



TEKNILLINEN TIEDEKUNTA

AUTOMAATTISESTI TARKASTETTAVAT TEHTÄVÄT

Taneli Lohi

KONETEKNIikka

Kandidaatintyö

Tammikuu 2022

TIIVISTELMÄ

Automaattisesti tarkastettavat tehtävät

Taneli Lohi

Oulun yliopisto, Konetekniikan tutkinto-ohjelma

Kandidaatintyö 2022, 27 s.

Työn ohjaaja yliopistolla: Toni Liedes

Tässä kandidaatintyössä perehdytään automaattisesti tarkistettavien tehtävien muodostamisen teoriataustaan, sekä luodaan esimerkkitehtäviä Koneiden tietotekniikka - opintojaksolle. Pääasiallisessa käsittelyssä ovat monivalintatehtävät ja niiden luonti, mutta myös avoimien tehtävätyyppien tarkistaminen ja tehtävien luonti on tarkastelussa. Työssä sivutaan myös koneoppimisalgoritmien avulla tehtävää tehtävien tarkistamista.

Helppoin tehtävätyppi tarkistaa automaattisesti on monivalintatehtävät ja niiden variantit, koska niissä ei ole tulkinnan varaa vastauksen oikeellisuuden suhteen. Tästä syystä monivalintatehtävät ovat todella suosittukysymystyyppi. Haasteena monivalintatehtävissä on kuitenkin uskottavien väärien vastausten keksiminen. Huonojen monivalintatehtävien takia monivalintatehtävillä on huono maine, mutta kirjallisuudessa on osoitettu, että tästä huolimatta hyvin laaditulla monivalintatestillä voidaan tehokkaasti mitata osaamista.

Avoimien kysymysten automaattitarkastaminen on vaikeampaa ja viriheherkempää, mutta oikeanlaisella suunnittelulla ja tehtävänannolla on tietyn tyyppisten avoimien kysymysten tarkistaminen mahdollista. Helppointen tarkistettavia avoimia tehtäviä ovat tehtävät, joissa vastauksena annetaan jokin yksi luku, sana tai matemaattinen kaava. Näissäkin tulee varautua joko ohjeistamaan tarkasti halutun vastauksen muoto, sekä ottamaan huomioon synonyymien tai erilaisten esitysmuotojen mahdollisuudet. Esseetyyppisten tehtävien tarkistaminen automaattisesti nykyisellä teknologialla on haastavaa.

Asiasanat: automaattisesti tarkastettavat tehtävät, monivalintatehtävät, avoimet kysymykset

ABSTRACT

Automatically assessing exercises

Taneli Lohi

University of Oulu, Degree Programme of Mechanical Engineering

Bachelor's thesis + 2022, 27 pp.

Supervisor at the university: Toni Liedes

This bachelor's thesis is review for creating automatically assessing exercise. In the second part I made automatically assessable exercises for Information technology and machines course. Multi-choice questions are primarily under discussion, but I also review constructed response questions and their creating and assessing. This thesis briefly also covers usage of machine learning as tool in assessing.

The easiest type of automatically assessing questions are multiple-choice questions because those have no room for interpretation when assessed. Therefore, multiple-choice questions are very widely used type of question in mass tests. Main difficulty of creating multiple-choice questions is creation of believable distractors. Because world is full of badly constructed multiple-choice questions, this question type has got some bad reputation. However, in literature is proved that well done multiple-choice questions is efficient method of test knowledge.

Assessing open questions (constructed response questions) is more difficult and it has higher failure rate. Nevertheless, with proper design and well-defined question definition those can also be automatically assessed. Easiest type of open question in viewpoint of automatically assessment is simple question that requires simple answer like number, word, or mathematical equation. While designing this type of question synonyms, different typecases and different styles to present same answer must be took into consideration. With current technology assessing essay type of questions is very challenging.

Keywords: automatically assessing, multiple response question, MRQ, constructed response question, CRQ, developing exercises

SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

SISÄLLYSLUETTELO

1 Johdanto	7
2 Kirjallisuuskatsaus	8
2.1 Automaattisesti tarkistettavat tehtävät	9
2.2 Tehtävätyypit.....	10
2.2.1 Monivalintatehtävät	10
2.2.2 Avoin kysymys	14
3 Tehtävien muodostaminen	16
4 Yhteenveto	25

LÄHDELUETTELO

1 JOHDANTO

Tämän kandidaatin tutkielman tarkoituksena on selvittää kirjallisuudesta, kuinka automaattisesti tarkastettavaksi sopivia tehtäviä tulisi muodostaa, sekä tapausesimerkkinä muodostaa koneiden tietotekniikka -opintojaksolle tehtäviä, jotka sopivat automaattisesti tarkastettaviksi.

Opiskelijoiden osaamisen testaaminen toimii perustana kurssien ja opintojaksojen arvioinnissa. Sen avulla voidaan määrittää, onko opiskelija saavuttanut asetetut osaamistavoitteet, ja voiko näin opiskelijalle antaa suoritusmerkinnän. Korkeakouluissa kursseilla saattaa olla satoja opiskelijoita, jolloin opetushenkilökunnan työtaakka on valtava, mikäli kurssilla on paljon manuaalisesti tarkistettavia suorituksia, kuten tenttejä, harjoitustöitä, laskuharjoituksia tai oppimispäiväkirjoja. Kurssin osallistujamäärän kasvattamisen esteenä saattaakin olla lisääntyneen työmäärän takia syntyvä resurssipula.

Ratkaisu tähän on automatisoida tehtävien tarkistamista, jolloin osallistujamäärää voidaan kasvattaa tai opetushenkilökunnalle jää enemmän resursseja kurssin kehittämiseen. Kaikenlaisia tehtäviä ei kuitenkaan voida automatisoida helposti, sillä tietokone ei osaa tulkita vastauksista mitä vastaaja tarkoitti, vaan vastauksen tulee olla yksiselitteinen.

Työssä käsitellään monivalintatehtävien ja niiden johdannaisten luomista, sekä avoimia kysymyksiä ja niiden automattitarkastettavuutta. Tapausesimerkkinä muodostetaan Koneiden tietotekniikka -opintojaksolle tehtäviä, sekä perustellaan, miksi tehtävistä on tehty sellaisia, kun ne ovat.

