

Ai-zootson documentation

MAT-75006 Artificial Intelligence 2014

Name: Juho Teperi

Student number: 218197

Selkeyden, sekä kiireen kannalta, kirjoitin tämä suurimmaksi osaksi suomeksi.

Running

Mukana lein-skripti, eli <http://leiningen.org>. Se hakee koneelle projektissa käytetyt kirjastot. Lein näyttää lintulassa antavan jonkun virheen mutta näyttäisi toimivan...

Ohjelmaa voi ajaa komennolla (kun leinin ajaa ensimmäisen kerran kestää hetki kun se lataa kirjastot).

```
./lein run
```

Ohjelman voi kääntää komennolla

```
./lein uberjar
```

Tämän jälkeen ohjelman voi ajaa komennolla (tästä ei välttämättä juurikaan hyötyä vrt. lein run, mutta syntynyttä .jar voi suorittaa millä tahansa koneella jossa on JRE):

```
java -jar target/ai-zootson-0.0.1-SNAPSHOT-standalone.jar
```

Ohjelman mukana on kattava kokoelma testejä, ne voi ajaa komennolla

```
./lein midje
```

Ohjelman käynnistyminen voi tuntua hitaalta mutta se johtuu vain JVM käynnistymisen hitaudesta. Lisäksi lintulassa ohjelman kääntäminen kestää minuutteja, kun järkevällä koneella se kestää sekunteja.

How does your implementation work?

Toteutus on tehty funktionaalisella Clojure kielellä joka on Lisp toteutus Java Virtual Machinelle. Toteutus hyödyntää core.logic-kirjastoa joka tarjoaa Clojurelle Prolog tyyllisen logiikkaohjelmoinnin. Lauseiden parsimiseen käytän instaparse-kirjastoa jolle voi syöttää BNF-määrittelyn tekstinä ja se luo siitä funktion joka jäsentää syötteen kielen mukaan.

Core.logic tarjoaa faktatietokannan:

<https://github.com/clojure/core.logic/wiki/Features#simple-in-memory-database>

Faktat kuvataan relaatioina esim. (has-prop "gorilla" "hair" true) ja sieltä voi tehdä hakuja joissa yksi tai useampi relaation muuttujista voi olla vapaa: (run* [q] (has-prop q "hair" true)) palauttaisi eläimet joilla on karvoitusta. Hauissa vois myös olla vapaita

muuttujia joista ei olla kiinnostuneita:

```
(run* [q]
  (fresh [x]
    (has-prop "gorilla" x y)))
```

Antaisi listan gorillan kaikista tiedoista, välittämättä niiden arvosta...

ps. *test* hakemistosta löytyy yksikkötestit. Niitä on lähes yhtä paljon kuin koodia ja kuvaavat hyvin miten mikäkin funktio toimii. Erityisesti *facts_test.clj* ja *questions_test.clj* ovat kiintoisia sillä niissä on testi jokaiselle esimerkin lauseelle joista näkyy mitenkä BNF parseri toimii.

<http://clojure.org/>

<https://github.com/clojure/core.logic>

<https://github.com/Engelberg/instaparse>

What AI methods have been used in your implementation?

- Looginen ohjelmointi?
- ei oikein mitään?

Were you able to use things learned during the course in your implementation?

En.

Kirjassa oli NLP liittyen todella pieni raapaisu enkä kurssin kalvoista löytänyt mitään.

Were the instructions for the assignment clear enough?

- Ehkä muutamilta kohdilta voisi olla selkeämpi
- Esim. ohjeita mistä faktoista mikäkin vastaus pitäisi pystyä päättämään
 - hieman epäselvää esim:
 - "Cheetah is the fastest land animal." -> "Are girls slower than a cheetah?"
 - "Which is smaller: dolphin or crayfish?", delfiiniä ei mikään fakta maininnut
 - jotain ohjetta miten sanojen eri taivutukset/synonyymit kannattaisi hoitaa
 - "swimmer" "swim" "poisonous" "venomous"

Is it OK to use your implementation in our AI research?

Kyllä

What did this assignment teach you, if anything?

- Loogista ohjelmointi
- Merkkijonojen parsimista BNF:llä
- Clojuresta taas paljon lisää
- NLP on todella vaikeata

How much time did you use for this assignment?

150h, enemmän kun olisi kerennyt tekemään niin ei olisi jäänyt näin myöhään ja keskeneräiseksi.

Possible other opinions or remarks concerning the assignment and suggestions for the future?

- Saako koodin laittaa julkiseksi?
- Kuinkahan moni tämän sai palautettua, tuntui todella vaikealta
- Haluasin tietää miten tämä olisi kannattanut tehdä. Oma ratkaisu meinasi hajota käsiin viimeisen päivän aikana.