Lekserite käitumisest

Vesal Vojdani
(TÜ Arvutiteaduse Instituut)

- Regulaaravaldised üle (lõpliku) tähestiku Σ

$$E ::= \varepsilon \mid \mathsf{a} \mid (E\ E) \mid (E \mid E) \mid E^\star$$

kus $\mathsf{a} \in \Sigma$.

- Regulaaravaldis E defineerib keele $L(E) \subseteq \Sigma^\star$

$$s \in L(E)$$

Regulaaravaldis defineerib keele

Leksimine!

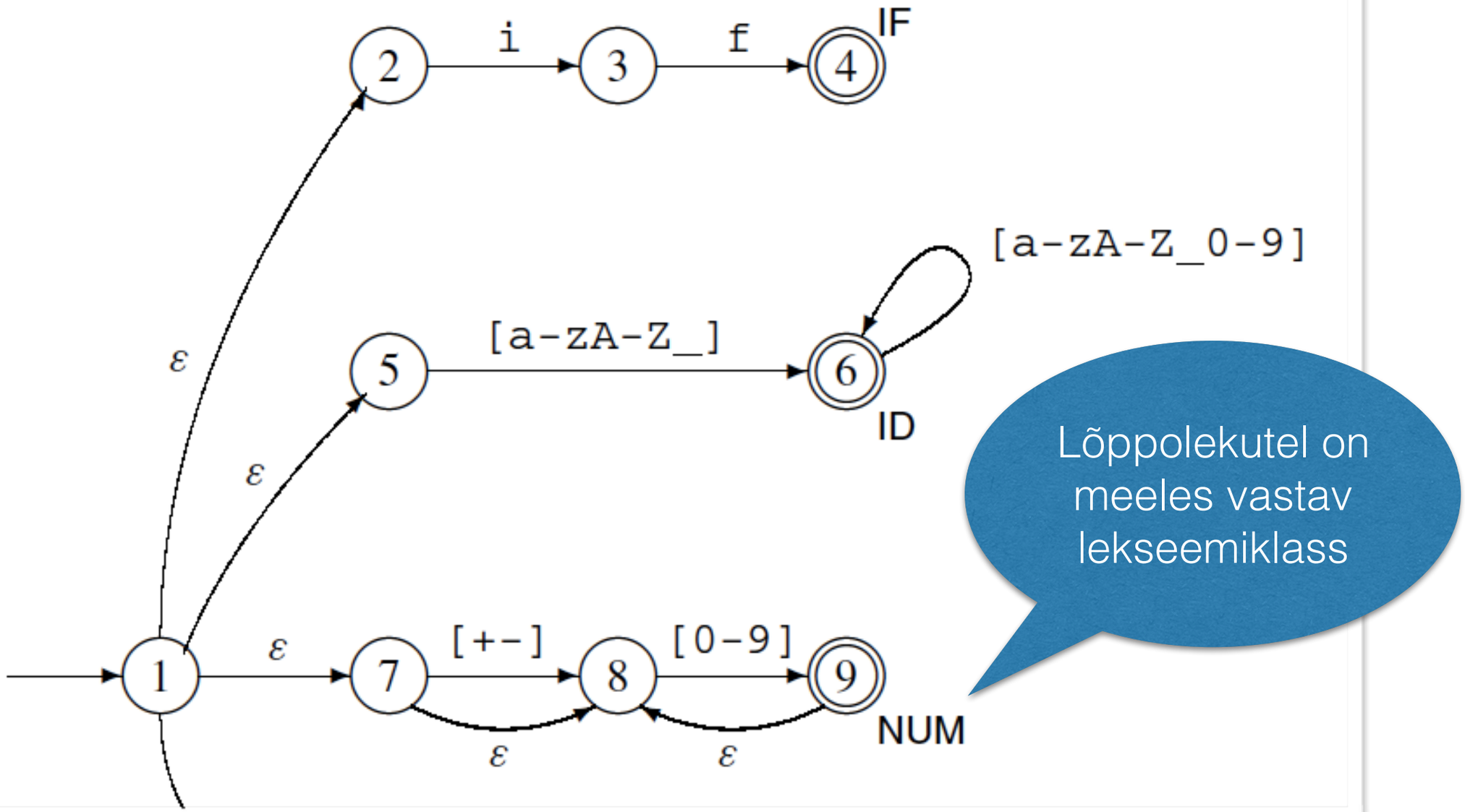
Probleemiks on siin “tüübiviga”. (lvalue/rvalue)

