

7. syyskuuta 2013

Jani Viherväs
jani.p.vihervas@student.jyu.fi

HARJOITUSTYÖ: CALCULATOR

TIEA241 AUTOMAATIT JA KIELIOPIT

Johdanto

Harjoitustyöni aiheena oli laskinohjelma, jolla pystyy laskemaan aritmeettisia lausekkeita. Kurssin luennoilla läpikäytyyn ohjelmaan [1] verrattuna omani laajentaa ohjelman toiminnallisuuksia mahdollistamalla desimaalilukujen, potenssi-, juuri- ja logaritmilaskujen sekä alustettavien muuttujien käytön. Lisäksi ohjelmalla voidaan esittää desimaaliluvut murtolukumuodossa sekä kielioppivirheen kohdalla käyttäjälle kerrotaan virheellisen merkin tarkka paikka kuvaavan virheviestin kera. Tarkemmin tietoa ohjelman käytöstä ja ominaisuuksista kerrotaan kappaleessa 2 sekä kappaleessa 1 käsitellään ohjelman käyttämiä automaatteja sekä kielioppeja.

1 Ohjelman toteuttamat automaattit

Ohjelman toiminta perustuu kahdella tasolla toimiviin automaatteihin siten, että ylemmän tason automaatti valitsee, mitä alemman tason ennustavaa jäsennysautomaattia tai funktiota käytetään. Ylätason automaatti `InputHandler`-luokka on normaali deterministinen äärellinen automaatti, jonka hyväksyvä tila saavutetaan `quit`-komennolla. Komennoilla `set`, `unset`, `frac`, `help` ja `variables` siirrytään niitä vastaaviin automaatteihin tai funktioihin ja komennolla `calc` sekä millä tahansa muulla kuin edellä mainituilla komennoilla siirrytään `Calculate`-automaattiin. Alemman tason automaattit toimivat täten ylemmän tason automaatin tilasiirtyminä, joista palataan aina automaatin aloitustilaan.

Alemman tason automaattit tunnistavat kullekin ominaisen kontekstitto-
man kieliopin, joita on esitelty tarkemmin kappaleessa 3. Yksinkertaisuudensa takia nimitän `Variables`-luokkaa funktioksi, sillä sen sisällä ei ole mitään syötteistä riippuvaista toiminnallisuutta. Ylemmän tason automaatin toteutustavasta johtuen ohjelma ratkeaa kaikilla syötteillä, koska jos `InputHandler`-luokalle ei anneta sen tunnistamaa syötettä, käskytetään `Calculate`-luokkaa, mikä pysähtyy heti, jos syötteessä on kielioppivirhe. Tyhjän syötteen kohdatessaan `Calculate` palauttaa tuloksen 0.

2 Ohjeita ohjelman käyttöön

2.1 Ohjelman kääntäminen ja ajaminen komentoriviltä

Ohjelma kääntyy Windows-tietokoneella komentoriviltä komennolla

```
csc Calculate.cs Calculator.cs CalculatorException.cs  
    CommandBase.cs ErrorHandler.cs Fraction.cs Help.cs  
InputHandler.cs SetVariable.cs StringParse.cs UnsetVariable.cs  
    Variable.cs Variables.cs.
```

Jos haluaa ajaa NUnit-yksikkötestejä, lisätään yllä olevan komennon perään

```
CalculateTests.cs SetVariableTests.cs StringParseTests.cs  
/r:nunit.framework.dll,
```

missä *nunit.framework.dll* on NUnit-testausjärjestelmän kirjastotiedosto. NUnitin voi asentaa omalle koneelle lataussivuilta [2] saatavalla asennustiedostolla.

Ohjelman voi tämän jälkeen käynnistää komennolla

```
Calculator.exe.
```

2.2 Ohjelman käyttö ja ominaisuudet

Käynnistyksen yhteydessä ohjelma neuvoo käyttäjää kertomalla saavansa lisäapua komentojen käyttöön `help`-komennolla sekä pystyvänsä sammuttamaan ohjelman `quit`-komennolla.

Pääkomennot:

- `calc`: Laskutoimituksen laskeminen, komentoa ei ole pakko kirjoittaa laskutoimituksen eteen.
- `set`: Muuttujan alustaminen.
- `unset`: Muuttujan poistaminen.
- `variables`: Tallennettujen muuttujien tulostaminen.
- `frac`: Laskutoimituksen tuloksen tulostaminen murtolukuna. Komentolle voidaan antaa parametriksi yksittäinen luku, muuttuja tai laskutoimitus. Suurin tarkkuus, millä reaaliluku voidaan muuttaa murtoluvuksi on $\frac{1}{10\,000}$.

