2.8..:9161208/..?=&=9#5F
ftp://.n:7945457//?9
http://Jy:98/9,3?===&#1q
http://G42:7Nz596e@6.4b//F/,?&I=0
ftp://.697..?===SU=
http://3d00:ud@.1dF9/2q//5
ftp://.d5...8:646#D
ftp://62ql1:40P63@4.:321727?=
http://.//,.,/
ftp://8zN3xl:3499l8@t036./,3?=&=40
http://B7j85D3:NvPd7M@.8.p.:5/,.,#e7JS
http://t4...:124///6,G.?=&&=#3F2Qx
http://YP6:zKG@.:946775?=#Zb7
http://./,31,,F.#693
ftp://7V:c4748C2@.//..,,,.?&&&&&2R
http://..:40123?=r=&7I
ftp://.74:4773362/./A#Et
ftp://67:3g5YNi@.5M.2..:06716?&=#3W758V6
ftp://i:cqj97@..2..3:362287?&=&&7f5#4
http://1:l@N..6..i:667//,,6,
http://7Oo0:518@3:4791089#962
ftp://zA35Qsu:56@..5..:997/,.
ftp://8.../5?&n#7i1C7G3
ftp://2:fm0@J.:6208/,Z/H#3GZ747b
http://2:7p54n14@8r09.1
ftp://XK3438:w169KkU@..5R.8?=6g
```

Learning Grammars



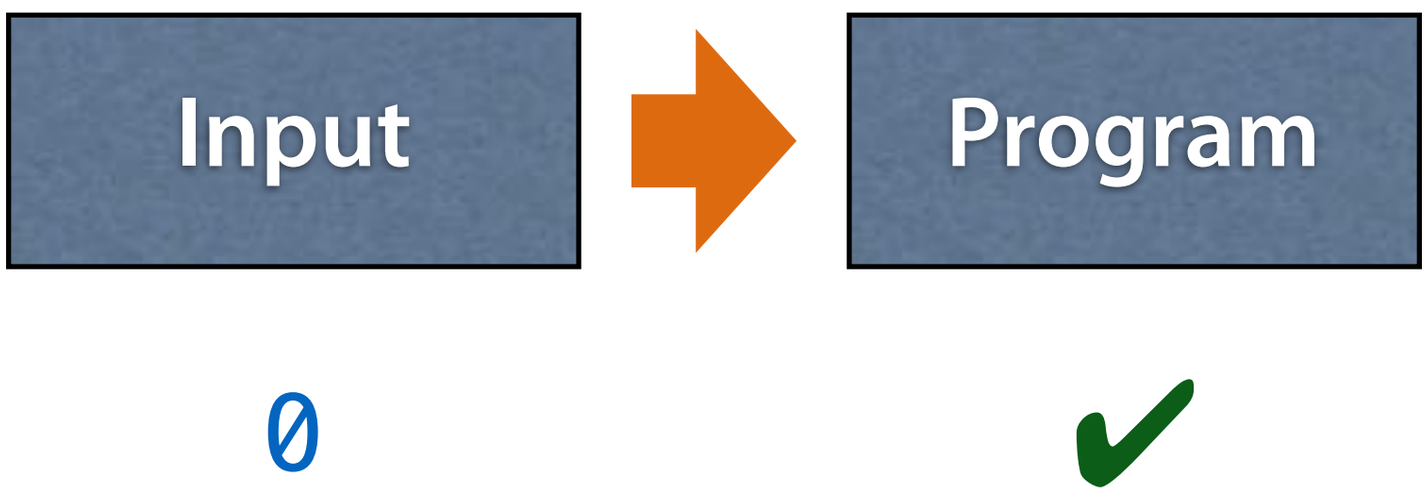
Höschele, Zeller "Mining Input Grammars from Dynamic Taints", ASE 2016

Dynamic Checks

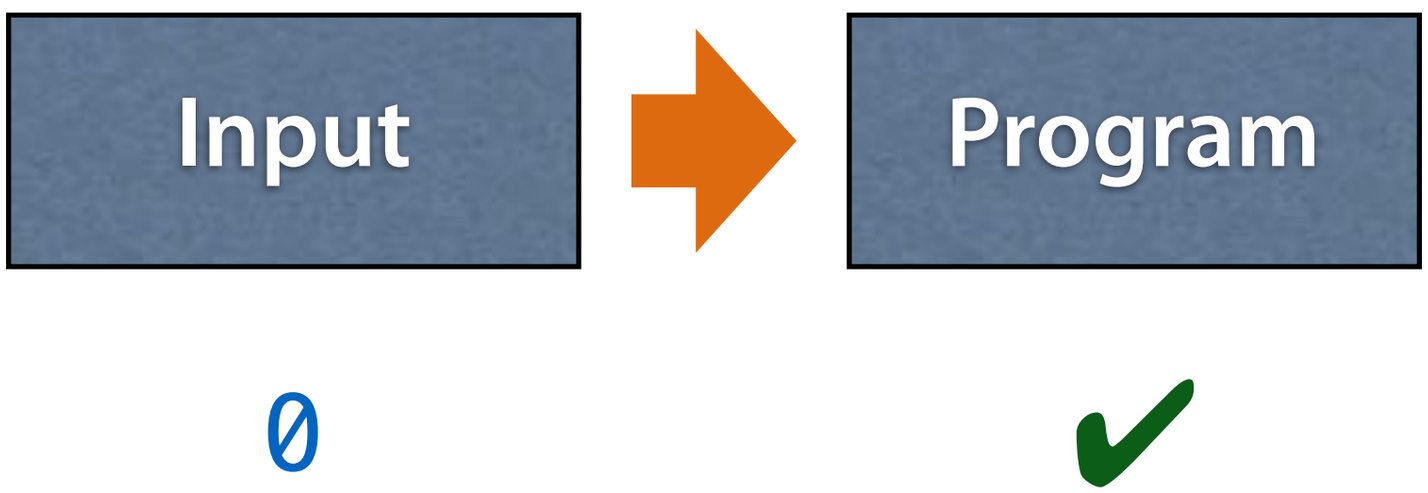


- checks for digit
- checks for "true"/"false"
- checks for ""