- x+++++y
  <ID:x>, <Op:++>, <Op:++>, <Op:+>, <ID:y>

- x+++ ++y
  <ID:x>, <Op:++>, <Op:+>, <Op:++>, <ID:y>

Leksiline spetsifikatsioon

- Keyword: ‘if’ | …
- Op: ‘++’ | ‘+’ | …
- Identifier: [a-zA-Z_][a-zA-Z_0-9]*
- …



Kombineeritud keel

$E = E_1 \mid E_2 \mid \ldots E_n$

Kuidas kasutada

- Sisendiks on $x_1 \ldots x_n$.
- Otsime kas leidub mõni *i*, et alamsõne $w = x_1 \ldots x_i$ kuuluks keelde L(E).
- Kui leidub, siis peab olema alamkeel $L(E_j)$, kuhu sõne kuulub. (Automaadi lõppolekus oli kirjas.)
- Eemaldame sisendist sõne w ja jätkame samamoodi, kuni sisend on tühi.

Maximal Munge!

- Mis juhtub, kui sobivad kaks alamsõne??
- Võib ju juhtuda, et $x_1 \ldots x_i$ ja $x_1 \ldots x_j$ mõlemad kuuluvad keelde L(E).
- Näiteks ‘++’ ja ‘+’ on mõlemad Java operaatorid.
- Loomulik konventsioon on valida pikim alamsõne, mis sobitub!

