

1 Inleiding

Het spel boter, kaas en eieren is goed te spelen op een computer met behulp van Minimax. Dit is echter heel anders dan de manier waarop mensen spelen. Een ander nadeel is dat er een heuristisch gebruikt moet worden, en dat bordposities dus moeten worden omgezet in een 1-dimensionale score. Bij een goede strategie worden echter meerdere factoren meegewogen, en afhankelijk van de situatie weegt iets zwaar of minder zwaar.

2 Advice Language

Een mogelijke hypothese is dat mensen gebruik maken van regeltjes, om goed boter kaas en eieren te spelen. Een uitwerking van dit idee zou gebruik kunnen maken van Advice Language [1]. Dit is een rule-based, goal oriented systeem voor spellen. Het idee is om veel kennis te verwerken in regels. Het voordeel is natuurlijk dat er veel minder rekenwerk nodig is dan bij minimax. Maar een bijkomend voordeel is dat het systeem de gebruikte regels kan tonen, de redeneer stappen als het ware.

Nadeel is wel dat het veel werk kan zijn om zulke regels op te stellen. Een voorbeeld van een regel in advice language¹:

```
advice(mate_in_2,  
      mate :  
      not rooklost and their_king_edge :  
      (depth = 0) and legal then (depth = 2) and checkmove :  
      (depth = 1) and legal).
```

Het eerste argument is de naam van het advies, zodat hij kan worden aangeroepen vanuit een regel. Het tweede argument is de “BetterGoal”, het uiteindelijk doel van dit advies. Daarna komt de “HoldingGoal”, dit doel moet steeds vervuld blijven wil dit advies zin hebben (in het voorbeeld: zonder toren kunnen we niet schaak zetten). De laatste twee argumenten zijn constraints voor de speler zelf en de tegenstander.

Zo zijn er dan nog enkele adviezen, voor andere situaties. Vervolgens is er een regel die deze zetten probeert in een bepaalde volgorde, als een conditie wordt voldaan:

```
edge_rule ::  
  if their_king_on_edge and our_king_close  
  then [ mate_in_2, squeeze, approach, keeprum, divide].
```

¹De notatie lijkt niet op Prolog, maar als de juiste operators gedefinieerd worden is het een correcte Prolog clause

```
else_rule ::  
    if true  
    then [ ... ]
```

3 Bronnen

[1] Bratko (2001), Prolog programming for AI.