- checks for '['
- checks for '{'

Dynamic Checks

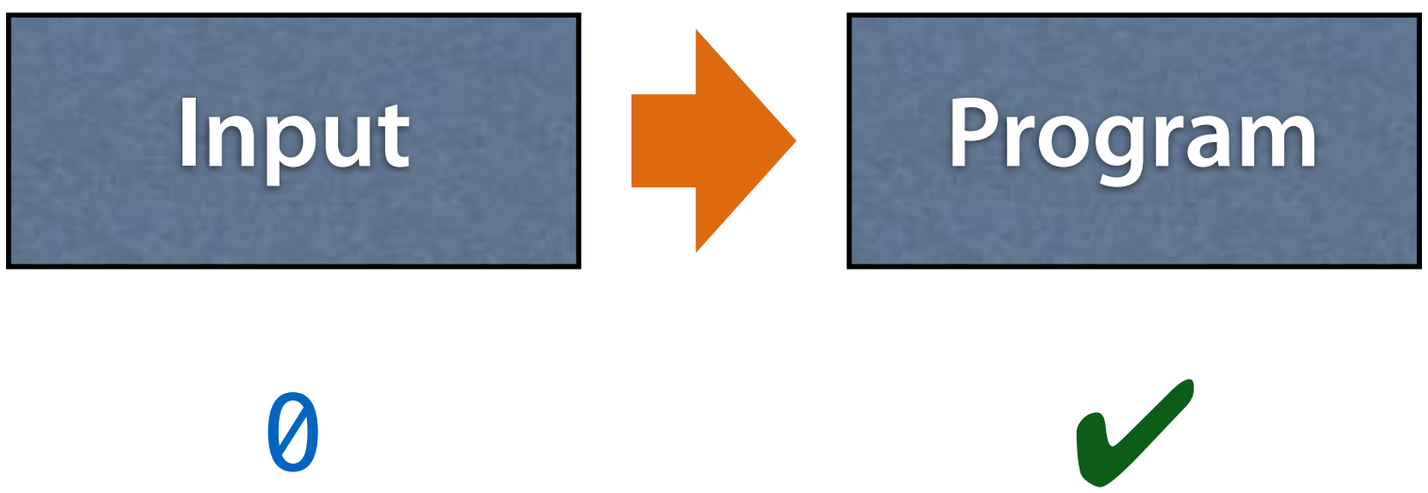


Dynamic Checks



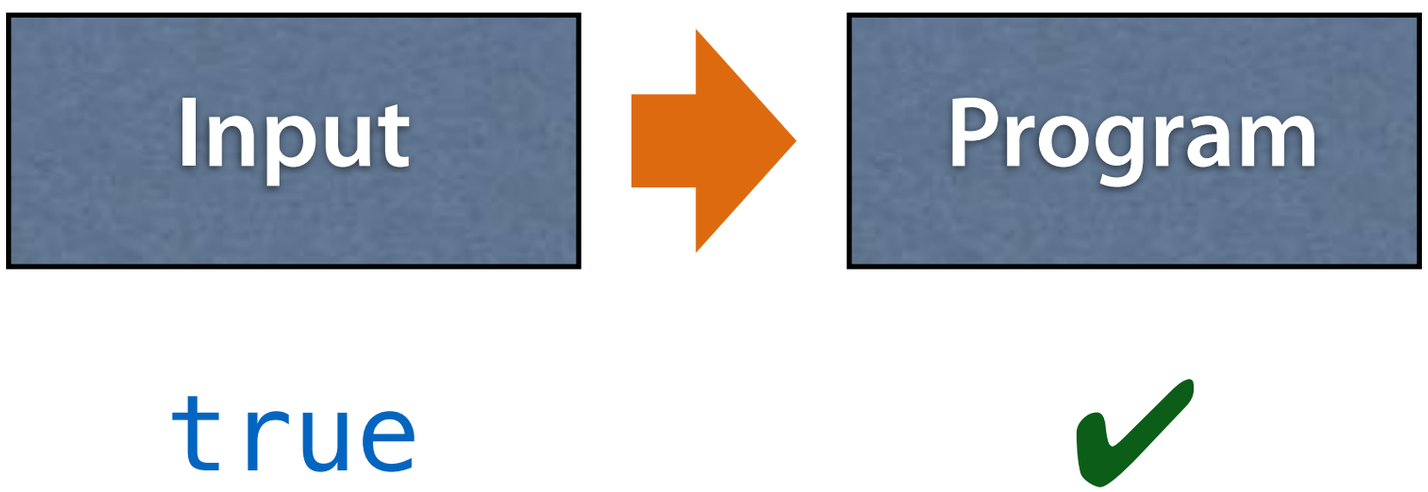
- checks for digit
- checks for "true"/"false"
- checks for ""
- checks for '['
- checks for '{'

Dynamic Checks

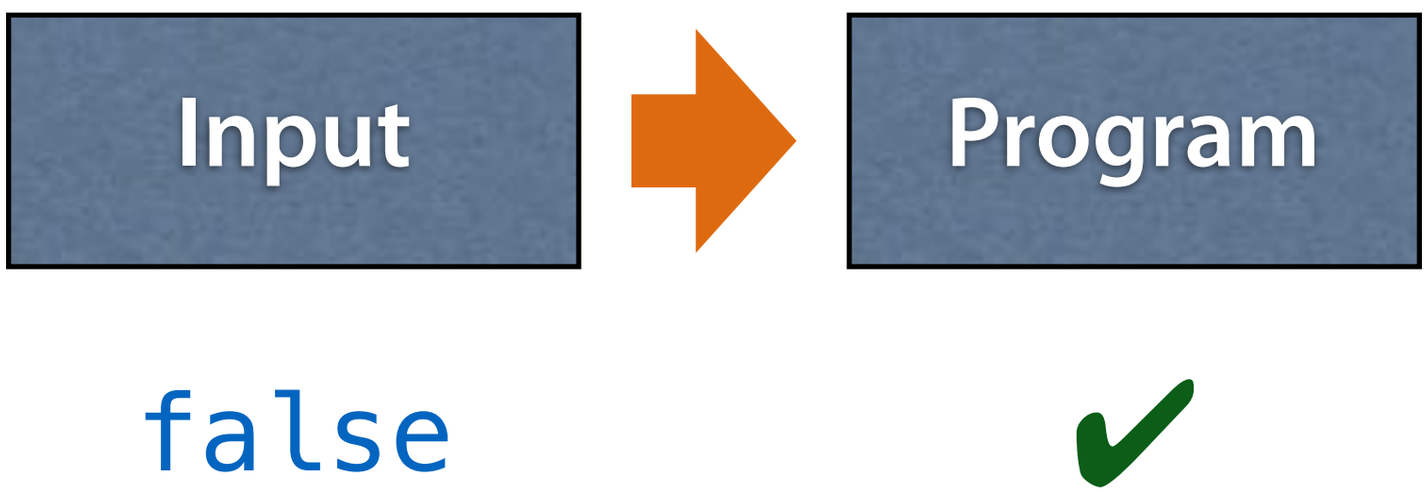


- checks for digit
- checks for "true"/"false"
- checks for ""
- checks for '['
- checks for '{'

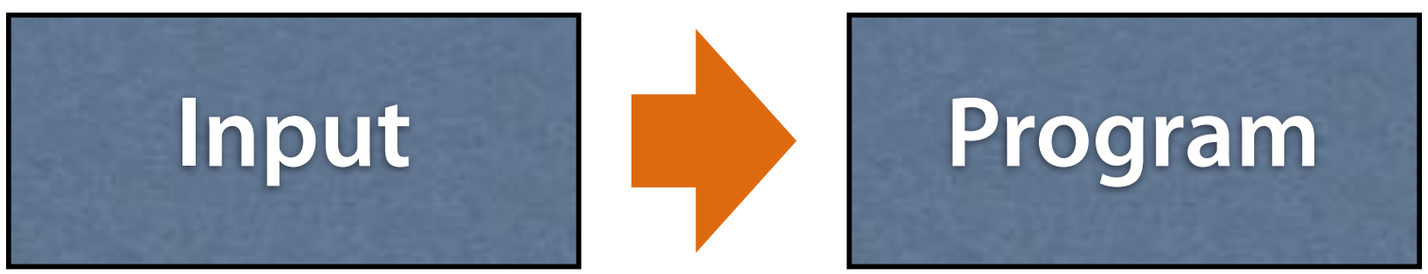
Dynamic Checks



Dynamic Checks



Dynamic Checks

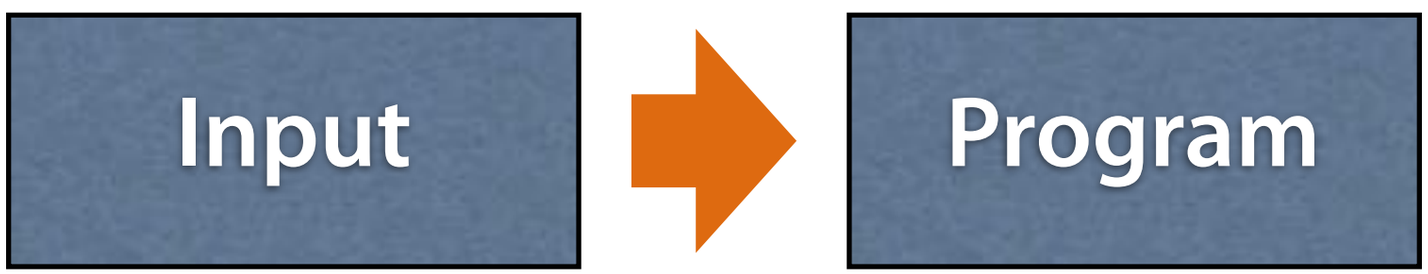


`false`

✔

- checks for digit
- checks for "true"/"false"
- checks for ""
- checks for '['
- checks for '{'

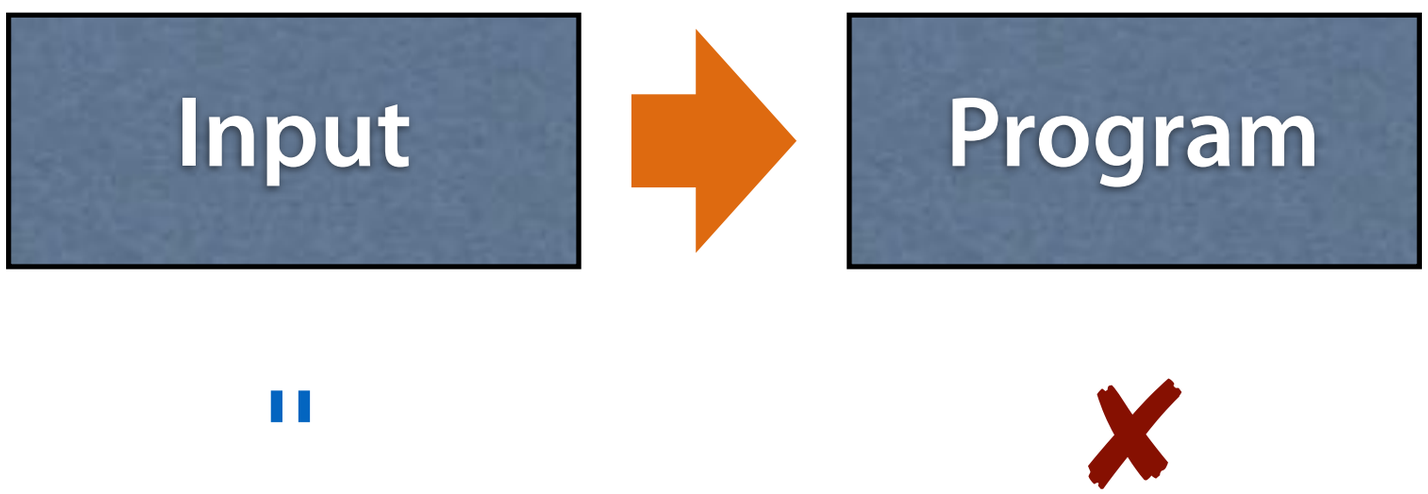
Dynamic Checks



`false` ✔

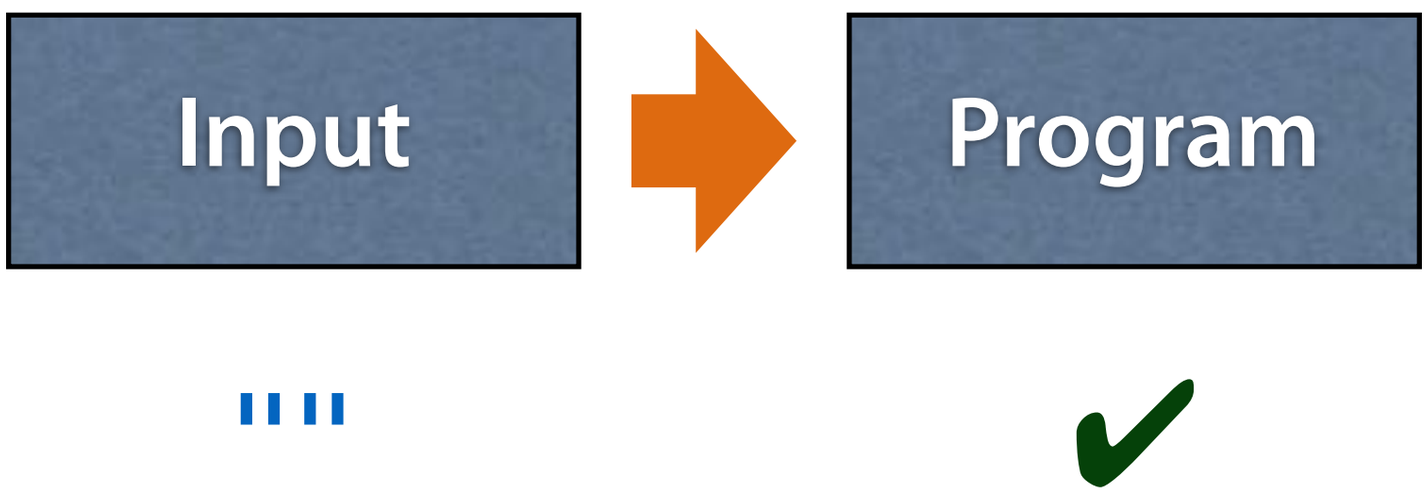
- checks for digit
- checks for "true"/"false"
- checks for ""
- checks for '['
- checks for '{'

Dynamic Checks



- checks for ""
- checks for '\'
- checks for character

Dynamic Checks



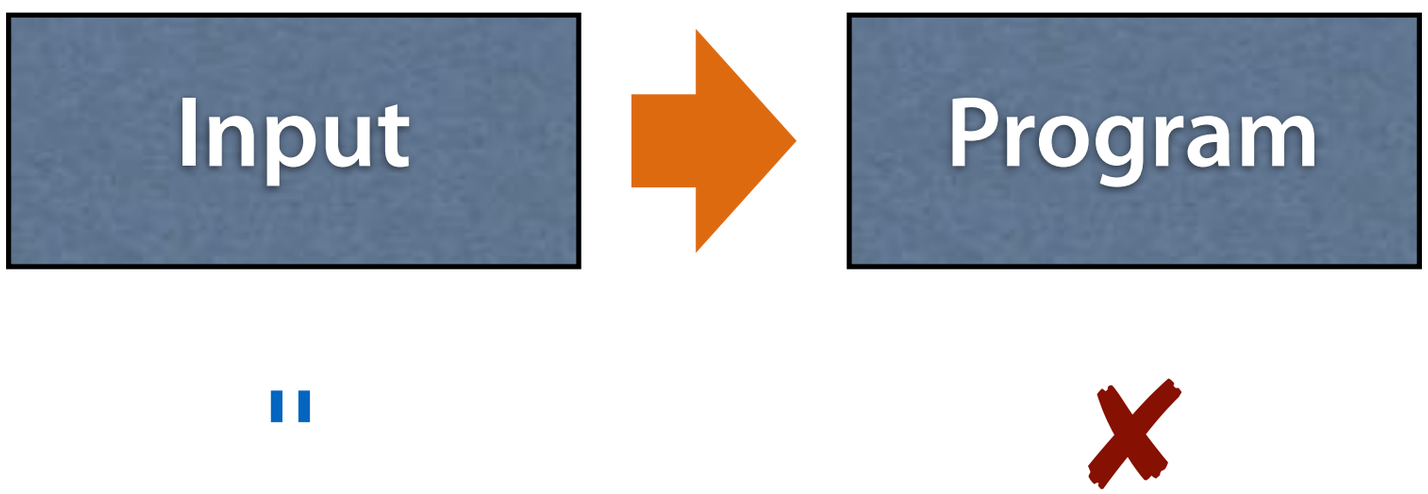
JSON Input