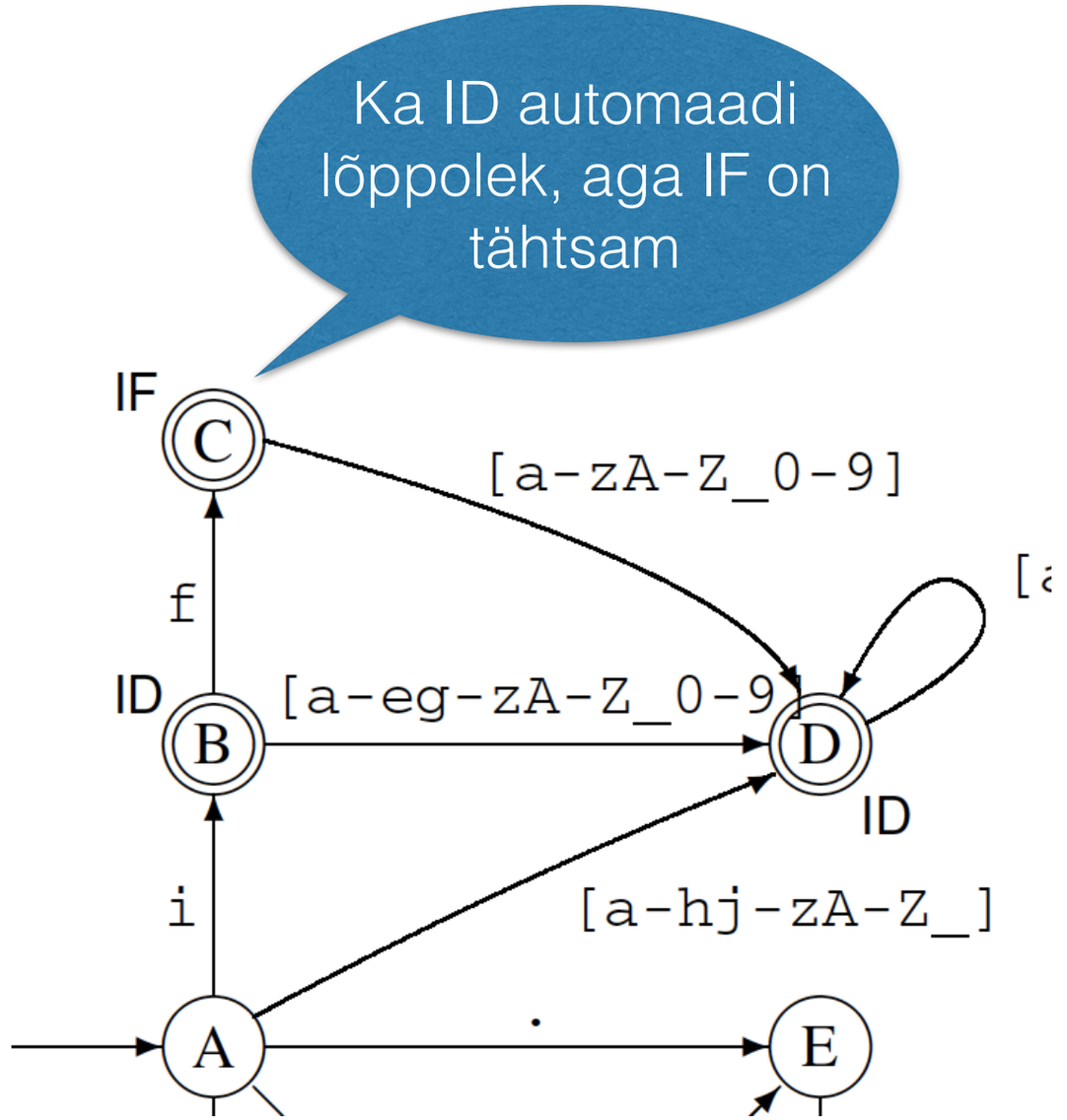
Milline lekseemiklass?

- Kas “`if`” on muutuja nimi või keyword?
- Prioriteedijärjekord: leksilise spetsifikatsiooni järjekord on oluline.
- NB! Kõigepealt valime pikim alamsõne ja alles siis valime lekseemiklass. (Seega, “`ifo`” on muutuja.)

Implementatsioonist

- Determiniseerime: jätame lõppolekutel meelde kõrgeima prioriteediga lekseem.

- Minimeerimine: lõppolekud on eristatavad, kui nad annavad erinevad lekseeme: {A,E}, {B,D}, {C}.



Pikim pigistus

- Automaat tuleb jooksutada nii kaugele kui saab, kasvõi sisendi lõpuni…
- Tuleb meeles pidada viimane lõppolek ja tagastada sellele vastav lekseem.
- Sisendis peab ka tagasi minema viimase sobitumise kohale.

Quiz!

Olgu järgmine leksiline spetsifikatsioon:

```
T1: a
T2: a*b
```

Millise sõne puhul on leksimine kõige aeglasem?
Ja kui aeglane see on (keerukus, kui *n* on sisendi pikkus)?

