

# **Samodejno računanje dvosmernih preslikav**

PROJEKT PRI PREDMETU MATEMATIKA S FUNKCIJSKIM  
PROGRAMIRANJEM

Urška Pangerc  
Melanija Vezočnik

Ljubljana, januar 2015

# DVOSMERNNA PRESLIKAVA

## Funkcija *get*

- Vhodni argument:
  - izvor
- Izhodni argument:
  - slika

## Funkcija *put*

- Vhodna argumenta:
  - izvor
  - posodobljena slika
- Izhodni argument:
  - posodobljeni izvor

## FUNKCIJA *GET*:

```
polovicka :: [a] -> [a]
```

```
polovicka seznam = take (length `div` 2) seznam
```

```
let s = [1, 2, 3, 4, 5, 6] ← izvorišče
```

```
let v = polovicka s ← slika ([1, 2, 3])
```

## FUNKCIJA *PUT*:

```
put_1 :: [a] -> [a] -> [a]
```

```
put_1 x x' | length x' == n
```

```
    = x' ++ drop n x
```

```
    where n = length x `div` 2
```

```
let v' = 7 : (drop 1 v) ← spremenimo sliko ([7, 2, 3])
```

```
let s' = put_1 s v'
```

```
    ← posodobljeni izvorišče ([7, 2, 3, 4, 5, 6])
```

**Ali je mogoče  
postopek  
avtomatizirati?**



**Ali je mogoče  
postopek  
avtomatizirati?**

**DA**



# IDEJA

V Haskell-u napišimo funkcijo višjega reda, ki kot argument vzame polimorfno funkcijo *get* in vrne ustrezno funkcijo *put*.



# BFF - kriterij

➤ Za funkciji *put* in *get* morata veljati:

1. Zakon **GetPut**

$$\mathit{put} \text{ izvor } (\mathit{get} \text{ izvor}) \equiv \text{izvor}$$

2. Zakon **PutGet**

$$\mathit{get} (\mathit{put} \text{ izvor slika}) \equiv \text{slika}$$

# BFF

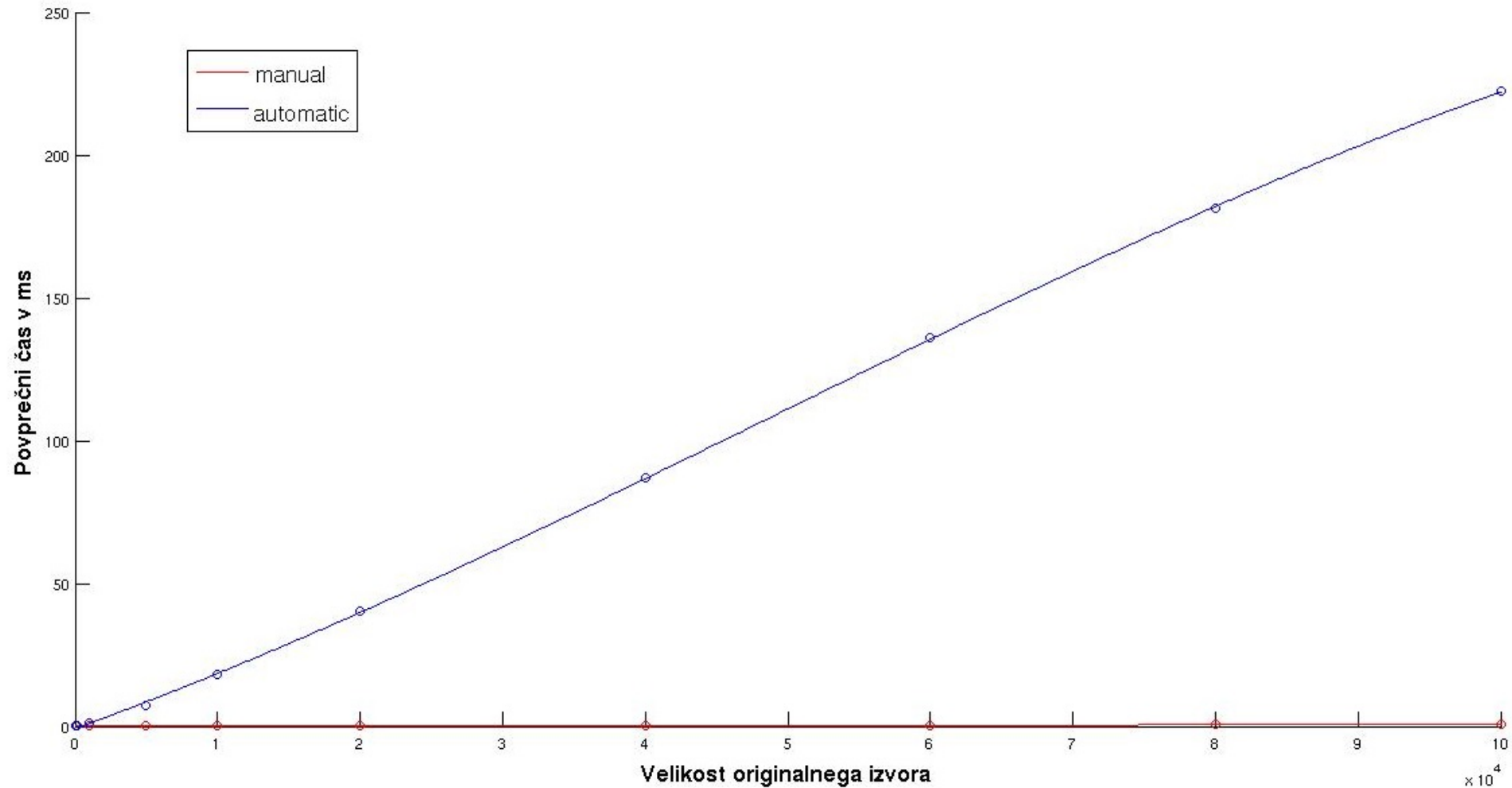
➤ Vhodni argumenti: funkcija *get*, izvor *s*, slika *v*

