

AKT eksami põhiosa (6. mai 2019)

Põhiosa eesmärgiks on koostada etteantud reeglitele vastav avaldise grammatika kasutades ANTLRi vahendeid, moodustada abstraktne süntaksipuu (kasutades etteantud AST klasse).

1. Grammatika koostamine (Ujukoma.g4)

Meie avaldised on siin täiesti tavalised aritmeetilised avaldised. Me üritame aga süntaktiliselt eraldada ujukoma ja täisarvu tüüpi avaldised. Mõlemal juhul on avaldis kas muutuja, literaal, aritmeetiline tehe või sulgudega ümbritsetud avaldis, aga peamine erinevus on siis literaalide ja muutujate definitsioonides ja siis tuleb lihtsalt keelata kahe erineva tüüpi avaldise kombineerimist. Seega peamine väljakutse on siin lubada $1+1$ ja $1.0+1.0$, aga mitte lubada $1+1.0$, sest üks on täisarv ja teine on ujukomaarv.

Järgmised on olulised reeglid nende avaldiste tüüpide määramiseks:

- Muutuja koosneb ühest ladina väiketähest. Sealjuures on meil kokkuleppe, et tähed *a-h* on reserveeritud ujukomamuutujatele ja *i-z* on täisarvu tüüpi muutuja.
- Täisarvuliteeralideks lubame märgita täiserve, aga me ei luba niisama nulle arvu ette, seega loeme täisarvuks 0, 1, 2, jne, aga ei tohi kirjutada 01 või 00. Ujukomaliteeraali puhul peab olema kasutatud punkt (mitte koma), millele peab järgnema vähemalt üks arv. Koma ette võib (aga ei pea) olema täisarv (millel on samad reeglid nagu täisarvuliteeraalil). Seega on lubatud 0.5 ja .5, aga mitte 5. või 05.5.
- Aritmeetiliste operatsioonide ja sulgavaldiste korral kehtivad aritmeetika tavalised assotsiatiivsused ja prioriteedid. Tuleb lihtsalt silmas pidada, et ainult sama tüüpi avaldised võib tehetega kombineerida. Siin on ka see oluline reegel, et **täisarvude jagamist me ei luba!**
- Tühikuid ja tabulaatoreid tuleb ignoreerida. (Selleks on vastav skip reegel meil ette antud!)

2. AST koostamine (UjukomaAst.java)

Siin ülesandes tuleb teisendada ANTLRi abil moodustatud *parse*-puu abstraktseks süntaksipuuks. AST moodustamisel tuleb kasutada *UjukomaNode.java* alamklasse. Kõige mugavam on kasutada selles samas ülemklassis olevad staatilised meetodid. Kui literaale välja arvata, mis on argumendi tüüpi järgi eristatavad, siis AST koostamisel tuleb avaldiste tüüpe ära unustada.

NB! Ära unusta enne ASTi ülesande juurde asumist genereerida lekser ja parser (*generateGrammarSource*). ASTi klasse ise muuta ei tohi.

Esitama peab failid: **Ujukoma.g4** ja **UjukomaAst.java**.