Proces algebra met interrupt mechanisme

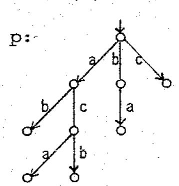
6.2 Proces algebra met interrupt mechanisme

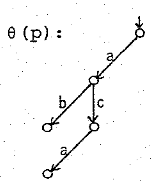
J.C.M. Baeten, J.A. Bergstra en J.W. Klop,
Centrum voor Wiskunde & Informatica

SAMENVATTING. We introduceren een mechanisme om prioriteiten te beschrijven in ACP (algebra van communicerende processen). Hierbij kan voorgeschreven worden dat sommige acties in een proces prioriteit hebben over andere acties, in toestanden waarin zich in het proces een non-deterministische keuze voordoet. Dit mechanisme, dat door middel van vergelijkingen beschreven wordt, kan gebruikt worden om de werking van 'interrupts' in een gedistribueerd systeem te modelleren. Dit wordt toegelicht aan de hand van een eenvoudig voorbeeld.

0. Inleiding. De basis van het in dit artikel beschreven mechanisme om acties met prioriteiten te behandelen, wordt gevormd door ACP, een axioma systeem voor processen die op asynchrone wijze coopereren via synchrone communicatie. Een informele uiteenzetting van ACP is gegeven tijdens het NGI-SION congres van 1984; zie [1]. Voor een meer volledige presentatie van ACP verwijzen we naar [2]. Anders dan in [1] worden in dit artikel geen onzichtbare stappen (τ-stappen) beschouwd; anders gezegd, abstractie van interne stappen is nu niet aan de orde.

Gegeven ACP met een alphabet A van acties, is het uitgangspunt van de onderhavige onderneming een partiële ordening $>$ op A. De intuitieve betekenis van $>$ is dat in een proces $a_1x_1 + \ldots + a_nx_n$ alleen die a_i $(i = 1,\ldots,n)$ gekozen kunnen worden waarvoor er geen a_j $(j = 1,\ldots,n)$ is met $a_j > a_i$. De operator die zorgt voor de toekenning van de door $>$ gegeven prioriteiten, is θ. Bijvoorbeeld, als $a > b$ en $a > c$, dan $\theta(a + b) = a$, en voor p als in Figuur 1(a) is $\theta(p)$ als in 1(b).





Figuur 1. (a) (b)

Om een algebraische karakterisering van θ te geven met uitsluitend vergelijkingen, die aan ACP toegevoegd het axioma systeem ACP_θ (algebra van communicerende processen met prioriteiten) opleveren, introduceren we een hulpoperator $\lhd$ ('tenzij') op processen. De betekenis van $x \lhd y$ ('x tenzij y') is als volgt: alle initiële acties van x die gemajoreerd worden (in de zin van >) door een initiële actie van y, worden geëlimineerd. Deze eliminatie gebeurt door dergelijke initiële acties van x te vervangen door δ, het symbool dat in ACP 'deadlock' van een proces aanduidt. Bijvoorbeeld, als $a > b > c$ dan $(ax + by + cz) \lhd (bp + cq) = ax + by + \delta z = ax + by$. Hierbij zijn $a,b,c \in A$ en x,y,z,p,q zijn willekeurige processen.

De inhoud van dit artikel is als volgt: Tabel 2 bevat het axioma systeem ACP_θ. Sectie 1 geeft een kort commentaar hierop, alsmede enkele voorbeelden ter illustratie. In sectie 2 wordt een meer uitgebreid voorbeeld behandeld, via ACP_θ, van een eenvoudige situatie waarin een interrupt mechanisme optreedt. Voor een volledige presentatie van ACP_θ, incluis een bewijs van de consistentie van dit systeem via een herschrijfregel argument, verwijzen we naar [3]. Hierin worden ook enkele andere iets meer uitgebreide voorbeelden van situaties met interrupts behandeld via ACP_θ.

1. Uiteenzetting van ACP_θ. In Tabel 2 variëren a,b,c over het actie alphabet A, dat steeds eindig wordt verondersteld. De variabelen x,y,z staan voor processen (elementen uit een proces algebra, i.e. een model van het axioma systeem). De signatuur van ACP_θ is als volgt:

$+$	*alternatieve compositie (som)*
$\cdot$	*sequentiële compositie (product)*
$\Vert$	*parallelle compositie ('merge')*
$\lfloor\!\lfloor$	*left merge*
$\vert$	*communicatie merge*
∂_H	*encapsulatie*
δ	*deadlock*
θ	*prioriteitstoekenning*
$\lhd$	*tenzij*

Tabel 1.

