

# Algoritmen en datastructuren 3

## Huffman algoritmen

TOBIAH LISSENS

Universiteit Gent

6 December 2017

### Samenvatting

*In dit verslag wordt een manier gegeven om een simpele schaakcomputer in prolog te schrijven. De invoer en uitvoer wordt aan de hand van Forsyth-Edwards Notation gedaan, dit is een korte voorstelling van het schaakbord. Eerst wordt kort aangehaald hoe deze invoer omgezet kan worden naar een interne prolog voorstelling aan de hand van Definate Clause Grammer. Vervolgens wordt een besproken hoe we met minimax de beste zet kunnen vinden in een spelboom. Als laatste worden enkele optimalisaties besproken zoals het gebruik van alfa-beta-snoeien. Het resultaat is een schaakcomputer die enkele spelen kan drawen tegen Stockfisch niveau 1.*

### I. INLEIDING

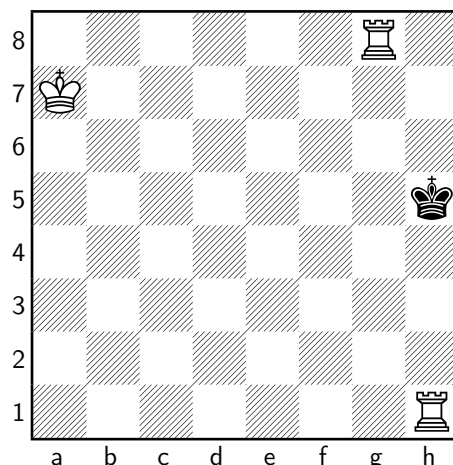
Schaken is een heel bekende denksport. Het evalueren van een schaakpositie behoort echter tot de complexiteitsklasse EXPTIME. Hierdoor is een optimale manier van schaken nog steeds een raadsel. Dit heeft programmeurs echter niet tegengehouden de uitdaging aan te gaan om schaakcomputers te schrijven die beter zijn als de menselijke meesters. Enkele bekende schaakcomputers zijn Stockfish 9, Houdini6 en Alpha Zero. De bedoeling van dit project is een schaakcomputer te schrijven die gelijkspel kan spelen tegen de schaakcomputer stockfish 9 op het laagste niveau.

### II. FEN INVOER EN UITVOER

Het parsen van de FEN<sup>1</sup> gebeurt aan de hand van DCG<sup>2</sup>. Deze manier van parsen is bijzonder handig omdat je bidirectioneel kan converteren van FEN string naar een interne representatie en van een interne representatie naar FEN.

<sup>1</sup>Forsyth-Edwards Notation

<sup>2</sup>Definite clause grammar



FEN 6R1/K7/8/7k/8/8/8/7R w KQkq - 10 50

De bovenstaande configuratie wordt intern met de volgende term voorgesteld.








```
fen_config(  
  board(  
    row(nil, nil, nil, nil, nil, nil, piece(w, rook), nil),  
    row(piece(w, king), nil, nil, nil, nil, nil, nil, nil),  
    row(nil, nil, nil, nil, nil, nil, nil, nil),  
    row(nil, nil, nil, nil, nil, nil, piece(b, king), nil),  
    row(nil, nil, nil, nil, nil, nil, nil, nil),  
    row(nil, nil, nil, nil, nil, nil, nil, nil)
```

row(nil, nil, nil, nil, nil, nil, nil, nil), lueren. Het kiezen van een goede evaluatie  
 row(nil, nil, nil, nil, nil, nil, nil, piece(w, castle(false, false, false, false), nil, 0, 50  
 ) ,w, castle(false, false, false, false), nil, 0, 50  
 )

### III. GENEREREN ZETTEN

Het genereren van zetten gebeurt voor de Loper, Toren, Koningin en Koning allemaal op dezelfde manier. Aan de hand van de movement regel hier kan een lijst van richtingen en een range<sup>3</sup> aan worden meegegeven. Zo zijn de richtingen voor de stukken als volgt:

Tabel 1: richtingen

stuk	richtingen	bereik
	-1/1, 1/1, -1/-1, 1/-1	8
	-1/0, 1/0, 0/-1, 0/-1	8
	 + 	8
		1

