TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Artur Hendrik Mägi

Veebipõhise automaatide õppevahendi loomine ainele “Automaadid, keeled ja translaatorid”

Bakalaureusetöö (9 EAP)

Juhendaja(d): Vesal Vojdani

Tartu 2024

Veebipõhise automaatide õppevahendi loomine ainele “Automaadid, keeled ja translaatorid”

Lühikokkuvõte:

Käesolevas bakalaureusetöös tulemusena loodi uus veebipõhine automaatide õppevahend AktFLAP Tartu Ülikooli ainele “Automaadid, keeled ja translaatorid”. Töös antakse ülevaade automaatide simulaatoritest ja nende kasutamisest õppetöös. Lisaks tuuakse välja veebirakendusele seatud nõuded ja kasutatavad tehnoloogiad. Viimaseks on kirjeldatud valminud õppevahendi osasid ning toodud välja rakenduse testimise protsess.

Võtmesõnad:

Automaatide simulaator, veebirakendus, õppevahend

CERCS: P175 Informaatika, süsteemiteooria

Creating a Web-Based Automata Learning Tool for the Course “Automata, Languages and Compilers”

Abstract:

As a result of this bachelor's thesis, a new web-based automata learning tool called AktFLAP was created for the University of Tartu course “Automata, Languages and Compilers”. The thesis provides an overview of automata simulators and their use in teaching. Additionally, the requirements set for the web application and the technologies used are outlined. Finally, the components of the developed learning tool and the details of the application testing process are described.

Keywords:

Automata simulator, web application, learning tool

CERCS: P175 Informatics, systems theory

Sisukord

Sissejuhatus

Tartu Ülikoolis õpetatavas aines “Automaadid, keeled ja translaatorid” kasutatakse lõplike automaatide õpetamisel rakendust JFLAP. Koduste ülesannete ja eksami raames loodud automaate saab testida Moodle keskkonnas ja kursuse repositooriumis olevate testide abil[1]. Sellel rakendusel on aga antud aine õpetamise raames mitmed puudused. Suurimaks puuduseks on asjaolu, et JFLAP rakenduse kasutamisel ei ole võimalik loodud automaate koheselt testida, vaid on vaja fail iga kord uuesti alla laadida ja seejärel üles laadida Moodlesse või kasutada kursuse repositooriumis olevaid Java teste. Teiseks puuduseks on rakenduse kasutajakogemus. Peamiseks probleemiks kasutajakogemuse juures on iganenud kasutajaliides. Lisaks sellele on mõningaid probleeme ka kasutajaliidesega interaktsioonides. Puudujääke on täpsemalt kirjeldatud teises peatükis.

Käesoleva bakalaureusetöö eesmärk on luua veebirakendus, mis võimaldab lõplike automaatide joonistamist ja käivitamist. Kasutajal peaks olema võimalus loodud automaate alla laadida. Lisaks peaks olema võimalus lugeda automaat sisse failist. See annab võimaluse kasutada samu faile nii JFLAP-is kui ka antud rakenduses. Lisaks sellele on loodavas veebirakenduses võimalik kasutajal praktikumide ülesannete raames loodud automaate koheselt avalike testidega testida, mis lihtsustab antud protsessi. Veel on eesmärgiks luua automaatide normaliseerimise funktsionaalsus, et lihtsustada automaatide omavahelist võrdlemist. Lisaks sellele on eesmärgiks luua võimalus mittedeterministlikke lõplike automaate teisendada deterministlikeks lõplikeks automaatideks. Viimaseks on plaanis parandada ka rakenduse kasutajakogemust. Veebirakendust ei ole vaja installeerida enda arvutisse, vaid on kättesaadav aadressil (ligipääsuks peab olema TÜ sisevõrgus või kasutama VPN ühendust): https://courses.cs.ut.ee/t/akt/Main/AktFlap

Käesolev bakalaureusetöö koosneb kuuest suuremast osast. Esmalt on välja toodud olulisemad mõisted ja terminid antud valdkonna kohta. Seejärel kirjeldatakse käsitletavat valdkonda üldisemalt. Kolmandaks tuuakse välja veebirakenduse olulisemad funktsionaalsed ja mittefunktsionaalsed nõuded. Neljandas peatükis kirjeldatakse rakenduses kasutatavaid tehnoloogiaid. Viiendas osas tutvustatakse juba valminud rakenduse tähtsamaid osasid. Viimases

[1] Kursuse info: https://courses.cs.ut.ee/2024/AKT/spring

ehk kuuendas osas käsitletakse rakenduse testimise protsessi ja võimalusi, milliste funktsionaalsuste lisamisega saaks õppevahendit veel edasi arendada ja täiendada.

Antud lõputöös on loetavuse parandamiseks kasutatud tehisintellekti ChatGPT-3.5[2]. Tehisintellekti pole kasutatud sisu loomiseks.

[2] https://openai.com/index/chatgpt/

1. Mõisted ja terminid

Lõplik olekumasin ehk **lõplik automaat** (ingl *finite automaton*) kujutab endast käitumismudelit, mis koosneb olekutest, üleminekutest ehk siiretest ja toimingutest. Olek on salvestatud informatsioon mineviku kohta, s.t sisendite muutuste kohta süsteemi käivitamisest kuni käesoleva hetkeni. Üleminek näitab oleku muutust ja seda kirjeldatakse tingimusega, mis peab olema täidetud, et üleminek oleks võimalik. Toiming on selle tegevuse kirjeldus, mida antud momendil on vaja teostada [1]. Aine "Automaadid, keeled ja translaatorid" raames käsitletakse keelte äratundmise automaate, mille toiminguteks on sisendi lugemine ja automaadi vastus on binaarne (jah/ei) vastavalt sellele, kas automaadi kirjeldatud keel aktsepteerib sisendit või mitte.

Deterministlik lõplik automaat ehk **DFA** (ingl *deterministic finite automaton*) on lõplik automaat, mis vastab järgmistele tingimustele [2]:

1. Iga üleminek on unikaalselt määratletud selle lähteoleku ja sisendsümboli alusel.
2. Iga ülemineku jaoks olekute vahel on vajalik sisendsümboli lugemine.

Mittedeterministlik lõplik automaat ehk **NFA** (ingl *nondeterministic finite automaton*) on lõplik automaat, mis ei pea vastama DFA-le seatud tingimustele. Ehk iga DFA on ka NFA [3].

Turingi masin (ingl *Turing machine*) on Alan Turingu 1937. aastal kirjeldatud arvuti matemaatiline mudel, millel on järgmised põhikomponendid [4]:

1. Arvuti mälu modelleeriv lõpmatu jadapöördusega mälu nn "lint".
2. Lõplikust arvust olekutest koosnev programm, kus iga arvutuse sisendandmed x esitatakse kodeerituna lindil ja töö käigus asendatakse väljundandmeid y = M(x) esitava koodiga ning iga arvutust saab esitada programmi olekute lõpliku jadaga, mis alati lõpeb kindlas lõppolekus.

DFA minimeerimine (ingl *Minimization of DFA*) on protsess, mille käigus algoritmiliselt vähendatakse DFA olekute arvu minimaalseks võimalikuks olekute arvuks [5].

JFLAP on tarkvara formaalkeelte teemade eksperimenteerimiseks, sealhulgas lõplike automaatide, Turingi masinate, mitut tüüpi grammatikate, parsimisega jne. Lisaks näidete loomisele ja testimisele võimaldab JFLAP teisendamist ühelt vormist teisele, näiteks mittedeterministliku lõpliku automaadi e NFA muutmine deterministlikuks lõplikuks automaadiks e DFA-ks, minimaalse oleku DFA-ks või regulaarsavaldiseks [6].

2. Automaatide simulaatorid

Järgnev materjal tugineb P. Chakraborty jt artiklile [7]. Automaatide simulaatorid on tarkvaralised vahendid, mis võimaldavad uurida ja analüüsida erinevat tüüpi automaate, näiteks lõplikud automaadid ja Turingi masinad. Need simulaatorid on kasulikud formaalkeelte, algoritmide ja automaaditeooria valdkondades ning neid saab kasutada nii õpetamise kui ka teadustöö eesmärkidel. Järgnevalt on välja toodud mõned olulisemad aspektid automaatide simulaatorite kohta:

1. Automaatide simulaatorid võimaldavad kasutajatel luua ja kirjeldada erinevaid automaatide mudeleid, määratledes nende olekud, üleminekureeglid ja algseisundi.
2. Enamik simulaatoreid pakub graafilist keskkonda, kus automaadi struktuuri saab visuaalselt esitada, mis muudab keerukamate automaatide mõistmise lihtsamaks.
3. Simulaatorid võimaldavad kasutajatel käivitada automaate, sisestada sisendeid ja jälgida, kuidas automaat vastavalt sellele käitub. See aitab mõista automaadi käitumist erinevate sisendite korral.
4. Automaatide simulaatorid on sageli kasutusel formaalkeelte ja automaaditeooria õpetamisel. Need võimaldavad õpilastel praktiliselt uurida ja kogeda teooria kontseptsioone.
5. Simulaatorid võimaldavad automaatide teisendamist, näiteks NFA-st DFA-sse või regulaaravaldisest automaadiks. See on oluline näiteks automaatide ekvivalentsuse seisukohast.

JFLAP on üks populaarsemaid automaatide simulaatoreid, mida kasutatakse formaalkeelte ja automaaditeooria õppes. Lisaks sellele on olemas mitmeid teisi simulaatoreid, mis pakuvad erinevaid funktsioone ja võimalusi automaatide uurimiseks. Automaatide simulaatorid on seega võimsad vahendid, mis aitavad nii õpilastel kui ka teadlastel süvitsi mõista automaatide käitumist ja nende rakendusi arvutiteaduses.

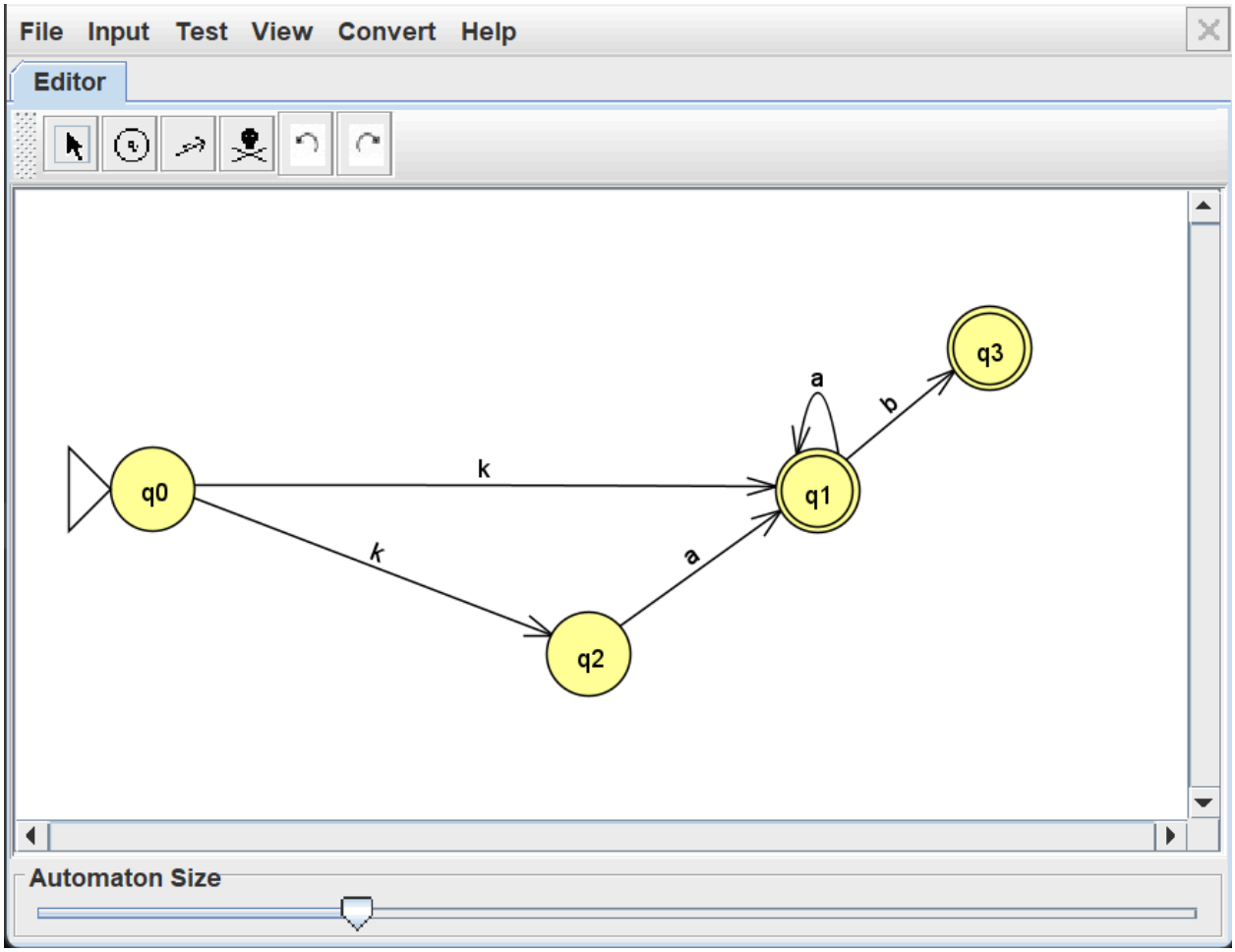
2.1 Automaatide simulaatorite kasutamine õppetöös

