LaTeX uitvoer genereren vanuit C programma's

Abstract
This article describes a simple way to generate LaTeX output from C programs.

Keywords
LaTeX, automatisch opmaken, C

Inleiding

Onlangs heeft de auteur een C-programma geschreven om mechanische en thermische eigenschappen van laminaten gemaakt van vezelversterkte kunststoffen te berekenen[1]. De uitvoer van dit programma bestaat uit een tweetal tabellen—die een overzicht van de opbouw van een laminaat en de globale eigenschappen weergeven—en een tweetal 6×6 matrices.

Om de resultaten van dit programma makkelijk in documenten te kunnen gebruiken, is het voorzien van de mogelijkheid om naast platte tekst ook HTML en LaTeX uitvoer te genereren. Deze laatste mogelijkheid is onderwerp van dit artikel.

Hoewel hier gebruik is gemaakt van de taal C, kan dezelfde aanpak ook zonder problemen gebruikt worden in andere programmeertalen.

Werkwijze

Uit oogpunt van flexibiliteit, is ervoor gekozen om géén compleet LaTeX document laten genereren, maar alleen datgene wat nodig is—bijvoorbeeld samengebonden in een `table`-omgeving. Aangezien het met behulp van `\input` heel makkelijk is om de gegenereerde code op te nemen in een document, leek mij dit de meest flexibele oplossing. Het geeft de gebruiker namelijk maximale vrijheid in het kiezen van documentstijlen en opties.

Bij uitvoeren van LaTeX-code vanuit C—en andere programmeertalen—moet rekening gehouden worden met een aantal valkuilen, waarover later meer. Daarom is het raadzaam om de volgende werkwijze aan te houden:

- □ maak een prototype van de gewenste uitvoer in LaTeX.
- □ vertaal dit naar uitvoerstatements in C broncode
- □ verfijn het resultaat

Op deze manier kun je het probleem opsplitsen in onafhankelijke delen, waarvan sommige geautomatiseerd kunnen worden.

Valkuilen

In *strings* in C[2] wordt de '\' gebruikt voor het aangeven van een aantal speciale karakters, met name aanhalingstekens en regeleindes, zie [KR90, blz. 51]. Om een letterlijke '\' in de uitvoer te krijgen, moet '\\' in de C tekenreeks gebruikt worden!

Voor het afdrukken van tekst word gebruik gemaakt van verschillende functies uit de C standaardbibliotheek, `puts` [KR90, blz. 336] en `printf` [KR90, blz. 331]. Met printf kun je ook gegevens uit het programma afdrukken, terwijl puts alleen maar letterlijke strings afdrukt. Ook voegt puts automatisch aan het eind van de regel een regeleinde (*newline*, '\n') toe, terwijl je die bij printf expliciet moet toevoegen. Als printf een '%' in de tekenreeks ziet, interpreteert hij dit als een *conversiespecificatie*—een plaats waar een variabele uit het programma afgedrukt moet worden. Om een letterlijke '%' in de uitvoer te krijgen, moet '%%' in deze tekenreeks staan. Dit alles is samengevat in tabel 1.

Omzetting	Opmerkingen
\% → %	
\ → \\	
" → \"	
' → \'	
% → %%	in `printf`

Tabel 1. omzettingen van LaTeX naar C-*strings*

Omzetten

Hoewel het natuurlijk mogelijk is om een dergelijke omzetting met de hand te doen, is het makkelijker en minder foutgevoelig om het aan de computer over

te laten. Het PERL-programma [SC97] weergegeven in figuur 1 is in staat om de conversie uit te voeren. Het echte werk wordt gedaan door de vervangingsopdrachten met behulp van *reguliere expressies* [SC97, hoofdstuk 7] zoals 's/\\%/%/g;'. Deze opdracht betekent: zoek alle tekenreeksen '\%' en vervang ze door '%'. Door een aantal van deze vervangingsopdrachten in de juiste volgorde te gebruiken, kan de LaTeX-code moeiteloos omgezet worden in C-statements die dezelfde uitvoer genereren.

```
#!/usr/bin/perl -pa
chomp;
s/\\%/%/g;
s/\\/\\\\/g;
s/\"/\\\"/g;
s/\'/\\\'/g;
if (/%[^%\s]/) {        # use printf
    s/%([\\\s])/%%\1/g;
    printf("printf(\"%s\\n\", );\n", $_);
} else {            # use puts
    printf("puts(\"%s\");\n", $_);
}
```

