

QI QUÆSTIONES INFORMATICAÆ

Volume 5 • Number 2

October 1987

M.J. Wagener	Rekenaar Spaaksintese: Die Omskakeling van Teks na Klank	1
E.C. Anderssen S.H. von Solms	A CAI Model of Space and Time with Special Reference to Field Battles	7
H.A. Goosen C.H. Hoogendoorn	A Model for Fault-Tolerant Computer Systems	16
E.M. Ehlers S.H. von Solms	Random Context Structure Grammars	23
C.S.M. Mueller	Set-Oriented Functional Style of Programming	29
P.J.S. Bruwer	User Attitudes: Main Reason Why Information Systems Fail	40
C.F. Scheepers	Polygon Shading on Vector Type Devices	46
G.R. Finnie	Novice Attitude Changes During a First Course in Computing: A Case Study	56
P.G. Clayton	Hands-On Microprogramming for Computer Science Students	63
	BOOK REVIEW	39
	CONFERENCE ANNOUNCEMENT	68

An official publication of the Computer Society of South Africa and of the South African Institute of Computer Scientists

'n Amptelike tydskrif van die Rekenaarvereeneging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

QUÆSTIONES INFORMATICÆ

An official publication of the Computer Society of South Africa
and of the South African Institute of Computer Scientists

'n Amptelike tydskrif van die Rekenaarvereniging van Suid-Afrika
en van die Suid-Afrikaanse Instituut van Rekenarwetenskaplikes

Editor

Professor G. Wiechers
INFOPLAN
Private Bag 3002
Monument Park 0105

Dr P.C. Pirow

Graduate School of Business Admin.
University of the Witwatersrand
P.O. Box 31170, Braamfontein, 2017

Editorial Advisory Board

Professor D.W. Barron
Department of Mathematics
The University
Southampton SO9 5NH, UK

Professor J.M. Bishop
Department of Computer Science
University of the Witwatersrand
1 Jans Smuts Avenue
2050 WITS

Professor K. MacGregor
Department of Computer Science
University of Cape Town
Private Bag
Rondebosch, 7700

Prof H. Messerschmidt
University of the Orange Free State
Bloemfontein, 9301

Professor S.H. von Solms
Department of Computer Science
Rand Afrikaans University
Auckland Park
Johannesburg, 2001

Professor M.H. Williams
Department of Computer Science
Heriot-Watt University, Edinburgh
Scotland

Production

Mr C.S.M. Mueller
Department of Computer Science
University of the Witwatersrand
2050 WITS

Subscriptions

Annual subscription are as follows:

	SA	US	UK
Individuals	R10	\$7	£5
Institutions	R15	\$14	£10

Computer Society of South Africa
Box 1714 Halfway House

Quæstiones Informaticæ is prepared by the Computer Science Department of the University of the Witwatersrand and printed by Printed Matter, for the Computer Society of South Africa and the South African Institute of Computer Scientists.

REKENAAR SPRAAKSINTESE: Die Omskakeling van Teks na Klank

M.J. Wagener

Departement Rekenaarwetenskap
Universiteit van Port Elizabeth

OPSOMMING

Verslag word gelewer oor die ontwikkeling en implementering van 'n stel letter-na-klank reëls vir die omskakeling van Afrikaanse teks na foneme. 'n Inleidende oorsig van rekenaar spraaksintese en 'n kort beskrywing van die verskillende metodes van teks na klank transkripsie, word ook gegee.

1. INLEIDING

Die afgelope vyf jaar het 'n nuwe afvoertoestel op die kommersiële mark verskyn, naamlik die elektroniese spraaksintiseerder. Daar is hoofsaaklik twee tipes spraaksintiseerders: die digitale spraaksintiseerder en die fonetiese spraaksintiseerder.

Die digitale spraaksintiseerder het die vermoë om 'n digitale voorstelling van stempatrone om te skakel in klank. Die gehalte van die spraak wat gelewer word, is direk afhanklik van die aantal waarnemings wat van die oorspronklike stempatrone gemaak word. Die digitale spraaksintiseerder kan dus spraak van baie hoë kwaliteit produseer indien groot hoeveelhede geheue beskikbaar is. Byvoorbeeld, as een greep benodig word vir 'n enkele waarneming en ag duisend waarnemings per sekonde gemaak word, sal ag kilogrepe geheue benodig word om een sekonde spraak te stoor.