```
{
 "v": true,
 "x": 25,
 "y": -36,
 …
}
```

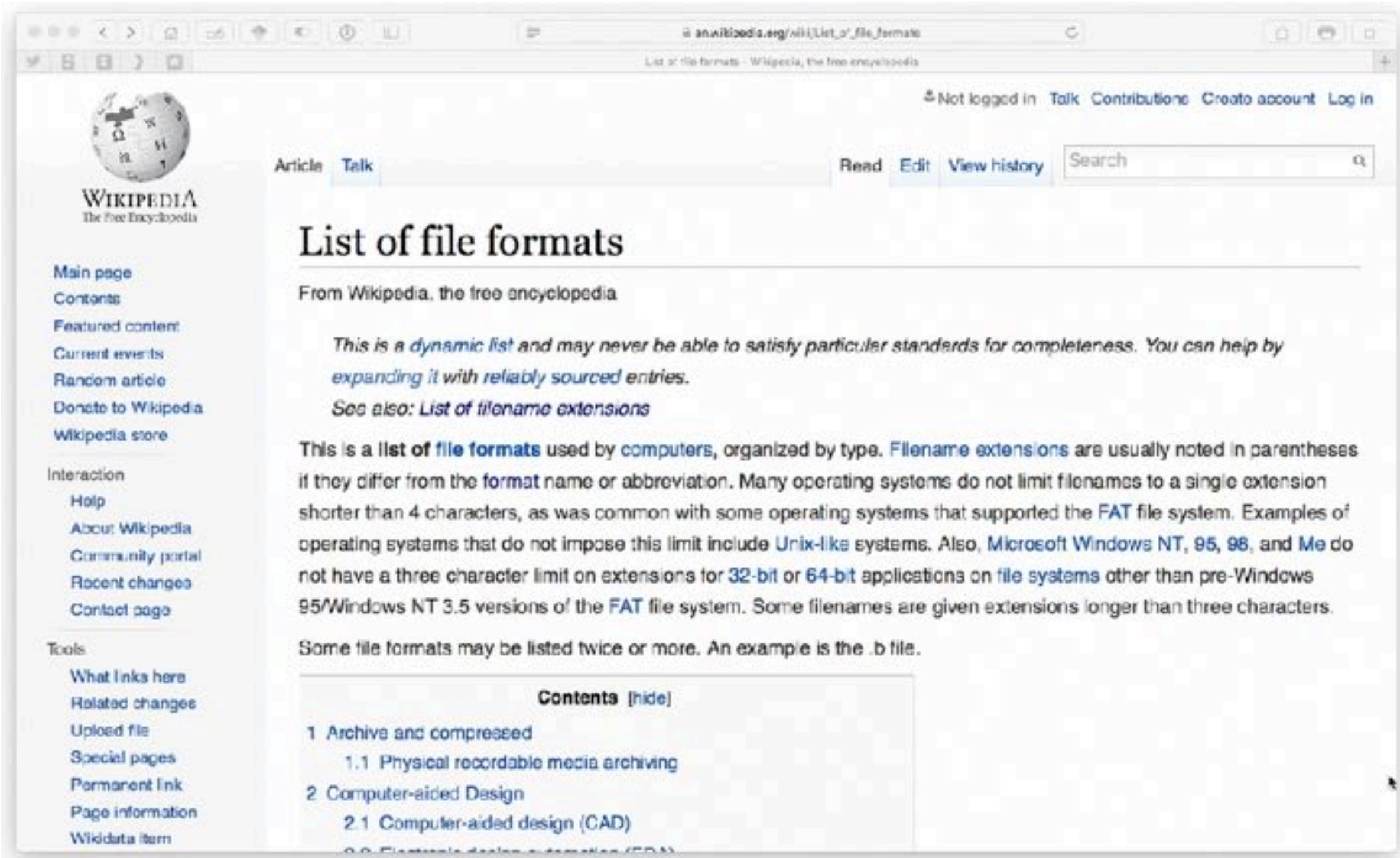
```
JSON ::= VALUE
VALUE ::= JSONOBJECT | ARRAY | STRINGVALUE |
          TRUE | FALSE | NULL | NUMBER
TRUE ::= 'true'
FALSE ::= 'false'
NULL ::= 'null'
NUMBER ::= ['-'] /[0-9]+/
STRINGVALUE ::= '"' INTERNALSTRING '"'
INTERNALSTRING ::= /[a-zA-Z0-9 ]+/
ARRAY ::= '['
 [VALUE [',' VALUE]+]
']'
JSONOBJECT ::= '{'
 [STRINGVALUE ':' VALUE
  [',' STRINGVALUE ':' VALUE]
  +]
'}'
```

Fuzz Testing

Dynamic Checks



File Formats



Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item

List of file formats

From Wikipedia, the free encyclopedia

This is a dynamic list and may never be able to satisfy particular standards for completeness. You can help by expanding it with reliably sourced entries.

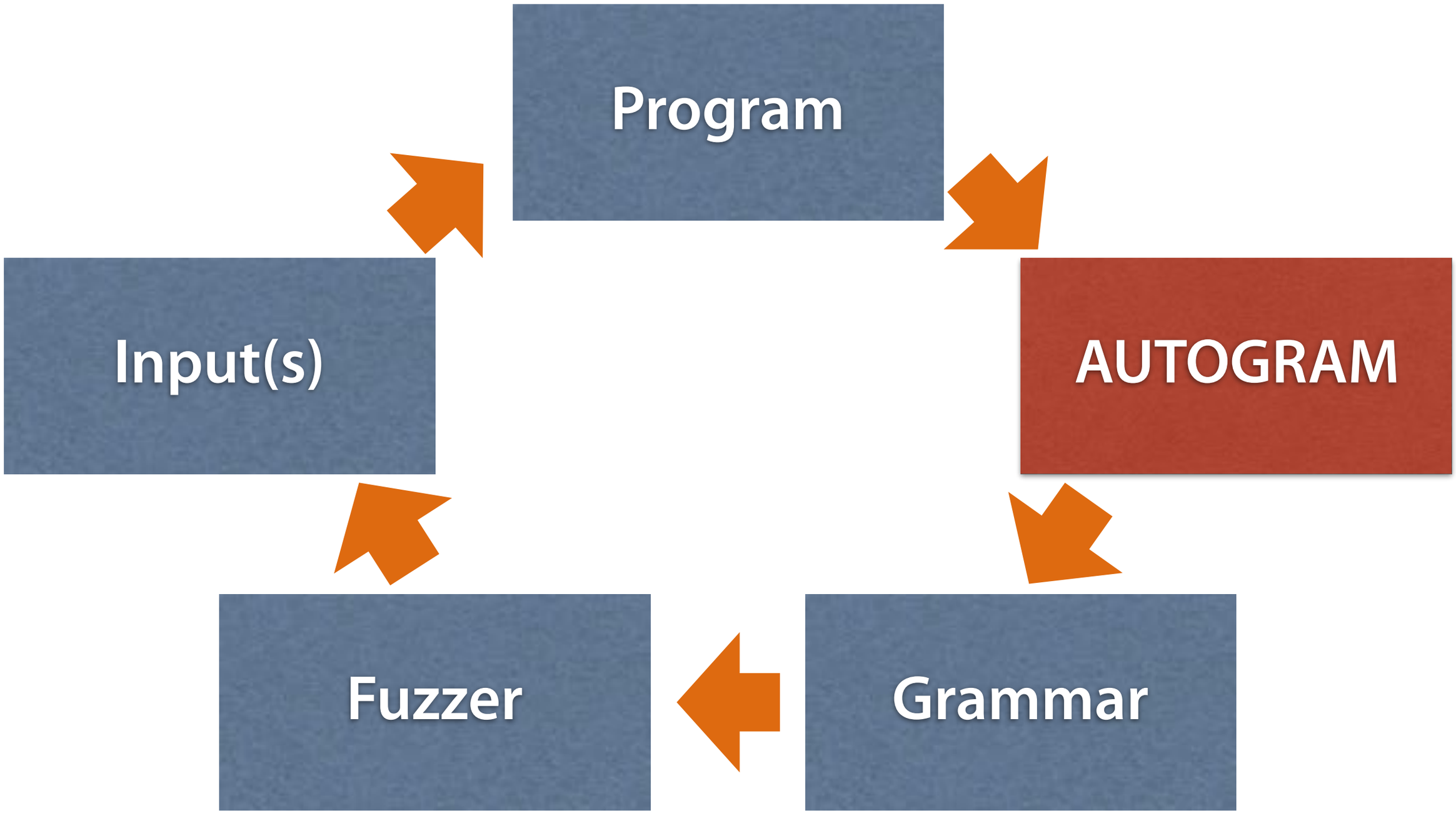
See also: List of filename extensions

This is a **list of file formats** used by computers, organized by type. Filename extensions are usually noted in parentheses if they differ from the format name or abbreviation. Many operating systems do not limit filenames to a single extension shorter than 4 characters, as was common with some operating systems that supported the FAT file system. Examples of operating systems that do not impose this limit include Unix-like systems. Also, Microsoft Windows NT, 95, 98, and Me do not have a three character limit on extensions for 32-bit or 64-bit applications on file systems other than pre-Windows 95/Windows NT 3.5 versions of the FAT file system. Some filenames are given extensions longer than three characters.

Some file formats may be listed twice or more. An example is the .b file.

Contents [hide]

Mining Input Grammars



https://www.st.cs.uni-saarland.de/models/autogram/

Testing Grammars

- *Test generation + dynamic tracking of comparisons* can infer input grammars
- Works even without any input samples
- Resulting grammars can be directly fed into *automated fuzzing tools*

fully automatic • scalable • practical

Mining Input Grammars

Testing
Program Behavior

fully automatic • scalable • practical

TECHNOLOGY

DARPA'S CYBER GRAND CHALLENGE ENDS IN TRIUMPH

A MACHINE NAMED MAYHEM TOOK HOME THE $2 MILLION PRIZE

By **Kelsey D. Atherton** August 8, 2016

Related Content

DARPA's Cyber Insider Threat Program Is the Agency's Great Hope for Ending Leaks

Hacking Phones With The

Mining Input Grammars

Testing
Program Behavior

fully automatic • scalable • practical

Mining Input Grammars

Learning
Program
Behavior

Testing
Program
Behavior

Checking
Program
Behavior

fully automatic • scalable • practical

Mining Input Grammars

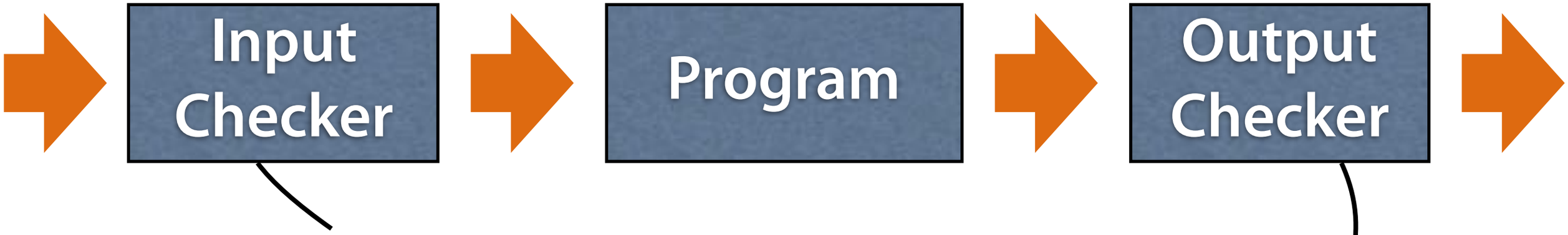
Checking
Program
Behavior

Modeling Behavior