ACP_θ

$x+y = y+x$	A1	$a \lhd b = a$ if not $(a<b)$	P1
$x+(y+z) = (x+y)+z$	A2	$a \lhd b = \delta$ if $a<b$	P2
$x+x = x$	A3	$x \lhd yz = x \lhd y$	P3
$(x+y)z = xz+yz$	A4	$x \lhd (y+z) = (x \lhd y) \lhd z$	P4
$(xy)z = x(yz)$	A5	$xy \lhd z = (x \lhd z)y$	P5
$x+\delta = x$	A6	$(x+y) \lhd z = x \lhd z + y \lhd z$	P6
$\delta x = \delta$	A7		
$a \mid b = b \mid a$	C1		
$(a \mid b) \mid c = a \mid (b \mid c)$.	C2		
$\delta \mid a = \delta$	C3		
$x \parallel y = x \mathbin{\lfloor\!\lfloor} y + y \mathbin{\lfloor\!\lfloor} x + x \mid y$	CM1	$\theta(a) = a$	TH1
$a \mathbin{\lfloor\!\lfloor} x = ax$	CM2	$\theta(xy) = \theta(x) \cdot \theta(y)$	TH2
$ax \mathbin{\lfloor\!\lfloor} y = a(x \parallel y)$	CM3	$\theta(x+y) = \theta(x) \lhd y + \theta(y) \lhd x$	TH3
$(x+y) \mathbin{\lfloor\!\lfloor} z = x \mathbin{\lfloor\!\lfloor} z + y \mathbin{\lfloor\!\lfloor} z$	CM4		
$(ax) \mid b = (a \mid b)x$	CM5		
$a \mid (bx) = (a \mid b)x$	CM6		
$(ax) \mid (by) = (a \mid b)(x \parallel y)$	CM7		
$(x+y) \mid z = x \mid z + y \mid z$	CM8		
$x \mid (y+z) = x \mid y + x \mid z$	CM9		
$\partial_H(a) = a$ if $a \notin H$	D1		
$\partial_H(a) = \delta$ if $a \in H$	D2		
$\partial_H(x+y) = \partial_H(x) + \partial_H(y)$	D3		
$\partial_H(xy) = \partial_H(x) \cdot \partial_H(y)$	D4		

Tabel 2.

In producten $x \cdot y$ wordt $\cdot$ meestal weggelaten. De partiële ordening $>$ is a priori gegeven. Steeds wordt van $>$ geeist dat $\delta < a$ voor alle $a \neq \delta$. (Merk op dat deze eis al besloten ligt in axioma A6, $x + \delta = x$, dat zegt dat de optie δ in aanwezigheid van andere opties in feite geen werkelijke optie is.)

De axioma's P1,P2 zijn in feite axioma schema's die staan voor eindig veel axioma's (eindig, omdat A eindig verondersteld is). Bijvoorbeeld als $A = \{a,b,\delta\}$, met $>$ gegeven door $a > b > \delta$, dan staat P1 voor de axioma's: $a \lhd a = a$, $a \lhd b = a$, $a \lhd \delta = a$, $b \lhd b = b$, $b \lhd \delta = \delta$, $\delta \lhd \delta = \delta$, terwijl P2 het axioma schema is met instanties: $\delta \lhd b = \delta$, $\delta \lhd a = \delta$, $b \lhd a = \delta$.

De interpretatie van de operator $\lhd$, boven reeds aangegeven, verklaart axioma's P1-6. Zo is P3 het gevolg van het feit dat alleen *initiële* acties van y er toe doen, en P5 zegt dat de 'filterende' werking van y in $x \lhd y$ alleen betrekking heeft op de *initiële* acties van x. Axioma P4 zegt dat het simultaan verkregen effect van y in $x \lhd y$ ook verkregen kan worden door delen (in de zin van +) van y in opeenvolging te laten werken.