Die tweede tipe spraaksintiseerder het die vermoë om foneme om te skakel in klank. Indien die fonetiese transkripsie van 'n woord in die fonetiese spraaksintiseerder ingevoer word, sal dit die betrokke woord "sê". Alhoewel hierdie tipe sintiseerder nie dieselfde kwaliteit spraak as die digitale sintiseerder lewer nie, vereis dit baie minder geheue. Die fonetiese transkripsie van die teks benodig ongeveer dieselfde hoeveelheid geheue as die teks self.

2. TEKS NA KLANK OMSKAKELING

Die maklikste manier om die fonetiese transkripsie van woorde te bepaal, is deur die woorde en hul fonetiese transkripsies in 'n woordeboek te stoor. Woorde kan getranskribeer word deur dit op te soek en die ooreenstemmende foneme te produseer. Hierdie metode verseker absolut akkurate en korrekte transkripsies en die gebruiker kan sy voorkeur in uitspraak van woorde bepaal. 'n Woordeboekstelsel het egter verskeie beperkende nadele.

- (1) Dit vereis genoegsame geheue om die woorde te stoor, asook effektiewe en vinnige middele om die woorde te onttrek. 'n Toename in die grootte van die woordeboek het ook 'n gepaardgaande afname in die spoed van woordherwinning tot gevolg.
- (2) Slegs woorde bevat in die woordeboek kan getranskribeer word.
- (3) 'n Minder beduidende nadeel is dat woorde met dieselfde spelling en verskillende uitspraak nie suksesvol hanteer kan word nie.

So 'n woordeboekstelsel is geïmplementeer deur die woorde en hul ooreenstemmende foneemstringe in 'n relasiedatabasis te stoor. 'n Volledige stelsel vir die instandhouding van die databasis asook die gebruik daarvan vir transkripsie, is ook voltooi. Die spesifieke implementasie het waardevolle hulp verleen met die beoordeling van die sukses van die letter-na-klank algoritme vir Afrikaans. (sien paragraaf 3)

'n Praktiese aanvaarbare metode van teks na klank omskakeling moet "slim" genoeg wees om te bepaal wat die uitspraak van 'n gegewe woord is. Die NRL stelsel van Elovitz, Johnson en ander ontwikkel vir Engels, beantwoord aan hierdie vereiste. Hierdie stelsel gebruik 'n stel van letter-na-klank reëls wat die uitspraak van 'n letter of groep letters bepaal deur te kyk na die konteks waarin die letter of groep letters in 'n woord voorkom. 'n Soortgelyke stelsel vir Afrikaans is ontwikkel, geïmplementeer en getoets. Die prestasie van so 'n stelsel verswak as die

groepe letters wat getranskribeer word te groot is of selfs volledige woorde gebruik word vir 'n spesifieke reël en kan in dié geval oorgaan tot 'n tipiese woordeboekstelsel. Dit is egter ook 'n voordeel van die metode dat ongewone woorde as reëls ingesluit kan word sonder om die logika en struktuur van die algoritme te benadeel. Die sukses van hierdie metode word bepaal deur die spesifieke taal waarvoor dit ontwikkel is: as daar 'n sterk verband tussen die spelling van woorde en die fonetiese transkripsie daarvan bestaan, kan beter resultate verkry word.

'n Derde bekende metode van teks na klank omskakeling doen 'n morfologiese ontleding van woorde. Enige woord kan opgebreek word in morfeme, wat basies ooreenstem met die voorvoegsels, stamwoorde en agtervoegsels van die woord. So 'n stelsel hou 'n woordeboek van morfeme. Dit is 'n beter uitweg as 'n woordeboek van woorde aangesien daar baie meer woorde in 'n taal bestaan as morfeme en die aantal morfeme in so 'n woordeboek 'n versadigingspunt bereik. Verder sal die weglatting van 'n morfeem uit die woordeboek nie die transkripsieproses beëindig nie, maar slegs 'n minder korrekte uitspraak tot gevolg hê. Die nadeel van morfologiese ontleding is dat dit nie altyd moontlik is om die morfeme van 'n betrokke woord te bepaal nie. Die proses van leksikale samestelling het gewoonlik tot gevolg dat die individuele morfeme van 'n woord verander word. Om die proses weer om te keer en die oorspronklike morfeme te verkry, word 'n stel samestellingsreëls gebruik. Die kompleksiteit van so 'n stel reëls is afhanklik van die grootte van die woordeboek van morfeme. Morfeme word getranskribeer deur weereens 'n stel van letter-na-klank reëls te gebruik of deur die foneemstringe van morfeme ook in die woordeboek van morfeme te stoor.