```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp' | ...
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= /[0-9]+/
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= /[a-z0-9=&]+/
REF ::= /[a-z]+/
```

```
REPLY ::= 'HTTP/1.1 ' CODE '\n' \
          HEADER+ '\n\n' DATA
CODE ::= '200 OK' | '404 Not Found'
HEADER ::= ...
DATA ::= ...
```

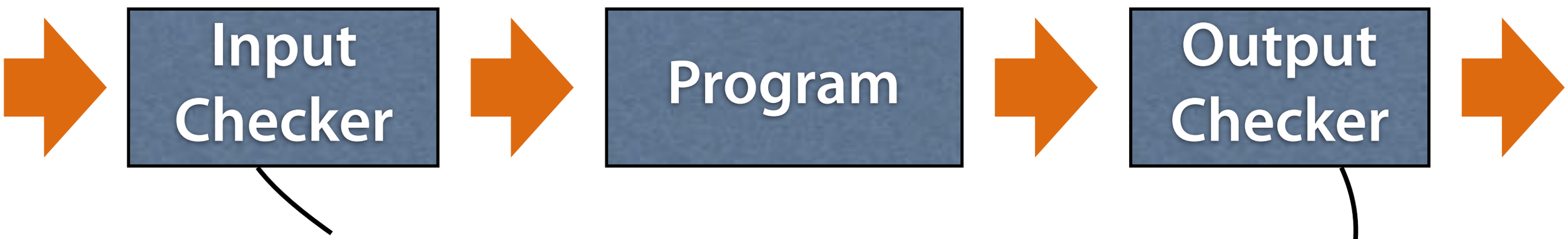
Checking Behavior




```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp' | ...
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= /[0-9]+/
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= /[a-z0-9=&]+/
REF ::= /[a-z]+/
```

```
REPLY ::= 'HTTP/1.1 ' CODE '\n' \
          HEADER+ '\n\n' DATA
CODE ::= '200 OK' | '404 Not Found'
HEADER ::= ...
DATA ::= ...
```

Resisting Attacks




```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp' | ...
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= /[0-9]+/
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= /[a-z0-9=&]+/
REF ::= /[a-z]+/
```

```
REPLY ::= 'HTTP/1.1 ' CODE '\n' \
          HEADER+ '\n\n' DATA
CODE ::= '200 OK' | '404 Not Found'
HEADER ::= ...
DATA ::= ...
```

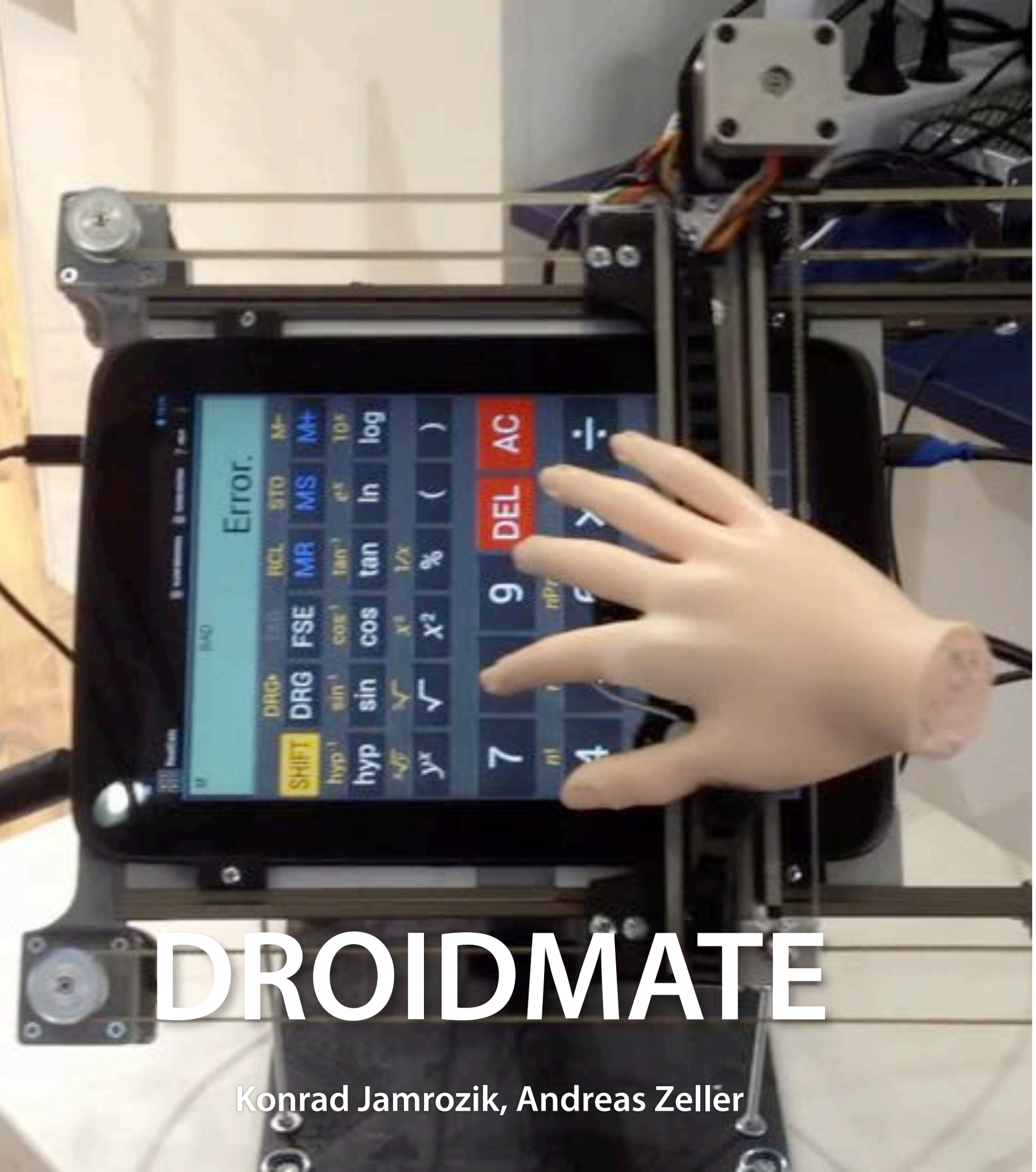
Resisting Attacks

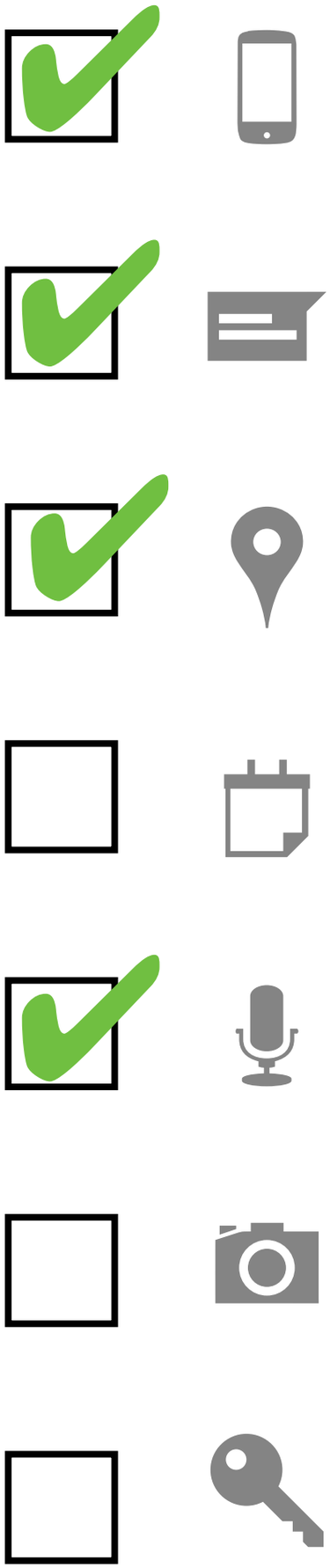