Van de axioma's TH1-3 voor θ is TH3 het meest bijzonder. Duidelijk is dat θ niet over + distribueert, omdat er in $\theta(x+y)$ een wisselwerking is tussen de restricties die x en y elkaar qua prioriteiten opleggen, terwijl in $\theta(x) + \theta(y)$ deze wisselwerking ontbreekt. Met een 'correctiefactor' hebben we echter wel een 'distributieve' wet:

$$\theta(x+y) = \theta(x) \lhd (x+y) + \theta(y) \lhd (x+y).$$

Dit is een equivalente formulering van TH3, want:

$$\theta(x) = \theta(x+x) \overset{\text{TH3}}{=} \theta(x) \lhd x + \theta(x) \lhd x = \theta(x) \lhd x$$

en hiermee

$$\theta(x) \lhd (x+y) \overset{\text{P4}}{=} (\theta(x) \lhd x) \lhd y = \theta(x) \lhd y.$$

<u>Voorbeelden</u>. Met de partiële ordening $b < a$, $c < a$ hebben we:

(i) $\theta(a+b) = \theta(a) \lhd b + \theta(b) \lhd a = a \lhd b + b \lhd a = a + \delta = a.$

(ii) $\theta(b+c) = \theta(b) \lhd c + \theta(c) \lhd b = b \lhd c + c \lhd b = b + c.$

(iii) $\theta(b(a+c)) = \theta(b) \cdot \theta(a+c) = b(\theta(a) \lhd c + \theta(c) \lhd a) =$

$b(a \lhd c + c \lhd a) = b(a + \delta) = ba.$

(iv) $\theta(a+b+c) = \theta(a) \triangleleft (b+c) + \theta(b+c) \triangleleft a =$

$(\theta(a) \triangleleft b) \triangleleft c + (\theta(b) \triangleleft c + \theta(c) \triangleleft b) \triangleleft a = (a \triangleleft b) \triangleleft c + (b \triangleleft c + c \triangleleft b) \triangleleft a =$

$a \triangleleft c + (b+c) \triangleleft a = a + \delta = a.$

(v) $\theta(a+b+c) = \theta(a+b) \triangleleft c + \theta(c) \triangleleft (a+b) = (\theta(a) \triangleleft b + \theta(b) \triangleleft a) \triangleleft c$

$+ \theta(c) \triangleleft a) \triangleleft b = (a \triangleleft b + b \triangleleft a) \triangleleft c + (c \triangleleft a) \triangleleft b = (a + \delta) \triangleleft c + \delta \triangleleft b =$

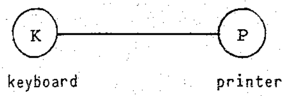
$a \triangleleft c + \delta \triangleleft c + \delta = a + \delta + \delta = a.$

De laatste twee voorbeelden tonen aan dat enige voorzichtigheid geboden lijkt omdat θ van een som op verschillende manieren afgebroken kan worden. In [3] is echter bewezen dat elke manier om de operatoren $\theta, \triangleleft$ uit een term te verdrijven tot hetzelfde resultaat leidt. (D.w.z. het op voor de hand liggende wijze met ACP_θ geassocieerde herschrijfsysteem is confluent en heeft de terminatie eigenschap.)

2. Uitgebreid voorbeeld. Zij D een eindige verzameling data en zij $\alpha = \langle d_0, d_1, d_2, \ldots \rangle$ een oneindige rij data uit D. De printer P (zie Figuur 2) moet deze rij data afdrukken, maar kan hierbij onderbroken worden door K (keyboard). Het alfabet A bestaat uit de volgende acties (waarbij de communicatie acties cursief zijn weergegeven):

acties van K:	k(BR)	(key in BREAK)
	k(SP)	(key in START PRINTING)
	s(BR)	(stuur BR naar printer)
	s(SP)	(stuur SP naar printer)

acties van P:	p(d)	(print d)
	r(BR)	(ontvang BR van keyboard)
	r(SP)	(ontvang SP van keyboard)



Figuur 2.

K is gedefinieerd door de recursievergelijking:

$$\boxed{K = (k(BR)\cdot \mathit{s(BR)} + k(SP)\cdot \mathit{s(SP)})\cdot K}$$

De printer P is gedefinieerd door het volgende stelsel van recursievergelijkingen:

$$\boxed{\begin{array}{ll} P = W_0 & \\ W_i = \mathit{r(SP)}\cdot P_i + \mathit{r(BR)}\cdot W_i & (i \geqslant 0) \\ P_i = p(d_i)\cdot P_{i+1} + \mathit{r(BR)}\cdot W_i + \mathit{r(SP)}\cdot P_i & (i \geqslant 0) \end{array}}$$

waarbij P_i de 'printing' toestand van P is na $d_0,d_1,\ldots,d_{i-1}$ geprint te hebben en W_i de 'waiting' toestand van P is na $d_0,d_1,\ldots,d_{i-1}$ geprint te hebben.

