De taal van acceptabele popmuziek

Elise Hopman & Tim Steenvoorden

18 januari 2012

1 Beschrijving

Veel popmuziek heeft als basis dezelfde vier akkoorden, vaak zelfs in dezelfde volgorde. Bekijk bijvoorbeeld eens The Axis of Awesome 4 Chords. In dit werkstuk gaan wij op zoek naar een formele beschrijving van simpele popliedjes. Hoe een liedje klinkt wordt grotendeels bepaald door de gebruikte akkoorden. Daarom beperken wij onze beschrijving van muziek tot het akkoordenschema. De vier gebruikte akkorden in het voorbeeldliedje zijn *E*, *B*, *C#m* en *A*, maar er zijn vergelijkbare compilaties met bijvoorbeeld *C*, *G*, *Am* en *F*. Dat deze compilaties hetzelfde klinken komt omdat de sprongen tussen de akkoorden in beide schema's hetzelfde zijn. Het zijn de sprongen tussen de akkoorden die ervoor zorgen dat deze liedjes makkelijk in het gehoor liggen, maar welke akkoorden je precies neemt maakt niet uit. Om onze beschrijving zo algemeen mogelijk te houden werken wij met de fictieve akkoorden a, b, c en d.

Laten we aannemen dat de overgangen $a \to b$, $b \to c$, $c \to d$ en $d \to a$ goed in het gehoor liggen. Dit zijn *mooie* overgangen. We krijgen wel saaie liedjes wanneer we alleen mooie overgangen toestaan. Het hele liedje ligt dan namelijk vast wanneer we het beginakkoord kiezen. Daarom keuren we ook de *matige* overgangen $a \to c$, $c \to a$, $b \to d$ en $d \to b$ goed. Deze klinken iets minder mooi, maar zijn zeker goed genoeg om in een liedje te verwerken. Een simpele grammatica hiervoor ziet er dan als volg uit:

```
G: S -> A | B | C | D
   A -> aB | aC | a
   B -> bC | bD | b
   C -> cD | cA | c
   D -> dA | dB | d
```

Deze grammatica voldoet aan de voorwaarden van een *reguliere grammatica* en dus is de bijbehorende taal regulier. Helaas genereert deze grammatica ook liedjes met alleen maar matige overgangen. Als we willen dat onze liedjes de Top 40 halen kunnen we eisen dat er meer *mooie* dan *matige* overgangen inzitten. Muziek die hieraan voldoet noemen we *acceptabel*. Wanneer $L(G)$ de taal is van grammatica G kunnen we onze taal karakteriseren door:

$$L := \left\{ w \in L(G) \mid n(w) < m(w) \right\}$$

Hierbij zijn $m(w)$ het aantal mooie overgangen en $n(w)$ het aantal matige overgangen in een woord. Merk op dat het aantal overgangen in een woord w altijd gelijk is aan $lengte(w) - 1$.

Voor we gaan bekijken of L regulier en/of contextvrij is bekijken we het nut van onze taal. Hoe realistisch is onze taal L?

Lengte van de liedjes Liedjes die in L zitten hebben een lengte van 1 tot n akkoorden, waar $n \in \mathbb{N}$. Dit is natuurlijk niet erg realistisch. Een popliedje duurt meestal tussen de twee en vier minuten, terwijl onze taal liedjes met elke mogelijke duur bevat. Merk op dat het lege liedje niet in L zit.

Variatie Met onze taal beschrijven we liedjes die maximaal vier akkoorden bevatten. Hoewel je hiermee veel muziek beschrijft zijn en natuurlijk nog veel meer liedjes die ingewikkelder in elkaar zitten. Ook zijn onze eisen aan het liedje over hoe je die vier akkoorden dan na elkaar gebruikt vrij streng. Waarschijnlijk worden in echte liedjes ook wel eens *lelijke* overgangen gemaakt die wij niet toestaan, bijvoorbeeld twee maten achter elkaar met hetzelfde akkoord.

Structuur De meeste liedjes zijn opgebouwd uit stukken die een paar keer terugkomen, zoals coupletten en het refrein. Meestal hoort dan bij elk couplet hetzelfde akkoordenschema. Onze taal houdt geen rekening met dit soort herhalingen. Een oplossing hiervoor zou kunnen zijn om L te gebruiken als beschrijving van de losse stukken (couplet, refrein, bridge, et cetera) en zelf de gewenste vorm van het liedje te maken met de losse stukken.