```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp' | ...
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= /[0-9]+/
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= /[a-z0-9=&]+/
REF ::= /[a-z]+/
```

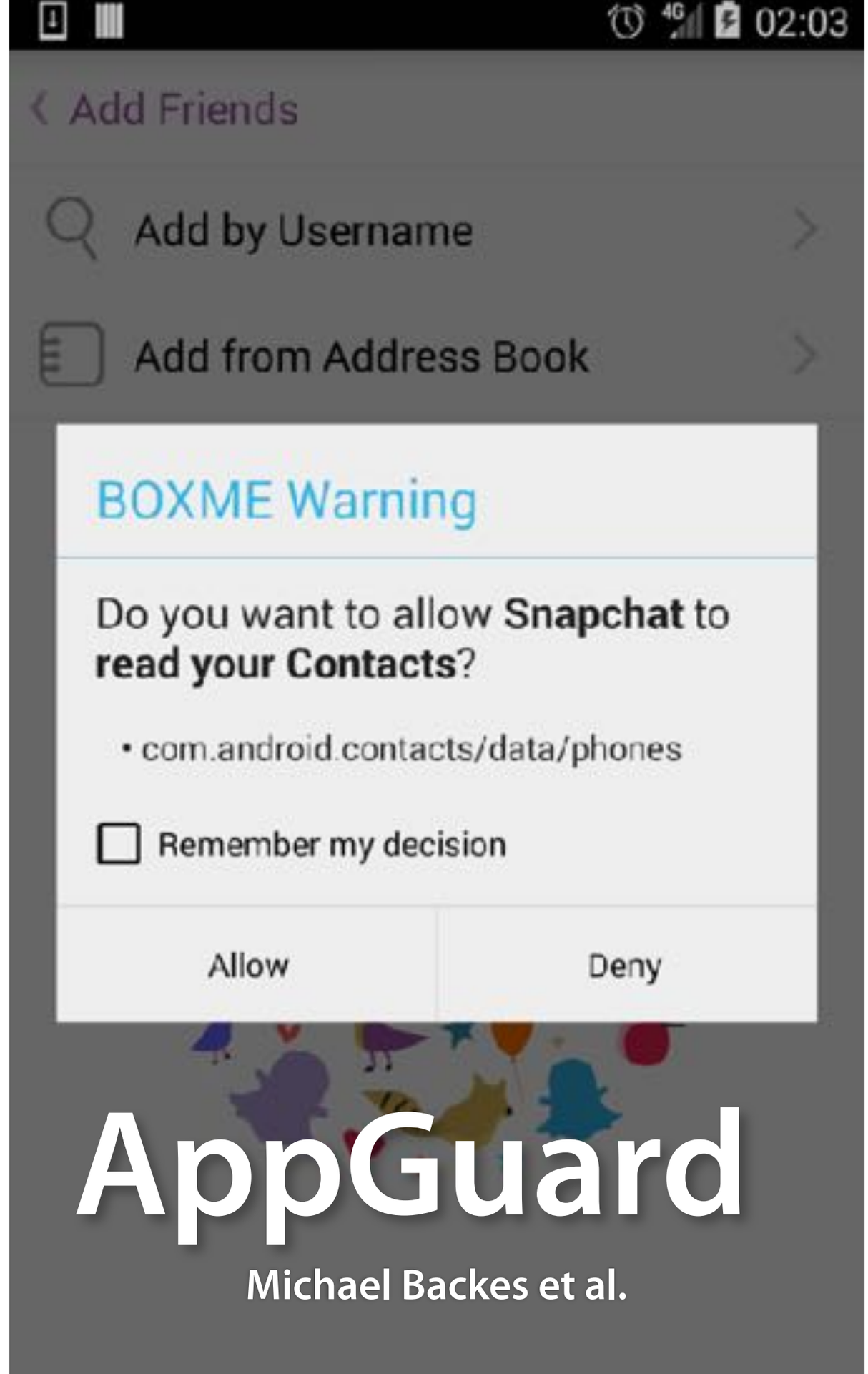
```
REPLY ::= 'HTTP/1.1 ' CODE '\n' \
          HEADER+ '\n\n' DATA
CODE ::= '200 OK' | '404 Not Found'
HEADER ::= ...
DATA ::= ...
```

DROIDMATE

Konrad Jamrozik, Andreas Zeller





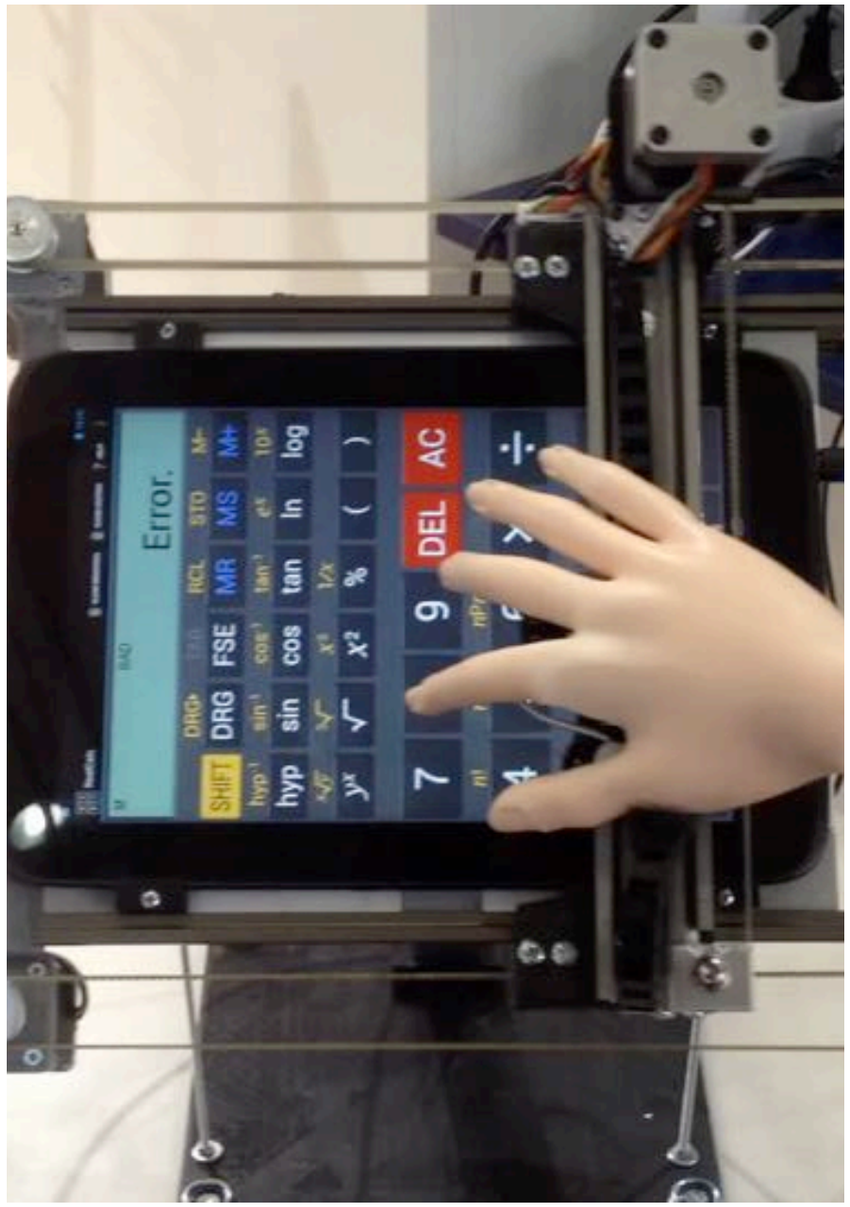


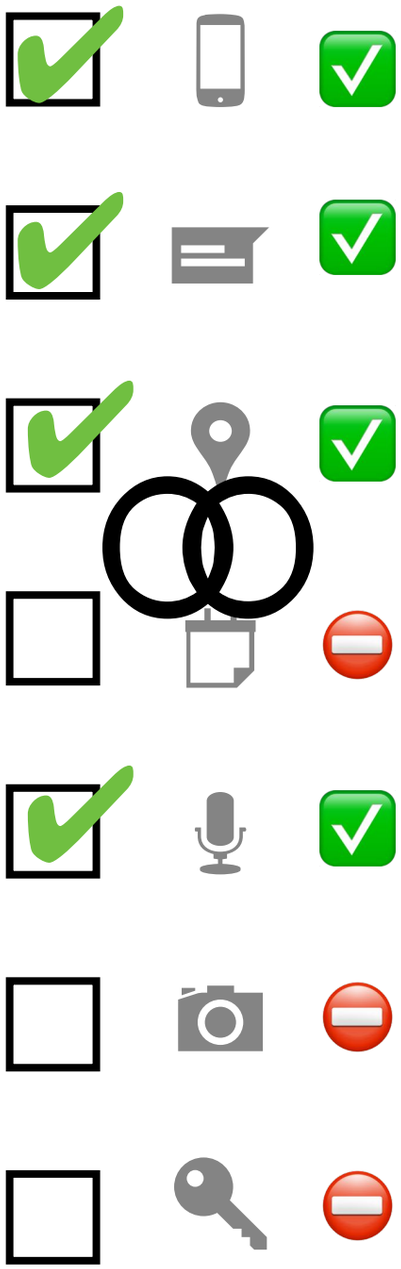
AppGuard