Järgnevalt on välja toodud mõned olulised aspektid automaatide simulaatorite kasutamise kohta õppetöös. Järgnev materjal tugineb P. Chakraborty jt artiklile [8]. Alates 20. sajandi keskpaigast on automaaditeooria muutunud arvutiteaduse oluliseks alustalaks, kuid kuna tegemist on keerulise valdkonnaga, siis on hakatud kasutama automaatide simulaatoreid antud teema õpetamiseks. Automaatide simulaatorid võimaldavad tudengitel simuleerida erinevate automaatide tööd, nagu näiteks lõplikud automaadid, Turingi masinad jne. Simulaatorid võimaldavad õpikust saadud või isekujundatud automaatide uurimist ja testimist. Tudengitel on võimalus luua suuri ja keerukaid automaate, mida käsitsi kavandada oleks keerukas. Enamik simulaatoreid pakub ka erinevaid simulatsioonirežiime. Kuid üheks olulisemaks eeliseks automaatide simulaatorite kasutamisel õppetöös on õpilaste huvi ja arusaamise suurendamine, samuti toovad need endaga kaasa paremaid õpitulemusi. Eelnevat väidet kinnitab ka T. Singh jt läbi viidud uuring [9], kus on välja toodud, et õpilaste arvates muudab simulaatorite kasutamine kursuse huvitavamaks ja kaasahaaravamaks. Nagu eelnevatest näidetest näha, siis kaasneb automaatide simulaatorite kasutamisega õppetöös mitmeid positiivseid tagajärgi. Seetõttu on ka Tartu Ülikooli arvutiteaduse instituudi õppeaines “Automaadid, keeled ja translaatorid” õppevahendina kasutusele võetud tarkvara JFLAP. Antud rakenduse puhul on aga tegemist allalaetava tarkvaraga, mistõttu ei ole selle kasutamine niivõrd mugav ja lihtne kui oleks veebirakenduse kasutamine. Seetõttu ongi tekkinud õppeaine “Automaadid, keeled ja translaatorid” raames vajadus sarnase funktsiooniga tarkvara järele, mis oleks aga veebirakendusena lahendatud. Lisaks sellele on oluline ka välja tuua T. Singh jt läbi viidud uuringu [9] teine tulemus, mille kohaselt annab simulaatorite kasutamine ka paremaid tulemusi õppimisel, eriti kui kasutada paralleelselt mitut erinevat rakendust, seda peamiselt just seetõttu, et alternatiivne tarkvara võimaldab kasutada õppevahendit ka teistes seadmetes peale arvuti. Eelnevat väidet kinnitab ka H. Mohamedi läbi viidud uuring [10], mille eesmärk oli testida automaatide simulaatorite efektiivsust õppevahendina ülikooli automaaditeooria kursuse raames. Selleks jagati üliõpilased nelja rühma, kus kaks rühma kasutasid simulaatoreid oma õppetöös ja kaks rühma ei kasutanud neid. Tulemused näitasid, et simulaatoreid kasutanud rühmad sooritasid kursusel paremaid tulemusi, kui need, kes simulaatoreid ei kasutanud. Eelnevad näited ilmestavad hästi teist tähtsat põhjust, miks veebipõhise automaatide simulaatori kaasamine õppeaine “Automaadid, keeled ja translaatorid” raames on oluline.

2.2 Rakenduse JFLAP puudused

Aines “Automaadid, keeled ja translaatorid” on hetkel automaaditeooria õpetamisel kasutusel ainult rakendus JFLAP. Antud rakendusel on olemas paljud õppeaine seisukohalt olulised funktsionaalsused. Rakenduses on võimalik automaate joonistada ja neid käivitada. Veel on olemas võimalus mittedeterministlikke automaatide teisendamine deterministleks automaatideks ja deterministlikke automaatide minimeerimine. Lisaks on võimalus automaate salvestada ja failist lugeda. Siiski on antud rakendusel puudu üks oluline funktsionaalsus, milleks on aine ülesannete kohene testimisvõimalus ja tagasiside saamine. Hetkel on aine raames kasutusel lahendus, kus automaate saab testida laadides faili alla ja seejärel kasutada kursuse repositooriumis olevaid eeldefineeritud teste või laadida fail üles Moodle keskkonda, kus on ka testid olemas. Selline lahendus on aga tudengile üsnagi ajakulukas, sest peale igat muudatust ülesande lahendamisel on vaja tagasiside saamiseks rakendusest väljuda ja kasutada teisi keskkondi. Antud lõputöö raames loodud õppevahendi üheks suurimaks eesmärgiks ongi lisada funktsionaalsus, mille abil on tudengitel koheselt võimalus automaate testida ja saada vastav tagasiside.

Teiseks probleemiks on JFLAP-i kasutajakogemus. Tegemist on üsnagi vana rakendusega, mistõttu rakenduse kasutajaliides ei ole tänapäevane (vt. Joonis 1). Lisaks sellele on mõningaid probleeme kasutajaliidese interaktsioonide disainis. Näiteks valides rakenduses režiimiks “Lisa olek”, “Lisa üleminek” või “Kustuta” ei ole võimalik joonistusala liigutada või juba olemasolevate olekute atribuute muuta. Selline lahendus ei ole aga väga intuitiivne ja kasutajasõbralik. Ka seda probleemi üritab lahendada antud lõputöö raames loodud õppevahend. Eesmärgiks on luua lihtne ja intuitiivne kasutajaliides, mille disain on tänapäevane.



Joonis 1. Kuvatõmmis JFLAP rakendusest

3. Veebipõhise õppevahendi nõuded

Antud peatükis on kirjeldatud käesoleva lõputöö raames loodud veebipõhise õppevahendi funktsionaalseid ja mittefunktsionaalseid nõudeid.

3.1 Funktsionaalsed nõuded

Järgnevalt on välja toodud nimekiri funktsionaalsetest nõuetest, millele õppevahend peab vastama:

1. Kasutajal peab olema võimalus lisada joonistusalale uusi automaadi olekuid.
2. Kasutajal peab olema võimalus lisada olemasolevate olekute vahele üleminekuid.
3. Kasutajal peab olema võimalus olemasolevaid olekuid ja üleminekuid kustutada.
4. Kasutajal peab olema võimalus olekuid joonistusalal liigutada.
5. Kasutajal peab olema võimalus joonistusala liigutada ning sisse ja välja suumida.
6. Kasutajal peab olema võimalus joonistatud automaati jooksutada, andes ette sisendsõne.
7. Automaadi jooksutamisel on kasutajale igal sammul kuvatud olekud, milles automaat saab sellel hetkel olla.
8. Automaadi jooksutamisel on kasutajale igal sammul kuvatud selleks hetkeks töödeldud sisendsõne.
9. Automaadi jooksutamisel peab kasutajal olema igal hetkel võimalus liikuda samm edasi ja ka samm tagasi.
10. Automaadi jooksutamisel peab kasutajal olema igal hetkel võimalus liikuda töötlemise algusesse ja ka töötlemise viimasesse sammu.
11. Automaadi jooksutamise viimasel sammul, kui sisendsõne on lõpuni töödeldud, peab kasutajale kuvama teadet selle kohta, kas antud automaat võtab sõne vastu või ei võta.
12. Kasutajal peab olema võimalus joonistatud NFA-d teisendada DFA-ks.
13. Kasutajal peab olema võimalus joonistatud DFA-d minimeerida.
14. Kasutajal peab olema võimalus joonistatud automaati alla laadida formaadis .jff, et sama automaati saaks kasutada ka rakenduses JFLAP.
15. Allalaetavate failide nimi peaks vastama hetkel valitud ülesandele. Näiteks kui on valitud Ülesanne 2, siis allalaetav fail peaks olema nimetatud ‘yl2.jff’.

16. Kasutajal peab olema võimalus automaati üles laadida failist, mille formaat on .jff, et saaks kasutada ka JFLAP rakenduses loodud automaate.
17. Kasutajal peab olema võimalus joonistatud automaati testida vastavalt defineeritud praktikumiülesannetele.
18. Kasutajal peab olema võimalus valida konkreetne praktikumiülesanne, mida testida.
19. Kasutajale peab olema kuvatud hetkel valitud praktikumiülesande kirjeldus.
20. Kasutajal peab olema võimalus valida rakenduse keelt (eesti ja inglise).
21. Süsteem peab kasutajat teavitama, kui automaadil puudub algolek ja kasutaja proovib automaati testida, jooksutada, teisendada DFA-ks või minimeerida DFA-d.
22. Süsteem peab kasutajat teavitama, kui automaat ei ole NFA ja kasutaja proovib teisendada NFA-d DFA-ks.
23. Süsteem peab kasutajat teavitama, kui automaat ei ole DFA ja kasutaja proovib minimeerida DFA-d.
24. Süsteem peab kasutajat teavitama, kui kasutaja testib loodud automaati vastavalt praktikumiülesannetele ja loodud automaat ei ole õige lahendus. Teavitus peab kuvama ka ühe sõna, mida loodud automaat ei aktsepteeri, aga peaks või ühe sõna, mida loodud automaat aktsepteerib, aga ei tohiks.
25. Süsteem peab kasutajat teavitama, kui kasutaja testib loodud automaati vastavalt praktikumiülesannetele ja loodud automaat on õige lahendus testitavale ülesandele.
26. Süsteem peab kasutajat teavitama, kui tekib tõrge *backend* serveriga ühendamisel.

3.2 Mittefunktsionaalsed nõuded

Järgnevalt on välja toodud nimekiri mittefunktsionaalsetest nõuetest, millele õppevahend peab vastama:

1. Kasutaja suudab ilma juhendita kasutada rakenduse põhilisi funktsionaalsusi, eeldusel, et varasemalt on kasutajal kogemus JFLAP töövahendiga.
2. Rakendus peab töötama enimkasutatavates veebilehitsejates.
3. Rakendus peab vastama päringutele maksimaalselt 5 sekundiga.

4. Kasutatavad tehnoloogiad

Järgnevalt on välja toodud rakenduse loomiseks kasutatavad tehnoloogiad.

4.1 *Frontend* tehnoloogiad

Antud alapeatükis on täpsemalt kirjeldatud veebipõhise õppevahendi loomisel kasutatavaid *frontend* tehnoloogiaid.