2 KIRJALLISUUSKATSAUS

Tehtävä muodostuu seuraavista osista:

- Johdanto johdattelee tehtävän tekijän aihepiiriin pariin
- Kysymys tai tehtävänanto kertoo mitä tehtävän tekijän odotetaan tekevän
- Pisteytettävä osio, jonka avulla määritellään, miten tehtävästä suoriuduttiin (Haladyna 2004)

Tehtävää luodessa tulee ensiksi määritellä tehtävän tarkoitus. Halutaanko mitata osaamista, toimia oppimisen tukena vai molempia?

Tehtävät harvemmin esiintyvät yksin, vaan niistä yleensä kootaan useamman tehtävän kokonaisuus eli testi. Testin tarkoitus voi olla sama kuin yksittäisen tehtävänkin, eli testata tai opettaa jotain asiaa. Yksittäisellä tehtävällä voidaan yleensä mitata vain muutaman asian osaamista, mutta hyvin laaditulla testillä voidaan testata laajempaa kokonaisuutta. Jokaisella tehtävällä on oma tarkoituksensa testissä: mitata tai opettaa jotain tarkasti määriteltyä asiaa. Suurempaa kokonaisuutta käsiteltäessä tehtävät lähestyvät aihetta eri näkökulmista ja keskittyen eri yksityiskohtiin.

Insinööritieteiden alalla tehtävillä on usein omat erityispiirteensä. Erityisesti luonnontieteitä soveltavilla opintojaksoilla halutaan mitata konseptien osaamista ja soveltamistaitoa. Konseptit saattavat olla monimutkaisia, mutta ne eivät kuitenkaan ole yleensä monitulkintaisia ja niiden soveltamisen osaamista voidaan mitata laskutehtävillä, joissa on vain yksi oikea vastaus; tehtävän ratkaisu. Tällaiset tehtävät soveltuvat hyvin automaattitarkistettaviksi, koska lopputuloksessa ei tulisi olla juurikaan tulkinnan varaa.

Mikäli kysytään vain laskutehtävän lopputulosta, tehtävä mittaa vain laskun kokonaan suorittamista oikeellisesti, jolloin osittaisen osaamisen osoittaminen on vaikeaa. Tämä voidaan ratkaista luomalla samasta laskutehtävästä useampi tehtävä, joissa kysytään välivaiheita, -tulosta tai käytettyjä menetelmiä ja periaatteita.

2.1 Automaattisesti tarkistettavat tehtävät

Opintojaksolla opitun asian soveltaminen on tärkeä askel siirryttäessä ymmärtää-osaamistasolta soveltaa-osaamistasolle. Moniin insinöörialan kursseihin sisältyy laskuharjoituksia tai viikkotehtäviä, joilla pyritään syventämään opiskelijan ymmärtämistä ja korottamaan osaamistasoa. Jotta opiskelija saa mahdollisimman paljon irti laskuharjoituksista tai viikkotehtävistä, tulee hänen saada palaute suorituksesta. Palautteeksi riittää usein tieto siitä menikö tehtävä oikein, mutta mieluummin olisi annettava syvällisempää palautetta tehtävän suorittamisesta. Jos opiskelija tekee virheen, joka johtaa väärään tulokseen, on opiskelijan oppimisen kannalta parempi, jos hän saa tiedon tai vihjeen mikä meni väärin. Mikäli tehtävät arvostellaan manuaalisesti, tämänkaltaisia vihjeitä on helppo antaa. Manuaalinen arvostelu vaatii kuitenkin paljon työtä kurssin opettajalta ja siihen käytetty aika on poissa opintojakson kehittämisestä ja muusta työstä. Automaattisesti tarkistettavat tehtävät keventävät opintojakson järjestäjien työtaakkaa ja mahdollistavat kurssin opiskelijamäärien kasvattamisen ilman työtaakan huomattavaa kasvua (Sychev et al. 2020).

Automaattisesti tarkistettavia tehtäviä on kahdentyypisiä: suljettuja ja avoimia tehtäviä. Suljetuissa tehtävissä opiskelija valitsee tehtävänannon perusteella annetuista vaihtoehdoista oikean vastauksen, eli ne ovat joko monivalintatehtäviä tai niiden johdannaisia. Avoimissa tehtävissä opiskelija tuottaa pyydetyn vastauksen annetun tehtävänannon perusteella. Suljetut tehtävät on helppo tarkastaa automaattisesti, koska opiskelijan vastaus on ennalta määrätty ja näin yksiselitteisesti oikein tai väärin. Avointen tehtävien tarkastaminen automaattisesti on vaikeampaa, koska opiskelijan itse tuottaessa vastauksessa voidaan käyttää synonyymejä ja sama asia voidaan esittää eri tavoin. Tietokone ei myöskään osaa automaattisesti korjata kirjoitusvirheitä eikä tulkita mitä opiskelija vastauksellaan tarkoittaa. Avointen tehtävien tehtävänannossa on tärkeää määritellä miten vastaus tulee antaa. Helpoimpia arvosteltavia ovat yksittäiset sanat tai lukuarvot, joista tehtävän laatija voi kirjata oikeiksi vaihtoehdoiksi synonyymit ja variaatiot. Monimutkaisempia avoimia kysymyksiä voidaan tarkistaa käyttämällä koneoppimis- tai tekoälyalgoritmeja, mutta näiden käyttöönotto ja juuri kyseiseen tehtävän tarkistamisen opettaminen on toistaiseksi erittäin työlästä.

2.2 Tehtävätyypit

Kuten aiemmin todettiin, automaattisesti tarkistettavat tehtävät voidaan jakaa karkeasti avoimiin ja suljettuihin tehtäviin. Yleisimmin käytetyt suljetut tehtävät ovat monivalintatehtäviä ja niiden johdannaisia. Näiden tarkistamisen helppous on tehnyt niistä suosittuja automaattisesti tarkistettavissa testeissä.

Avoimien kysymysten tarkistaminen automaattisesti on vaikeampaa, mikä aiheuttaa rajoitteita kysymysten tyyppille. Avoimista kysymyksistä helpoiten tarkastettavia ovat tehtävät, joiden vastaus pelkistyy yksittäiseen sanaan tai lukuarvoon.