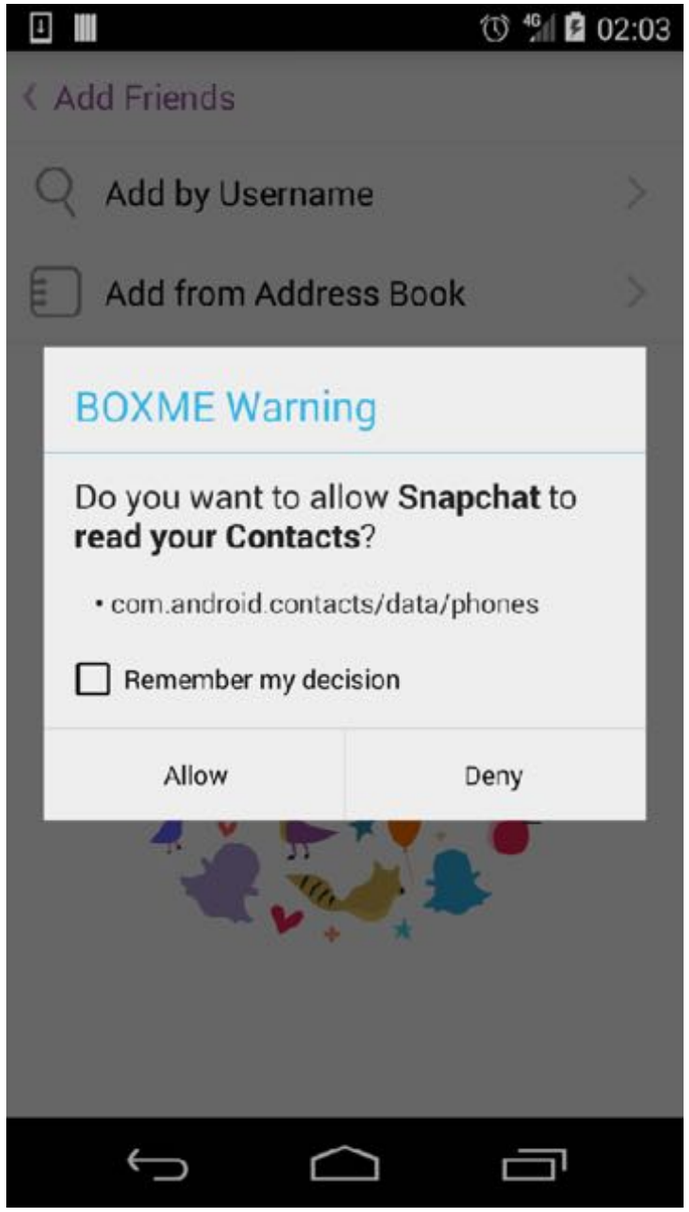
Michael Backes et al.

@AndreasZeller

BOXMATE


Error.
DEL
AC




02:03
Add Friends
Add by Username
Add from Address Book
BOXME Warning
Do you want to allow **Snapchat** to **read your Contacts**?
• com.android.contacts/data/phones
Remember my decision
Allow
Deny

Mining Sandboxes

prevents *unexpected behavior changes*

prevents *latent malware*

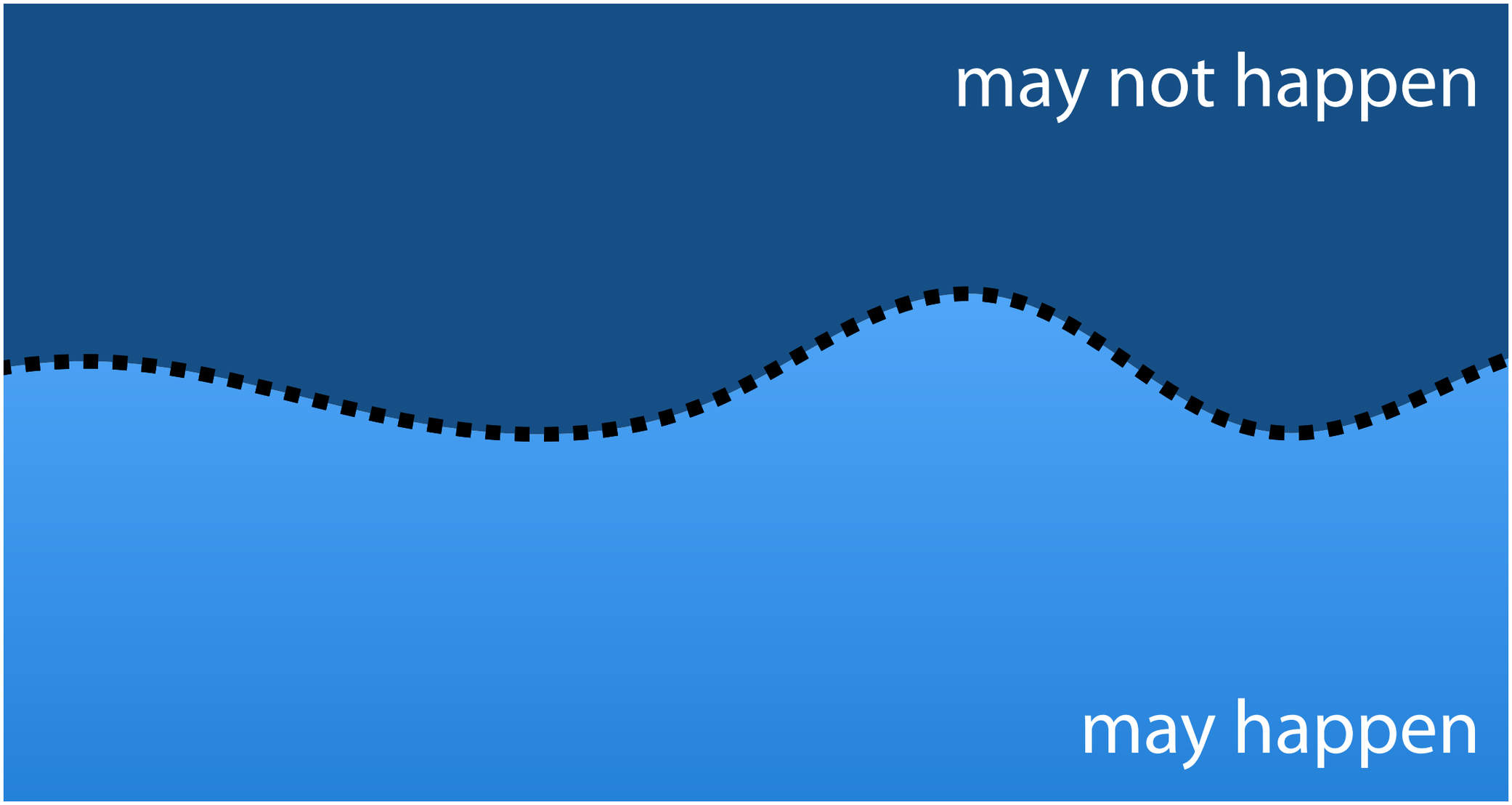
closes *backdoors* and *exploits*

works on adversarial and obscure code

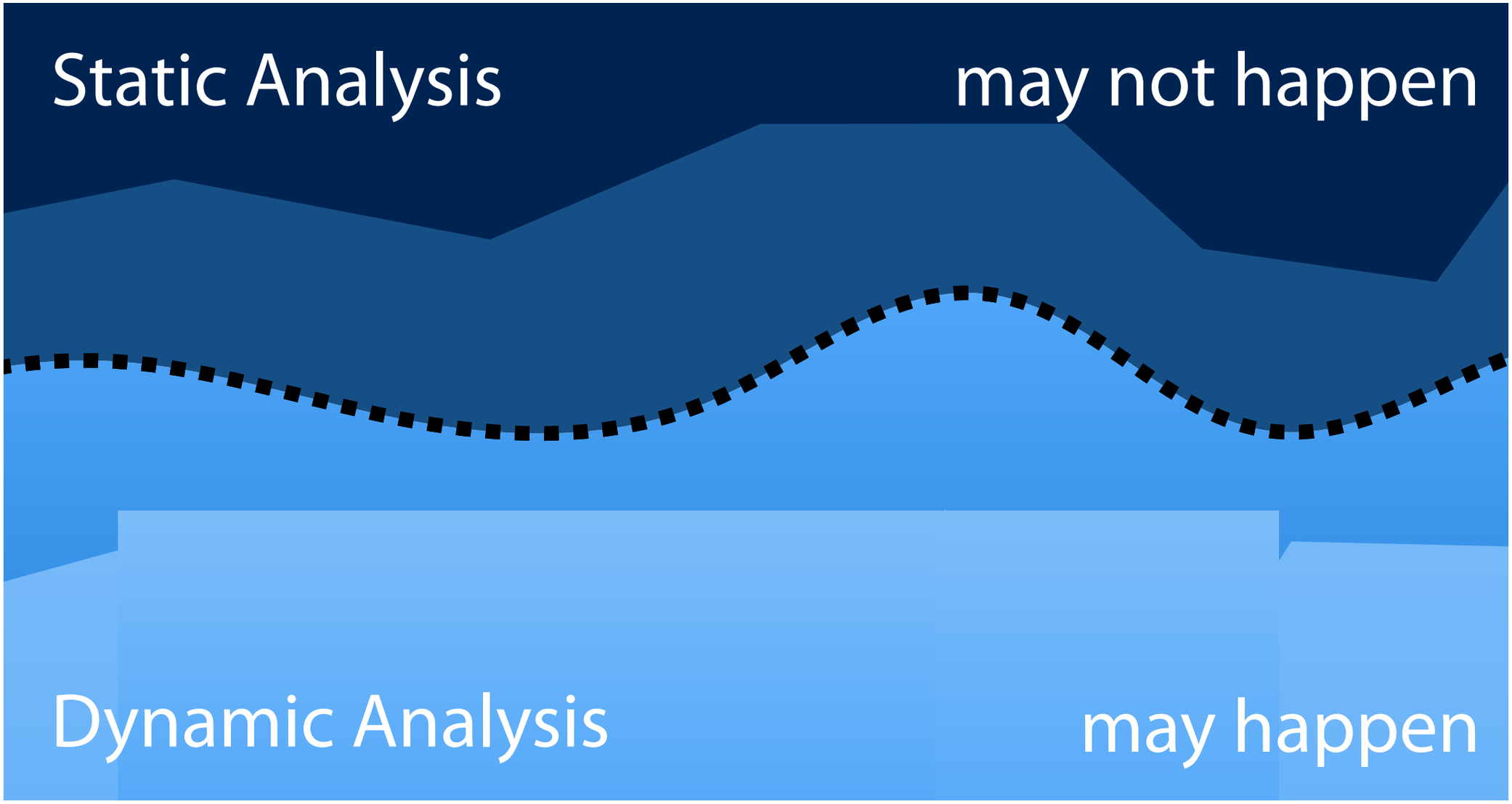
produces *guarantees from testing*