4.1.1 JavaScript

JavaScript on Netscape'i poolt välja töötatud skriptikeel, mis võimaldab veebiautoritel luua interaktiivseid veebisaite. Kuigi JavaScriptil on mitmeid ühiseid jooni Javaga, on see siiski täiesti iseseisev keel. JavaScript suudab suhelda HTML-keeles kirjutatud lähtekoodiga ja võimaldab muuta veebilehed dünaamiliseks [11]. JavaScript toetab ka objektorienteeritud, imperatiivset ja deklaratiivset programmeerimist [12]. Antud lõputöö raames loodava rakenduse kliendipool on kirjutatud JavaScriptis. Lisaks puhtale JavaScriptile kasutatakse rakenduses ka erinevaid teeke.

4.1.2 CSS

CSS ehk kaskaadlaadistik on veebilehtede valmistajatele ja kasutajatele mõeldud laadistik. Laadilehed (ingl *style sheets*) kirjeldavad, kuidas HTML dokumente esitada kuvaril, printeril või kõnesüntesaatorist kostva kõnena. Laadilehed lubavad kasutajal muuta sadade dokumendilehtede väljanägemist üheainsa CSS faili muutmise teel. Laadileht koosneb reeglitest, mis teatavad brauserile, kuidas dokumenti kuvada. Andes laadilehe elementidele mitmesuguseid väärtusi, saab ära määrata fonte, värve, fooni omadusi, teksti omadusi, sisubokside omadusi, klassifikatsiooni omadusi, ühikuid jms [11].

4.1.3 React

React on JavaScripti teek, mida kasutatakse kasutajaliideste loomiseks. Kogu Reacti loogika põhineb komponentidel, mis on modulaarsed ja taaskasutatavad [13].

4.1.4 vis.js

vis.js on dünaamiline ja brauseripõhine visualiseerimise teek. Teek on loodud suurte dünaamiliste andmekogudega töötamiseks ning võimaldab andmete manipuleerimist ja interaktsiooni andmetega. Teek koosneb komponentidest DataSet, Timeline, Network, Graph2d ja Graph3d [14]. Antud lõputöö raames loodavas õppevahendis on kasutatud komponenti Network, et visualiseerida veebilehel joonistatud automaate. Network on tippudest ja servadest koosnevate võrgustike visualiseerimiseks kasutatav komponent. Visualiseerimine on lihtsasti kasutatav ja toetab elementide kohandamist vastavalt vajadusele. Network komponent kasutab visualiseerimiseks HTML Canvas elementi [15].

4.1.5 i18next

i18next on JavaScripti raamistik, mis võimaldab hallata veebirakenduse keelt ja sellega seonduvat. Raamistik laeb tõlked automaatselt tõlgete failist ja keele muutmisel teeb ise automaatselt vastavad muudatused veebilehel kuvatavates tekstides [16]. Antud lõputöö raames loodavas õppevahendis on raamistik kasutusel eesti ja inglise keele toe haldamiseks.

4.1.6 Font Awesome

Font Awesome on fontide ja ikoonide *toolkit*, mis on loodud CSS-i baasil. Selle kasutamiseks on olemas ametlik Reacti komponent, mis muudab integreerimise rakendusse väga lihtsaks [17]. Antud lõputöö raames loodavas rakenduses on *toolkit* kasutusel nuppudele ikoonide lisamiseks.

4.2 *Backend* tehnoloogiad

Antud alapeatükis on täpsemalt kirjeldatud veebipõhise õppevahendi loomisel kasutatavaid *backend* tehnoloogiaid.

4.2.1 Java

Java on laialdaselt kasutatav objektorienteeritud programmeerimiskeel, mis loodi 1990-aastate keskpaigas. See on disainitud lihtsaks, platvormist sõltumatuks ja turvaliseks, mistõttu on sellel mitmeid kasutusalasid, näiteks veebirakenduste arendus [18]. Antud lõputöö raames loodava õppevahendi *backend* on arendatud kasutades Javat koos Spring Boot raamistikuga.

4.2.2 Spring Boot

Spring Boot on Java raamistik, mida kasutatakse eraldiseisvate Spring rakenduste loomiseks. Springi eelis on kiire *startup* ja *shutdown* ning lisaks ka optimeeritud käitamine [19]. Spring Boot kiirendab arendusprotsessi veelgi, sest rakenduse üles seadmine on lihtsustatud, kasutades automaatset konfigureerimist. Tegemist on ühe maailma populaarseima Java raamistikuga [20].

4.2.3 dk.brics.automaton

Dk.brics.automaton on Java *package*, mis sisaldab DFA/NFA implementatisooni ja toetab erinevaid automaatidega tehtavaid operatsioone, nagu näiteks determineerimine, minimeerimine jne [21]. Antud *package* on kasutusel ka kursusel "Automaadid, keeled ja translaatorid" ülesannete lahendusautomaatide testimiseks[3]. Käesoleva lõputöö raames loodava õppevahendi puhul on kasuatatud *package*'it järgmiste funktsionaalsuste implementeerimisel:

1. NFA teisendamine DFA-ks.
2. DFA minimeerimine.
3. Joonistatud automaadi ekvivalentsuse testimine praktikumiülesannete lahendusautomaatidega (eeldab lahendusautomaatide olemasolu serveris).

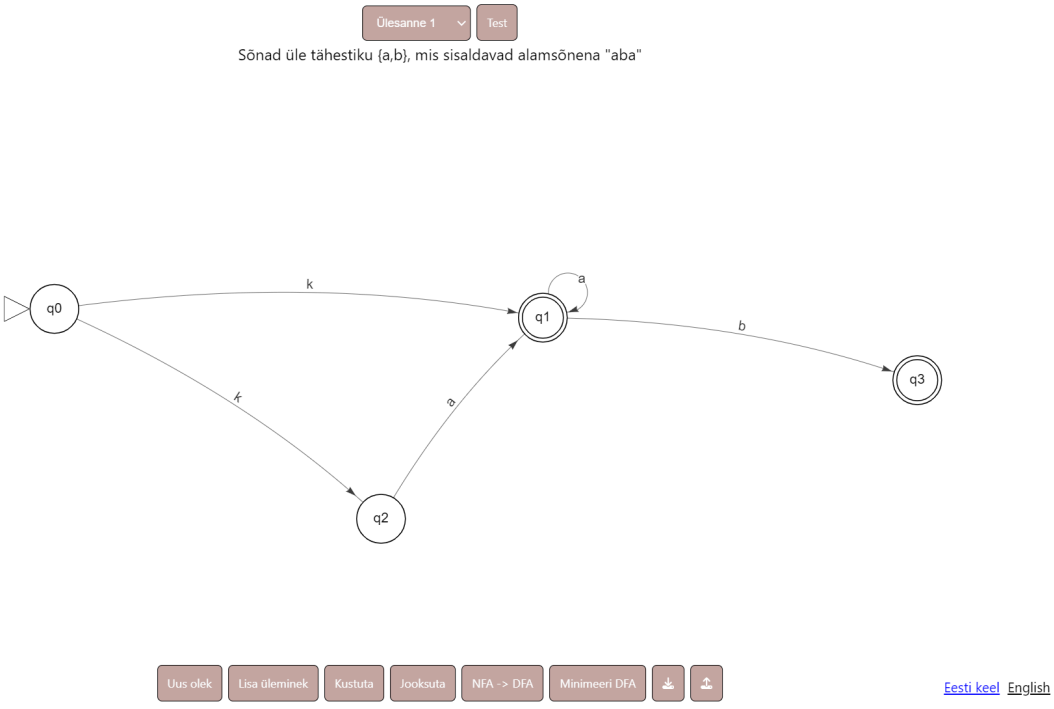
[3] Kursuse repositoorium: https://github.com/sws-lab/akt2024

5. Rakenduse osad

Valminud veebipõhine õppevahend koosneb kahest peamisest vaatest: “Automaadi joonistamise vaade” ja “Automaadi jooksutamise vaade”. Järgnevalt tutvustatakse neid täpsemalt ja kirjeldatakse, kuidas saab nendes leiduvaid funktsionaalsusi kasutada.

5.1 Automaadi joonistamise vaade

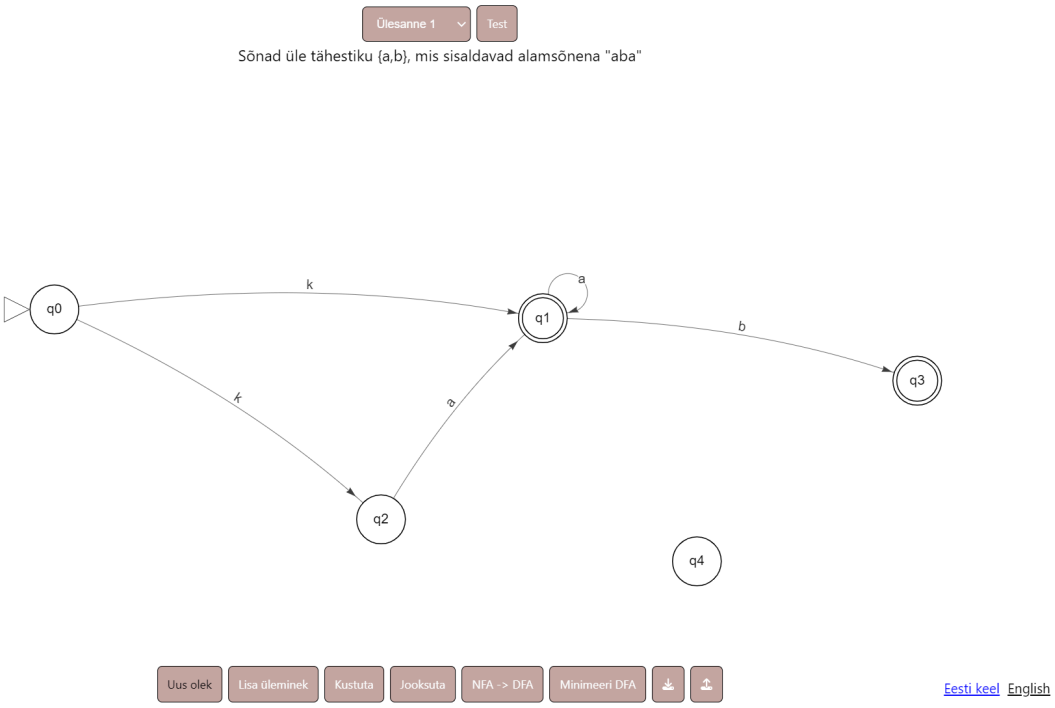
Valminud veebipõhise õppevahendi peamiseks vaateks on “Automaadi joonistamise vaade” (vt. Joonis 2). Antud vaates on kasutajal võimalik lisada uusi olekuid ja üleminekuid ning neid kustutada. Teiseks võimaluseks on olekute muutmine alg- ja lõppolekuteks. Veel on olemas võimalus joonistatud automaati teisendada NFA-st DFA-ks ja DFA-d minimeerida. Lisaks saab valminud automaati alla laadida ning automaati üles laadida failist. Oluliseimaks lisatud funktsionaalsuseks võrreldes JFLAP rakendusega on võimalus koheselt testida loodud automaate. Lisaks on võimalus õppevahendi keelt muuta eesti ja inglise keeleks. Järgnevates alapeatükkides on täpsemalt kirjeldatud neid funktsionaalsusi.