3. TEKS-NA-KLANK ALGORITME VIR AFRIKAANS

Die doel van 'n teks-na-klank algoritme is om die gaping tussen die geskrewe taal en die uitspraak daarvan te probeer oorbrug. Mense skryf nie 'n taal soos hulle dit praat nie. Dit is egter ook verkeerd om te sê dat daar geen verband tussen ortografie en uitspraak bestaan nie, want ons is tog in staat om te lees. 'n Meganiese omskakelingsalgoritme kan dus poog om die menslike leesproses na te boots. Die probleem is egter dat wanneer 'n persoon lees, sien hy nie net 'n string losstaande letters nie, maar wel woorde en die woorde het sekere betekenis in die konteks waarin dit voorkom. Die semantiek van 'n woord bepaal tot 'n mindere mate die uitspraak daarvan.

Die algoritme wat ontwikkel is vir Afrikaans, beskou 'n letter of 'n groep letters in die konteks van die woorde waarin dit voorkom, en maak daarvolgens 'n afleiding oor die uitspraak daarvan. Dit neem geensins die semantiek van die woorde in ag nie — dit sal dus byvoorbeeld nie die verskil in uitspraak tussen "ge'klik" en "gek'lik" kan bepaal nie. Daar word ook geensins gepoog om intonasie en ritme toe te pas nie. Die algoritme, op sigself, is dus 'n suiwer vertaling van letters na foneme. Die algoritme bestaan uit die toepassing van 'n stel reëls, en word aangegee in aanhangsel B. Elke reël het die volgende formaat:

$$A[B]C = D$$

B is die letter of letters wat omskep word in D — een of meer foneme soos aangegee in aanhangsel A. A en C is weereens 'n letter of groep letters wat B onderskeidelik moet voorafgaan en volg in die woorde. In die reëls self bevat A en C ook karakters wat nie as sulks in die konteks voorkom nie, maar wel ander groeppe karakters voorstel. Bv. 'n \$ in 'n reël dui enige medeklinker in die konteks aan en 'n & 'n klinker. Hierdie karakters word aangegee in aanhangsel B. Die reëls is gesorteer in groeppe volgens die eerste letter in B. Vir die transkripsie van 'n lettergreep in die teks, word daar slegs na die groep reëls wat ooreenstem met die eerste letter in die lettergreep, gekyk. Die reëls word een vir een, beginnende by die eerste reël in die groep, getoets en die eerste korrekte een word dan gebruik. (Die laaste reël in 'n groep sal altyd pas en die soekproses beëindig.) Die algoritme vir die transkripsie van 'n woorde is soos in figuur 1.

Die ontwikkeling van die stel reëls is begin deur 'n basiese stel reëls saam te stel. Hierin is bevat die reëls waarvoor daar 'n sterk verband tussen letters en foneme bestaan. Byvoorbeeld waar 'n medeklinker herhaal word, word slegs een klank gehoor, en 'n medeklinker alleen het gewoonlik 'n standaard uitspraak wat maklik afleibaar is van die letter. Dubbelklinkers soos "aa" en dubbelklanke soos "ooi" en "eu" kan ook sonder probleme getranskribeer word. Dit is egter enkel klinkers en veral die "e" waarvan die uitspraak in 'n spesifieke konteks moeilik bepaalbaar is. Om 'n nuwe reël op te stel moet daar gesoek word na 'n spesifieke konteks waarin 'n letter baie voorkom en dan ook meerendeels 'n spesifieke transkripsie het. Om dit met die hand te doen, is haas onmoontlik en dit was in die oopsig waar die woordeboekstelsel van onskatbare waarde was. Die woordeboek is gelaai met 'n groot steekproef van Afrikaanse woorde. Transkripsies is

gedoen met beide metodes en sodoende kon foute gemaak deur die stel reëls maklik uitgelig word en analyseer word om te kyk of spesifieke foute nie herhaal word nie. Verder is die suksespersentasie van die stel reëls ook bepaal en kon die byvoeging of weglatting van 'n reël onmiddellik gemotiveer word. Op so 'n wyse is die stel reëls uitgebrei en verfyn om maksimale sukses te verkry.

```

I = 1
J = lengte(WOORD)
terwyl I <= J
{
    beskou eerste reël met B[1] = WOORD[I]
    reël gevind = vals
    terwyl die korrekte reël nie gevind is nie
    (
        LEN = lengte(B)
        as WOORD[I,I+LEN-1] = B
            as (linker konteks pas) en (regter konteks pas)
            produseer D
            reël gevind = waar
        beskou volgende reël met B[1] = WOORD[I]
    )
    I = I + LEN
}