2.2.1 Monivalintatehtävät

Monivalintatehtävät on yleisesti käytetty tehtävätyyppi, joka sopii erinomaisesti automaattisesti tarkistettavaksi. Monivalintatehtävät taipuvat monenlaiseen käyttötarkoitukseen. Niillä voidaan mitata tietoa, ymmärtämistä, ongelmanratkaisukykyä ja monia muita oppimiseen liittyviä tietoja ja taitoja (Ebel ja Frisbie 1991). Haladyna (2004) toteaa luovuuden olevan yksi harvoista asioista, jota ei voida mitata monivalintakysymyksillä.

Bjork et al. (2015) mukaan monivalintatehtävät ovat kaikkialla läsnä, mikä kertoo hyvin niiden yleisyydestä opitun testaamisessa. Monivalintatehtäviä käytetään työkaluna moniin eri tarkoituksiin. Esimerkiksi pääsykokeissa ja opiskelijavalinnoissa, sertifikaattien ja pätevyysien myöntämisen työkaluna, arvosanojen ja kurssisuoritusten lähteenä, mitattaessa mitä ei ole vielä opittu, sekä myös työelämässä erilaisten kyselyjen muodossa (Haladyna 2004). Monivalintatehtävät ovat helppo ja vähän resursseja vaativa tapa toteuttaa suuren osallistujamäärän testaaminen. Monivalintatehtäviä voidaan käyttää testaamisen lisäksi oppimista edistävän välineenä, jolloin tarkoituksena ei ole testata osaamista, vaan johdattaa tehtävien tekijä aiheen äärelle ja opettaa jotain aiheesta.

Monivalintatehtävät muodostuvat kysymysrungosta ja vaihtoehdoista, joista testattava valitsee annetun ohjeistuksen mukaan joko yhden tai useamman. Riippuen kysymyksestä ja vaihtoehdoista voidaan luoda erityyppisiä monivalintakysymyksiä, joilla on omat heikkoudet ja vahvuudet. Monivalintakysymyksille on kuitenkin tyypillistä se, että vastaus tai vastaukset löytyvät suoraan tehtävästä. Testin tekijän tulee vain poimia oikea

tai oikeat vastaukset annetuista vaihtoehdoista. Monivalintatehtävissä on aina myös oikean tai oikeiden vastausten lisäksi yksi tai useampi väärä vastaus (Haladyna 2004)

Vaikeutena monivalintatehtävien luomisessa on muodostaa uskottavalta näyttäviä vääriä vastauksia. Väärien vastausten tulee olla sellaisia, että henkilö, jolla ei ole tietoa kysytystä aiheista, ei voi erottaa oikeaa vastausta vääristä, mutta mikäli tietoa on, vastaukset ovat selvästi vääriä. Vastausten tulee siis olla selkeästi vääriä, mutta muistuttaa oikeata vastausta ulkoasun, kieliopin ja tyylin osalta. (Haladyna 2004)

Tutkimusten mukaan (Vegada et al. 2016, Tarrant et al. 2010) monivalintatehtävissä vaihtoehtojen määrän tulisi olla 3 (yksi oikea ja kaksi väärää), sillä vaihtoehtojen määrän kasvattaminen ei kasvata testin luotettavuutta, mutta lisää testin laatimiseen vaadittavaa työpanosta sekä pidentää testin tekemiseen tarvittavaa aikaa. Tutkimusten mukaan yli kolmen vaihtoehdon kysymyksissä on yleensä yksi tai useampi heikko vastaus, jotka testin tekijöillä on helppo sivuuttaa varmasti vääränä vaihtoehtona, jolloin kyseisestä vaihtoehdosta ei ole mitään hyötyä, vaan kysymys toimisi identtisellä tavalla, vaikka vaihtoehto jätettäisiin pois.

Monivalintatehtävistä on olemassa paljon erilaisia variaatioita. Kysymystyyppin valintaan vaikuttaa se mitä halutaan mitata. Jokaisella kysymystyyppillä on omat heikkoutensa ja vahvuutensa. Kysymystyyppistä ja kysymyksen laatimistavasta riippuen voidaan mitata pinnallisen osaamisen ja muistamisen lisäksi myös syvempää ymmärtämistä ja kokonaisuuksien hallintaa, vaikkakin sellaisten testien tuottaminen on vaikeampaa kuin triviaalien asioiden muistamista mittavien testien (Frederiksen 1984).

Perinteistä monivalintatehtävää on kolmea tyyppiä:

- Kysymys–vastaus (esimerkki 2.2.1)
 - o Runkona on kysymys, johon valitaan vaihtoehdoista vastaus
- Aukkotehtävä (esimerkki 2.2.2)
 - o Runkona lause, joka jää kesken, tai siinä on aukko keskellä sanaa.
 - o Vaihtoehdoista valitaan lauseen järkeväksi täydentävä
- Paras vastaus
 - o Runkona on kysymys, johon valitaan vaihtoehdoista oikein vastaus.

(Haladyna 2004)

Statmanin (1988) mukaan aukkotehtävissä lauseen alussa ja keskellä olevaa aukkoa ei suositella käytettäväksi, koska opiskelijan tulee käydä koko lausetta läpi palasina, testaten vaihtoehtoa tyhjään paikkaan. Stressaavat testiolut voivat aiheuttaa ongelmia lyhytkestoisen muistin kanssa, mikä aiheuttaa mahdollisesti virheen tehtävän ratkaisussa. Tällaisen aukkotehtävän ratkaisemiseen menee myös paljon aikaa verrattuna samaa asiaa testaaviin erityyppisiin kysymyksiin.

Missä CPU:n osassa laskutoimitukset tapahtuvat?

- a) RAM
- b) ALU (*)
- c) ACC

Esimerkki 2.2.1 kysymys–vastaus monivalintatehtävä

Rekisterin R2 sisältö saadaan nollattua suorittamalla XOR operaatio ____ ja tallentamalla tulos rekisteriin R2.