Jamrozik, Zeller: "Mining Sandboxes", ICSE 2016

Program Analysis

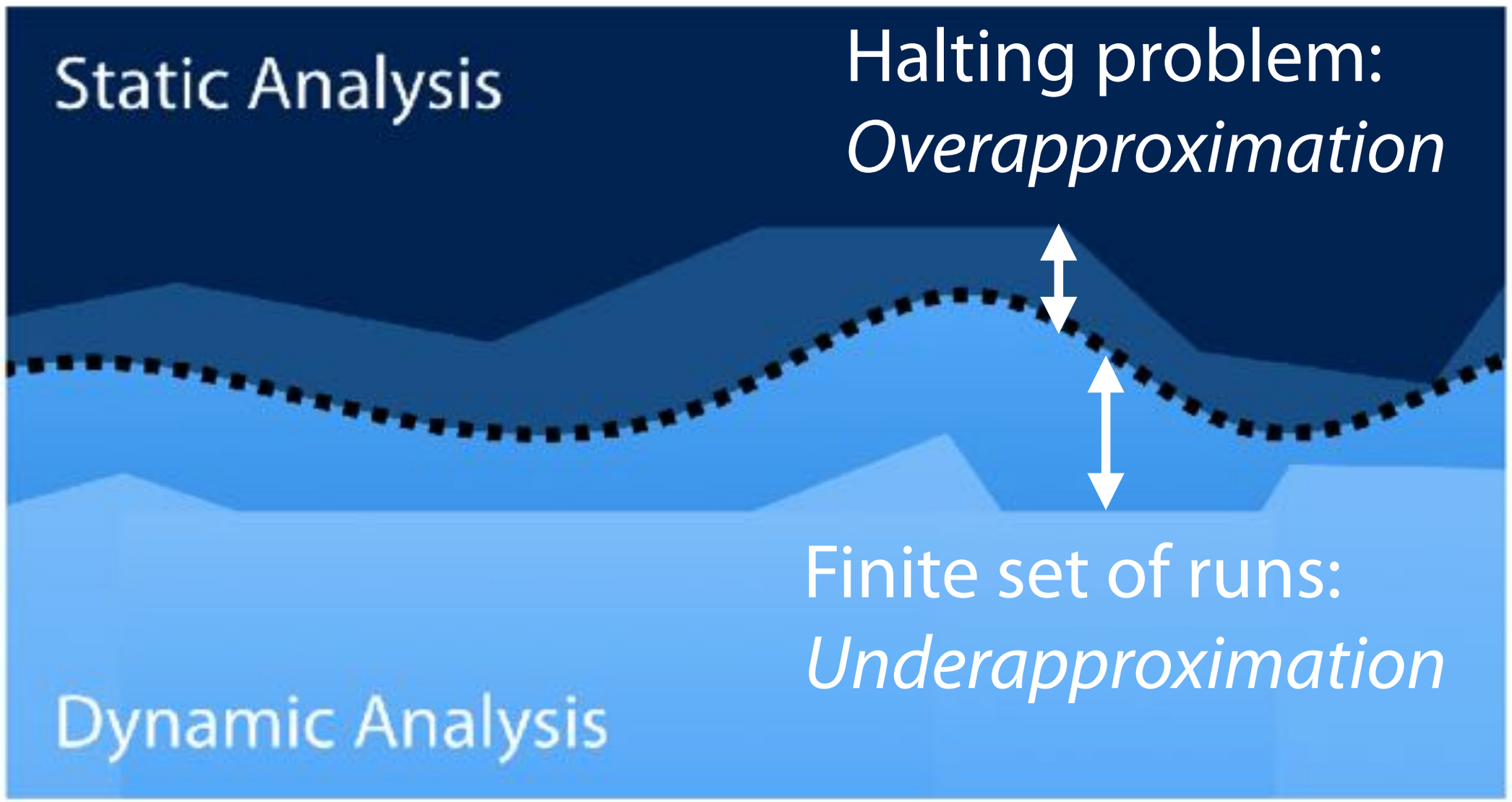


Program Analysis



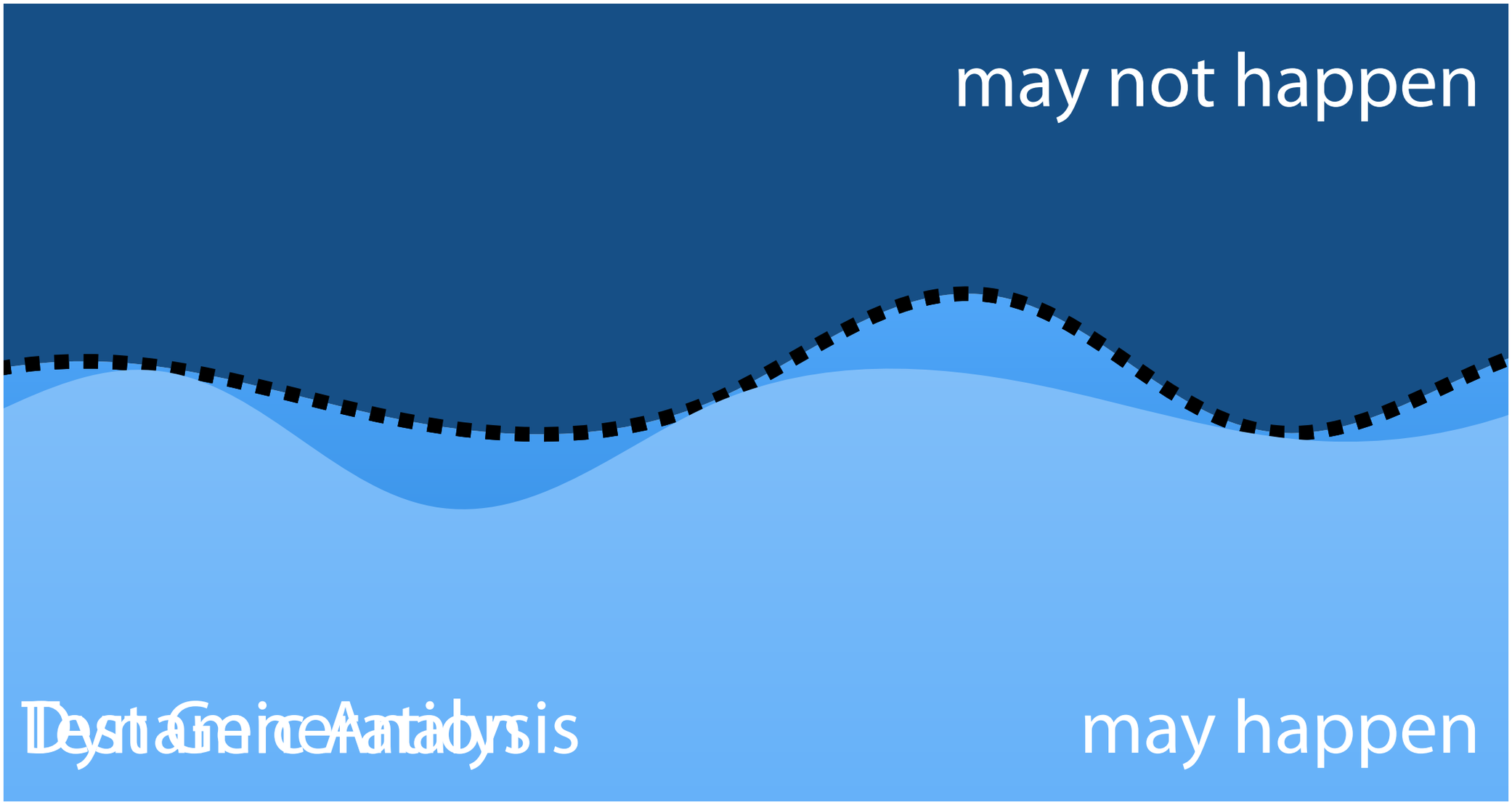
@AndreasZeller

Program Analysis



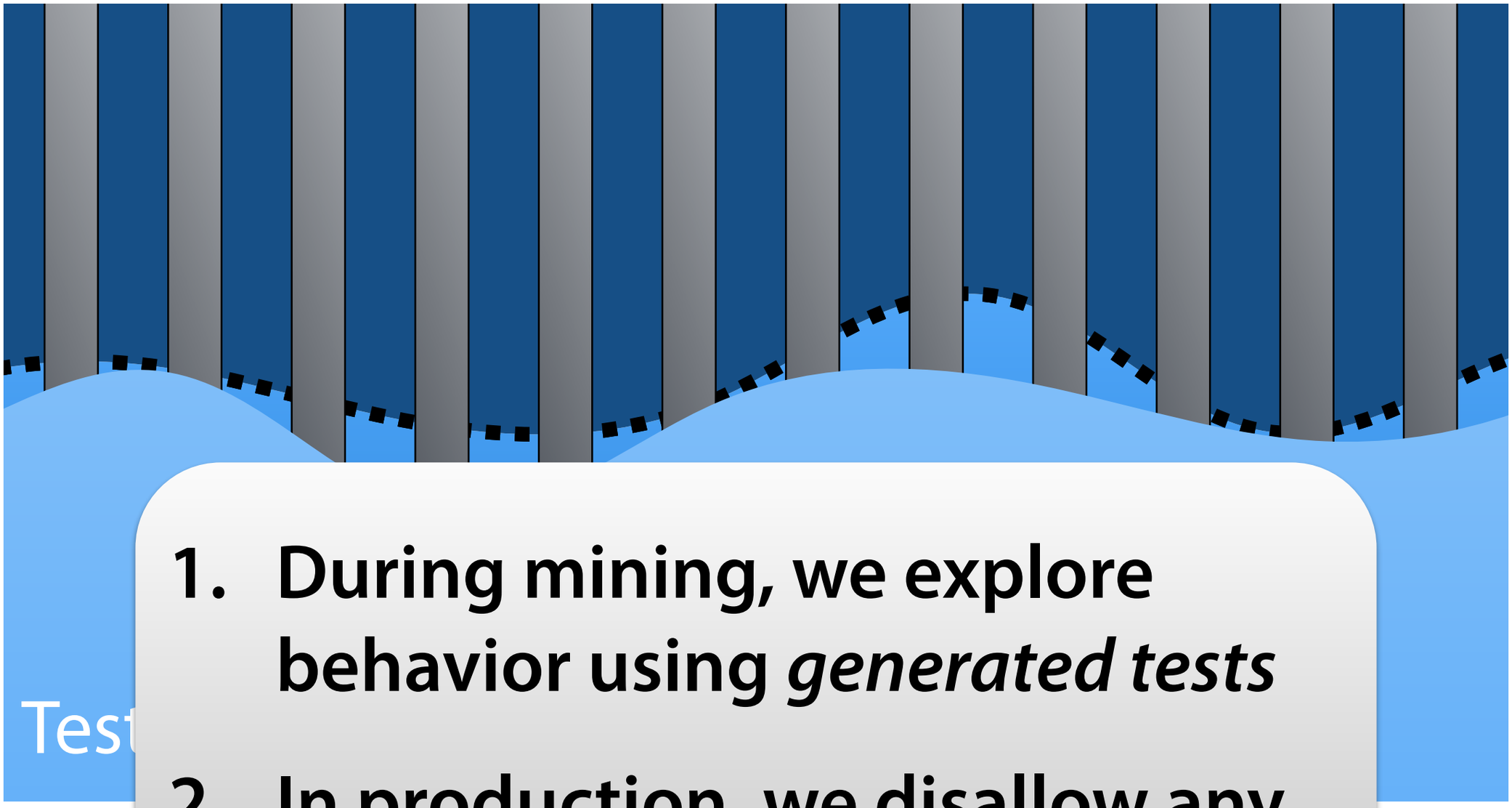
@AndreasZeller

Mining Behavior



@AndreasZeller

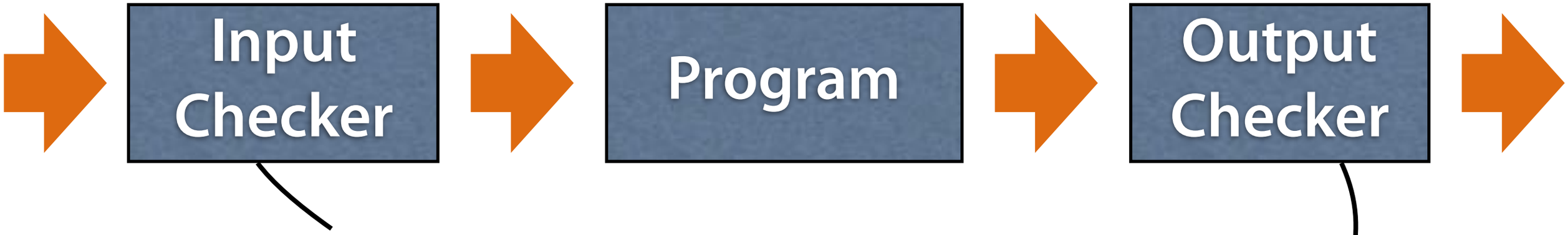
Test Complement Exclusion



1. During mining, we explore behavior using *generated tests*
2. In production, we disallow any behavior *not seen during testing*

@AndreasZeller

Guarantees from Testing