Figuur 1. conversie script

Neem als voorbeeld de LaTeX-code weergegeven in figuur 2.

```
\begin{table}[htbp]
  \centering
  \begin{tabular}{lll}
    \toprule % gebruik het booktabs package
    Symbool & Waarde\\
    \midrule
    $E_1$ & \%g MPa\\
    $E_2$ & \%g MPa\\
    $E^f_1$ & \%g MPa\\
    $E^f_2$ & \%g MPa\\
    $E^m$ & \%g MPa\\
    $\nu_{12}$ & \%g -\\
    $\nu^f_{12}$ & \%g -\\
    $\nu_m$ & \%g -\\
    $V^f$ & \%g \%\\
    $\eta$ & 0,5\\
    \bottomrule
  \end{tabular}
  \caption{\label{symbcomp}mechanische
    eigenschappen UD laminaten}
\end{table}
```

Figuur 2. voorbeeld invoer

Na omzetting door het in figuur 1 weergegeven script, ziet dit er uit als in figuur 3.

Hiermee is de omzetting natuurlijk nog niet compleet. De gegenereerde code moet nog opgenomen worden in een C-functie, en in de printf functies moeten nog de juiste variabelen toegevoegd worden achter de comma's. Maar het meest foutgevoelige gedeelte van het werk is al gedaan.

```
puts("\\begin{table}[htbp]");
puts("  \\centering");
puts("  \\begin{tabular}{lll}");
puts("    \\toprule % gebruik het booktabs package");
puts("    Symbool & Waarde\\\\");
puts("    \\midrule");
printf("    $E_1$ & %g MPa\\\\\n", );
printf("    $E_2$ & %g MPa\\\\\n", );
printf("    $E^f_1$ & %g MPa\\\\\n", );
printf("    $E^f_2$ & %g MPa\\\\\n", );
printf("    $E^m$ & %g MPa\\\\\n", );
printf("    $\\nu_{12}$ & %g -\\\\\n", );
printf("    $\\nu^f_{12}$ & %g -\\\\\n", );
printf("    $\\nu_m$ & %g -\\\\\n", );
printf("    $V^f$ & %g %%\\\\\n", );
puts("    $\\eta$ & 0,5\\\\");
puts("    \\bottomrule");
puts("  \\end{tabular}");
puts("  \\caption{\\label{symbcomp}mechanische");
puts("    eigenschappen UD laminaten}");
puts("\\end{table}");
```

Figuur 3. voorbeeld uitvoer

Conclusies

Toen de gewenste lay-out van de uitvoer eenmaal vast stond, was het relatief makkelijk om het produceren van de gewenste LaTeX-code op te nemen in het programma. De kwaliteit van de uiteindelijke uitvoer weegt zeer zeker op tegen de tijd die het kost om het op te zetten.

Het feit dat LaTeX-commando's uit platte tekst bestaan—en uitstekend gedocumenteerd zijn—maakt het genereren hiervan kinderspel. Samen met het `\input` mechanisme geeft het een ongekend gemak en flexibiliteit in het opnemen van programmaresultaten in documenten.

Referenties

[KR90] Brian Kernighan and Dennis M. Ritchie. *C handboek*. Academic Service, 1990. ISBN 90-6233-488-1.

[SC97] Randal L. Schwartz and Tom Christiansen. *Learning Perl*. O'Reilly & Associates, 1997. ISBN 1-56592-284-0.

Notes

1. `lamprop`, te vinden op `http://www.xs4all.nl/~rsmith/software/`
2. en vele andere programmeertalen.

Roland Smith
`rsmith@xs4all.nl`