- a) itsensä rekisteri R2 kanssa
- b) nollatavun kanssa
- c) ykköstavun kanssa

Esimerkki 2.2.2 aukkotehtävä

Yhdistäminen (esimerkki 2.2.3) on tehtävätyyppi, jossa tehtävänannon väittämät tulee yhdistää vaihtoehtojen väittämiin tai kuvauksiin niin, että yhdistetyt väittämät ja vaihtoehdot muodostavat totta olevan lauseen tai vaihtoehto muuten kuvaa väittämää. Tyypillisesti vaihtoehtojen väittämiä tai kuvauksia on yksi tai useampi enemmän kuin tehtävänannon kysymyksiä, jolloin yksi vastausvaihtoehto jää ylimääräiseksi. Vaihtoehtojen tulee olla samankaltaisia, jotta asiaa ymmärtämätön ei voi sulkea pois osia vaihtoehdoista vain ulkoasun perusteella. (Haladyna 2004) Tällainen tehtävätyyppi sopii moniin eri tarkoituksiin, esimerkiksi voidaan opetella vieraan kielen sanoja, testata koneen osien ja niiden toiminnallisuuksien ymmärtämistä. Yhdistämistehtävä sopii myös moniin muihin aihepiireihin, missä kahden asian välillä on yhteys.

Haladynan mukaan (2004) yhdistämisformaatin etuja ovat tehtävien helppo luonti, koska vääriä vastauksia ei tarvitse luoda. Samaan tehtävään voidaan sisällyttää useiden perinteisen monivalintatehtävän oikeat vastaukset, ja testattavalla kuluu vähemmän aikaa tehtävän tekemiseen kuin samansisältöisten usean perinteisen monivalintatehtävän tekemiseen.

1. Huolehtii käskyn suorittamisen eri vaiheista (Stepperi)
2. Tahdistaa käskyjen suorittamisen (Kello)
3. Suorittaa matemaattisloogiset operaatiot (ALU)
4. Toimii käskyjen ja datan säilytyspaikkana (RAM)
5. Säilytetään tietoa tilapäisesti (Rekisteri)

- a) MAR
- b) kello
- c) rekisteri
- d) stepperi
- e) ALU
- f) RAM

Esimerkki 2.2.3 yhdistäminen

Vaihtoehto-kysymystyyppi (Alternative Choice, AC) on monivalintatehtävä, jossa on vain kaksi vastausvaihtoehtoa, joista toinen on oikein ja toinen väärin. Haladyna painottaa kirjassaan (2004), että vaikka AC-kysymystyyppissä on vain kaksi vaihtoehtoa, se ei ole oikein-väärin (true-false, TF) -kysymys, vaan vastaajan pitää vertailla vaihtoehtoja, mitä TF-kysymyksissä ei tarvitse tehdä. AC-kysymyksiä on helppo tehdä, koska niihin tarvitsee keksiä oikean vastauksen lisäksi vain yksi hyvä harhauttaja. Ebel (1982) esittää AC-kysymystyyppin olevan tavallista monivalintatehtävää parempi, koska AC-tehtävän tekemiseen kuluu vähemmän aikaa, joten samassa ajassa voidaan kysyä useampia kysymyksiä, mikä taas nostaa testin luotettavuutta. AC-kysymykset ja niiden vastaukset ovat yleensä tavallista monivalintaa lyhyempiä, joten kysymykset keskittävät testattavan huomion paremmin testattavaan asiaan. (Ebel 1982) Heikkoutena AC-formaatissa on arvaamisen vaikutus testaukseen. Todennäköisyys vastata oikein puhtaasti arvaamalla on 50 %. Mikäli tätä kysymystyyppiä käytetään, testin suositeltava läpikäyminen on yli 50 %. (Haladyna 2004)

Oikein-väärin (True-False, TF) -kysymystyyppissä esitetään väittämä, johon vastataan valitsemalla, onko se oikea vai väärä. TF-kysymysformaatin käyttö jakaa tutkijoiden mielipiteitä. Haladyna (2004) suosittelee käyttämään TF-formaattia harkiten tai ei ollenkaan, koska kuten muillakin kahden vaihtoehdon kysymyksillä, niillä on liian paljon heikkouksia. Yleisimmäksi ongelmaksi Haladyna mainitsee TF- kysymysten yleensä mittaavan yleensä triviaalisen tiedon muistamista, eikä syvempää ymmärtämistä. Ebel ja Frisbie (1991) ovat kuitenkin eri mieltä. Heidän mukaansa oikein käytettynä TF-formaatilla voidaan mitata ymmärtämistä yhtä hyvin kuin tavanomaisella monivalintatehtävälläkin.

2.2.2 Avoin kysymys

Avoin kysymys (Constructed response question, CR,) on kysymystyyppi, jossa tehtävän tekijän tulee itse tuottaa vastaus annetun tehtävänannon perusteella. Avoin kysymys -tehtävät koostuvat tehtävänannosta sekä tilasta mihin vastaaminen tehdään. Tehtävänanto voi olla puhtaasti tekstimuodossa ja hyvin samankaltainen kuin monivalintatehtävissäkin, jolloin tehtävä on kuin monivalintakysymys, missä ei ole vaihtoehtoja vaan testin tekijän tulee itse kirjoittaa oikea vastaus. Toisaalta avoin kysymys -tehtävä voi olla nuolen lisääminen kuvaajaan haluttuun kohtaan, tai esseiden kirjoitus, matemaattisen tehtävän ratkaisu, kuvaajan piirtäminen. Jopa musiikkiesityksestä suoriutuminen, taideportfoliot ja vieraan kielen sujuvuuden arviointi voidaan sisällyttää avoimiin kysymyksiin. (Lukhele et al. 1994)

Insinöörityeiden ollessa kyseessä, on syytä rajata taiteelliset tehtävyyt pit pois ja keskittyä lyhyesti vastattavaan, esseetyyppiseen ja matemaattisluonnontieteellisten ongelmien ratkaisuun. Wainerin ja Thissenin tutkimuksessa (1993) tuli esille, että samantasoisesta avoimista kysymyksistä koostuvan testin luominen vaatii 4–40 kertaa enemmän aikaa sekä arvioiminen maksaa jopa tuhansia kertoja enemmän kuin vastaavan monivalintatehtävistä koostuvan testin manuaalinen arvioiminen.

Avointen tehtävien ajatellaan kuitenkin kuvaavan paremmin akateemisen tutkimuksen ja työelämän vaatimuksia ja ne muistuttavat niissä tehtävää työtä. Tämän takia niiden ajatellaan olevan arvokkaita opetuksessa ja testaamisessa työelämästä huolimatta. (Lukhele et al. 1994)

Avoin kysymys -tehtävyyppi on niin laaja, ettei niiden automaattitarkistettavuuteen kokonaisuutena voi ottaa kantaa. Seuraavaksi käsitellään kuitenkin erilaisia alatyyppejä automaattitarkistettavuuden näkökulmasta.