De communicatie functie is gedefinieerd door

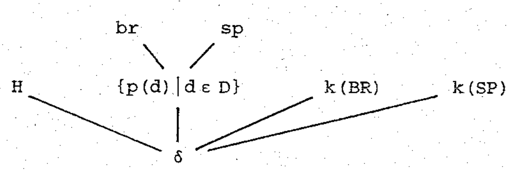
$$\mathit{s(BR)} \mid \mathit{r(BR)} = br$$

$$\mathit{s(SP)} \mid \mathit{r(SP)} = sp$$

en dit zijn de enige communicaties, i.e. alle andere communicaties geven δ.

We zijn nu geinteresseerd in de cooperatie $K\|P$ waarbij onsuccesvolle communicaties weggewerkt dienen te worden. D.w.z. we beschouwen $\partial_H(K\|P)$ met $H = \{\mathit{s(BR)},\mathit{r(BR)},\mathit{s(SP)},\mathit{r(SP)}\}$.

Vervolgens definiëren we een partiële ordening > op A die de prioriteiten vastlegt. De enige prioriteiten zijn die van 'br' en 'sp' over de print acties p(d): br *moet* het printen interrumperen, en *s(SP) moet* ontvangen worden door de printer omdat anders K geblokkeerd is (tussen k(SP) en *s(SP)*) en K daardoor geen 'break' opdracht kan geven. De partiële ordening > wordt gegeven door Figuur 3 waarin $a > b$ als a boven b ligt. (Als V een verzameling is betekent $V > a$ dat $v > a$ voor alle $v \in V$.)



Figuur 3.

Het bedoelde systeem wordt nu beschreven door

$$\theta(\partial_H(K\|P)).$$

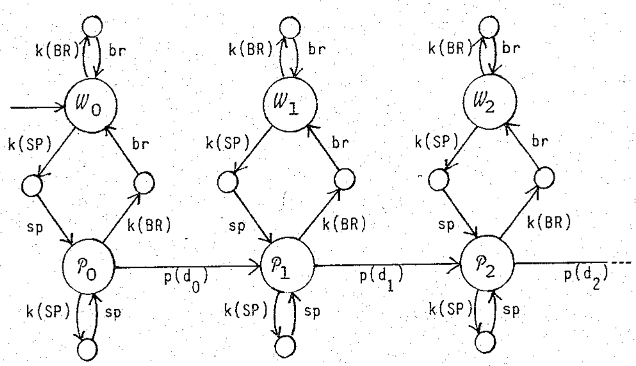
Stellen we nu, als afkorting,

$$\mathcal{P}_i = \theta(\partial_H(K\|P_i))$$
$$\mathcal{W}_i = \theta(\partial_H(K\|W_i)) \qquad (i \geqslant 0)$$

dan komen we met enig rekenwerk waarbij uitsluitend de axioma's van ACP_θ gebruikt worden, tot de volgende vergelijkingen (voor $i \geqslant 0$):

$$\mathcal{W}_i = k(BR)\cdot br\cdot\mathcal{W}_i + k(SP)\cdot sp\cdot\mathcal{P}_i$$
$$\mathcal{P}_i = p(d_i)\cdot\mathcal{P}_{i+1} + k(BR)\cdot br\cdot\mathcal{W}_i + k(SP)\cdot sp\cdot\mathcal{P}_i.$$

Deze vergelijkingen, die de proces graaf (toestandsovergangsdiagram) als in Figuur 4 voorstellen, definiëren juist het intuitief verwachte resultaat van de beschouwde cooperatie.



Figuur 4.

LITERATUUR

[1] BERGSTRA, J.A. & J.W. KLOP, Proces algebra met communicatie en abstractie, Proceedings NGI-SION 1984 Informatica Symposium, p.262-271, NGI Amsterdam 1984.

[2] BERGSTRA, J.A. & J.W. KLOP, Algebra of Communicating Processes, Report CS-R8421, Centrum voor Wiskunde en Informatica, Amsterdam 1984. To be published in: Proceedings of the CWI Symposium Mathematics and Computer Science (eds. J.W. de Bakker, M. Hazewinkel and J.K. Lenstra), North-Holland, Amsterdam 1985.

[3] BAETEN, J.C.M., BERGSTRA, J.A. & J.W. KLOP, Syntax and defining equations for an interrupt mechanism in process algebra, Report CS-R8503, Centrum voor Wiskunde en Informatica, Amsterdam 1985.