Hoorcollege II

- Programma's en gedrag
- Tot en met de programmeertalen PGLEc en PGLEcw

3 Gedragsexpressies voor Programma's met Inactie

Twee programma's kunnen gelijk gedrag vertonen en toch niet structureel congruent zijn. Bv.:

$+a;!;!$ en $a;!;!$ en $-a;!;!$ en $a;!$,

$\#2;u;!$ en $!;u;!$,

a^ω en $+a^\omega$ en $-a^\omega$ (er 'gebeurt' steeds een a),

$+a;u^\omega$ en $a;u^\omega$ en $-a;u^\omega$.

3.1 Primitieven van BPPA

- BPPA (basic polarized process algebra) is gebaseerd op een collectie A van basisinstructies.
- De elementen van A worden *acties* genoemd.
- BPPA heeft als primitieven twee constanten om terminatie en inactie te modelleren, en twee methoden van samenstelling (operatoren) om grotere *gedragsexpressies* te kunnen maken.

3.1 Primitieven van BPPA

Terminatie. S (stop)

Inactief gedrag. D (divergentie of inactie)

Postconditionele compositie. Voor elke actie $a \in A$ en gedragsexpressies P en Q

$$P \trianglelefteq a \trianglerighteq Q$$

Actieprefix. Voor elke $a \in A$ en gedragsexpressie P

$$a \circ P$$

Afspraak: $a \circ P = P \trianglelefteq a \trianglerighteq P$.

3.1 Primitieven van BPPA

- S en D zijn bevat in BPPA voor elke keuze van A.
- Elk gedrag wordt verkregen uit S en/of D door middel van postconditionele compositie.
- Een enkel element van A is geen gedragsexpressie.

Voorbeeld:

$$S \trianglelefteq a \trianglerighteq (S \trianglelefteq b \trianglerighteq (c \circ D)).$$

Afspraak: actieprefix *bindt sterker* dan postconditionele compositie (dus we mogen

$$S \trianglelefteq a \trianglerighteq (S \trianglelefteq b \trianglerighteq c \circ D)$$

schrijven).

3.2 Projecties en oneindig gedrag

Projectieoperatoren op eindig gedrag:

$$\begin{aligned}
\pi_0(P) &= D,\\
\pi_{n+1}(S) &= S,\\
\pi_{n+1}(D) &= D,\\
\pi_{n+1}(P \trianglelefteq a \trianglerighteq Q) &= \pi_n(P) \trianglelefteq a \trianglerighteq \pi_n(Q),
\end{aligned}$$

Dus

$$\pi_{n+1}(a \circ P) = a \circ \pi_n(P).$$

3.2 Projecties en oneindig gedrag

Voorbeelden (we gebruiken in herhaalde toepassingen van actieprefix geen haakjes):

$$\pi_2(a \circ D) = a \circ D,$$

$$\pi_3(b \circ S \trianglelefteq c \trianglerighteq d \circ e \circ f \circ S) = b \circ S \trianglelefteq c \trianglerighteq d \circ e \circ D,$$

$$\pi_4(c \circ D \trianglelefteq a \trianglerighteq b \circ e \circ f \circ f \circ D) = c \circ D \trianglelefteq a \trianglerighteq b \circ e \circ f \circ D.$$

3.2 Projecties en oneindig gedrag

Een *projectieve rij* $(P_n)_{n\in\mathbb{N}}$ is een rij $P_0, P_1, P_2, \ldots$ met voor elke $n \in \mathbb{N}$:

- P_n is een eindig gedrag (dus een BPPA-expressie), en
- $\pi_n(P_{n+1}) = P_n$.