Yksinkertaisten kysymysten, joihin on vain yksi oikea vastaus, tarkistaminen voidaan automatisoida kohtalaisen helposti tunnistamalla säännöllisten lausekkeiden (regular expression, regex) tai suoraan merkkijonojen vastaavuuden avulla vastauksen oikeellisuus. Vastausten ollessa numeerisia, tulee tehtävänannossa kertoa haluttu tarkkuus ja yksikkö, sillä muuten tehtävän laatijan tulisi merkitä oikeiksi vastauksiksi todella suuri määrä erilaisia esitysmuotoja. Vastausten ollessa kirjallisia, yksittäisen sanan tai yksinkertaisen matemaattisen lausekkeen tarkistaminen on vielä kohtalaisen

helppoa, sillä vastausvaihtoehdoissa tulee ottaa huomioon vain isot alkukirjaimet ja mahdollisesti kirjoitusvirheet sekä matemaattisissa lausekkeissa erilaiset esitystavat.

Avoimia kysymyksiä voidaan arvioida perinteisten menetelmien sijaan myös tekoälysovelluksilla. LaVoie et al. (2019) tutkivat Latent Semantic Analysis (LSA) koneoppimisalgoritmia seuraustestin arvioimiseen niin, että samat kysymykset arvioi LSA-algoritmi ja treenattu ihmisarvostelija. Testissä pyydettiin osallistujia listaamaan epätavallisen tilanteen seuraukset. Kysymyksen asettelu oli ”Mitä saattaisi olla seurauksena jos...”. Algoritmin opetusdatana käytettiin 420 osallistujan testejä, jotka harjaantuneet ihmiset olivat arvioineet. Varsinaiseen kokeeseen osallistui 1443 testin tekijää, joista jokainen vastasi viiteen kysymykseen. Tutkimuksen tuloksena LSA-algoritmin ja ihmisen arvioiman testituloksen korrelaatio oli 0,94, kun kahden ihmisarvioijan välillä korrelaatio oli 0,98, eli algoritmilla saatiin lähes yhtä hyviä tuloksia kuin ihmistyövoimalla. Ongelmana koneoppimisalgoritmin käytössä voi kuitenkin olla riittävän suuren määrän käyttötarkoitukseen soveltuvan opetusdatan hankkiminen. (LaVoie et al. 2019)

3 TEHTÄVIEN MUODOSTAMINEN

Käytännön osuudessa luodaan teoriataustan pohjalta tehtäviä Koneiden tietotekniikka -opintojaksolle. Tehtävät käsittelevät 8-bittisen tietokoneen rakennuspalikoita, toimintalogiikkaa ja ohjelmointia. Tarkoituksena on, että tehtävät voidaan tarvittaessa siirtää pienellä vaivalla Koneiden tietotekniikka -opintojakson Moodle-työtilaan automaattisesti tarkistettaviksi tehtäviksi, joilla voidaan testata opiskelijoiden osaamista ja/tai tukea oppimista.

Seuraavan sivun tehtävien 1–3 tarkoitus on opettaa testin suorittajalle, millaisia loogisia portteja on olemassa ja miten niiden toimintalogiikka voidaan kuvata totuustaulun avulla, sekä testata osaako testin tekijä lukea totuustauluja. Tarkoituksena on myös esitellä erilaisia monivalinta-tehtävätyyppejä esimerkkien avulla.

Ensimmäinen kysymys (tehtävä 1) on perinteinen monivalintatehtävä. Vastatakseen tähän kysymykseen testin tekijän tulee tutkia annettua AND-operaation totuustaulua ja valita sen pohjalta vaihtoehto. Kysymys toimii harjoituskysymyksenä, sillä kysymyksen avulla testin tekijä tutustuu AND-operaatioon ja mahdollisesti osaa muodostaa yhteyden englannin kielisen AND ja suomen kielen ”ja” välille. Osaamista testaavana kysymyksenä se on ongelmallinen, koska oikean vastauksen voi lukea totuustaulusta (ellei tarkoituksena ole nimenomaan testata totuustaulun lukemista), sekä koska testin tekijä voi kysymysrungossa olevan englanninkielisen AND sanan avulla sulkea pois a-vaihtoehdon, koska siinä esiintyy ”tai” sana. Mikäli tätä kysymystä aikoisi käyttää osaamista mittaavana tehtävänä kannattaisi a-vaihtoehto tiputtaa pois, jolloin formaatti muuttuisi vaihtoehtoiseksi valinnaksi. Esimerkistä huomataan, että sama kysymys voi olla joko hyvä tai huono, käyttötarkoituksesta riippuen.

Toinen kysymys on oikein–väärin-kysymys, jossa tutustutaan NAND-operaatioon ja erityisesti siihen tapaukseen, kun ulostulona on nolla. Tehtävän tarkoituksena on testin tekijän tutustuttaminen NAND-operaatioon. Tehtävä voitaisiin toteuttaa myös vaihtoehtoisena valintana, jolloin oikein ja väärin vaihtoehtojen tilalle laitettaisiin kysymysrungon viimeinen termi 1 ja toiseksi vaihtoehdoksi 0. Tämä ei kuitenkaan toisi mitään lisäarvoa tehtävään.

Kolmas tehtävä on yhdistämistehtävä. Tällä formaatilla voidaan kysyä helposti useita kohtalaisen yksinkertaisia asioita. Formaattilla saadaan pakattua monta kysymystä pieneen tilaan ja vääriä vastauksia ei tarvitse keksiä. Kun osaan kysymyksistä on vastattu, jäljellä olevien vaihtoehtojen määrä pienenee. Jotta testin tekijä ei saisi viimeistä vastausta poissulkemalla muihin kysymyksiin sopivat vastaukset, on vaihtoehtoja yksi enemmän kuin kysymyksiä.

Perinteinen monivalintatehtävä:

Tutki annettua AND-operaation totuustaulua ja valitse oikea vaihtoehto.

1. AND-operaatiossa C saa arvon 1, kun

- A tai B saavat arvon 1.
- A ja B saavat molemmat arvon 1.*
- A ja B saavat arvon 0.

Vaihtoehtoinen valinta:

Tutki NAND-operaation totuustaulua ja täydennä lause valitsemalla oikea vaihtoehto:

2. NAND-operaatiossa C saa arvon 0, kun A ja B saavat arvon 1

- Oikein *
- Väärin.