1. a
2. aaaaaaaaaa
3. aaaaaaaaaab

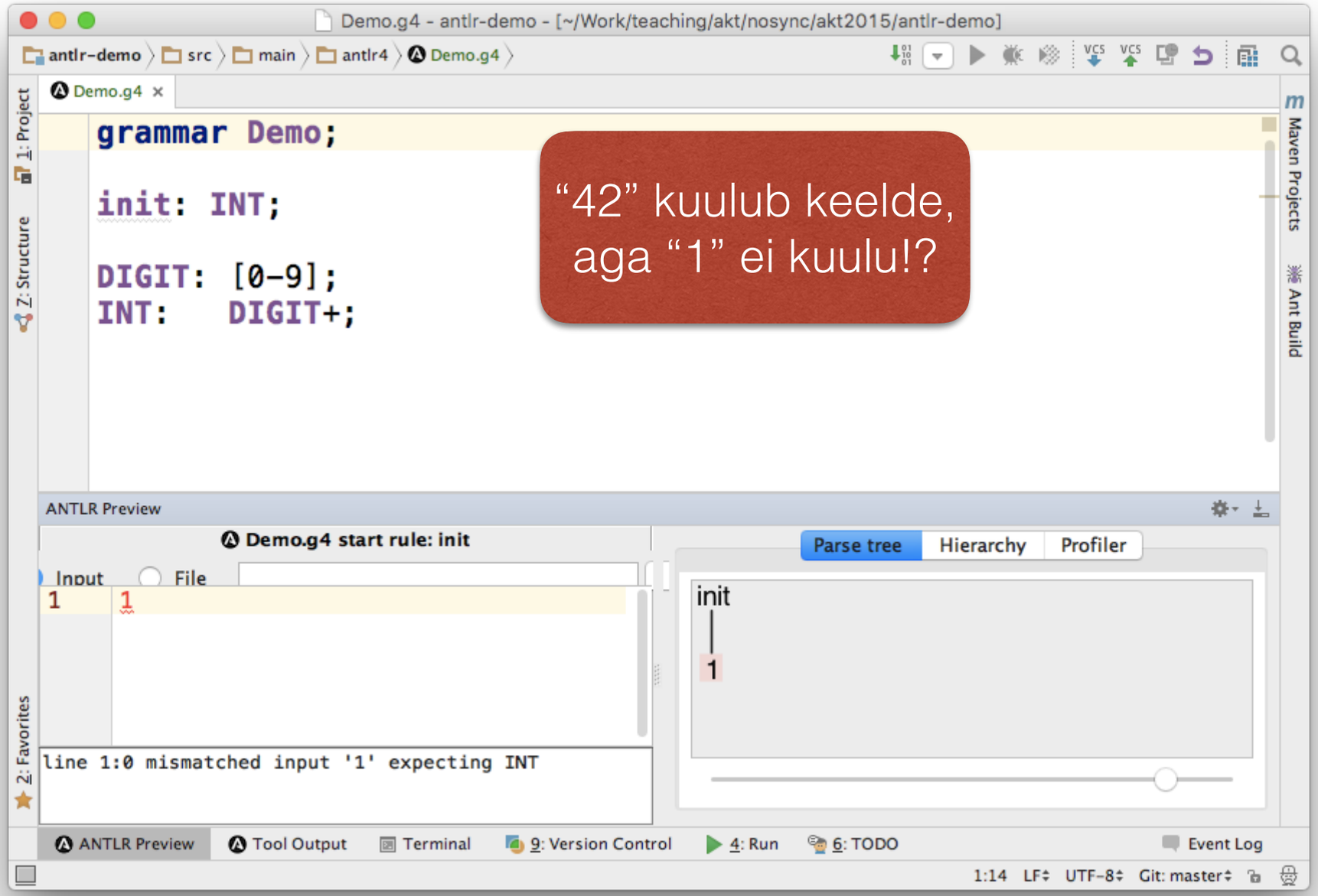
Hoiatus: ANTLR

- See temaatika muutub relevantseks, kui hakkame ANTLR'iga grammatikad kirjutama.

- ANTLR'is on lekseri reeglid ja grammatika reeglid ühes ja samas failis.