Joonis 2. Automaadi joonistamise vaade

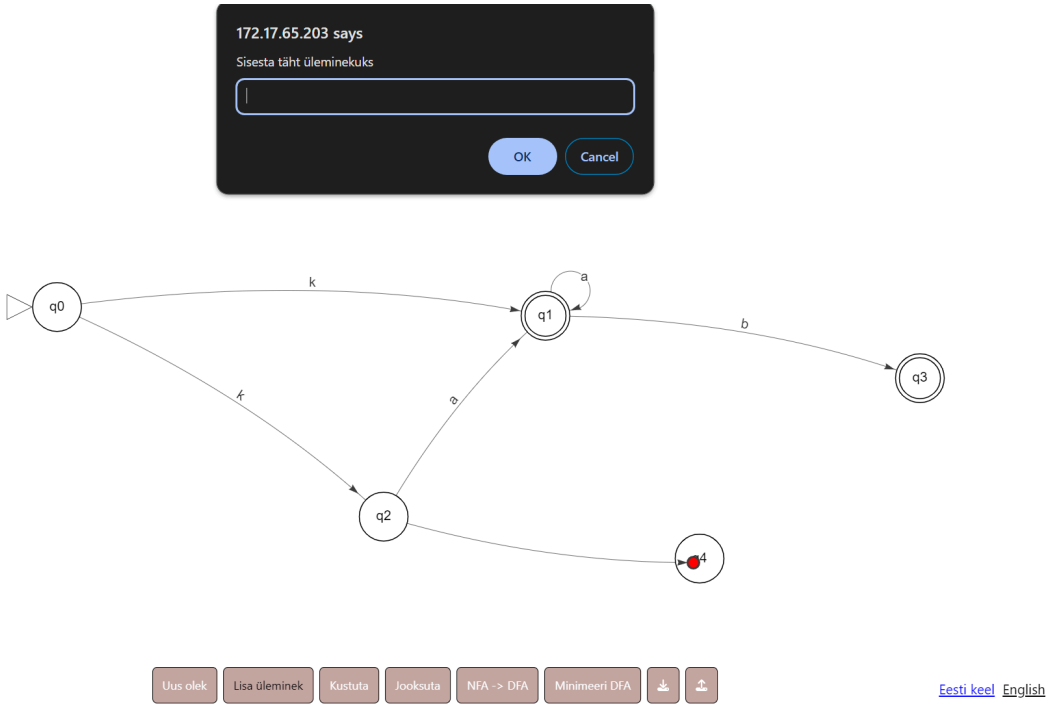
5.1.1 Olekute ja üleminekute lisamine ning kustutamine

Olekute lisamiseks peab kasutaja valima vastava režiimi vajutades nupule “Uus olek” ning seejärel tegema vasaku hiirevajutuse joonistusalal (vt. Joonis 3).



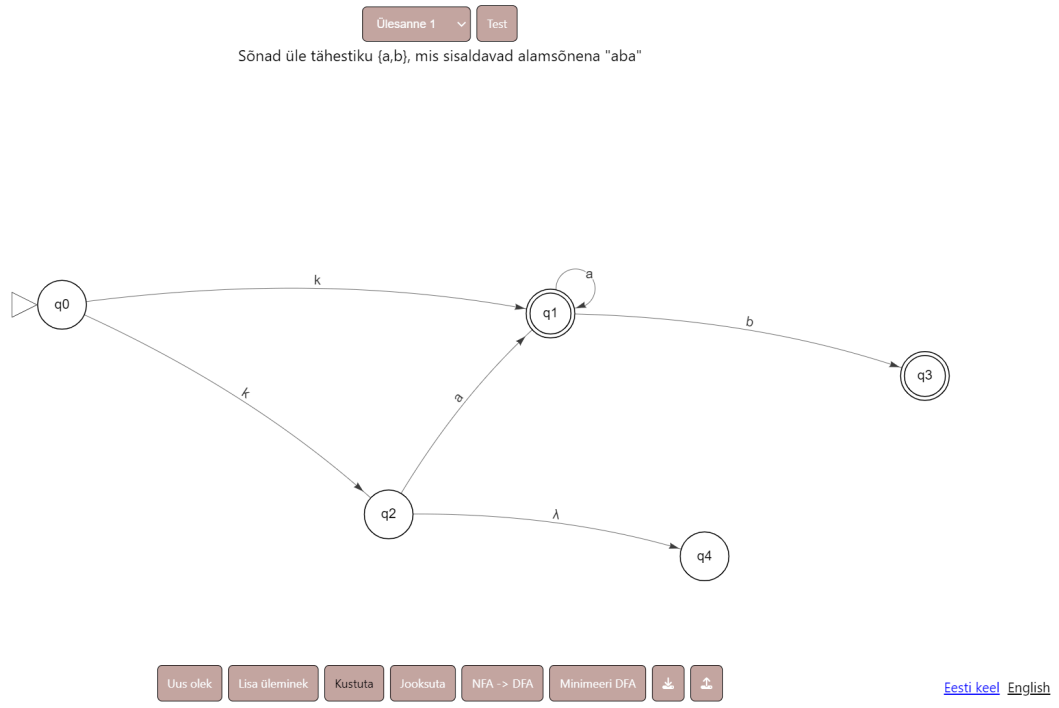
Joonis 3. Oleku lisamise režiim uue lisatud olekuga (q4)

Üleminekute lisamiseks peab kasutaja valima vastava režiimi vajutades nupule “Lisa üleminek” ning seejärel vajutama vasaku hiireklahviga soovitud olekule ja hiireklahvi all hoides liigutama hiire ülemineku sihtoleku peale. Peale seda küsitakse kasutajalt ülemineku sümbolit (vt. Joonis 4), kusjuures lubatud ei ole *whitespace* sümbolite sisestamine ega mitme sümboli korraga sisestamine. Jättes sisestusala tühjaks on tegemist epsilonüleminekuga (tähistatud sümboliga λ).

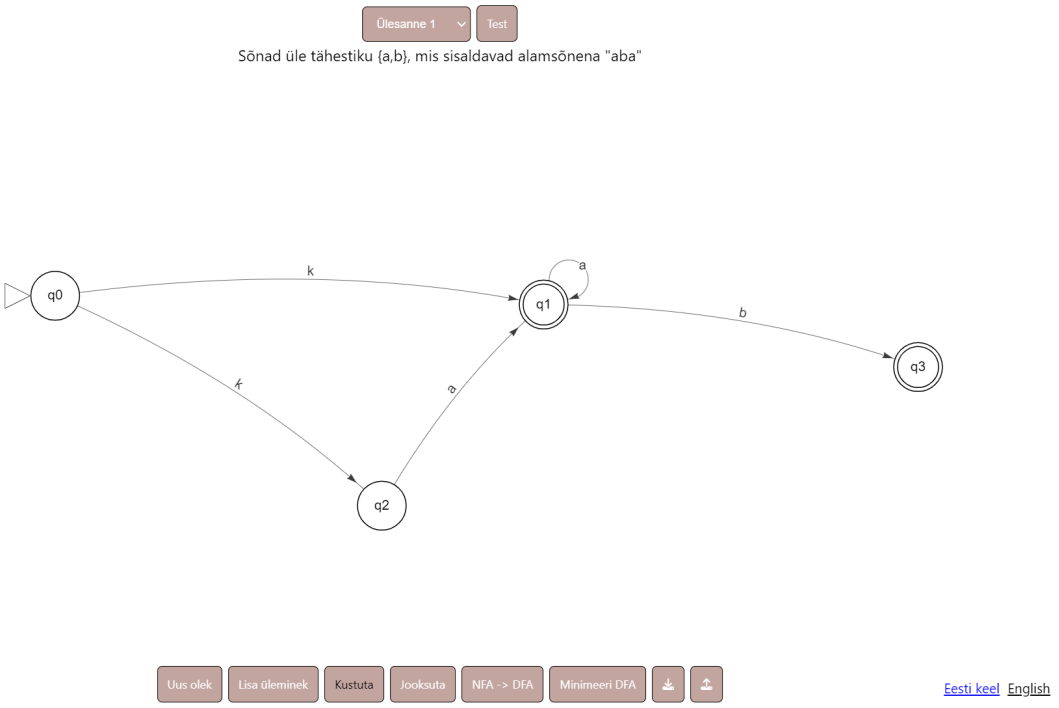


Joonis 4. Ülemineku lisamisel sümboli küsimine

Olekute ja üleminekute kustutamiseks peab kasutaja valima vastava režiimi vajutades nupule "Kustuta" ning seejärel tegema vasaku hiirevajutuse olekul või üleminekul, mida kustutada soovib. Kusjuures oleku kustutamisel kustutatakse automaatselt ka kõik selle olekuga seotud üleminekud (vt. Joonised 5 ja 6).



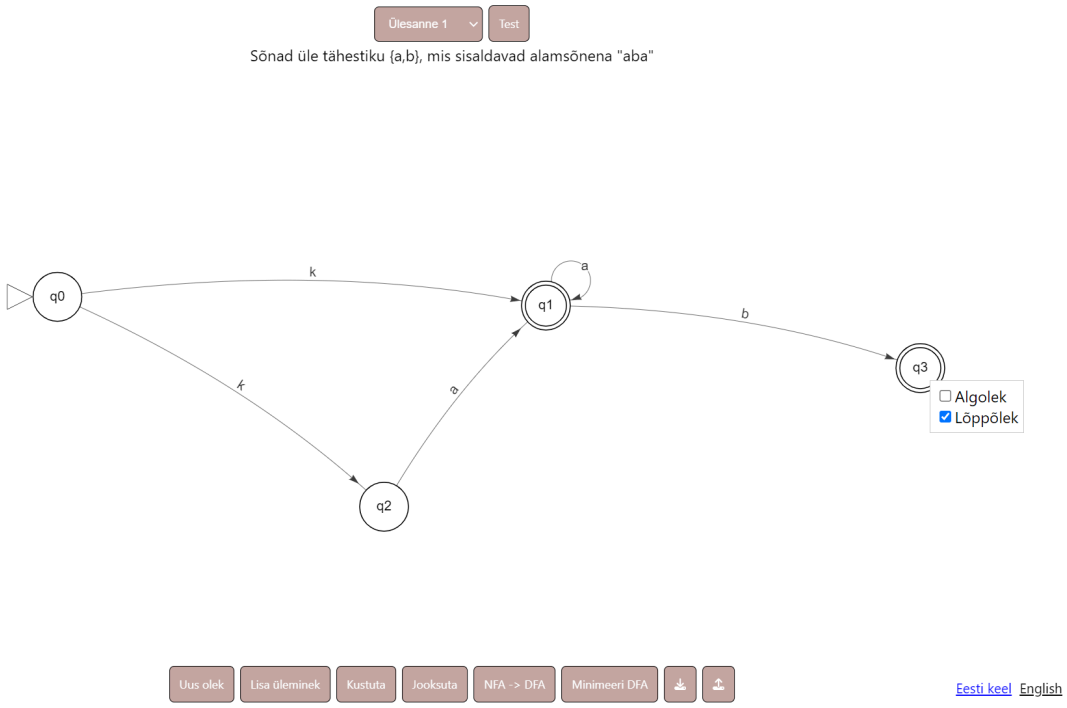
Joonis 5. Automaat kustutamise režiimis enne kustutamise teostamist



Joonis 6. Automaat kustutamise režiimis peale oleku q4 kustutamist

5.1.2 Olekute muutmine alg- ja lõppolekuteks

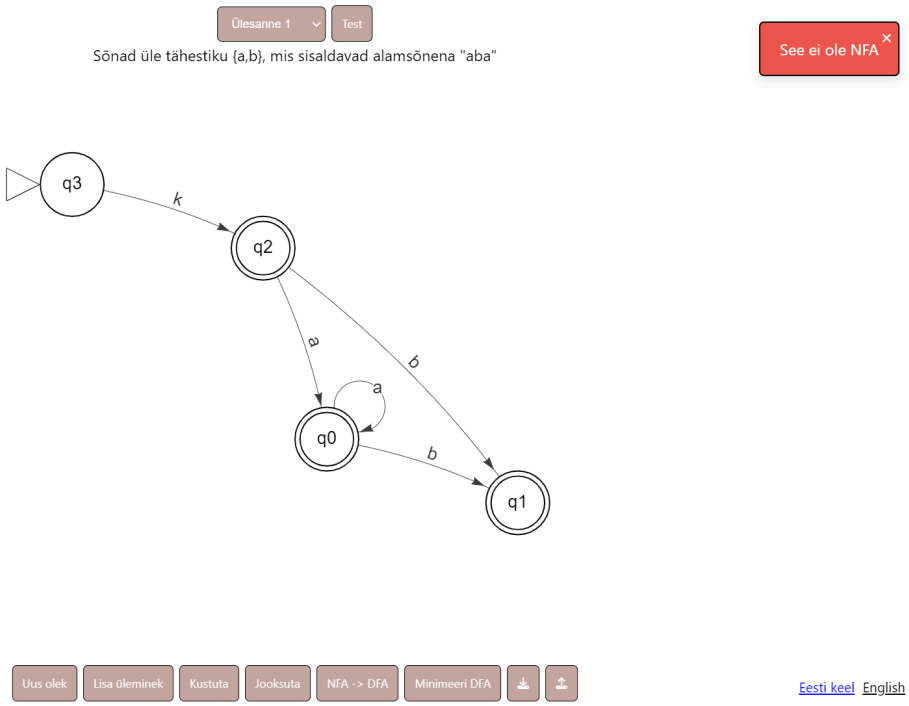
Kui kasutaja soovib muuta olemasolevat olekut alg- või lõppolekuks tuleb tal teha parem hiirevajutus vastaval olekul ja seejärel valida hüpikmenüüst vastav omadus (vt. Joonis 7). Kuna automaadil saab olla vaid üks algolek, siis oleku algolekuks muutmisel eemaldab rakendus automaatselt varasemalt algolekult (kui selline olek leidub) selle omaduse.