AND

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

NAND

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

OR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

XOR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

NOT

A	C
0	1
1	0

Yhdistäminen:

3. Yhdistä looginen portti ja sen kuvailu

a	b	c	d

- a. C saa arvon 1 **vain** kun A:n ja B:n arvot ovat erisuuria (XOR)
- b. On AND operaation käänteisoperaatio (NAND)
- c. Operaatio vaatii **vain** yhden sisääntulon (NOT)
- d. C saa arvon 1 kun joku sisääntulo saa arvon 1 (OR)

1. AND
2. NAND
3. OR
4. XOR
5. NOT

Tehtävät 4 ja 5 käsittelevät 8-bittisen tietokoneen rekistereitä ja miten niitä käytetään käskyn noutamiseen RAM-muistista.

Tehtävä 4 on yhdistelytehtävä, jossa tehtävänannossa kuvaillaan rekisterejä ja tehtävän tekijän tulee valita annetuista rekisterin nimistä kuvausta vastaava rekisteri. Lisäksi vaihtoehtoissa on yksi ylimääräinen rekisteri, joka ei sovi mihinkään kuvaukseen. Oikeat vastaukset ovat suluissa kysymyksen perässä.

Yhdistäminen-tehtävyyypin tehtäviä on helppo tehdä, koska monivalintatehtävän vaikein osa, eli uskottavan väärän vastauksen muodostamista ei tarvitse tehdä niin paljon kuin jos jokaisesta rekisteristä olisi tehnyt oman monivalintatehtävän. Vastausvaihtoehdot ovat kaikki lyhyitä kirjainyhdistelmiä, joten asiaa tuntemattoman on hankala arvata oikea ratkaisu.

Tehtävä 5 on avoimen kysymystyyppin (CR) aukkotehtävä, jossa annettuun tekstiin tulee täyttää CPU:n osat niin, että lauseista tulee järkeviä. Oikeat vastaukset on esitetty suluissa aukkopaiikkojen jälkeen. Avoin kysymys eroaa monivalinnasta sillä, että vaihtoehtoja ei anneta valmiina vaan testin tekijän tulee itse muodostaa vastaus. Mikäli tehtävä

muutettaisiin monivalintatehtäväksi antamalla vaihtoehdot ja yksi harhautus, tulisi siitä tehtävän 4 kaltainen yhdistämistehtävä.

Tekstissä käydään läpi käskyn noutaminen muistista vaiheittain. Testin tekijän tulee käydä läpi käskyn noutamisen logiikka ja CPU:n osien käyttötarkoitus käskyn noutamisessa, jotta pystyy vastaamaan tehtävään. Tehtävä ei vastaa täysin aukkotehtävän suunnittelun ohjeita (Haladyna 2004), koska aukkokohtia on lauseiden keskellä. Haladynan mukaan lauseen keskellä tai alussa oleva aukko paikka vaatii testin tekijältä enemmän työmuistia kuin lopussa oleva aukko paikka, mikä saattaa stressaavassa testitilanteessa johtaa helpommin epäonnistumiseen. Ohjeenmukaisia aukkotehtäviä on kuitenkin vaikea tehdä, mikäli haluaa täydennettävästä tekstikokonaisuudesta yhtenäisen.

4. Yhdistä kuvailu ja rekisteri

1. Rekisteriin säilötään ALU:n tekemän laskutoimituksen tulos odottamaan käyttöä. (ACC)
2. Rekisteristä löytyy seuraavan käskyn osoite. (IAR)
3. RAM-muistin rekisteri, joka säilöo annetun käskyn osoitteen. (MAR)
4. Rekisteri sisältää RAM-muistista noudetun käskyn. (IR)
5. Rekisteriin voidaan säilöä bitti odottamaan myöhempää käyttöä. (R0)
6. Rekisteri mahdollistaa kahden tavun operaation. (TMP)

a. IAR	1.	2.	3.	4.	5.	6.
b. R0						
c. ACC						
d. TMP						
e. BUS 1						
f. MAR						
g. IR						

5. Täytä käskyn noutamiseen muistista liittyvän tekstin aukkoihin oikeat CPU:n osat.

1. Enabloidaan ____ (IAR), jolloin seuraavan käskyn osoitetieto julkaistaan väylälle. Samalla enabloidaan ____ (BUS 1) ja koska ALU:n operaatiokoodi on 000 (ADD) lasketaan yhteenlasku, jossa edellisen kierroksen osoitetietoon lisätään 1 ja tallennetaan saatu seuraavan ohjelmakierroksen osoite asetettuun ____ (ACC)-rekisteriin. Myös ____ (MAR) rekisteri asetetaan, ja sinne tallennetaan meneillään olevan ohjelmakierroksen osoitetieto.
2. ____ (RAM) enabloidaan, jolloin muistista kopioituu osoitetietoa vastaava käsky väylälle. ____ (IR) asetetaan, jolloin käsky siirtyy väylältä kyseiseen rekisteriin.
3. ____ (ACC) enabloidaan ja ____ (IAR) asetetaan, jolloin uusi osoitetieto tallentuu ja prosessori on valmis noutamaan uuden käskyn.

Tehtävässä 6 on binäärisillä käskykoodeilla toteutettu ohjelma, sekä selite käskyille. Tehtävässä 6.1 tehtävän tekijän tulee muuntaa käskyt ihmiselle helpommin luettavaan koodikieliseen muotoon. Tehtävässä 6.2 ja 6.3 ohjelman toimintaa tulkitaan ja vastataan sen perusteella annettuihin kysymyksiin. 6.2 ja 6.3 on olennaista tunnistaa silmukkarakenne, minkä perusteella voidaan sanoa kuinka monta kertaa kysytyt asiat tapahtuvat. Tehtävän alla on 6.1 oikea vastaus. 6.2 ja 6.3 vastaukset löytyvät suluista kysymyksen perässä. Tehtävän alla on myös ohjelman toimintaa kuvaava pseudokoodi, mikä voidaan antaa testin tekijälle suorituksen jälkeen havainnollistamaan koodin toimintalogiikkaa, sekä miten sama toiminnallisuus voitaisiin toteuttaa koreamman tason ohjelmointikielellä.