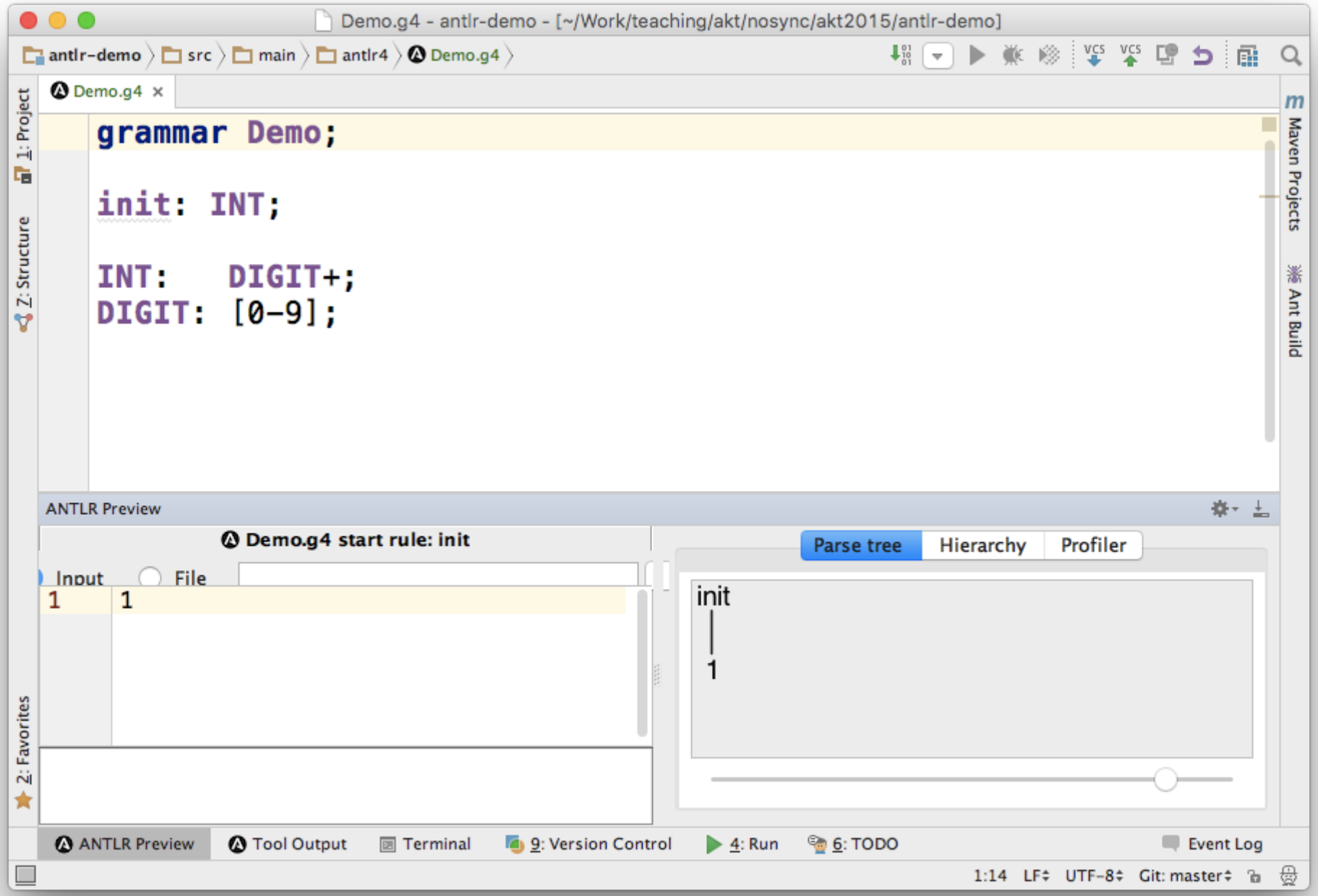
- Lekseri reeglid algavad suurte tähtedega…

- Lekseri reeglite vahel valitakse pikima pigistuse alusel, aga grammatika reeglitel mitte!