- `help [command]`: Ilman parametria ja tunnistamattomalla syötteellä komento tulostaa yleisavustuksen kaikista komennoista ja paramet-
rin kanssa se tulostaa yksityiskohtaisemman avustuksen. Esimerkiksi
`help set` tulostaa tarkemman avustuksen liittyen muuttujien alusta-
miseen.
- `quit`: Ohjelman sammuttaminen.

Ohjelmalla pystyy laskemaan aritmeettisiä laskutoimituksia reaali- ja ko-
konaisluvulla seuraavilla operaatioilla:

- `+`, `-`: Yhteen- ja vähennyslaskut.
- `*`, `/`: Kerto- ja jakolaskut.
- `^`, `root / root[n]`: Potenssiin korotukset ja juurilaskut ¹. Juurilas-
kuissa voidaan myös antaa parametrina asteluku n , minkä oletuksena
käytetään $n = 2$ (neliöjuuri).
- `log / log[n]`: Logaritmilaskut, parametrina voidaan antaa kanta-
luku n , minkä oletuksena käytetään Neperin lukua e (luonnollinen
logaritmi).
- `(,)`: Sulutukset.

Valmiin n :n juuren laskumetodin puuttuessa `.Net`:n kirjastoista, käytin
seuraavaa muunnoskaavaa [3]:

$$\sqrt[n]{a} = a^{\frac{1}{n}}.$$

Laskutoimituksissa voidaan myös käyttää `set`-komennolla alustettuja muut-
tujia ja niiden arvoja. Ohjelmaan on tallennettu kolme vakiomuuttujaa,
joita ei voi ylikirjoittaa:

1. `pi`: $\pi \approx 3,14159\,26535\,9$,
2. `e`: Neperin luku $e \approx 2,71828\,18284\,59045$ ja

¹Laskutoimitusten yksinkertaistamiseksi on oletettu, että $\sqrt[n]{x} > 0$. Imaginäärilukujen
laskutoimituksia ei tueta, jolloin jos $x < 0$, ohjelma tulostaa virheilmoituksen.

3. ans: Edellisen laskutoimituksen tulos. Ennen ensimmäistä laskutoimitusta ans on alustettu nolllaksi.

Esimerkki laskutoimituksesta:

```
set x = 1/3
x = 0,3333333333333333

set x = x - 1/6
x = 0,1666666666666667

x * root[3] 8 + log e
1,3333333333333333

frac ans
1 1/3
```

3 Alemman tason automaatit

3.1 Calculate: calc

Laajensin luennoilla esitettyä [4] aritmeettisten lausekkeiden kielioppia lisäämällä potenssi-, juuri- sekä logaritmioperaatiot sekä muuttujat kielioppiin:

$$E \rightarrow c \mid v \mid -E \mid E + E \mid E - E \mid E * E \mid E / E \mid E^E \mid \sqrt[E]{E} \mid \log_E E \mid (E),$$

missä v (*variable*) tarkoittaa muuttujan nimeä.

Operaattoreiden luokkiin jaossa käytin jälkiprujussa esitettyä [5] luokkajakoja ja laajensin sen kattamaan yllä mainitut operaattorit:

Lausekkeet (*expressions*) Yhteen- ja vähennyslaskut.

Termit (*terms*) Kerto- ja jakolaskut.

Eksponentiaalit (*powers*) Potenssiin korotukset.

Tekijät (*factors*) Negaatio, juuri- ja logaritmilaskut.

Sijoitin juuri- ja logaritmilaskut tekijäluokkaan, sillä \sqrt{x} ja $\log x$ ovat prefiksioperaattoreita. Noudattamalla luentojen menetelmää [6] saadaan automaatin käyttämä kontekstiton kielioppi muutettua yksiselitteiseksi:

$$\begin{aligned} E &\rightarrow T \mid E + T \mid E - T \\ T &\rightarrow P \mid T * P \mid T / P \\ P &\rightarrow F \mid P^F \\ F &\rightarrow C \mid -F \mid \sqrt[F]{F'} \mid \log_F F' \\ F' &\rightarrow \varepsilon \mid [C] \\ C &\rightarrow c \mid v \mid (E) \end{aligned}$$

Muutetaan eksponentiaalioperaattorit tietokoneella kirjoitettavaan muotoon:

$$\begin{aligned} E &\rightarrow T \mid E + T \mid E - T \\ T &\rightarrow P \mid T * P \mid T / P \\ P &\rightarrow F \mid P \wedge F \\ F &\rightarrow C \mid -F \mid \text{root } F' F \mid \log F' F \\ F' &\rightarrow \varepsilon \mid [C] \\ C &\rightarrow c \mid v \mid (E), \end{aligned}$$

Juuri- ja logaritmilaskuissa hakasulkeissa olevat arvot toimivat myös tyhjinä. Tällöin käytetään oletuksena juurilaskuissa arvoa 2 (neliöjuuri) ja logaritmilaskuissa käytetään luonnollista logaritmia (kantelukuna Neperin luku $e \approx 2,71828\,18284\,59045$). Lisätietoa laskuoperaatioista on kappaleessa 2 sekä käskyttämällä ohjelmalle `help calc`.