Tehtävän tarkoitus on antaa esimerkki, miten yksinkertainen ohjelma voidaan toteuttaa käskyinä, saada tuntumaa, miten käskyjä voidaan käyttää yhdessä, sekä avata miten muistissa voidaan liikkua myös hyppien aiempiin muistiositteisiin vain eteenpäin menemisen sijaan. Ensimmäisessä tehtävässä tutustutaan ohjelman komponentteihin. Tämä osa ei ole kovinkaan haastava, vaan lähinnä mekaaninen käskyjen sijoittaminen taulukkoon ja niiden toiminnan kertaaminen. Toisessa- ja kolmannessa tehtävässä taas tulkitaan ohjelman toimintaa. Kolmanteen tehtävään on piilotettu ansa: ALU-käskyt alkavat aina bitillä 1, josta ne voidaan tunnistaa. Koodissa on myös numero 150 (1001 0110), mikä ei ole ALU-käsky vaan numero binääriformaattissa.

Koska käskyjen syntaksin ja niitä vastaavien binäärikoodien ulkoa opettelu ei ole tarkoituksenmukaista on suositeltavaa käyttää tämän tyyppistä tehtävää ympäristössä, missä materiaalin käyttö on sallittua tai kooditaulukko tulisi antaa tehtävän liitteenä.

6. Ohjelmakoodin tulkinta

RAM osoite	Käsky	Käsky koodikielisenä	Selitys
50	0010 0001	<i>DATA R1, 15</i>	Kirjoitetaan rekisteriin R1 seuraavassa muistiosoitteessa sijaitseva luku
51	1001 0110		150
52	1110 1010		Nollataan rekisterin R2 sisältö (XOR)
53	0010 0011		Kirjoitetaan rekisteriin R3 seuraavassa muistiosoitteessa sijaitseva luku
54	0000 0001		1
55	1000 1110		Lasketaan R3 ja R2 yhteen ja tallennetaan vastaus rekisteriin R2
56	0110 0000		Nollaa lippurekisteri
57	1111 0110		Vertaa R1 ja R2 toisinsa, tallenna tulos "A larger" tai "Equal" lippujen avulla lippurekisteriin
58	0101 0001		Jos "Equal" lippu on nostettuna hyppää seuraavassa muistiosoitteessa olevaan muistiosoitteeseen
59	0011 1110		62
60	0100 0000		Hyppää seuraavassa muistiosoitteessa olevaan muistiosoitteeseen
61	0011 0111		55
62		<i>Seuraava käsky...</i>	

6.1 Täydennä ohjelman käskyt taulukkoon koodikielellä annettujen selitysten perusteella muistiosoitteen 50 antaman mallin mukaisesti.

6.2 Kuinka monta kertaa lippurekisteri nollataan ohjelman ajon aikana? (150)

6.3 Kuinka monta ALU-käskyä suoritetaan ohjelman ajon aikana? (301)

```

50 0010 0001 DATA R1, 15
51 1001 0110 1001 0110
52 1110 1010 XOR R2, R2
53 0010 0011 DATA R3
54 0000 0001 0000 0001/1
55 1000 1110 ADD, R3, R2
56 0110 0000 CLF
57 1111 0110 CMP, R1, R2

58 0101 0001 JE

59 0011 1110 0011 1110/62
60 0100 0000 JMP
61 0011 0111 0011 0111/55

```

```

1 R1 = 150
2 R2 = 0
3 R3 = 1
4 while(true):
5     R2 = R2 + R3
6     if (R2 == R1)
7         break
8 ...

```

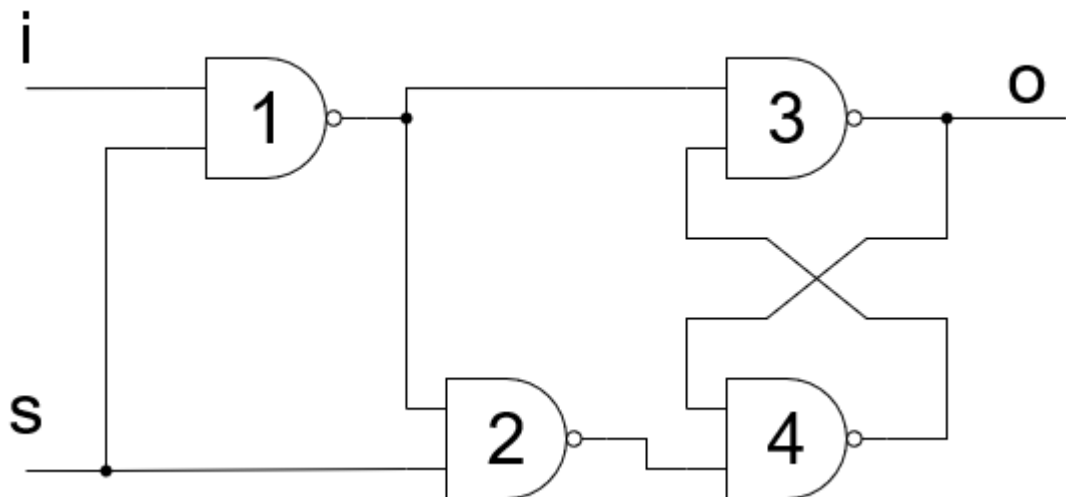
Kuva 1 Koodin toimintaa kuvaava pseudokoodi

Oikeat vastaukset tehtävään 6.1.

Tehtävät 7–10 ovat vaihtoehtoinen-valinta (Alternative Choice) tehtäviä, joissa tehtävän suorittajan tehtävä on annetun muistipaikan loogista toimintaa kuvaavan kaavion (kuva 2) avulla päätellä, miten ulostulon (o) arvoon vaikuttaa sisääntulon (i) ja asetussignaalin (s) arvot. Vaihtoehtoinen valinta on monivalinnan erikoistyyppi, jossa on vain kaksi vaihtoehtoa. Tämän kysymystyyppin etuna pidetään sekä tehtävän luomisen, että ratkaisemisen nopeutta, jolloin jää aikaa useammalle tehtävälle.

Jotta tehtävän voi ratkaista, tulee opiskelijan tuntea NAND-portin toiminta. Tehtävän tarkoituksena on antaa esimerkki loogisten porttien käytöstä konkreettisesti hyödyllisessä sovelluksessa, opetella lukemaan signaalien etenemistä porttien välillä, sekä havainnollistaa miten tietokone voi tallentaa tietoa muistipaikkoihin.