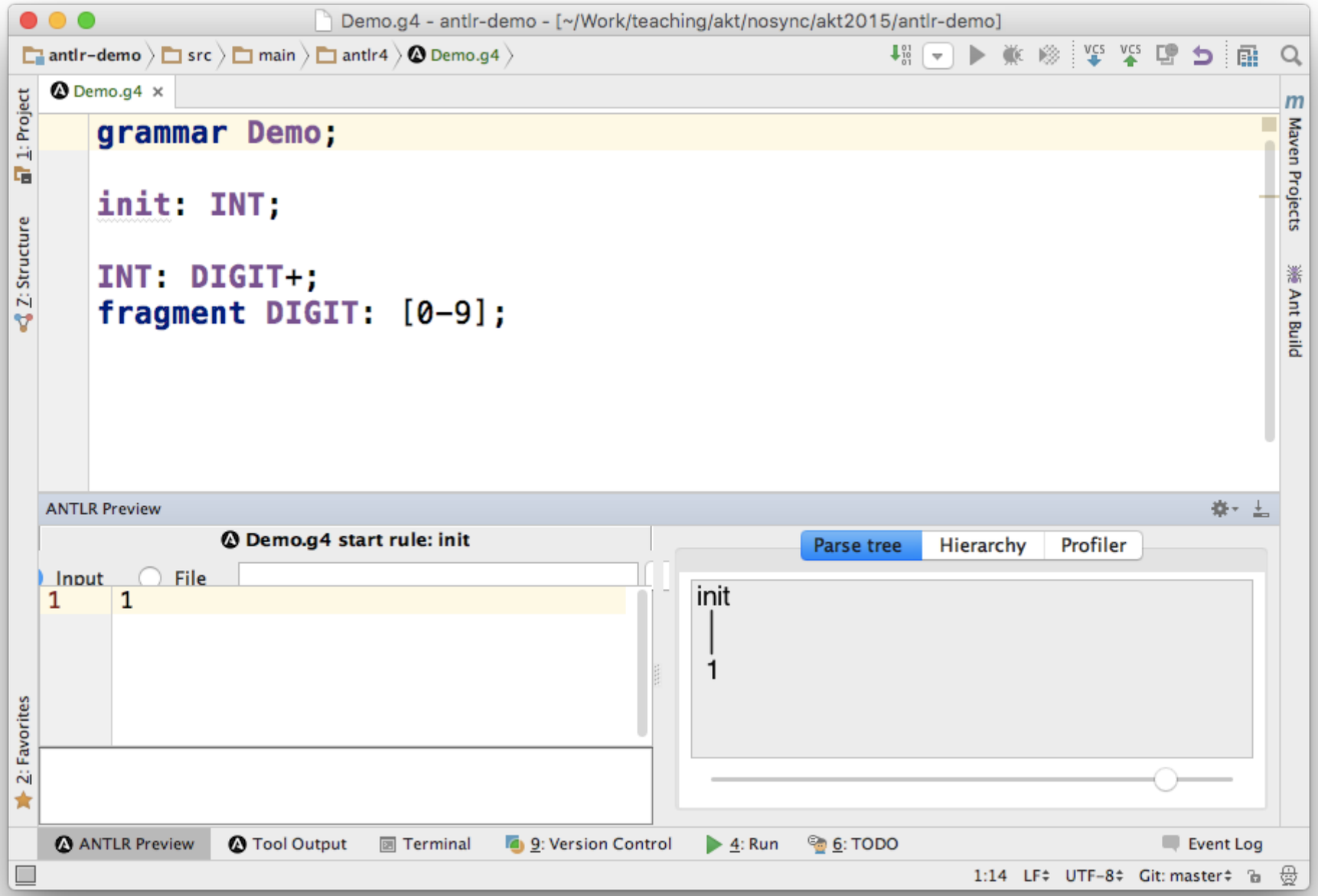
Danger!

ANTLR lubab lekseri reeglid kasutada teiste sees.
(See aga ei muuda, kuidas lekseeme valitakse!)



Järjestame ümber

ANTLR lubab lekseri reeglid kasutada teiste sees.
(See aga ei muuda, kuidas lekseeme valitakse!)



Palju kindlam

Fragmendi puhul ei ole DIGIT enam ise lekseem.
(Teeb lihtsalt loetavamaks, aga ei sega lekseri tööd!)