Poistetaan välitön vasen rekursio:

$$\begin{aligned}
E &\rightarrow TE' \\
E' &\rightarrow \varepsilon \mid +TE' \mid -TE' \\
T &\rightarrow PT' \\
T' &\rightarrow \varepsilon \mid *PT' \mid /PT' \\
P &\rightarrow FP' \\
P' &\rightarrow \varepsilon \mid ^FP' \\
F &\rightarrow C \mid -F \mid \text{root } F'F \mid \log F'F \\
F' &\rightarrow \varepsilon \mid [C] \\
C &\rightarrow c \mid v \mid (E)
\end{aligned}$$

NULLABLE, FIRST ja FOLLOW²:

	NULLABLE	FIRST	FOLLOW
E		$-, \text{root}, \log, c, v, ($	$), \dagger$
E'	*	$+, -$	$), \dagger$
T		$-, \text{root}, \log, c, v, ($	$+, -,), \dagger$
T'	*	$*, /$	$+, -,), \dagger$
P		$-, \text{root}, \log, c, v, ($	$*, /, +, -,), \dagger$
P'	*	$^$	$*, /, +, -,), \dagger$
F		$-, \text{root}, \log, c, v, ($	$^, *, /, +, -,), \dagger$
F'	*	$[$	$\text{root}, \log, c, v, (, ^, *, /, +, -,), \dagger$
C		$c, v, ($	$], ^, *, /, +, -,), \dagger$

² \dagger -merkkinä käytetään myös ;-merkkiä sallimaan SetVariable-automaatin käyttämän useamman muuttujan alustuksen. En kuitenkaan kirjannut ;-merkkiä kielioppiin, sillä se ei varsinaisesti kuulu Calculate-automaatin kielioppiin.

Ennustava jäsennostaulukko:

Primäärioperaattorit:

	c	v	$[$	$]$	$($	$)$	\dagger
E	$E \rightarrow TE'$	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'						$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow PT'$	$T \rightarrow PT'$			$T \rightarrow PT'$		
T'						$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
P	$P \rightarrow FP'$	$P \rightarrow FP'$			$P \rightarrow FP'$		
P'						$P' \rightarrow \varepsilon$	$P' \rightarrow \varepsilon$
F	$F \rightarrow C$	$F \rightarrow C$			$F \rightarrow C$		
F'	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow [C]$		$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$
C	$C \rightarrow c$	$C \rightarrow v$			$C \rightarrow (E)$		

Laskuoperaattorit:

	$+$	$-$	$*$	$/$	\wedge	root	log
E		$E \rightarrow TE'$				$E \rightarrow TE'$	$E \rightarrow TE'$
E'	$E \rightarrow +TE'$	$E \rightarrow -TE'$					
T		$T \rightarrow PT'$				$T \rightarrow PT'$	$T \rightarrow PT'$
T'	$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$	$T \rightarrow *PT'$	$T \rightarrow /PT'$			
P		$P \rightarrow FP'$				$P \rightarrow FP'$	$P \rightarrow FP'$
P'	$P' \rightarrow \varepsilon$	$P' \rightarrow \varepsilon$	$P' \rightarrow \varepsilon$	$P' \rightarrow \varepsilon$	$P' \rightarrow \wedge FP'$		
F		$F \rightarrow -F$				$F \rightarrow \text{root } F'F$	$F \rightarrow \log F'F$
F'	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$
C							

3.2 Muuttujien alustaminen ja poistaminen

3.2.1 SetVariable: set

Kieliopin luokkajako:

Lausekkeet (*expressions*) Muuttujan alustus.

Arvo (*value*) Muuttujan arvo.

Nimi (*name*) Muuttujan nimi.

