

Konstrukcija kompilatora - Septembar 2018.

1. Unaprediti već postojeći interpreter koji podržava cele neoznačene brojeve u kompilator sa mogućnošću prevođenja na LLVM.

- (a) (već urađeno) Interpreter ima mogućnost štampanja i računanja vrednosti celobrojnih izraza. Nad brojevima su podržani bitovski operatori i (*and*), ili (*or*), ekskluzivno ili (*xor*), negacija (*not*) i levo i desno siftovanje (*shl* i *shr*), sa prioriteto i asocijativnošću kao u programskom jeziku C. Brojevi se zadaju u dekadnom ili heksadekadnom sistemu.

```
main() {
    # stampanje dekadno
    print 4 or 1226 and 0xf;
    print not 23 and 0xff;
    print 16 shl 4 shr 1;
    print 16 shl (4 shr 1)
}
```

- (b) (već urađeno) Interpreter ima mogućnost memorisanja vrednosti izraza u promenljivu čije se ime sastoji samo od malih slova.

```
main() {
    # dodela
    set tmp to 127 shl 2;
    print tmp or 1024 shl 1
}
```

- (c) (već urađeno) Rezultat izraza je moguće ispisati i u heksadekasnom sistemu, u zavisnosti od promenljive *flag*. Ako promenljiva nije definisana, ili ako joj je vrednost 0, ispis je dekadni, inače je heksadekadni.

```
main() {
    # stampanje heksadekadno
    set flag to not 0;
    print tmp or 1024 shl 1
}
```

- (d) Dodati mogućnost da program sadrži još neku funkciju, osim *main()*, pri čemu se pretpostavlja da sve one vraćaju celobrojnu vrednost (vrednost poslednjeg izraza), a broj argumenta može biti 0 ili više.

```
f(x, y, z) {
    set a to x or y;
    a or z
}
one() {
    1
}

main() {
    set tmp to call f(8 or 2, 4, 1);
    print tmp;
    set tmp to call one();
    print tmp
}
```

- (e) Dodati petlje.

```
rotate_left(x, n) {
    set first_bit_mask to 1 shl 31;
    set i to 0;
    while (i < n) {
        set first_bit to x and first_bit_mask;
        set x to x shl 1 or first_bit shr 31;
        set i to i + 1
    };
    x
}

main() {
    set flag to not 0;
    set tmp to 10;
    print tmp;
    set tmp to call rotate_left(tmp, 4);
    print tmp
}
```