```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp' | ...
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= /[0-9]+/
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= /[a-z0-9=&]+/
REF ::= /[a-z]+/
```

```
REPLY ::= 'HTTP/1.1 ' CODE '\n' \
          HEADER+ '\n\n' DATA
CODE ::= '200 OK' | '404 Not Found'
HEADER ::= ...
DATA ::= ...
```

Checking Grammars

- Enforce behaviors seen during testing
- Effective protection against known and unknown attacks
- Challenge of *false alarms* can be addressed by grammar assessment + better testing

fully automatic • scalable • practical

Mining Input Grammars

Checking
Program Behavior

fully automatic • scalable • practical

Mining Input Grammars

Learning
Program
Behavior

Testing
Program
Behavior

Checking
Program
Behavior

fully automatic • scalable • practical

Modeling Behavior


```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp' | ...
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= /[0-9]+/
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= /[a-z0-9=&]+/
REF ::= /[a-z]+/
```

```
REPLY ::= 'HTTP/1.1 ' CODE '\n' \
          HEADER+ '\n\n' DATA
CODE ::= '200 OK' | '404 Not Found'
HEADER ::= ...
DATA ::= ...
```

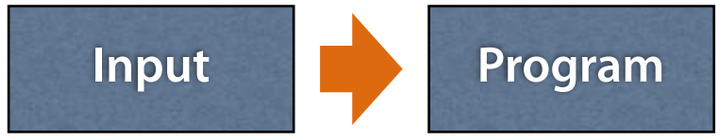
Mining Grammars

```
java.net.URL.set(protocol, host, port, authority, userinfo, path, query, ref)
    | http  user:password@www.google.com:80/command foo=bar&lorem=ipsum fragment
  param: protocol
    | http
  param: host
    |                    www.google.com
  param: port
    |                                   80
  param: authority
    |        user:password@www.google.com:80
  param: userinfo
    |        user:password
  param: path
    |                                  /command
  param: query
    |                                           foo=bar&lorem=ips
  param: ref
    |
```

M. Höschele

```
URL ::= PROTOCOL '://' AUTHORITY
AUTHORITY ::= USERINFO '@' HOST
```

Learning Behavior



0 ✔

- checks for digit
- checks for "true
- checks for ""
- checks for '['
- checks for '{'

A. Kampmann

Testing Behavior

```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp'
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= '80'
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= 'foo=bar&lorem=ipsum'
REF ::= /[a-z]+/
```

```
http://6F35:PkT5v@2.5/,,
http://.g:8
http://C.Ta.2./p.,//1.#14cq5
http://.37...g:776/,.,
http://.:07//,.8B,#eUN027
http://87.:2117//?&=&&38#207
http://S1t26c:7223i@.1..:16207
ftp://wb428:lr@00.8y.#5W7V9U2
ftp://012304:xt9Ut@k:285?250===K
```

N. Havrikov

Checking Behavior


```
URL ::= PROTOCOL '://' AUTHORITY PATH ['?' QUERY] ['#' REF]
AUTHORITY ::= [USERINFO '@'] HOST [':' PORT]
PROTOCOL ::= 'http' | 'ftp' | ...
USERINFO ::= /[a-z]+:[a-z]+/
HOST ::= /[a-z.]+/
PORT ::= /[0-9]+/
PATH ::= /\/[a-z0-9.\/]*/
QUERY ::= /[a-z0-9=&]+/
REF ::= /[a-z]+/
```

```
REPLY ::= 'HTTP/1.1 ' CODE
          HEADER+ '\n\n'
CODE ::= '200 OK' | '404
HEADER ::= ...
DATA ::= ...
```

K. Jamrozik

https://www.st.cs.uni-saarland.de/

erc Google