Verder kan het paard geïmplementeerd worden door de huidige posite met een positie uit: -2/-1 -1/-2 1/-2 2/-1 -2/1 -1/2 1/2 2/1 op te tellen. De pion bestaat uit heel veel uitzonderingen waarvoor elk een apparte regel is gemaakt. Verder word het checken op schaak staan na een zet gedaan door alle mogelijke volgende borden te genereren en te controleren of de koning nog op het veld staat. Deze manier van controleren op schaak is alles behalve efficiënt, maar dit wordt bij het opbouwen van de spelboom bij minimax toch maar uitzonderlijk gebruik van gemaakt.



### IV. MINIMAX







### V. ALPHA-BETA

### VI. EVALUATIE

Bij het afsluiten van een zoekboom op een zeker diepte moeten we een positie kunnen eva-

<sup>3</sup>het maximaal aantal vakjes dat ze kunnen opschuiven

lueren. Het kiezen van een goede evaluatie functie is norm belangrijk. In deze implementatie hebben we gekozen voor een simplified chess evaluatie functie die in *Simplified evaluation function* staat uitgelegd. Het komt er op neer dat elke stuk de waarde zoals in Tabel 2 wordt toegekend. De meeste waarden zijn vanzelfsprekend behalve deze voor  en . De meeste schaakboeken kennen deze elke een score van 300 toe maar om te voorkomen dat deze stukken worden geruild voor 3 pionen. En om er voor te zorgen dat bisschop paren meer waard zijn dan paarden paren krijgen die een iets wat afwijkende waarde.

stuk	waarde
	100
	320
	330
	500
	900
	20000

Tabel 2: Waarden van stukken

Verder is bij het evalueren ook de positie van de stukken belangrijk. Het aantal velden dat ze bedrijven en het samen hangen van pionenen enzovoort. Hiervoor wordt er gebruik gemaakt van positie tabellen. Dit zijn tabellen die voor elke coördinaat een bonus waarde of penalty toekennen.

### VII. BESPREKING

In Tabel.?? zien we de de schaakinator2000 die 20 games tegen de stockfish engine en 20 games tegen de pychess engine gespeeld heeft. Hier zien we de dat stockfish engine het beduidend beter doet dan de schaakinator 2000 en pychess.

Verder kunnen we nog enkele mogelijke verbeteringen opsommen. De huidige versie van patbot kan geen onderscheid maken tussen gamefases. We zouden een heuristiek voor kunnen schrijven die de gamefase bepaalt bv door het aantal aanwezige stukken/pionen. Aan de

hand van de gamefase zouden we dan onze pos/stuk-score kunnen aanpassen. Een koning in het begin en einde aan de rand is goed maar in het eindspel is het meestal voordelig deze in het spel te betrekken.

Verder maakt de huidige schaakcomputer ook nog gebruik van semi-legale-zettengeneratie dit wil zeggen dat we bij het genereren van de moves in de spelboom niet checken of onze koning schaak staat na een bepaalde zet. Het probleem hiermee is dat wanneer de schaakbot zijn tegenstander schaak mat wil zetten hij niet kan weten of deze mat of pat staat. Door het gebruik van de gamefases zouden we in het eindspel kunnen overschakelen naar legale move generatie checken op schaak is iets zwaarder maar in het eindspel is het aantal mogelijke zetten ook een stuk kleiner.

**Tabel 3:** *Resultaten op 20 games*

match	win	loss	draw
pychess vs chesinator2000			
stockfish 9 vs chesinator2000			
stockfish 9 vs pychess			

## VIII. CONCLUSIE

Zoals verwacht is het schrijven van een goede schaakcomputer een hele grote uitdaging die buiten de scope van dit vak. Het is dus ook logisch dat deze schaakcomputer niet fantastisch goed presteerd. Desalnietemin is deze schaakcomputer een mooie proof of concept. Die andere schaakcomputers op hun laagste niveau als het mee zit kan verslaan.

## REFERENTIES

Michniewski, Tomasz. *Simplified evaluation function*. URL: <https://chessprogramming.wikispaces.com/Simplified+evaluation+function> (bezocht op 18-06-2018).