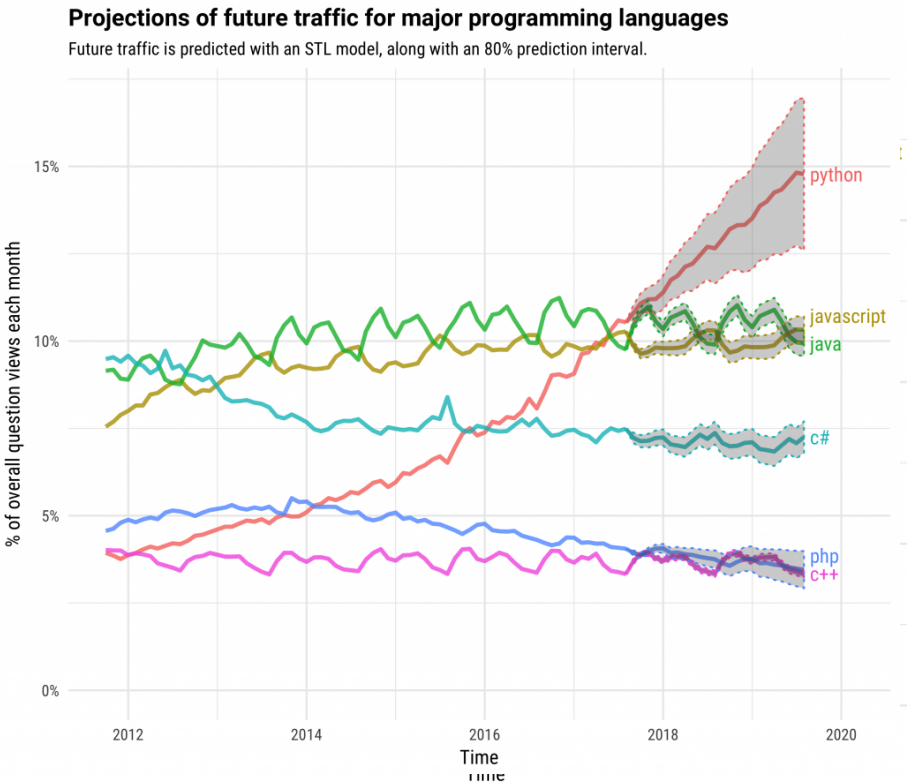
MaxSMT-Based Type Inference for Python 3

Mostafa Hassan, German University in Cairo
Caterina Urban
Marco Eilers
Peter Müller

ETH zürich

python™


Projections of future traffic for major programming languages
Future traffic is predicted with an STL model, along with an 80% prediction interval.
% of overall question views each month
15%
10%
5%
0%
2012
2014
2016
2018
2020
Time
python
javascript
java
c#
php
c++

Motivation

```
def get_percentage(db, ids, name)
    for id in ids:
        count, entry = db.get_entry(id)

        if name in entry.categories:
            return TOTAL_COUNT / count
    return -1
```

Type safety:

Static analysis:

Motivation

```
def get_percentage(db: Database,
                   ids: List[int],
                   name: str) -> float:
    for id in ids:
        count, entry = db.get_entry(id)

        if name in entry.categories:
            return TOTAL_COUNT / count
    return -1
```

PEP 484 -- Type Hints

PEP:	484
Title:	Type Hints
Author:	Guido van Rossum <guido at python.org>, Jukka Lehtosalo <jukka.lehtosalo at iki.fi>, Łukasz Langa <lukasz at python.org>
BDFL-Delegate:	Mark Shannon
Discussions-To:	Python-Dev <python-dev at python.org>

Type safety:

Static analysis:

Challenge

```
def get_val(o):
    return o.val
```

```
class A:
    def __init__(self):
        self.val = 15

class B:
    def __init__(self):
        self.val = “Hello”
```

Challenge

```
def get_val(o: A) -> int:
    return o.val
```

```
class A:
    def __init__(self):
        self.val = 15

class B:
    def __init__(self):
        self.val = “Hello”
```