Onze taal L kunnen we dus als een vrij simpele generator voor popmuziek gebruiken door zelf geschikte akkoorden in te vullen voor a, b, c en d. Daarbij moet de gebruiker zelf rekening houden met de lengte van het akkoordenschema en handmatig de schema's voor couplet en refrein in de gewenste volgorde en herhalingen zetten.

2 Regulier?

We gaan met het pomplemma bewijzen dat onze taal L niet regulier is. Stel dat L wel regulier is. We weten dan dat er een DFA met een eindig aantal, zeg k, toestanden is die L accepteert. Volgens het pomplemma geldt dan voor elk woord $z \in L$ met $lengte(z) \leq k$ dat z geschreven kan worden als $z = uvw$ met $lengte(uv) \leq k$ en $lengte(v) > 0$ zodat $uv^iw \in L\ \forall i \in \mathbb{N}$. We bekijken het woord $z = (ac)^k(abcd)^k$. De $lengte(z) = 2k + 4k$, het aantal overgangen is dan $2k - 1$ zoals we hierboven hebben uitgelegd. De overgangen $a \to c, c \to a$ zijn matig en komen $2k$ keer voor. Dit is inclusief de overgang van $(ac)^k$ naar $(abcd)^k$. De overgangen in het tweede deel van het woord zijn allemaal mooi. Dit zijn er $4k - 1$. We weten dat de $lengte(z) \geq k$, dus kunnen we het pomplemma op z toepassen. Volgens het pomplemma kunnen we z schrijven als uvw. Daarbij geldt dat $lengte(uv) \leq k$ en zit uv in het eerste gedeelte van het woord. Hier hebben we een gevalsonderscheiding nodig:

1. uv eindigt op een a, dus $uv = acac \dots a$ en $w = cacac \dots ac(abcd)^k$

   Nu zijn er weer twee gevallen te onderscheiden voor de $lengte(v) = j$, namelijk:

   a) j is oneven

      In dit geval begint en eindigt v met een a en eindigt u op een c. Dan geldt dat $uv^0w = acac \dots ccaca \dots ac(abcd)^k$, maar de overgang cc tussen u en w is niet toegestaan dus $uv^0w \notin L$

   b) j is even (dus $j \geq 2$)

      In dit geval pompen we $k+1$ keer. We voegen dan jk nieuwe matige overgangen toe. In totaal hebben we dan $2k + jk = (j+2)k > 4k$ want $j \geq 2$ matige overgangen, en dat is meer dan de $4k-1$ mooie overgangen. Dus $uv^{k+1}w \notin L$.

2. uv eindigt op een c, dus $uv = acac \dots c$ en $w = acac \dots ac(abcd)^k$

   Deze gaan analoog aan de twee gevallen hierboven met de rol van a en c omgewisseld.

We hebben aangetoond dat ons woord z in alle gevallen niet pompbaar is. Onze aanname dat L regulier is was dus verkeerd. Hiermee hebben we bewezen dat L regulier is.

3 Context vrij?

Om te bewijzen dat onze taal L context vrij is, construeren we een *stapelautomaat* die de taal herkent. Voordat we deze machine maken, bekijken we eerst een eenvoudiger geval van taal L, namelijk de taal

$$L' := \Big\{ w \in \{m, n\} \mid n_n(w) < n_m(w) \Big\}.$$

Dit zijn woorden uit het alfabet $\Sigma = \{m, n\}$ die bestaan uit *mooie* (m) en *matige* (of *niet mooie*, n) overgangen waarbij er meer mooie dan matige overgangen zijn. Deze taal lijkt dus erg op onze taal voor popmuziek. Zoals bij de beschrijving al is aangestipt, hebben woorden in deze taal een lengte die 1 kleiner is dan een vergelijkbaar woord in taal L. Daarnaast verwijderen we ook informatie wanneer we een woord uit L 'vertalen' naar een woord uit L', namelijk de akkoorden.

Om een automaat voor L' te maken, moeten we op de stapel bijhouden hoeveel mooie overgangen we voorbij zien komen, en hoeveel matige. Laten we hiervoor de letters M en N gebruiken. Nu lezen we een woord w uit L'. Wanneer we een m tegenkomen kunnen we het volgende doen:

- Een M op de stapel zetten om bij te houden dat we een mooie overgang hebben gelezen.
- Een N van de stapel halen zodat we matige overgangen 'ongedaan maken'.

- De m gratis lezen, aangezien er meer mooie dan matige overgangen mogen zijn.