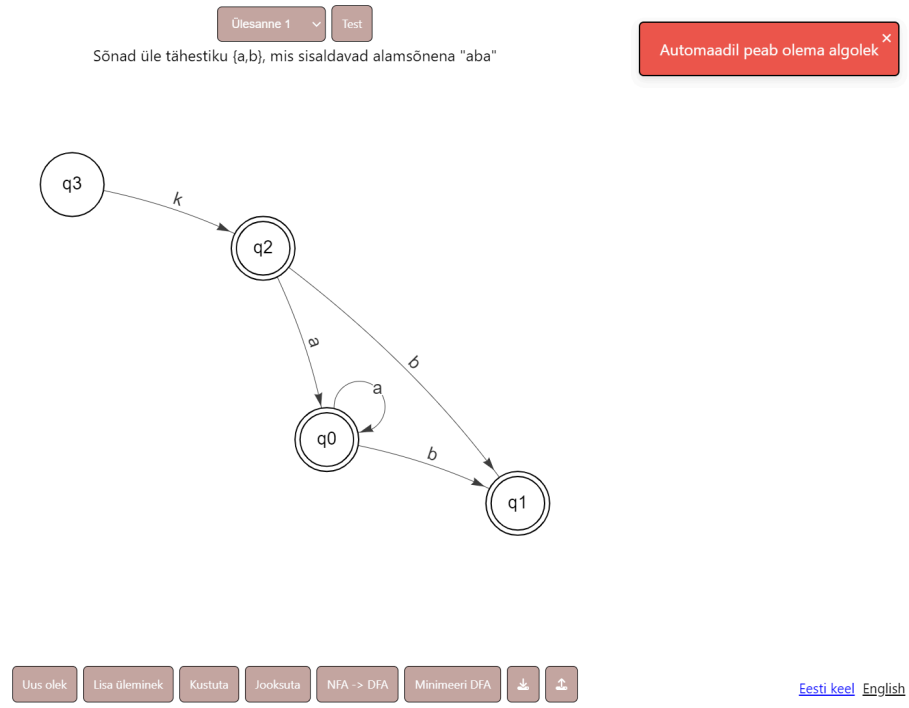
Joonis 7. Alg- ja lõppolekuks muutmise hüpikmenüü

5.1.3 NFA teisendamine DFA-ks

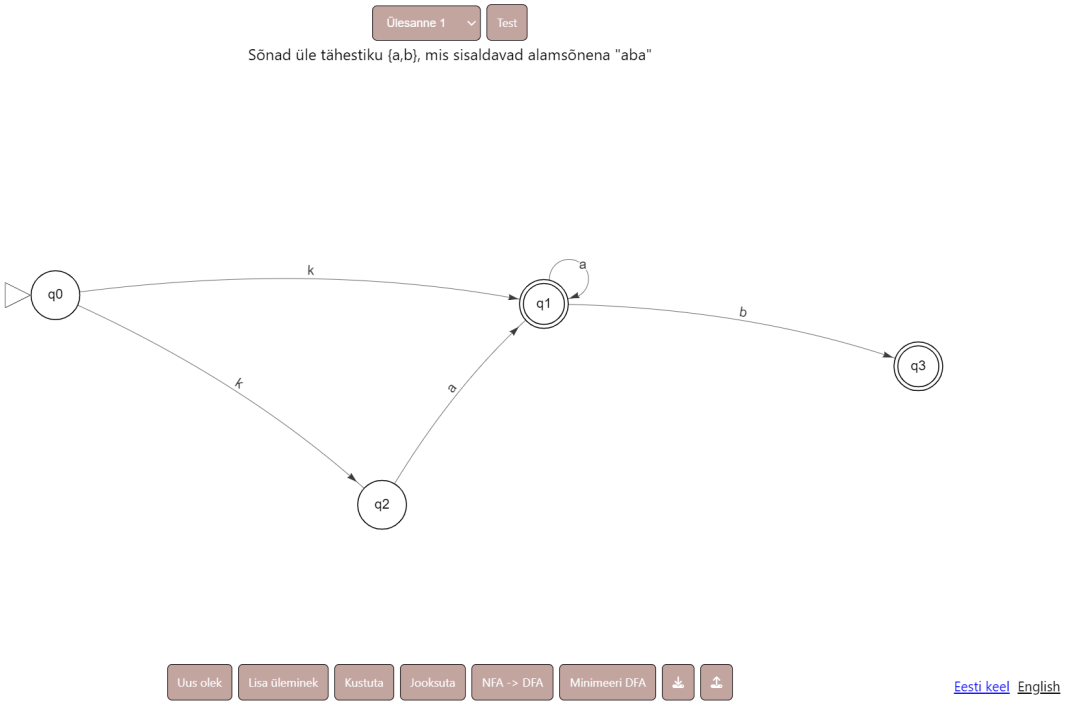
NFA DFA-ks teisendamiseks peab kasutaja vajutama nupule “NFA -> DFA”. Kui tegemist pole NFA-ga või automaadil puudub algolek, siis kuvatakse kasutajale vastav teade (vt. Joonised 8 ja 9). Peale kontrollide läbimist saadetakse automaadi info *backendile*, kus toimub teisendus, mille järel saadetakse *frontendile* tagasi uus automaat ja kuvatakse seda kasutajale (vt. Joonised 10 ja 11).



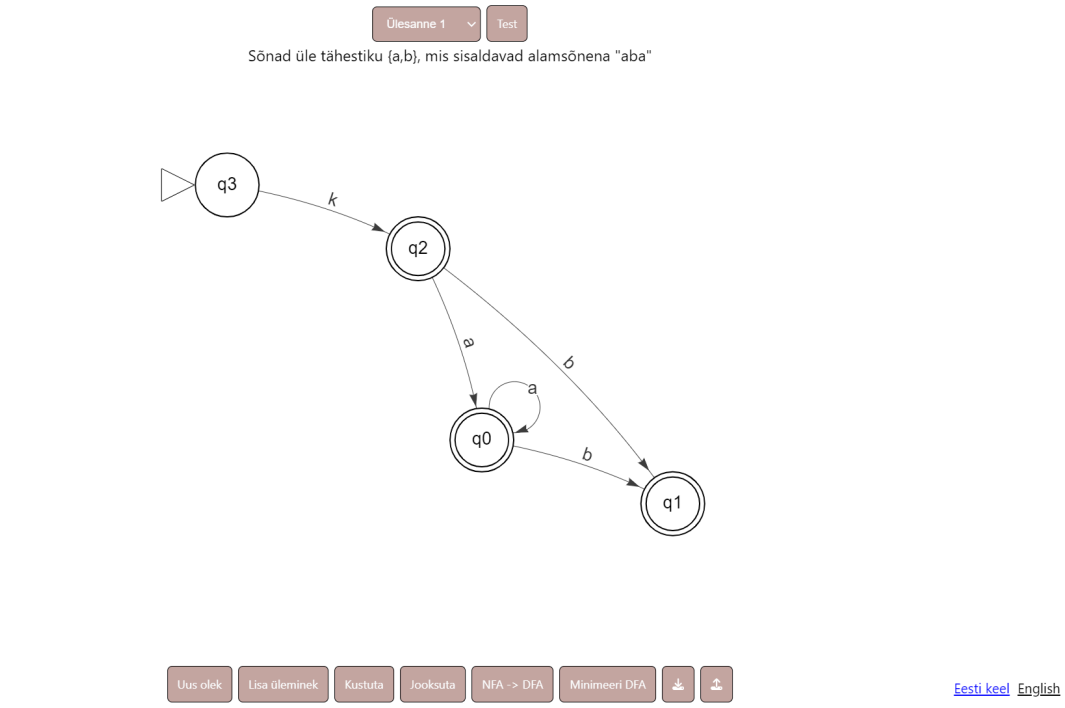
Joonis 8. Kuvatav teavitus, kui tegemist pole NFA-ga



