

Kompiliatorių testavimas

DOMANTAS JADENKUS, 3 GRUPĖ



Kompiliatorių bug'ai

Kompilatoriai – tai yra tiesiog programos.

Gali turėti bug'ų taip pat kaip ir visos kitos programos.

Kompiliatorių bug'ai paprastai yra reti.

Bug'ų rūšys

Lūžimas (ICE – internal compilation error).

Nepraleidžia teisingų programų.

Praleidžia neteisingas programas.

Blogai generuoja kodą (miscompilation) – teisingos programos sukompiliavus elgiasi ne taip kaip turėtų.

Kodėl bug'ai reti?

Daug programuotojų visame pasaulyje naudoja kompiliatorius.

Kompiliatoriaus bug'ai yra labai prioritizuojami – kompiliatorius kuris neveikia su paprastomis programomis yra bevertis.

Dauguma žmonių sprendžia normalias problemas rašydami normalų kodą.

Sprendžiama problema yra formali ir sprendimai yra gerai ištirti.

Daug testų.

Kada galima su jais susidurti?

Naudoti kompiliatorių kurio niekas nenaudoja.

Naudoti netestuotą kompiliatorių.

Rašyti keistą, neįprastą kodą.

Pavyzdys

```
1  #include <iostream>
2
3  void f(bool x = !(std::cout << "hi\n")) {}
4
5  int main() {
6      f();
7  }
```

Pavyzdys

```
$ g++ example.cpp -o example
```

```
$ ./example
```

```
hi
```

```
hi
```

Testavimo strategijos

Kaip atrodo tikrų kompiliatorių testai?

- Rust compiler
- Roslyn
- CompCert

Rust



Rust

`src/test/` yra apie 7500 programų, taip pat kai kurie kompiliatoriaus moduliai turi unit testus.

Iš programų įdomiausios:

- `src/test/compile-fail` (apie 2500 programų) – programos kurios turi nesusikompiliuoti. Tikrinama ar kompiliavimo klaidos yra tokios kokių tikimasi.
- `src/test/run-pass` (apie 3000 programų) – programos kurios turi sėkmingai susikompiliuoti, ir nenulūžti paleidus.

Rust – compile-fail testu pavyzdys

```
1  fn main () {  
2      loop {  
3          break 'a;  
4          //~^ ERROR E0426  
5          //~| NOTE undeclared label ``a`  
6      }  
7  }
```

Rust – run-pass testo pavyzdys

```
1  fn test<F>(f: F) -> usize where F: FnOnce(usize) -> usize {
2      return f(22);
3  }
4
5  pub fn main() {
6      let y = test(|x| 4 * x);
7      assert_eq!(y, 88);
8  }
```

Roslyn

Nér logo ☹️

Roslyn

.NET compiler platform („Roslyn“) – kompiliatoriai ir kodo analizavimo API C# ir VB.NET programavimo kalboms.

70000 testų!

Dauguma testų yra panašios formos:

- Iškviešti API ir duoti kažkokią paprastą programą.
- Patikrinti ar gauti rezultatai yra tokie kokių tikimasi.

CompCert

CompCert – ANSI C kompiliatorius.

Skirtas naudoti embedded sistemoms kurioms reikia patikimumo.

Gali kompiliuoti kodą PowerPC, ARM, x86 procesoriams.

CompCert

test/ kataloge radau 117 testų:

- 24 mažos programos.
- 3 didesnės programos.
- 90 regression testų.

CompCert

CompCert yra parašytas su Coq.

Coq – programavimo kalba leidžianti kartu su programa pateikti lemas ir jų įrodymus.

Vietoj testavimo, kompiliatoriaus korektiškumas yra tiesiog įrodytas.

Java ir Scala „bugas“

Surastas prieš metus.

Parodė, kad Java ir Scala tipų sistemos yra unsound.

```
1  class Unsound {
2      static class Constrain<A, B extends A> {}
3      static class Bind<A> {
4          <B extends A>
5          A upcast(Constrain<A,B> constrain, B b) {
6              return b;
7          }
8      }
9      static <T,U> U coerce(T t) {
10         Constrain<U,? super T> constrain = null;
11         Bind<U> bind = new Bind<U>();
12         return bind.upcast(constrain, t);
13     }
14     public static void main(String[] args) {
15         String zero = Unsound.<Integer,String>coerce(0);
16     }
17 }
```

Klausimai?