➤ KORAKI:

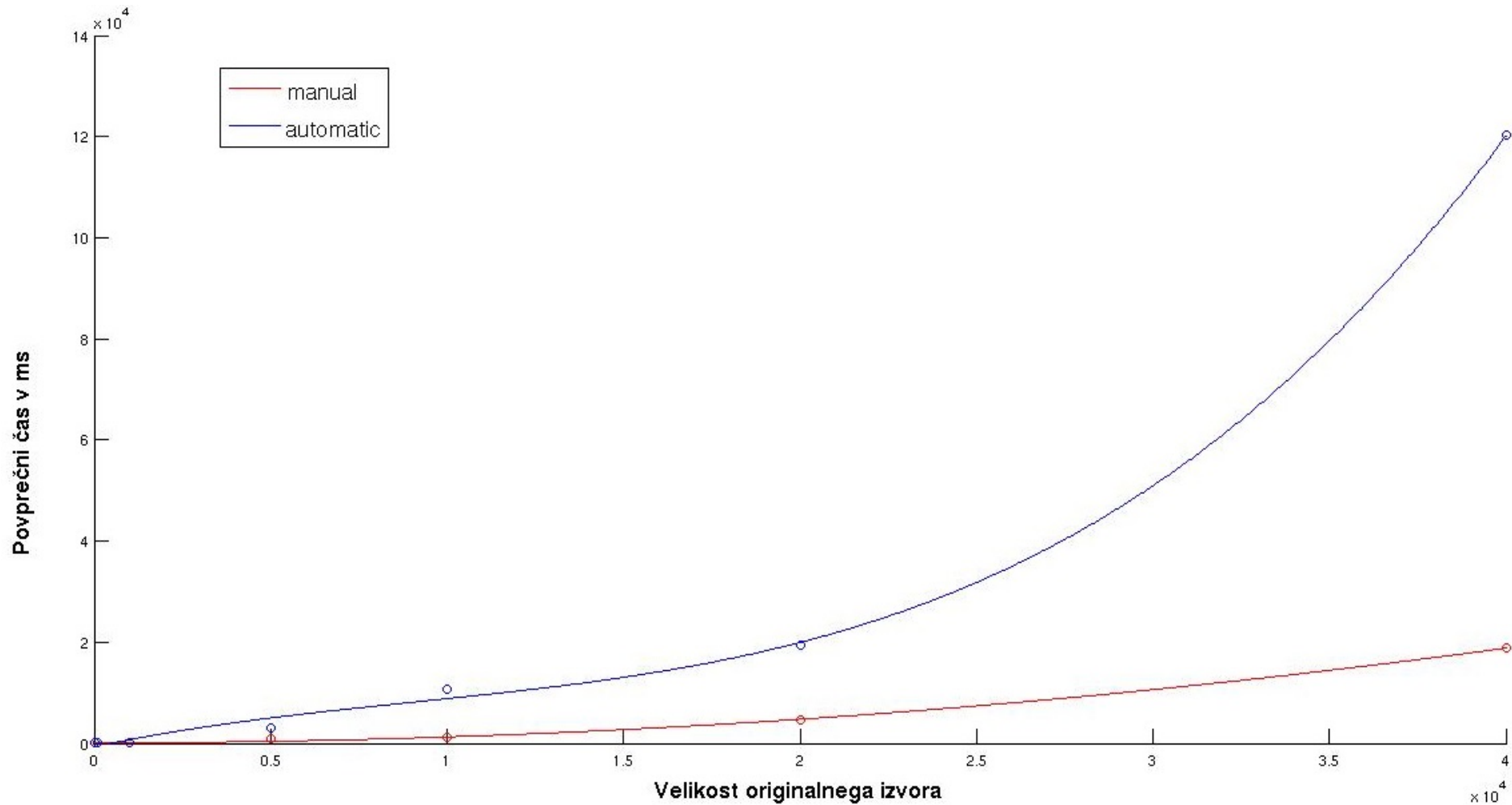
1. iz izvora *s* naredi predlogo  $s' = [0..n]$ , če  $n + 1 = \text{length}(s)$ , in asociacijo *g*, med ustreznimi vrednostmi iz *s'* in *s*.
2. zažene *get* na *s'*, dobi predlogo slike *v'* in sproducira asociacijo *h* med *v'* in *v*.
3. združi asociaciji *g* in *h* v *h'*, kjer ima *h* prednost, ko je index predlog najden v *h* in *g*.
4. na koncu naredi posodobljen izvor z zapolnitvijo vseh pozicij na  $[0..n]$  z ustreznimi vrednostmi glede na *h'*