Joonis 9. Kuvatav teavitus, kui automaadil puudub algolek



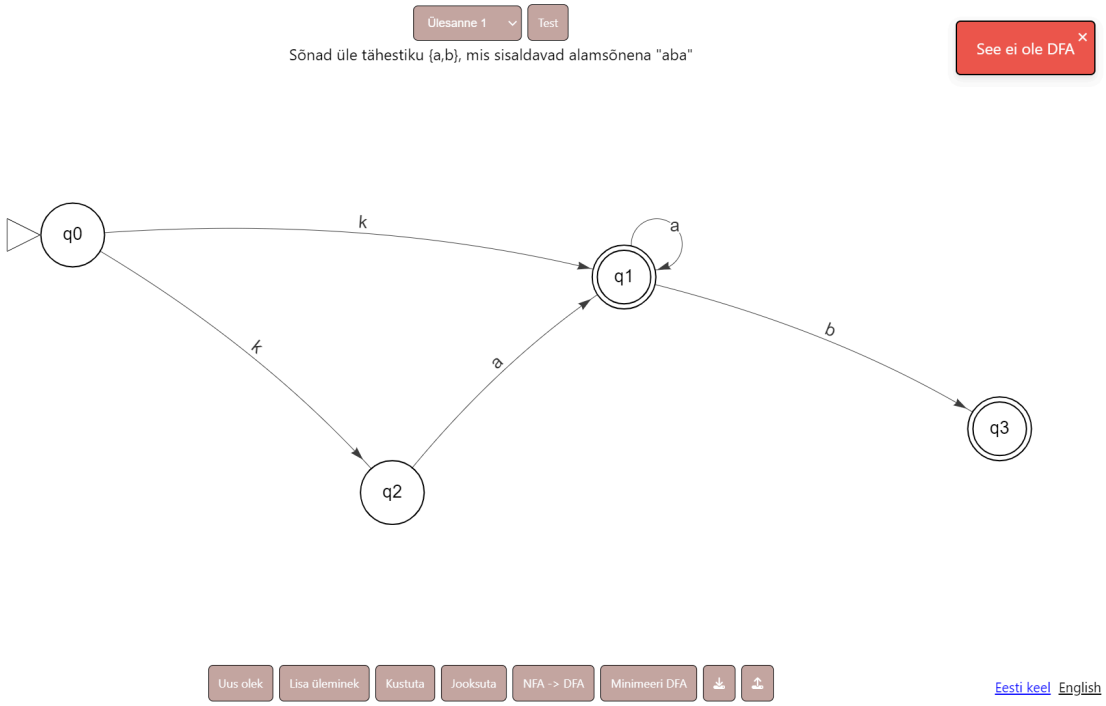
Joonis 10. NFA enne DFA-ks teisendamist



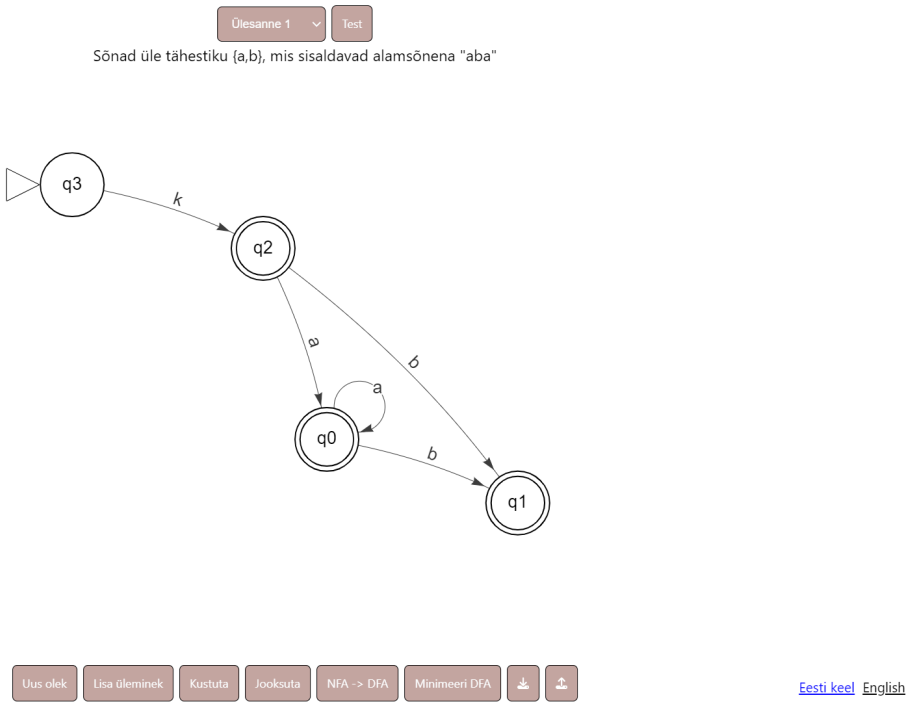
Joonis 11. Eelmisel joonisel kujutatud automaat pärast DFA-ks teisendamist

5.1.4 DFA minimeerimine

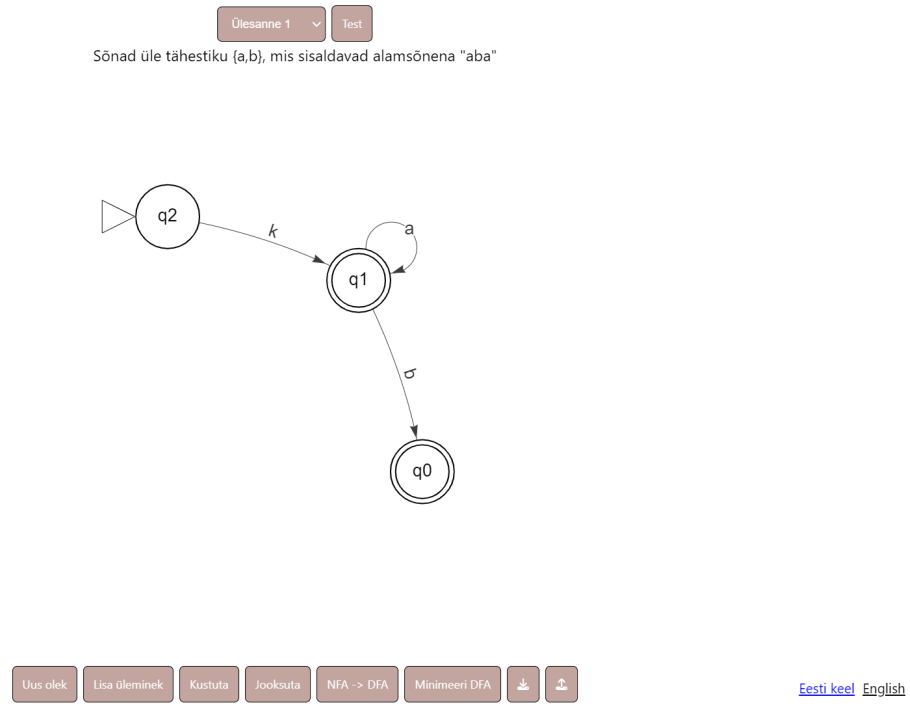
DFA minimeerimiseks peab kasutaja vajutama nupule “Minimeeri DFA”. Kui tegemist pole DFA-ga või automaadil puudub algolek, siis kuvatakse kasutajale vastav teade (vt. Joonis 12). Peale kontrollide läbimist saadetakse automaadi info *backendile*, kus toimub DFA minimeerimine, mille järel saadetakse *frontendile* tagasi uus automaat ja kuvatakse seda kasutajale (vt. Joonised 13 ja 14).



Joonis 12. Kuvatav teavitus, kui tegemist pole DFA-ga



Joonis 13. DFA enne minimeerimist



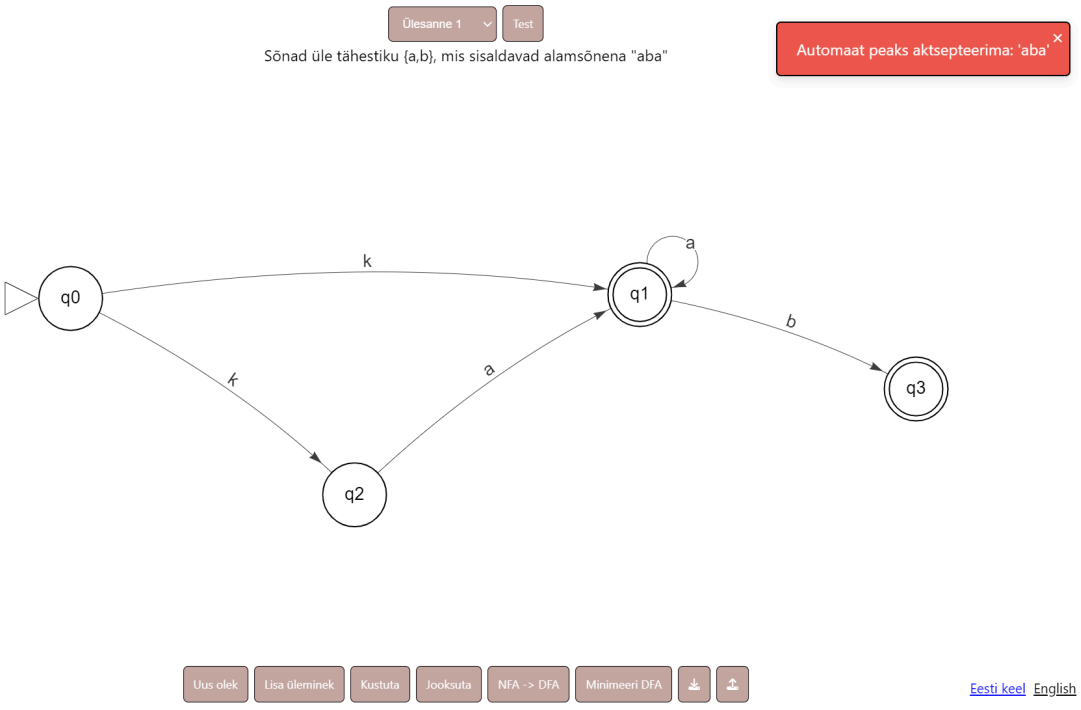
Joonis 14. Eelmisel joonisel kujutatud DFA pärast selle minimeerimist

5.1.5 Automaadi alla- ja üleslaadimine

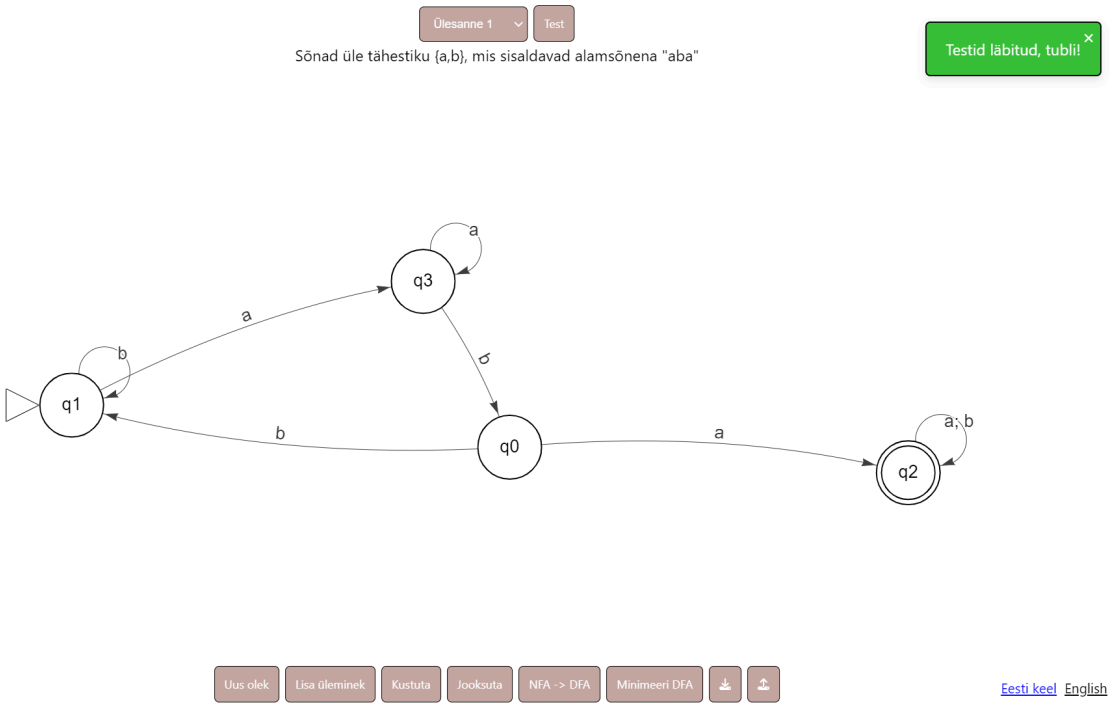
Automaadi alla ja üleslaadimiseks peab kasutaja vajutama vastavat nuppu. Allalaetavale failile määratakse automaatselt nimi vastavalt ekraani ülaosas valitud ülesandele, näiteks kui on valitud Ülesanne 2, siis allalaetav fail salvestatakse nimega ‘yl2.jff’. Failist automaadi üleslaadimiseks peab faili sisu vastama kindlale vormile (vt. Lisa II).

5.1.6 Automaadi testimine

Automaadi testimiseks peab kasutaja valima ekraani ülaosas olevast menüüst soovitud ülesande (ülesande kirjeldus on näha antud menüü all) ning seejärel vajutama nupule “Test”. Seejärel saadetakse automaadi info *backendile*, kus testitakse automaadi ekvivalentsi lahendusautomaadiga ning *frontendile* saadetakse tagasi teade õnnestumise või mitteõnnestumise kohta. Kui automaat ei vasta ülesande kirjelduses defineeritud automaadile, siis kuvatakse kasutajale veateada, mis sisaldab ühte sõne, mida automaat peaks aktsepteerima või ühte sõne, mida automaat ei tohiks aktsepteerida (vt. Joonis 15). Kui automaat automaat vastab ülesande kirjelduses defineeritud automaadile, siis kuvatakse kasutajale teade õnnestumise kohta (vt. Joonis 16).