```

figuur 1

Die alfabet foneme wat gebruik word in die stel reëls is saamgestel deur die Internasionale Fonetiese Vereniging van Parys. Die spesifieke karakters wat gebruik word om elke foneem voor te stel, is gewysig om saam te val met die karakterstel van 'n standaard sleutelbord. Die stel foneme wat hanteer kan word deur 'n spesifieke spraaksintiseerder sal egter nie presies met hierdie alfabet ooreenstem nie. Spraaksintiseerders het 'n kleiner stel foneme en sluit slegs die foneme in wat gebruik word in die taal waarvoor dit ontwikkel is. 'n Verdere omskakelingsalgoritme is dus nodig om 'n afbeelding tussen "logiese" en "fisiiese" foneme te verkry. 'n Algoritme soortgelyk aan die letter-na-klank algoritme is ontwikkel om die "logiese" foneme, soos aangegee in aanhangsel A, af te beeld op die "fisiiese" foneme van die VOTRAX SC-OI spraaksintesevlokkie. Alhoewel daarin geslaag is om elke "logiese" foneem af te beeld, het die uitspraak van Afrikaanse woorde 'n duidelike Amerikaanse aksent.

4. GEVOLGTREKKINGS

Die stel van 161 letter-na-klank reëls vir Afrikaans het bewys dat alhoewel daar nie 'n direkte ooreenkoms tussen ortografie en uitspraak bestaan nie, dit tog moontlik is om vir enige gegewe teks byna korrekte transkripsies te produseer sonder die gebruik van 'n lywige en lomp woerdeboek. (Resultate van 'n prestasiemeting van die stel reëls sal in 'n volgende verslag bespreek word.) Die grootte van die stel reëls maak dit ook moontlik om 'n spraaksintese stelsel vir 'n mikrorekenaar te ontwikkel en die transkripsieproses intyds te bedryf.

Verdere byvoegings en veranderinge van die reëls mag moontlik beter transkripsies tot gevolg hê, maar sal geen noemenswaardige verbetering in die finale uitspraak veroorsaak nie. Die byvoeging van algoritmes vir intonasie en ritme asook die gebruik van 'n spraaksintiseerder wat spesifiek vir Afrikaans ontwikkel is, sal die kwaliteit van uitspraak sodanig verbeter dat dit die gebruik daarvan in die praktyk regverdig.

BIBLIOGRAFIE

1. Witten, Ian H., [1982], *Principles of Computer Speech*, Academic Press, New York.
2. Heath Company, [1981], *Voice Synthesis*, Michigan.
3. Elovitz, H.S., Johnson, R., Mchugh, A., Shore, J.E., [1976], Letter-to-Sound Rules for Automatic Translation of English Text to Phonetics, *IEEE Transactions on Acoustics, Speech and Signal processing*, ASSP-24, 6, 446-459, December.
4. Le Roux, T.H., Pienaar, P. de V., [1976], *Uitspraak-Woerdeboek van Afrikaans*, J L van Schaik, Pretoria .

