Improving Trace-Based JIT Optimisation Using Whole-Program Information
Vandercammen, Maarten; De Roover, Coen

Publication date:
2016

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Improving Trace-Based JIT Optimisation using Whole-Program Information

Maarten Vandercammen, Coen De Roover

Software Languages Lab, Vrije Universiteit Brussel, Belgium

---

### Program

```python
function f(a) {
    var b = a > 0 ? 1 : 2;
    return a + b;
}
```

```python
var n = 100;
var result = 0;
var a = user_input();
function loop() {
    while (n > 1) {
        result += f(a);
        n -= 1;
    }
}
loop();
a = 10;
loop();
a = 18;
loop();
```

---

### Execution Trace

```python
loop:
LOAD n
LOAD_CONST 1
CMP_GREATER
GUARD_TRUE
LOAD a
LOAD_CONST 0
CMP_GREATER
GUARD_TRUE
LOAD result
ADD_INT
LOAD n
DEC
JUMP loop
```

---

### Initial Analysis

- Function `f(a)`
- Variable `b` and `c`
- Constant `1` and `2`

---

### Run-time Analysis

1. Analysis Launch Point?
2. Extent of Scope?

---

### Future Work

1. Analysis Launch Point?
2. Extent of Scope?