

Laiska evaluointi

Atte Keinänen / Informaatioverkostot
SCI3027.kand Tietotekniikan kandiseminaari
12.12.2017

Ohjaaja: Juha Sorva



Laiska evaluointi on evaluointisemantiikka

(synonyymeja: evaluointistrategia, suoritusstrategia)

Mitä ”evaluointisemantiikka” tarkoittaa?

- Määrittää säännöt ohjelmointikielen funktiokutsujen eri vaiheiden evaluoinnille eli lausekkeiden arvojen laskemiselle
- Kaikkein kiinnostavin kysymys, johon evaluointistrategia voi vastata: **Milloin ja miten funktiot parametrilausekkeet evaluoidaan?**

Mitä "evaluointisemantiikka" tarkoittaa?

- Määrittää säännöt ohjelmointikielen funktiokutsujen eri vaiheiden evaluoinnille eli lausekkeiden arvojen laskemiselle
- Kaikkein kiinnostavin kysymys, johon evaluointistrategia voi vastata: **Milloin ja miten funktiot parametrilausekkeet evaluoidaan?**

```
def printValuePlusOne(parameter):  
    parameter = parameter + 1  
    print "Value is: " + parameter
```

```
printValuePlusOne(335) # Literaali parametrina  
printValuePlusOne(25 * 12) # Aritmeettinen lauseke parametrina  
printValuePlusOne(someDefinedVariable) # Muuttujalauseke parametrina
```

Vertailu: Ahne ja laiska evaluointi

Ahne evaluointi

- Käytössä käytännössä kaikissa suosituimmissa ohjelmointikielissä
- **Milloin parametrilausekkeet evaluoidaan?** Heti, kun funktiota kutsutaan, ennen kuin funktion ohjelmakoodin suoritus alkaa

Laiska evaluointi

- Käytössä varsin harvoissa ohjelmointikielissä, joista tunnetuin esimerkki on Haskell
- **Milloin parametrilausekkeet evaluoidaan?** Vasta sitten, kun funktion ohjelmakoodi tarvitsee parametrin arvoa.