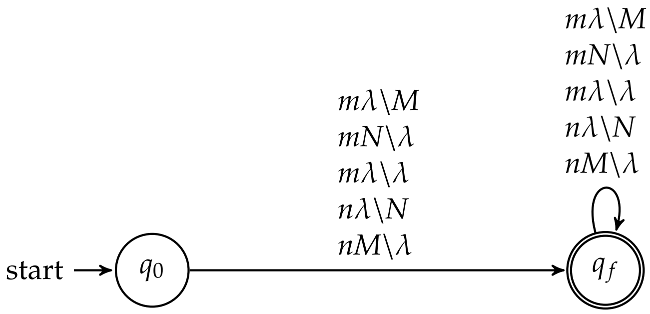
Voor het lezen van een n geldt grofweg het omgekeerde:

- Een N op de stapel zetten om bij te houden dat we een matige overgang hebben gelezen.

- Een M van de stapel halen zodat we mooie overgangen 'ongedaan maken'.

In dit geval mogen we geen gratis n-en lezen, dan zouden er immers meer matige dan mooie overgangen in het woord zitten! Let ook op het feit dat het lege woord λ *niet* in L' zit. Met andere woorden: het lege liedje is niet acceptabel. Wiskundig wordt dit uitgedrukt door het kleiner-dan-teken in de taaldefinitie.

Wanneer we bovenstaande redenatie volgen kunnen we de volgende automaat construeren voor L':



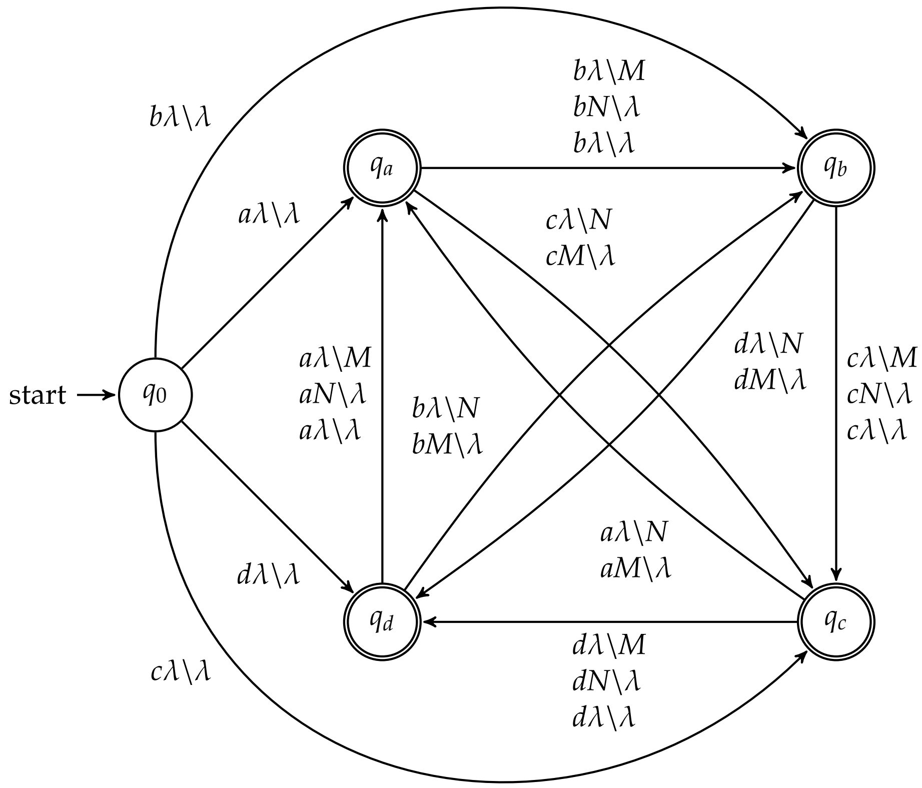
De twee toestanden zorgen ervoor dat we het lege woord niet kunnen lezen. In q_f kunnen we een lus maken aangezien zowel het woord als de stapel leeg moeten zijn om een woord te accepteren.

Deze automaat kunnen we uitbreiden naar onze taal voor popmuziek L. Het alfabet Σ bestaat nu uit $\{a, b, c, d\}$ in plaats van $\{m, n\}$. Sommige overgangen zijn nu wel toegestaan, andere niet. Maar het belangrijkste waar we op moeten letten is of een overgang mooi of matig is. Dit kunnen we op exact dezelfde manier bijhouden als bij de automaat hierboven:

- We zetten een M op de stapel of heffen een N op wanneer we een mooie overgang maken.

- We zetten een N op de stapel of heffen een M op wanneer we een matige overgang maken.

- Mooie overgangen mogen we gratis lezen.



Dit is een geldige stapelautomaat en hiermee is bewezen dat de taal voor acceptabele popmuziek L *niet* context vrij is.