Flash fill en Equation Discovery*

Jeroen Craps & Tom De Groote
KU Leuven
Leuven, Belgie
jeroen.craps@student.kuleuven.be
tom.degroote@student.kuleuven.be

Abstract

Deze paper beschrijft een applicatie voor het vinden van een vergelijking in een set van relatief willekeurige getallen die door een gebruiker zijn gespecifïeerd. Het doel van deze applicatie is om gebruikers te helpen in het vinden van een passende vergelijking voor de overige gegevens aan de hand van enkele voorbeelden. De applicatie voorziet dus de gebruiker om meerder voorbeelden te geven met welke de applicatie rekening moet houden. Op basis van enkele experimenten bleek deze applicatie in een beperkte tijd toch satisfieerbare resultaten te voorzien aan de gebruiker.

1 Introductie

1.1 Naburige problemen

Steeds vaker wordt er van een computer verwacht dat deze complexere taken kan uitvoeren. Terwijl het voor de gebruikers meestal duidelijk is wat er gewenst wordt, is dit voor de computer niet altijd het geval. Een recent voorbeeld hiervan is de 'FlashFill' functionaliteit dit momenteel aanwezig is in Excell 2014. Hierbij is het de bedoeling dat de computer het gewenste patroon, dat door de gebruiker gekend is, op een set van data herkent en kan toepassen.

Een ontbrekende functionaliteit hierin is wel het herkennen van wiskundige bewerkingen, het 'vinden' van een passende vergelijking. Het bepalen van dit soort van vergelijking is geen onbekend onderzoeksdomein. In 2002 verscheen er een paper die een oplossing voor het 'Countdown Problem' gaf. Bij het 'Countdown' is een Brits televisie programma waarbij de deelnemers een wiskunde vergelijking zoeken met de gegeven nummers om zo dicht mogelijk tot bij een bepaald streefdoel te komen. Een belangrijke beperking op dit probleem zijn wel dat elk gegeven cijfer maximaal éénmalig mag voorkomen. Indien een gebruiker bepaalde waarde meermaals wilt gebruiken is dit niet meer volgens de regels van 'Countdown'. Aangezien de gebruiker sommige waarden meermaals wil kunnen gebruiken moet er een alternative manier van aanpak gevolgd worden.

*Voor verandering vatbaar.

De vraag die ondezocht wordt in deze paper is nu: "Gegeven een aantal voorbeelden, is het mogelijk om een passende vergelijking te vinden binnen beperkte tijd?"

1.2 Context-vrije grammatica

Een eerste belangrijk punt is de manier waarop een vergelijking wordt voor gesteld. Hiervoor wordt er context-vrije grammatica (CFG) gebruikt. Volgens de 'formele taal theorie' is een CFG een formele grammatica waarbij elke productieregel er uit ziet als volgt: $V \rightarrow w$. Waarbij V een enkel niet-terminaal symbool is en w een tekenreeks is van niet-terminale en terminale symbolen. 'Context vrij' betekend dat de productieregels kunnen toegepast worden los van de context waarin het niet-terminale symbool zich bevind. In een gegeven tekenreeks kan V vervangen worden door de tekenreeks van w.

$$\begin{array}{|l|}\hline E \rightarrow E + E \\ E \rightarrow E - E \\ E \rightarrow E \times E \\ E \rightarrow E \div E \\ E \rightarrow 1, 2, \ldots, 9, a, b, \ldots \\ \hline \end{array}$$

In bovenstaande figuur staat een voorbeeld van een CFG dat vergelijkingen kan voorstellen. Indien een nieuwe wiskundige operatie gewenst is is deze eenvoudig toe te voegen aan de reeds bestaande grammatica. Gemakkelijke aanpasbaarheid is een voordeel van het werken met een context-vrije grammatica, omdat de productieregels volledige onafhankelijk van elkaar kunnen staan. Deze CFG is de basis waarop er gewerkt wordt om een oplossing te vinden tot het probleem.

1.3 Boomstructuur

Aangezien er op voorhand niet geweten kan worden welke uitwerking van de CFG er nodig zal zijn om een oplossing te voorzien, wordt er door herhaaldelijke toepassing van de productieregels op de begin term E een boomstructuur gecreëerd waarin alle mogelijkheden opgenoemd worden. De operatie zorgt voor een exponentiële groei r^d waarbij d staat voor de diepte van de boom en r voor het aantal mogelijke productieregels van de vorm $E \rightarrow E\ operand\ E$. Het verloop van deze boom staat beschreven in volgende figuur en tabel.

Vanwege de grote berekeningstijden op diepere niveau's van de boom en de onafhankelijkheid van het probleem tot de boom, is het ten zeerste aangeraden om deze boom slechts éénmalig op voorhand te bepalen. Een belangrijk voordeel van deze manier van opstellen is dat er geen enkele mogelijkheid kan ontbreken.

2 Eerste implementatie

2.1 Brute force

Deze opgestelde boom wordt gebruikt om een vergelijking te vinden passende bij verscheidene gebruikersvoorbeelden. Het invullen van elke mogelijkheid uit de boom met de gegeven variabelen van de gebruiker heeft een complexciteit van $b^d * v^{d+1}$. Voor elke E symbool moet elke mogelijkheid ingevuld worden om de volledige term te laten evalueren. Indien deze term evalueert tot de gezochte doelwaarde voor het voorbeeld, dan is dit een mogelijke vergelijking voor het gebruikers probleem.
Indien dat deze vergelijking dan voldoet aan de overige voorbeelden is dit een oplossing voor de gebruiker en is het probleem dus opgelost. En is het mogelijk om de overige gebruikersdata te vervolledigen aan de hand van deze gevonden vergelijking.

2.2 Gebruikersdata

De meeste applicatie worden geschreven om een bepaald gebruikersprobleem op het lossen. Hiervoor bestaat dus al de nodige gebruikersdata waaraan het algoritme zich kan testen. In het geval van dit probleem is bestaat er geen gebruikersdata, omdat er momenteel nog geen belangstelling is voor deze techniek. Alle data waarop experimenten zullen uitgevoerd worden zullen aan de hand van zelfgegenereerde data zijn. De manier waarop deze data genereerd wordt is gestructureerd met een zekere willekeur op het bepalen van kleine variaties.

3 Pruning

Tijdens het overlopen van de boom komen er veel knooppunten meerdere keren voor. Deze redundante knooppunten nemen niet alleen plaats in op de diepte van het huidige niveau. Hieruit worden dan met behulp van de productieregels ook enkel redundante knooppunten gecreëerd. Een voorbeeld van een redundante knoop is $E + E \times E$, indien $E \times E + E$ reeds gekend is.

3.1 Manier

Sommige termen zijn voor ons redundant op een bepaald niveau, maar zijn de nakomelingen van deze term niet redundant op hun niveau. Na het bepalen van de knopen op een bepaalde diepte, is er controle op knopen die redundant zijn op hun niveau. Maar ook een controle op knopen waarvan de nakomelingen niet equivalent zijn, deze knopen worden gebruikt om het volgende niveau te bepalen. De eerste lijst wordt gebruikt om zijn eigen niveau van de boom te beschrijven.

3.2 Resulaten

De gevolgen van deze eenvoudig vorm van pruning zijn enorm. Door de exponentiële groei van de boom zijn er veel minder knopen op diepere niveau's.

Pruned	Normal
1	1
4	4
15	16
52	64
186	256
699	1024
2717	4096
10742	16384

Aangezien er veel minder punten zijn is de tijd nodig om de volledige boom te doorlopen en op te stellen ook significant gedaald.

Acknowledgments