De Logo Taalwereld

Mattias Campe

Tegenstellingen

Brief Front IsoLab

Beste,

ik merk dat U Logo gebruikt, maar wij, van het “Front IsoLab”, vinden dat een slechte keuze:

(1)Logo is een moeilijke taal, voor niemand geschikt.

(2)MSWLogo heeft een aartslelijke interface.

(3)Een schildpad is een lelijk dier, een auto is veel toffer.

Tot schrijfs,
Geert Conver,
namens Front IsoLab

Doel oefening

- Doel
  - herschrijven van een ingegeven tekst, gebruik makend van tegenstellingen
- Concreet voorbeeld
  - *Logo is een <u>moeilijke</u> taal, voor <u>niemand</u> geschikt*
  - *Logo is een <u>makkelijke</u> taal, voor <u>iedereen</u> geschikt*
- Algoritme
  - Woord voor woord overlopen
  - Kijken of woord in de lijst “te vervangen” staat
    - Nee? => Schrijf woord uit
    - Ja? => Schrijf tegenstelling uit

Pseudocode

- zin regel

```
to zin regel
   als (regel = leeg)
      stop
   schrijf uit:
      (verwerking eerste woord)
         + zin (regel - eerste woord)
end
```

- vergelijk woord tegen

```
to vergelijk woord tegen
   als (tegen = leeg)
      uitschrijven woord
   anders als (woord = eerste woord tegen)
      schrijf tweede woord tegen uit
      anders als (woord = tweede woord tegen)
         schrijf eerste woord tegen uit
      anders
         schrijf uit vergelijk (woord)(tegen-eerste 2 woorden)
end
```

Vergelijken 1 woord

- Voorstelling tegenstellingen

```
Make "tegen [[moeilijke makkelijke] [niemand iedereen]]
```

- Controleren 1 woord: `vergelijk :woord :tegen`

```
to vergelijk :woord :tegen
   ifelse (emptyp :tegen) [
      output :woord
   ][
      ifelse (:woord = first first :tegen) [
         output last first :tegen
      ][
         ifelse (:woord = last first :tegen) [
            output first first :tegen
         ][
            output vergelijk :woord butfirst :tegen
         ]
      ]
   ]
end
```

Vergelijken zin + conclusie

- Implementatie zin: `zin :regel :tegen`

```
to zin :regel :tegen
   ifelse (emptyp :regel) [
      output []
   ][
      output fput (vergelijk first :regel :tegen) ~
         (zin (butfirst :regel) :tegen)
   ]
end
```

- Mogelijke uitbreidingen:
  - leestekens: bv. “toffer” ≠ “toffer.”
  - inlezen vanaf bestand
  - synoniemen: bv. [[toffer leuker]]

Pig Latin

Voorbeeld Pig Latin

- Voorbeelden
  - Mijpijn zupus zoupou graapaag knuputsepelepen, maapaar ipik wipil liepiever eepeen gepezepelschapapspepel spepelepe.
  - ijnMay uszay ouzay aaggray utselenknay, aarmay ikay ilway ieverlay eenay ezelschapspelgay elenspay.
- Origineel
  - Mijn zus zou graag knutselen, maar ik wil liever een geschelschapsspel spelen.

Wat is Pig Latin

- Definitie
  - Een publieke “geheime” taal, vooral populair bij kinderen.
  - soms ook “Dog Latin” genoemd
  - ontstaan in 18e eeuws Engeland, als combinatie van Latijn (of Grieks) en Engels
  - Gebruikt bij oa. Napster om filters te misleiden
- Links
  - Pig Latin lessen
  - Pig Latin vertaler
  - Geschiedenis Pig Latin
  - Verdere links

Implementatie

Pigwoord: zet alle letters achteraan het woord totdat een klinker ontmoet wordt, laat dan volgen door *ay*

```
to pigwoord :woord
   if (isklinker first :woord) [
      output word :woord "ay
   ]
   output pigwoord word (butfirst :woord) (first :woord)
end
```

```
to isklinker :letter
   output memberp :letter [A a E e I i O o U u]
end
```

- Piglatin

```
to piglatin :tekst
   if (emptyp :tekst) [
      output []
   ]
   output sentence (pigwoord first :tekst) ~
      (piglatin butfirst :tekst)
end
```

Bug

- *"Ik wil liever tv kijken"*

```
to pigwoord :woord
   if (isklinker first :woord) [
      output word :woord "ay
   ]
   output pigwoord word (butfirst :woord) (first :woord)
end
```

  - ✔ *Medeklinkers zorgen voor oneindige lus*

- enkelmedeklinkers :woord

```
to enkelmedeklinkers :woord
   ifelse (emptyp :woord) [
      output "true
   ][
      ifelse (memberp first :woord [A a E e I i O o U u]) [
         output "false
      ][
         output enkelmedeklinkers butfirst :woord
      ]
   ]
end
```

Implementatiefix

- Als een woord enkel uit medeklinkers bestaat, dan is het woord + *ay*

```
to piglatin :tekst
   if (emptyp :tekst) [
      output []
   ]
   ifelse (enkelmedeklinkers first :tekst) [
      output sentence word (first :tekst) "ay ~
         (piglatin butfirst :tekst)
   ] [
      output sentence (pigwoord first :tekst) ~
         (piglatin butfirst :tekst)
   ]
end
```

Uitbreidingen

- Uitbreidingen
  - enkel klinkers: “*ei*” -> “*ei**h**ay*” ipv “*eiay*”?
  - hoofdletters: “***G**raag*” -> “***A**ag**g**ray*” ipv “*aagGray*”
  - qu: “***qu**otiënt*” -> “*otiënt**qu**ay*” ipv “*uotiëntqay*”
  - leestekens: “wil.” -> “ilway.” ipv “il.*way*”
  - tekst inlezen vanaf een bestand (zie volgende slide)

Piglatin: IO

Piglatin en IO

- Doel
  - uitbreiding van het programma, zodat een tekst vanaf een bestand kan gelezen worden, waarbij de piglatin vertaling weggeschreven wordt in een ander bestand.

```
to piglatinIO :input
   make "output word :input "_piglatin.txt
   openread :input
   setread :input
   openwrite :output
   setwrite :output
   while [not EOFP] [
      pr piglatin readlist
   ]
   setread []
   setwrite []
   close :input
   close :output
end
```