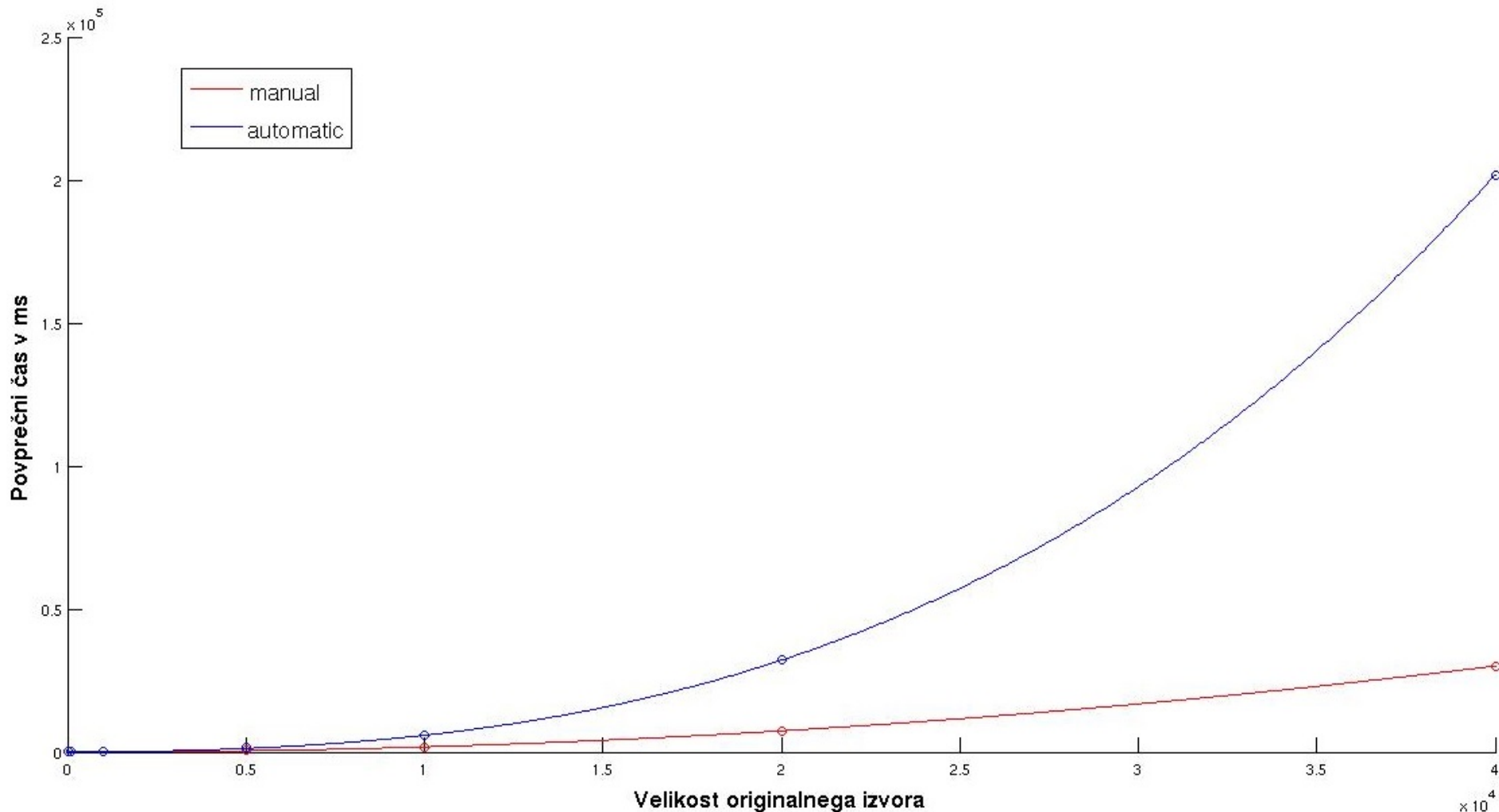
# PRIMERJAVA – put\_1 vs. Bff polovicka



# PRIMERJAVA – put\_3 vs. Bff zbrisiDvojnike (vsi elementi različni)



# PRIMERJAVA – put\_3 vs. Bff zbrisiDvojnike (vsak element ponovljen dvakrat)





Hvala za vašo pozornost.