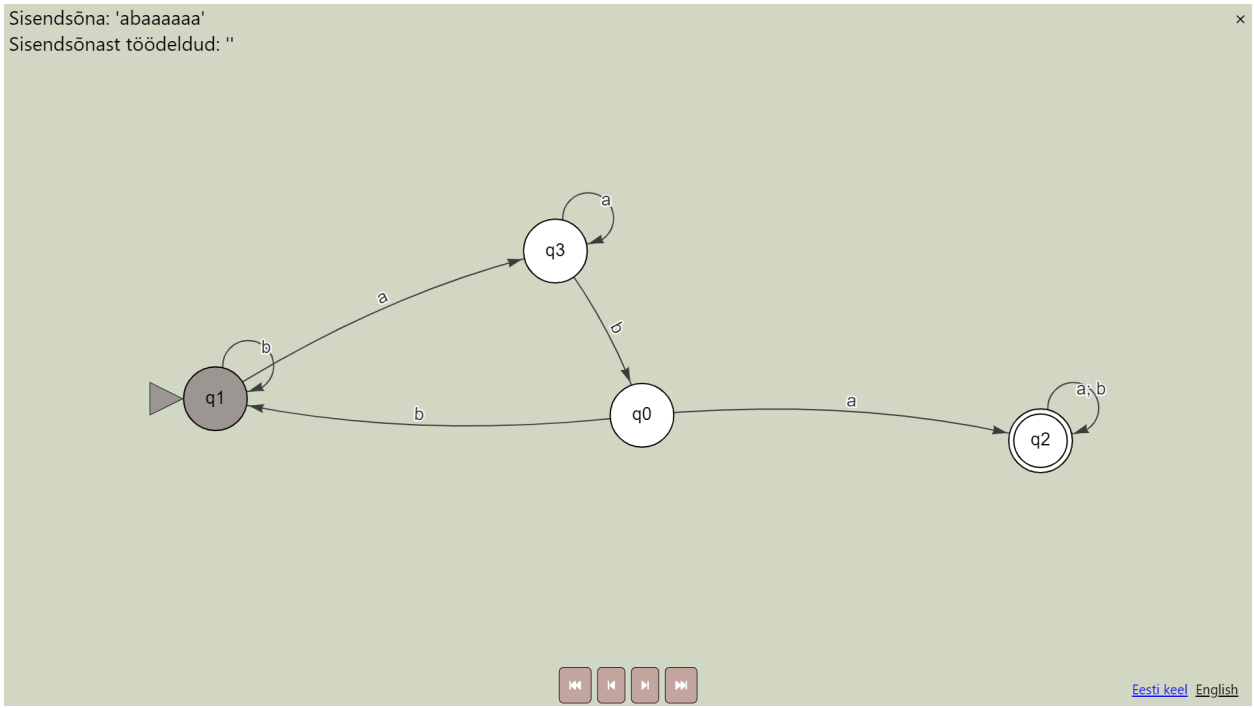
Joonis 15. Veateade testi mitteläbimise korral



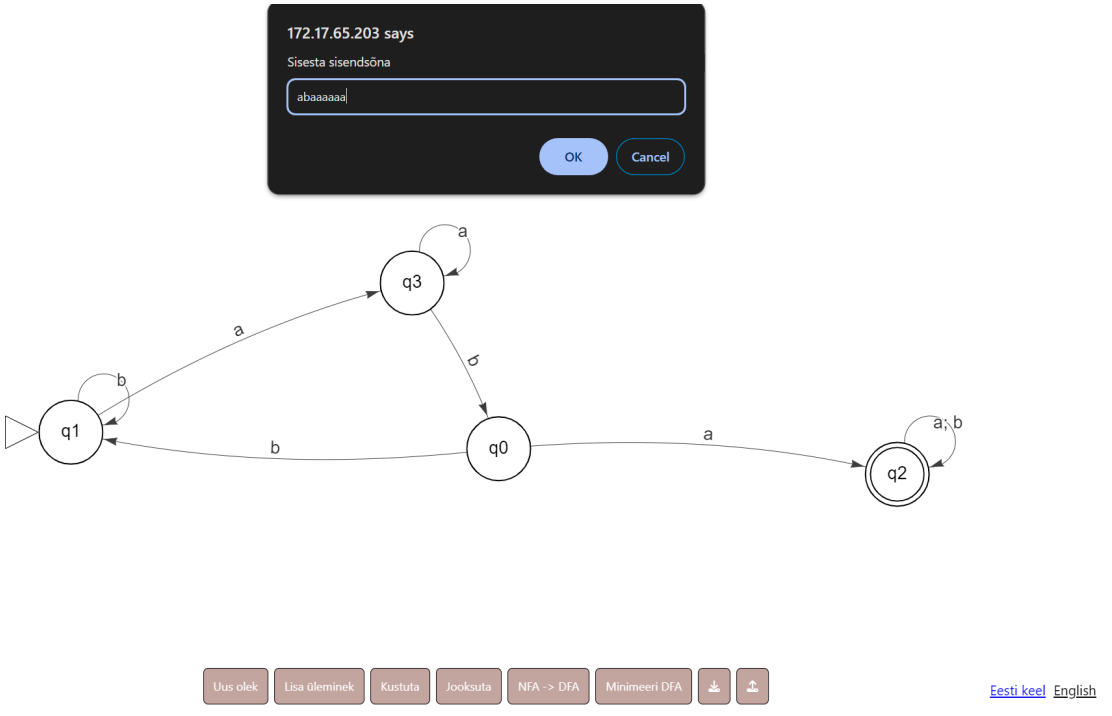
Joonis 16. Teade testi eduka läbimise kohta

5.2 Automaadi jooksutamise vaade

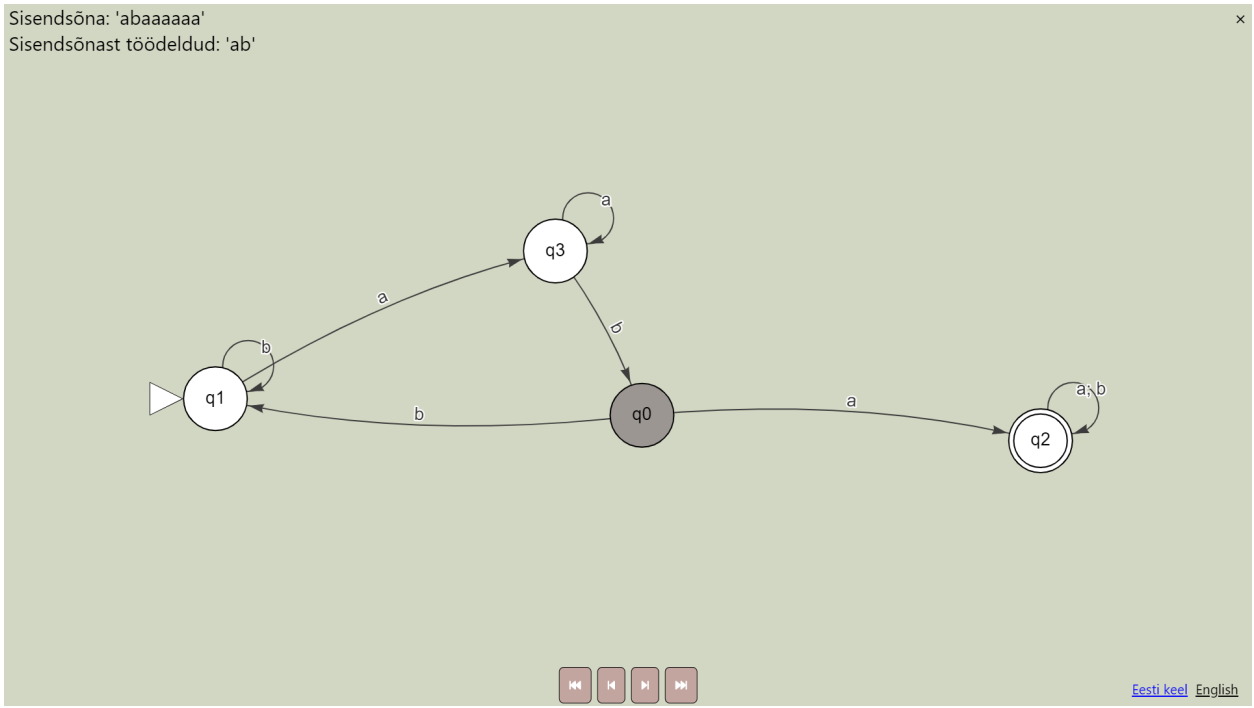
Valminud veebipõhise õppevahendi teiseks oluliseks vaateks on “Automaadi jooksutamise vaade” (vt. Joonis 17). Antud vaates on kasutajal võimalik simuleerida automaadi tööd sisendsõnel. Antud funktsionaalsuse kasutamiseks peab kasutaja “Automaadi joonistamise vaates” vajutama nupule “Jooksuta”, misjärel küsib süsteem kasutajalt sisendsõne (vt. Joonis 18). Automaadi jooksutamisel on kasutajal võimalik teha samm edasi ja samm tagasi ning vajadusel liikuda ka töötlemise algusesse ja lõppu. Igal sammul on kasutajale kuvatud selleks hetkeks töödeldud sisendsõne ning olekud, milles automaat viibida saab (vt. Joonis 19). Viimasel sammul, kui sisendsõne on töödeldud, on kasutajale kuvatud ka teade selle kohta, kas automaat aktsepteerib sisendsõne või mitte (vt. Joonis 20). Ka selles vaates on võimalus õppevahendi keelt muuta eesti ja inglise keeleks.



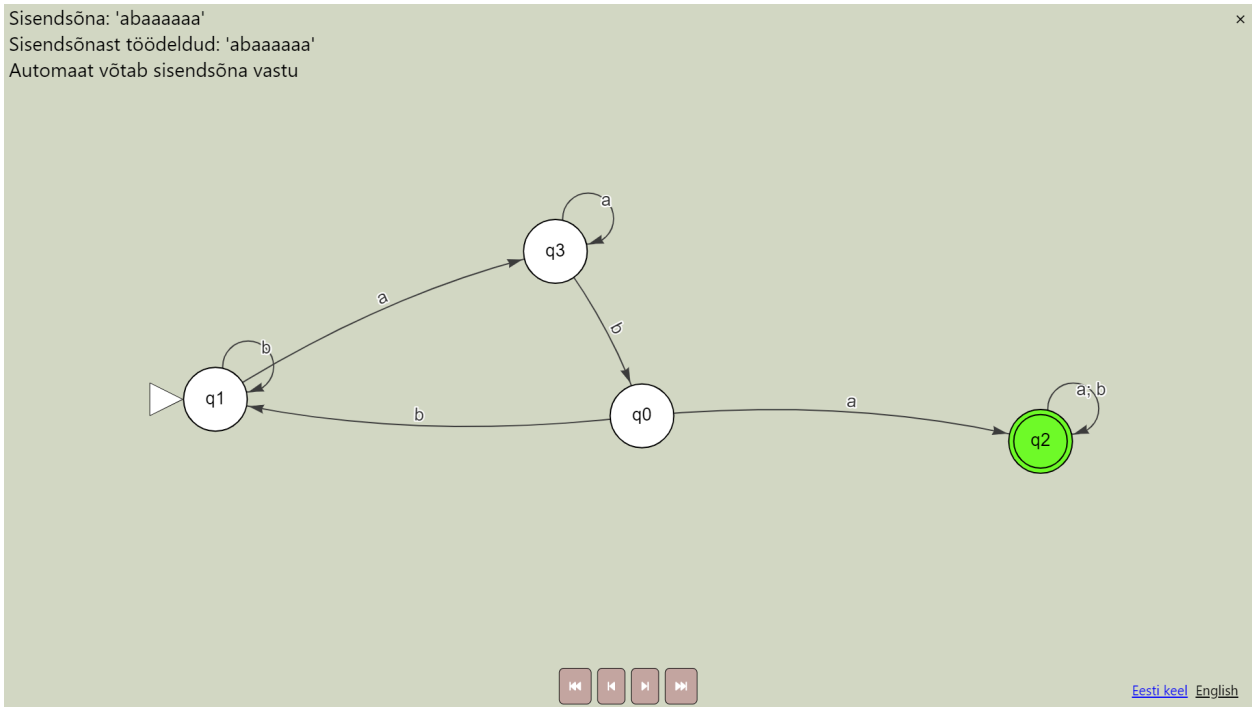
Joonis 17. Automaadi jooksutamise vaade



Joonis 18. Automaadi jooksutamise alustamine



Joonis 19. Olekud, kus automaat viibib pärast seda, kui sisendsõnest on töödeldud ‘ab’



Joonis 20. Kuvatav teade peale sisendsõne lõpuni töötlemist

6. Rakenduse testimine

Eesmärgiga välja selgitada rakenduse probleemikohti ja võimalikke tuleviku edasiarendusi, viidi läbi õppevahendi kasutatavuse testimine. Kasutatavuse testimine näitab, kui lihtne on kasutajatel rakendust kasutada oma eesmärkide täitmiseks, mis aitab omakorda muuta rakenduse võimalikult intuitiivseks [22].