Muistipaikan toiminta



Kuva 2 Muistipaikan looginen toteutus

Alkutilanne: $s = 0$, $i = 0$, $o = 0$. Edellisen kysymyksen tilanne säilyy seuraavassa kysymyksessä, ellei toisin mainita.

7. Nostetaan s:n arvoksi 1, mikä on o:n arvo?

- a) 1
- b) 0 *

8. Muutetaan i:n arvoksi 1, mikä on o:n arvo?

- a) 1 *
- b) 0

9. Muutetaan s:n arvoksi 0, mikä on o:n arvo?

- a) 1 *
- b) 0

10. Muutetaan i:n arvoksi 0, mikä on o:n arvo?

- a) 1 *
- b) 0

4 YHTEENVETO

Automaattisesti tarkistettavia tehtäviä on kahden tyyppisiä: avoimia ja suljettuja. Suljettujen tehtävien tarkistaminen on helpompaa kuin avoimien sillä niiden tarkistamiseen riittää, että järjestelmälle määritetään, mikä vaihtoehdoista on oikein, ja mikäli opiskelijan valinta ei vastaa ohjelmoitua valintaa, vastaus on väärin. Suljetut tehtävät ovat yleensä monivalintatehtäviä tai niiden johdannaisia. Suurimpana haasteena monivalintatehtävissä on uskottavien väärin vastausten keksiminen. On todettu, että hyvä vaihtoehtojen määrä on kolme: yksi oikea ja kaksi väärää. Suuremmalla määrällä väärin vastauksia, tulee helposti tehtyä ilmiselvästi väärin vastauksia, jolloin olisi sama, jos kyseistä vastausta ei olisi ollenkaan. Hyvien väärin vastausten tekeminen on vaativaa ja vaihtoehtojen kasvattaminen kolmesta neljään ei lisää merkittävästi tehtävän hyvyttä.

Monivalintatehtäviä ja niiden johdannaisia on monenlaisia ja ne sopivat kaikki yhtäläisesti automaattitarkastettaviksi. Esimerkkeinä voidaan mainita aukkotehtävä, jossa annetuista vaihtoehdoista valitaan aukkoon sopiva. Esimerkkinä johdannaisesta on oikein-väärin-kysymystyyppi, jossa esitetään väite, johon vastataan, onko se totta vai ei.

Avoimia kysymystyyppisiä on todella monenlaisia. Niistä automaattisesti tarkastettaviksi sopivat lähinnä tehtävät, joihin voidaan ajatella olevan yksiselitteinen vastaus. Tällaisia tehtäviä voivat olla esimerkiksi laskutehtävät tai monivalinta-aukkotehtävää vastaava aukkotehtävä, mihin pitää täyttää jokin yksittäinen sana. Näissä tehtävissä tulee kiinnittää erityisesti huomiota tehtävänantoon, niin että ei jää epäselvää mitä haetaan, ja missä muodossa vastaus tulee antaa. Tämän lisäksi tehtävän laatijan tulee varautua, että vastaus saatetaan esittää eri ulkoasuilla (esimerkiksi ensimmäinen kirjain isolla, tai kaikki pienellä), tai vastauksessa saatetaan käyttää synonyymejä, mikäli tehtävän annossa ei vaadita nimenomaan tiettyä kirjoitusasua.

Monimutkaisempia avoimia kysymyksiä voidaan arvioida koneoppimisalgoritmien avulla. Latent Semantic Analysis (LSA) algoritmilla on saavutettu hyviä tuloksia. Haasteena näissä on kuitenkin vielä niiden spesifisyys. Koneoppimisalgoritmin opettamiseen tarvitaan paljon resursseja ja silloinkin se soveltuu vain opetusaineiston määrittämisen sisällön tarkastamiseen. Mikäli koneoppimis- ja tekoälymenetelmät kehittyvät tulevaisuudessa niin, että ne pystyvät tarkastamaan monenlaisia avoimia tehtäviä, niin silloin ne olisivat hyvä työkalu automaattiseen arviointiin.

LÄHDELUETTELO

Bjork, E.L., Soderstrom, N.C. ja Little J.L., 2015. Can Multiple-Choice Testing Induce Desirable Difficulties? Evidence from the laboratory and the Classroom. *The American Journal of Psychology*, 128 (2), S. 229–239

Ebel, R., 1982. Proposed solutions to two problems of test construction. *Journal of Educational Measurement*, 19, S. 267–278

Ebel, R., Frisbie, D., 1991. *Essentials of educational measurement*. Englewoods Cliffs, NJ: Prentice-Hall

Frederiksen, N., 1984. The real test bias. Influences of testing on teaching and learning. *American Psychologist*, 39, S. 193–202

Haladyna, T.M., 2004. *Developing and Validating Multiple-choice Test Items*. Mahwah, N.J.: Routledge.

LaVoie, N.m Parker, J., Legree, P., Ardison, S., Kilcullen, R. 2019 Using Latent Semantic Analysis to Score Short Answer Constructed Responses: Automated Scoring of the Consequences Test. *Educational and Psychological Measurement*, 80, S. 399–414

Lukhele, R., Thissen, D., Wainer, H. 1994. On the Relative Value of Multiple-Choice, Constructed Response, and Examinee-Selected Items on Two Achievement Tests. *Journal of Educational Measurement*, 31, S. 234–250

Statman, S., 1988. Ask a clear question and get a clear answer. An enquiry into the question/answer and sentence completion formats of multiple choice items. *System*, 16, 367-376

Sychev, O., Anikin, A., Prokudin, A., 2020. Automatic grading and hinting in open-ended text questions. *Cognitive System Research*, 59, S. 264–272

Tarrant, M., Ware, J., 2010. A comparison of the psychometric properties of three- and four-option multiple-choice questions in nursing assessment. *Nurse Education Today*, 30, S. 539–543

Vegada, B., Shukla, A., Charan, J. ja Desai, C. 2016. Comparison between three option, four option and five option multiple choice question test for quality parameters: a randomized study. *Indian Journal of Pharmacology* 48, S. 571–575

Wainer, H., Thissen, D. 1993. Combining Multiple-Choice and Constructed-Response Test Scores: Toward a Marxist Theory of Test Construction. *Applied Measurement in Education*, 6, S. 103–118