Challenge

```
def get_val(o: A) -> int:
    return o.val
```

```
get_val(B())
```

```
class A:
    def __init__(self):
        self.val = 15

class B:
    def __init__(self):
        self.val = “Hello”
```

Challenge

```
def get_val(o: B) -> str:
    return o.val
```

```
get_val(B())
```

```
class A:
    def __init__(self):
        self.val = 15

class B:
    def __init__(self):
        self.val = “Hello”
```

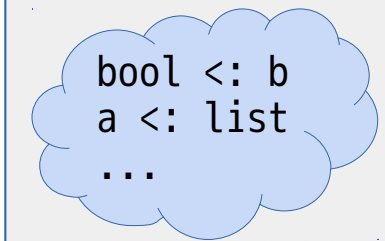
- No principal (most general) types
- Subtyping/polymorphism

Typpete

```
def main(a):
    b = a[0] == 0
    ...
```


```
def main(a: List[int])
        ->int:
    ...
```

```
Type error: argument
in line 3
```



```
type ptype = int | ...
const a: ptype
const b: ptype
subtype(bool, b)
...
```

```
b = bool
...
```

```
Sat. Model:
a = list(int), b = bool, ...
```

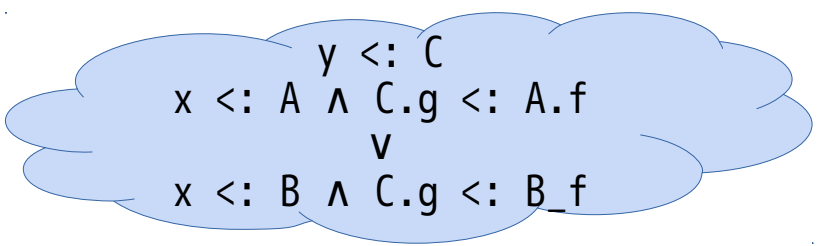
```
subtype(bool, b)
...
```

```
Unsat
```

SMT-Encoding

```
x.f = y.g
```

```
class A:
    def __init__(self, ...):
        self.f = ...
class B: ... # has field f
class C: ... # has field g
```




```
type ptype = int | str | A | ...
fun subtype (ptype, ptype) bool
const x: ptype
const y: ptype
const A_f: ptype
const B_f: ptype
const C_g: ptype
subtype(y, C)
subtype(x, A) ∧ subtype(C_g, A_f)
∨
subtype(x, B) ∧ subtype(C_g, B_f)
```

MaxSMT: Selecting Solutions

Problem: Arbitrary choice of solution

```
def get_true():
    return True
```

```
def get_true() -> float:
    return True
```

```
const get_true_return: ptype
subtype(bool, get_true_return)
```

MaxSMT: Selecting Solutions

Problem: Arbitrary choice of solution

```
def get_true():
    return True
```

```
def get_true() -> bool:
    return True
```

```
const get_true_return: ptype
subtype(bool, get_true_return)
```

```
get_true_return = bool
```

MaxSMT: Error Reporting

Problem: Error reporting

```
def incr_and_double(i):
    return 2 * i + 1

incr_and_double([3])
```

```
Unsat
```

```
const get_true_return: ptype
subtype(i, int)
subtype(int, incr_and_double_return)
...
```

```
...
```

MaxSMT: Error Reporting

Problem: Error reporting

```
def incr_and_double(i):
    return 2 * i + 1

incr_and_double([3])
```

```
def incr_and_double(i: int) -> int:
    return 2 * i + 1

incr_and_double([3])

# Error: argument subtype in line 4
```

```
const get_true_return: ptype
subtype(i, int)
subtype(int, incr_and_double_return)
...
```

Conclusion

- Evaluated on components of four open source projects, 400-770 LOC
  - Consistent speedup with MaxSMT
  - Constraint solving takes 3-15 seconds
- In the paper: Subtype encoding in SMT, generics
- Available open source
  - www.github.com/caterinaurban/Typpete

- Future work: Extend MaxSMT encoding to gradual typing

Try it out!

www.github.com/caterinaurban/Typpete