AANHANGSEL A : Lys van fonetiese karakters

	Engels	Afrikaans		Engels	Afrikaans
klinkers					
a	father	<i>vat</i>)i	<i>toy</i>	
a:		<i>vate</i>):i	<i>nôi</i>	
A	sat		ui	<i>groei</i>	
e		<i>melaats</i>			
e:		<i>mening</i>			
E	get	<i>les</i>			
E:		<i>sê</i>	b	<i>back</i>	<i>bak</i>
]	about	<i>wig</i>	c	<i>dime</i>	<i>paadjie</i>
):		<i>wîe</i>	d	<i>dime</i>	<i>doen</i>
i	beet	<i>vlieg</i>	f	<i>fault</i>	<i>vis</i>
i:		<i>vlieë</i>	g	<i>goat</i>	<i>ghoen</i>
I	bit		H		<i>hand</i>
o		<i>orals</i>	h	<i>hand</i>	
o:		<i>bode</i>	j	<i>young</i>	<i>ja</i>
O		<i>deur</i>	k	<i>coat</i>	<i>kat</i>
Q		<i>brug</i>	l	<i>laugh</i>	<i>lam</i>
Q:		<i>brûe</i>	m	<i>sun</i>	<i>mond</i>
u	full	<i>ploeg</i>	n	<i>sun</i>	<i>nag</i>
u:	fool	<i>ploëë</i>	J		<i>kantjie</i>
y		<i>rusie</i>	N	<i>sung</i>	<i>ding</i>
y:		<i>uur</i>	p	<i>pack</i>	<i>pad</i>
])	lawn	<i>os</i>	r	<i>rate</i>	<i>rooi</i>
):		<i>môre</i>	s	<i>sue</i>	<i>son</i>
&	not		F	<i>leash</i>	<i>sjarmant</i>
^	but		z	<i>leisure</i>	<i>genre</i>
R	murder		t	<i>time</i>	<i>tien</i>
~		<i>kans</i>	T	<i>char</i>	<i>tjek</i>
			D	<i>just</i>	
			v	<i>vault</i>	<i>water</i>
			V		<i>twee</i>
			x		<i>goed</i>
			Z	<i>zoo</i>	<i>Zoeloe</i>
dubbelklanke					
au	how		\$	<i>father</i>	
ai	hide	<i>aitsa</i>	@	<i>think</i>	
a:i		<i>raai</i>	w	<i>wall</i>	
e:u		<i>leeu</i>	wh	<i>where</i>	
ji	gate	<i>lei,ly</i>	%	<i>silent</i>	<i>stil klank</i>
o:i		<i>mooi</i>		<i>sound</i>	
ou	lone	<i>koud</i>			
Qy		<i>muis</i>			

AANHANGSEL B : Letter-na-klank reëls vir Afrikaans

Die spesiale karakters wat in kolomme A en C gebruik word, het die volgende betekenisse:

- * - die begin of einde van 'n woord
- \$ - 'n medeklinker
- & - 'n klinker
- : - twee soortgelyke klinkers
- ; - 'n B, G, M of T (slegs in kolom A)
- ; - 'n frikativ (slegs in kolom C)
- . - twee karakters waarmee 'n woord nie kan begin nie (slegs in kolom C)

A	B	C	D	A	B	C	D
	AAI	a:i		;	E	R.]
*	AAN	;	a:	V	E		E
	AA	a:		V	ER]r
	AI	ai			Eë		e:]
*	A	*	a:		E	\$&&]
	A	SIE	a:	&	E	\$E]
	A	\$&&	a		E		e:
	A	\$&	a:	:	E]
	AE		a:]	&\$	E]
&\$	A	*	a		E	\$&	e:
	A	*	a:		E	.	E
	A		a		E	\$\$	E
					E]
	BB	b			FF		f
*	B	*	be:		FV		f
	BESIG		be:s]x	*	F	*	Ef
	B		b		F		f
*	C	*	se:	*	G	G	g
	C		k	*	GH		xe:
				\$	G	*	g]
					GE	*	x
	D	D			G		
A	DJIE	ici					
O	DJIE	ici					
	DJIE	ci		*	H	*	Ha:
*	D	*	de:		H		H
	DT	t			IE		i
	DR	dr			Ie		i]
	DW	dv		*	IN	;]~
	DERS	d]rs			I	*	i
	DERE	d]r]			I	\$IE	i
	DES	d]s			I	&	i
	DELIK	d]l]jk			I]
	DEL	d]l					
D	&	d					
D		t		*	J	*	je:
					J		j
	EEU	e:u					
	EE	e:			KK		k
	EI	ji		*	K	*	ka
	EU	O			KERE		k]r]
E	SIE*	e:			KEL		k]l
E	\$E*	e:			K		k
*	E	*	e:				
E	*]			LL		l
E	R*]		*	L	*	El
;	E	\$*	E		L		l
;	E		E				