Ylläolevan luokkajaon mukainen kontekstiton kielioppi:

$$\begin{aligned} E &\rightarrow N = V E' \\ E' &\rightarrow \varepsilon \mid ; E \\ N &\rightarrow s \\ V &\rightarrow v, \end{aligned}$$

missä v tarkoittaa arvoa (engl. *value*), mikä saadaan Calculate-automaatin kieliopilla³ ja s tarkoittaa merkkijonoa (engl. *string*). Kielioppi ei ole vasenrekursiivinen, jolloin se sopii sellaisenaan käytettäväksi ennustavassa jäsennyksessä.

NULLABLE, FIRST ja FOLLOW:

	NULLABLE	FIRST	FOLLOW
E		s	\dagger
E'	\star	$;$	\dagger
N		s	$=, ;$
V		v	$;, \dagger$

Ennustava jäsennystaulukko:

	s	v	$=$	$;$	\dagger
E	$E \rightarrow N = V E'$				
E'				$E' \rightarrow ; E$	$E' \rightarrow \varepsilon$
N	$N \rightarrow s$				
V		$V \rightarrow v$			

3.2.2 UnsetVariable: unset

Kieliopin luokkajako:

Lausekkeet (*expressions*) Lauseke muuttujan poistamiselle.

Nimi (*name*) Muuttujan nimi.

³Toisin sanoen $V \rightarrow E^*$, missä E^* on Calculate-automaatin kieliopin E -välimerkki.

Välimerkit (*punctuation*) Välimerkit muuttujien nimien välillä.

Ylläolevan luokkajaon mukainen kontekstiton kielioppi:

$$\begin{aligned} E &\rightarrow NE' \\ E' &\rightarrow \varepsilon \mid PE \\ P &\rightarrow , \mid ; \\ N &\rightarrow s \end{aligned}$$

Tämäkään kielioppi ei ole SetVariable-automaatin kieliopin tapaan vasenrekursiivinen.

NULLABLE, FIRST ja FOLLOW (erottimena käytetty \mid -merkkiä):

	NULLABLE	FIRST	FOLLOW
E		s	\dagger
E'	\star	$, \mid ;$	\dagger
P		$, \mid ;$	$s \mid \dagger$
N		s	$, \mid ; \mid \dagger$

Ennustava jäsennostaulukko:

	s	$,$	$;$	\dagger
E	$E \rightarrow NE'$			
E'		$E' \rightarrow PE$	$E' \rightarrow PE$	$E' \rightarrow \varepsilon$
P		$P \rightarrow ,$	$P \rightarrow ;$	
N	$N \rightarrow s$			

3.3 Fraction: frac, Help: help ja Variables: variables

Fraction- sekä Variables-automaattien kieliopeissa ei ole mitään sen enempää dokumentoitavaa; Fraction käyttää Calculate-automaatin kielioppia lausekkeen laskemiseen ja muokkaa laskun tuloksen murtoluvuksi sekä Variables-funktio vain tulostaa kaikki tallennetut muuttujat nimiin ja arvoineen. Help toisaalta tulostaa yleisen avustuksen kaikista ope-

raatioista tyhjällä sekä tunnistamattomalla syötteellä, kun taas tunnistetulla komennon nimellä kyseenomaisen komennon yksityiskohtaisemman avustuksen.

Viitteet

- [1] Antti-Juhani Kaijanaho, laskinohjelma, http://users.jyu.fi/~antkaij/opetus/auki/2013/luennot/laskin_ag.c, 2013, viittauspäivä 7.9.2013.
- [2] NUnit-frameworkin lataussivut: <http://nunit.org/?p=download>
- [3] Randall D. Knight, *Physics for Scientists and Engineers: A Strategic Approach*, Appendix A, 2008, toinen painos.
- [4] Antti-Juhani Kaijanaho, kurssin TIEA241 *Automaatit ja kielio-pit* luentokalvot *Säännöllisistä kielistä kontekstittomiin* sivu 12, 3.6.2013, <http://users.jyu.fi/~antkaij/opetus/auki/2013/luennot/I2II.pdf>, viittauspäivä 7.9.2013.
- [5] Antti-Juhani Kaijanaho, kurssin TIEA241 *Automaatit ja kielio-pit* jälkipuju *Täydentäviä muistiinpanoja ennustavasta jäsennyksestä* sivut 1&2, 10.6.2013, <http://users.jyu.fi/~antkaij/opetus/auki/2013/luennot/10.pdf>, viittauspäivä 7.9.2013.
- [6] Antti-Juhani Kaijanaho, kurssin TIEA241 *Automaatit ja kielio-pit* luentokalvot *Ennustava jäsennyks* sivut 8&9, 10.6.2013, <http://users.jyu.fi/~antkaij/opetus/auki/2013/luennot/II-LL.pdf>, viittauspäivä 7.9.2013.