Videoitu demo

github.com/attekei/kandi/raw/master/demo.mp4

Kandityöni sisältö

Tutkimuskysymyksiä

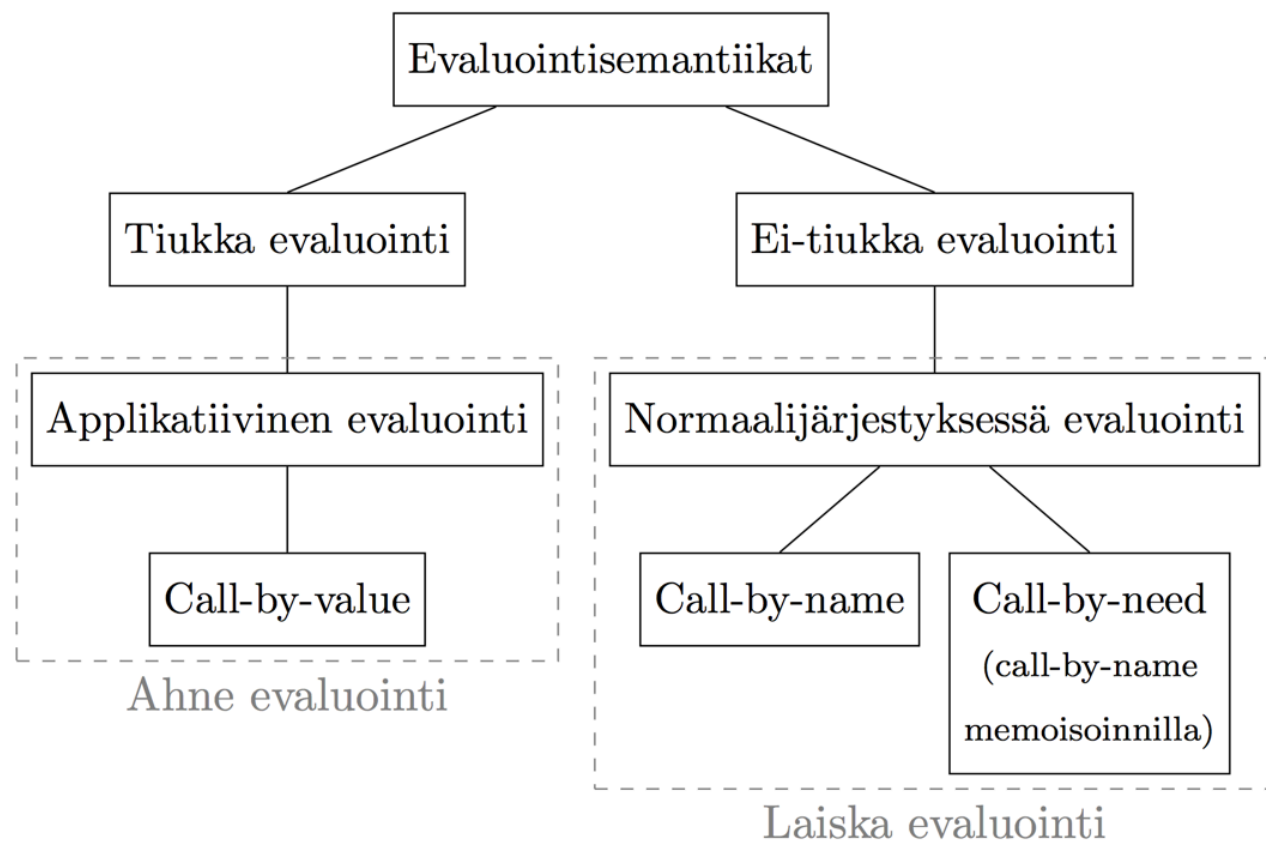
1. Millaista käsitteistöä laiskan evaluoinnin yhteydessä käytetään?
2. Millaisia sovellutuksia laiskalle evaluoinnille on moderneissa ohjelmointikielissä?
3. Millaisia etuja ja haittoja laiskaan evaluointiin liittyy?

Millaista käsitteistöä laiskan evaluoinnin yhteydessä käytetään?

Mitä jos ei arvoa alilausekkeella?

Milloin parametrit evaluoidaan?

Miten parametrit välitetään koodille?



Millaisia sovellutuksia laiskalle evaluoinnille on moderneissa ohjelmointikielissä?

Sovellutus	Evaluointisemantiikka	Toteutukset	
		Ohjelmointikielissä	Apukirjastoissa
Laiskat sekvenssit	Applikaatiivinen evaluointijärjestys	Clojure, F#, Haskell, Python, Ruby, Scala	Immutable.js, Lazy.js
Call-by-name -muuttujat	Call-by-name	Scala (funktion parametrina)	-
Call-by-need -muuttujat	Call-by-need	Scala (luokan instanssimuuttujana)	-
Laiskat futuurit/lupaukset	Call-by-need	C++11	Fluture (JavaScript-kirjasto)
Käyttäjän laatimat kontrollirakenteet	Applikaatiivinen evaluointi tai kääntämisvaiheen makrot	Clojure (makrot), Haskell, Scala (makrot, kokeellinen tuki)	-
Generaattorit	Funktion suorituksen keskeyttäminen (vrt. applikaatiivinen evaluointi)	JavaScript, Python, Ruby	-

Millaisia etuja ja haittoja laiskaan evaluointiin liittyy?

- Haskell-ohjelmointikielellä laaditut ohjelmat voivat käyttää arvaamattomasti muistia, ja evaluoimattomat lausekkeet voivat aiheuttaa muistivuotoja
- Laiskasti evaluoiduissa ohjelmointikielissä myös ajonaikaisten ongelmien selvittäminen on hankalaa, koska funktiokutsupinojen seuraaminen on vaikeaa
- Vaikuttaa soveltuvan parhaiten selkeästi rajattujen ongelmien ratkaisemiseen, minkä esimerkiksi laiskat sekvenssit mahdollistavat

Kiitos!

Lataa kandini täältä:
github.com/attekei/kandi