	A	B	C	D		A	B	C	D
*	MM	*	m		*	TJ			T
M		*	Em		A	TЛЕ		ici	
MES			m]s		O	TЛЕ		ici	
M			m			TЛЕ		ci	
				*		T	*	te:	
						TT		t	
\$&	NN		n			TERE		t[r]	
&	NG		N			TES		t]s	
A	N	K	N			T		t	
O	NTЛЕ		iJci						
A	NTЛЕ		iJci						
O	NDЛIE		iJci			UU		y:	
A	NDЛIE		iJci			UI		Qy	
O	NDЛIE		Jci		*	U	*	y	
	NDЛIE		Jci			U	*	y	
*	NTЛIE		Jci			U	\$IE	y	
N	*		En			U	\$&	y:	
N			n			U		Q	
	OOI		o:i		*	V	*	fe:	
O		g	o:			V		f	
OEI			ui						
OE			u:						
OE			u			WW		v	
OU			ou		*	W	*	ve:	
OO			o:			WELS		v]ls	
*	ON	;)~			WEL		vEl	
*	O	*	o:			W		v	
O	*		o:						
O	\$I		o:						
O	\$&&		o		*	X	*	Eks	
O	\$E		o:			X		s	
O	\$&		o						
ONDER)n]r		*	Y	*	ji	
O)			Y		ji	
*	PP		p						
P	*		pe:		*	Z	*	ZEt	
P			p			Z		Z	
*	Q	*	ky			.		%%%%	
Q			k			,		%%	
						;		%%	
R	R	*	Er			:		%%%%	
*	R		r]m			!			
*&	RM		r]m			"			
\$&	RM		r]n			?		%%%%	
*&	RN		r]n		*	'N	*]	
\$&	RN		r]n			,			
R			r		*	-	*	minQs	
						-			
						ê		E:	
						ë		J	
SS			s			ö)	
SJ			F			ô)	
*	S	*	Es			ü		y	
S			s			û		Q:	
						ï		i	
						î		i	
						+		plQs	
						*		ma:l	

NOTES FOR CONTRIBUTORS

The purpose of the journal will be to publish original papers in any field of computing. Papers submitted may be research articles, review articles and exploratory articles of general interest to readers of the journal. The preferred languages of the journal will be the congress languages of IFIP although papers in other languages will not be precluded.

Manuscripts should be submitted in triplicate to:

Prof. G. Wiechers
INFOPLAN
Private Bag 3002
Monument Park 0105
South Africa

Form of manuscript

Manuscripts should be in double-space typing on one side only of sheets of A4 size with wide margins. Manuscripts produced using the Apple Macintosh will be welcomed. Authors should write concisely.

The first page should include the article title (which should be brief), the author's name and affiliation and address. Each paper must be accompanied by an abstract less than 200 words which will be printed at the beginning of the paper, together with an appropriate key word list and a list of relevant Computing Review categories.

Tables and figures

Tables and figures should not be included in the text, although tables and figures should be referred to in the printed text. Tables should be typed on separate sheets and should be numbered consecutively and titled.

Figures should also be supplied on separate sheets, and each should be clearly identified on the back in pencil and the authors name and figure number. Original line drawings (not photocopies) should be submitted and should include all the relevant details. Drawings etc., should be submitted and should include all relevant details. Photographs as illustrations should be avoided if possible. If this cannot be

avoided, glossy bromide prints are required.

Symbols

Mathematical and other symbols may be either handwritten or typewritten. Greek letters and unusual symbols should be identified in the margin. Distinction should be made between capital and lower case letters; between the letter O and zero; between the letter I, the number one and prime; between K and kappa.

References

References should be listed at the end of the manuscript in alphabetic order of the author's name, and cited in the text in square brackets. Journal references should be arranged thus:

1. Ashcroft E. and Manna Z., The Translation of 'GOTO' Programs to 'WHILE' programs., *Proceedings of IFIP Congress 71*, North-Holland, Amsterdam, 250-255, 1972.
2. Bohm C. and Jacopini G., Flow Diagrams, Turing Machines and Languages with only Two Formation Rules., *Comm. ACM*, 9, 366-371, 1966.
3. Ginsburg S., Mathematical Theory of Context-free Languages, McGraw Hill, New York, 1966.

Proofs

Proofs will be sent to the author to ensure that the papers have been correctly typeset and *not* for the addition of new material or major amendment to the texts. Excessive alterations may be disallowed. Corrected proofs must be returned to the production manager within three days to minimize the risk of the author's contribution having to be held over to a later issue.

Only original papers will be accepted, and copyright in published papers will be vested in the publisher.

Letters

A section of "Letters to the Editor" (each limited to about 500 words) will provide a forum for discussion of recent problems.