6.1 Testimise läbiviimine

Testimiseks küsitleti 4 aine "Automaadid, keeled ja translaatorid" õppejõudu, kes pidid täitma etteantud testülesande. Testülesanne koosnes järgnevatest osadest:

1. Minna õppevahendi veebileheküljele ja valida lahendatavaks ülesandeks Ülesanne 1.
2. Joonistada ülesande kirjeldusele vastav automaat.
3. Teha automaat minimaalseks DFA-ks.
4. Testida loodud automaadi korrektsust, seni kuni automaat läbib testid.
5. Jooksutada automaati sisendiga etteantud sisendsõnega.

Küsitud küsimused olid järgmised:

1. Kui intuitiivne oli automaadi joonistamise protsess? (Skaalal 1-5, kus 1 tähendas "Üldse mitte intuitiivne" ja 5 tähendas "Väga intuitiivne")
2. Palun põhjendage eelnevat hinnangut.
3. Kui intuitiivne oli automaadi minimaalseks DFA-ks teisendamise protsess? (Skaalal 1-5, kus 1 tähendas "Üldse mitte intuitiivne" ja 5 tähendas "Väga intuitiivne")
4. Palun põhjendage eelnevat hinnangut.
5. Kui intuitiivne oli automaadi jooksutamise protsess? (Skaalal 1-5, kus 1 tähendas "Üldse mitte intuitiivne" ja 5 tähendas "Väga intuitiivne")
6. Palun põhjendage eelnevat hinnangut.
7. Kas said aru ülesande täitmise elementidest? (ikoonid, nupud jne). Too välja, mis sind nende juures häiris või mis meeldis.

Ühel testijal oli võimalik testida ainult rakenduse *frontendi*, seega ei andnud tema hinnangut DFA minimeerimise kohta.

6.2 Testimise tulemused

Järgnevates alapeatükkides on osade kaupa välja toodud testküsimustikule saadud vastused.

6.2.1 Automaadi joonistamine

Automaadi joonistamise protsessi intuitiivsuse hindeks anti keskmiselt 3,5. Peamised väljatoodud probleemid ja ettepanekud olid järgmised:

1. Erinevate režiimide (olekute lisamine, kustutamine jne) vahel liikumine võiks olla võimalik ka klaviatuuri kasutades.
2. Joonistusala sisse- ja väljasuumimine on liiga tundlik/kiire.
3. Kustutamine võiks toimida ka *backspace* või *delete* klahvi vajutamisel.

6.2.2 Automaadi muutmine minimaalseks DFA-ks

Automaadi minimaalseks DFA-ks muutmise protsessi intuitiivsuse hindeks anti keskmiselt 4,33. Peamised väljatoodud probleemid ja ettepanekud olid järgmised:

1. Kui automaat on juba minimaalne, siis võiks kasutajat sellest teavitada.
2. Lisaks protsessi tulemusele võiks kasutajale kuvada ka samme, kuidas algoritm selleni jõuab.

6.2.3 Automaadi jooksutamine

Automaadi jooksutamise protsessi intuitiivsuse hindeks anti keskmiselt 4,5. Peamised väljatoodud probleemid ja ettepanekud olid järgmised:

1. Aktiivsete olekute kuvamiseks võiks kasutada mõnda muud värvi peale halli, sest hall üldiselt sümboliseerib mitte-aktiivseid elemente.
2. Võiks olla näha igal sammul ka eelmise sammu aktiivsed olekud, et oleks paremini jälgitav, kuidas automaat sisendit töötleb.

6.2.4 Ülesande täitmise elemendid

Üldiselt saadi ülesande täitmise elementidest hästi aru. Peamised väljatoodud probleemid ja ettepanekud olid järgmised:

1. Automaadi jooksutamise vaatest väljumiseks kasutatav nupp ei olnud hästi märgatav.

2. Ainult ikooniga nuppudele (üleslaadimine ja allalaadimine) võiks olla lisatud ka tekstiline *tooltip*.
3. Võiks olla nupp, mis kustutab kogu automaadi korraga.
4. Võiks lisada *Undo* nupu.
5. Üleminekutele võiks saada lisada ka tähti ä, ö, ü, õ.

6.2 Implementeeritud muudatused

Testimise käigus saadud tagasiside põhjal valis antud lõputöö autor koheselt implementeeritavateks muudatusteks järgmised:

1. Lisati üleminekutel lubatavate tähtede hulka ä, ö, ü ja õ.
2. Vähendati joonistusala sisse- ja väljasuumimise kiirust.
3. Muudeti automaadi jooksutamisel aktiivsete olekute värvi.
4. Suurendati automaadi jooksutamise vaatest väljumiseks kasutatavat nuppu, et see oleks paremini märgatav.
5. Lisati automaadi alla- ja üleslaadimise nuppudele *tooltip*.
6. Lisati funktsionaalsus, mis teavitab kasutajat, kui DFA on juba minimaalne.

6.3 Tuleviku edasiarendused

Järgnevalt on välja toodud ettepanekud ja võimalused, kuidas tulevikus rakendust edasi arendada:

1. Lisada funktsionaalsus, mis võimaldab erinevate režiimide vahel liikuda ka klaviatuuri kasutades.
2. Lisada funktsionaalsus, mis võimaldab kustutamist ka *backspace* või *delete* klahvi vajutamisega.
3. Lisada funktsionaalsus, mis lisaks protsessi tulemusele kuvab kasutajale ka samme, kuidas algoritm selleni jõuab.
4. Lisada funktsionaalsus, mis võimaldab kasutajal näha igal sammul ka eelmise sammu aktiivsed olekud, et oleks paremini jälgitav, kuidas automaat sisendit töötleb.
5. Lisada nupp, mis kustutab kogu automaadi korraga.
6. Lisada *Undo* nupp.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua veebipõhine automaatide õppevahend ainele "Automaadid, keeled ja translaatorid", mis võimaldaks lõplike automaatide joonistamist ja käivitamist. Lisaks oli eesmärgiks luua võimalus loodud automaate alla ja üles laadida. Peamiseks eesmärgiks oli lisada uus funktsionaalsus, mis võimaldaks kasutajal praktikumide ülesannete raames loodud automaate koheselt avalike testidega testida. Veel oli eesmärgiks luua automaatide normaliseerimise funktsionaalsus ja mittedeterministlikke lõplike automaatide teisendamine deterministlikeks lõplikeks automaatideks. Viimaseks eesmärgiks oli parandada rakenduse kasutajakogemust.

Töös tuvustati üldisemalt automaatide simulaatoreid ja nende kasutamist õppetöös. Toodi välja peamised probleemid JFLAP rakenduses. Toodi välja loodava rakenduse nõuded.

Väljatoodud nõuete põhjal loodi uus veebipõhine automaatide õppevahend. Tutvustati selleks kasutatud tehnoloogiad. Lisaks viidi läbi kasutatavuse testimine ning täiustati tagasiside põhjal veebipõhist õppevahendit ja toodi välja võimalused tuleviku edasiarendusteks.

Viidatud kirjandus

[1] AKIT. Andmekaitse ja infoturbe portaal. https://akit.cyber.ee/term/1395.

[2] Javatpoint. (DFA) Deterministic finite automata. https://www.javatpoint.com/deterministic-finite-automata (13.05.2024).

[3] Javatpoint. (NFA) Non-Deterministic finite automata. https://www.javatpoint.com/non-deterministic-finite-automata (13.05.2024).

[4] AKIT. Andmekaitse ja infoturbe portaal. https://akit.cyber.ee/term/12034-turing-machine.

[5] Javatpoint. Minimization of DFA. https://www.javatpoint.com/minimization-of-dfa (14.05.2024).

[6] JFLAP. https://www.jflap.org/ (12.12.2023).

[7] Chakraborty, Pinaki, Prem Chandra Saxena, Chittaranjan Padmanabha Katti (2011). Fifty years of automata simulation: a review. *Inroads*, 2, 59-70. https://doi.org/10.1145/2038876.2038893 (14.05.2024).

[8] Chakraborty, Pinaki, Prem Chandra Saxena, Chittaranjan Padmanabha Katti (2012). Automata simulators: Classic tools for computer science education. *British Journal of Educational Technology*, 43, 11-13. https://doi.org/10.1111/j.1467-8535.2011.01243.x (13.12.2023).

[9] Singh, Tuhina, Simra Afreen, Pinaki Chakraborty, Rashmi Raj, Savita Yadav, Dipika Jain (2019). Automata Simulator: A mobile app to teach theory of computation. *Computer Applications in Engineering Education*, 27, 1064-1072. https://doi.org/10.1002/cae.22135 (13.12.2023).

[10] Hamada, Mohamed (2013). Turing Machine and Automata Simulators. *Procedia Computer Science*, 18, 1466-1474. https://doi.org/10.1016/j.procs.2013.05.314 (13.12.2023).

[11] e-Teatmik: IT ja sidetehnika seletav sõnaraamat. http://www.vallaste.ee/.

[12] MDN Web Docs. JavaScript. https://developer.mozilla.org/en-US/docs/Web/JavaScript (9.05.2024).

[13] React. https://react.dev/ (9.05.2024).

[14] vis.js. https://visjs.org/ (9.05.2024).

[15] vis.js. Network. https://visjs.github.io/vis-network/docs/network/ (10.05.2024).

[16] i18next. Introduction. https://www.i18next.com/ (10.05.2024).

[17] Font Awesome Docs. Set Up with React. https://docs.fontawesome.com/web/use-with/react/ (10.05.2024).

[18] Net-Informatics. Java tutorial. https://net-informations.com/java/default.htm (13.05.2024).

[19] Spring. Why Spring?. https://spring.io/why-spring (13.05.2024).

[20] Maple, Simon, Andrew Binstock (2018). JVM Ecosystem report 2018 - About your Platform and Application. https://snyk.io/blog/jvm-ecosystem-report-2018-platform-application/ (13.05.2024).

[21] dk.brics.automaton. https://www.brics.dk/automaton/ (13.05.2024).

[22] ExperienceUx. What is usability testing?. https://www.experienceux.co.uk/faqs/what-isusability-testing/ (14.05.2024).

Lisad

I. Rakenduse lähtekood

Õppevahendi *frontendi* lähtekood asub lingil https://github.com/arturhendrik/automata_frontend ja *backendi* lähtekood lingil https://github.com/arturhendrik/automata_backend

II. Üleslaetava automaadifaili sisu näidis

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<structure>
  <type>fa</type>
  <automaton>
    <!--The list of states.-->
    <state id="0" name="q0">
      <x>275</x>
      <y>272</y>
      <initial/>
    </state>
    <state id="1" name="q1">
      <x>519</x>
      <y>266</y>
      <final/>
    </state>
    <!--The list of transitions.-->
    <transition>
      <from>0</from>
      <to>1</to>
      <read>a</read>
    </transition>
    <transition>
      <from>1</from>
      <to>0</to>
      <read>b</read>
    </transition>
  </automaton>
</structure>
```

III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Artur Hendrik Mägi**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Veebipõhise automaatide õppevahendi loomine ainele "Automaadid, keeled ja translaatorid"**, mille juhendaja on **Vesal Vojdani**, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Artur Hendrik Mägi

15.05.2024