

# Projectideeën Machines en Berekenbaarheid

## The One Generator To Rule Them All

Het programma heeft als input:

- een CFG (via XML of GUI-gebaseerd ingeven of zo)
- een manier om een parsetree om te zetten naar wat je eigenlijk wil (bijvoorbeeld een pdf in het geval van baggerpapers) (mogelijks inclusief een manier om terminals in functie te zetten van bepaalde extra input, zoals zinnen in functie van keywords)

Het programma heeft als output:

- Zoveel omgezette parsetrees als je wil, op basis van derivations van CFG (en mogelijks extra input voor terminals omzetting)

Het programma kan gebruik maken van (tot nu toe) 3 manieren van derivations:

- uniforme verdelingen (elke regel heeft evenveel kans),
- stochastische CFG (elke regel heeft een bepaalde kans),
- stochastische context-gebondenheid (een eigen probeersel, zie PoC in fancy\_idee)
  - zelf tabellen ingeven (saai, en vrij veel werk)
  - at random genereren en gebruiker geeft scores, programma leert bij (cool)
  - optioneel kunnen we er CNF in smijten en zien wat dat geeft :)
- extra manieren? → uitbreidingsmogelijkheden

We gebruiken voorbeelden van hieronder (voornamelijk papers, maar L-systemen zijn ook cool om te doen (maar moet aanpassingen krijgen...))

eigenlijk YACC<sup>-1</sup>

extra vragen: Hoe relateert unambiguous talen hiertoe? (LR talen bv) Wat voor effect geeft dat?

## Helix 2.0

- Voordelen
    - Awesome basis om van te vertrekken
    - gebruikt CYK, CFG, SCFG, DPDA's
  - Nadelen
    - het eigenlijke biologisch nut is ver te zoeken (behalve RNA secondary structure → al gedaan)
    - te weinig werk?
- 

## BEEP (v2)

Spelen met componenten als stack, tape, counter, finite control, ...

- Voordelen
    - Als alles zou werken vrij cool
    - Veel GUI-werk (*vooral* als we in C++ moeten werken)
  - Nadelen
    - kan enkel de concepten uit de cursus schetsen, maar omzettingen zijn onpraktisch
- 

## Generator projecten

### Muziek, afbeeldingen, baggerpapers

- Voordelen
  - kan nice zijn mits een goede CFG
  - veel uitbreidingsmogelijkheden
- Nadelen
  - weinig links buiten de CFG en at random (desnoods SCFG) derivien
  - uitbreidingsmogelijkheden zijn meestal buiten de realm van de cursus

**CFG zelf opbouwen → Overmoeilijk, ik zou zeggen praktisch AI-compleet**

---

## Turing machine/PDA/DPDA benchmarking

Input een algo (aka turing machine/pda/dpda) en een manier om random input te genereren die minstens af en toe een juiste string genereert

- Voordelen
  - vrij cursus-gerelateerd
  - (vind ik zelf (Evert) nog wel cool)
- Nadelen
  - Stijn zegt 'aaaaargghhh'

- Saai, theoretisch
- 

## Extra ideeën die extra uitwerking vereisen

- Siri met CFG's ...