Voorbeeld:

$$D, a \circ D, a \circ a \circ D, \ldots$$

Afspraak:

$$(P_n)_{n\in\mathbb{N}} = (Q_n)_{n\in\mathbb{N}} \text{ als } P_n = Q_n \text{ voor alle } n \in \mathbb{N}.$$

3.3 Vergelijkingen voor gedragsextractie

Gedragsextractie op programmaobjecten:

$$
\begin{array}{rcl@{\qquad}rcl}
|!| & = & S & |!;X| & = & S \\
|a| & = & a \circ D & |a;X| & = & a \circ |X| \\
|+a| & = & a \circ D & |+a;X| & = & |X| \trianglelefteq a \trianglerighteq |\#2;X| \\
|-a| & = & a \circ D & |-a;X| & = & |\#2;X| \trianglelefteq a \trianglerighteq |X| \\
|\#k| & = & D & |\#0;X| & = & D \\
 & & & |\#1;X| & = & |X| \\
 & & & |\#k+2;u| & = & D \\
 & & & |\#k+2;u;X| & = & |\#k+1;X|
\end{array}
$$

3.3 Vergelijkingen voor gedragsextractie

Merk op: als $X =_{sc} Y$, dan geldt ook dat $|X| = |Y|$.

Enkele voorbeelden:

$$|(\#0)^\omega| = |\#0; (\#0)^\omega| = D,$$

$$\begin{aligned} |-a; b; c| &= |\#2; b; c| \trianglelefteq a \trianglerighteq |b; c| \\ &= |\#1; c| \trianglelefteq a \trianglerighteq b \circ |c| \\ &= |c| \trianglelefteq a \trianglerighteq b \circ c \circ D \\ &= c \circ D \trianglelefteq a \trianglerighteq b \circ c \circ D. \end{aligned}$$

3.3 Vergelijkingen voor gedragsextractie

Indien de vergelijkingen oneindig vaak kunnen worden toegepast zonder tot gedrag te leiden, spreken we af dat het geëxtraheerde gedrag gelijk is aan D.

Bijvoorbeeld:

$$|(\#1)^\omega| = |\#1; (\#1)^\omega| = |(\#1)^\omega| = \ldots = D,$$

$$|(\#2; a)^\omega| = |\#2; a; (\#2; a)^\omega| = |\#1; (\#2; a)^\omega| = |(\#2; a)^\omega| = \ldots = D.$$

3.3 Vergelijkingen voor gedragsextractie

Ook mogelijk is dat de vergelijkingen voor gedragsextractie oneindig vaak kunnen worden toegepast en zo tot een steeds groter gedrag leiden.

Bijvoorbeeld:

$$|a^\omega| = |a; a^\omega| = a \circ |a^\omega| = a \circ a \circ |a^\omega| = \ldots$$

In zulke gevallen spreken we af dat het gedrag van X wordt gerepresenteerd door de projectieve rij $(\pi_n(|X|))_{n\in\mathbb{N}}$.

3.4 Equivalentie en congruentie op gedragsexpressies

- X en Y zijn *gedragsequivalent*, indien $|X| = |Y|$.
- Gedragsequivalentie van X en Y wordt genoteerd met $X \equiv_{ge} Y$.

Bijvoorbeeld:

$$\begin{aligned} |a; \#2; \#1; b; c| &= a \circ |\#2; \#1; b; c| \\ &= a \circ |\#1; b; c| \\ &= a \circ |b; c| \\ &= |a; b; c|. \end{aligned}$$

Dus $a; b; c \equiv_{ge} a; \#2; \#1; b; c$.

3.4 Equivalentie en congruentie op gedragsexpressies

- Gedragsequivalentie is geen congruentie. Bijvoorbeeld: $+a \equiv_{ge} a$, maar

$$|+a; b| = b \circ D \trianglelefteq a \trianglerighteq D \neq a \circ b \circ D = |a; b|.$$

- *Gedragscongruentie* is de grootste congruentierelatie die bevat is in gedragsequivalentie.

- Gedragscongruentie van X en Y wordt genoteerd met $X =_{gc} Y$.

- Het is bewijsbaar dat $X =_{gc} Y$ als voor alle $n, m \in \mathbb{N}$,

$$\#n; X; !^m \equiv_{ge} \#n; Y; !^m.$$

5 Programmeertalen

Ter herinnering: PGLB kent als methode van samenstelling alleen ; (aaneenschakeling) en naast de primitieve instructies van PGA, i.e.,

$$a, \pm a, !, \#n$$

de terugwaartse sprong $\backslash\#n$.

Definitie: voor elk PGLB-programma X geldt dat $|X|_{pglb} = |\texttt{pglb2pga}(X)|$.

Net zo: voor elk PGLA-programma X geldt dat $|X|_{pgla} = |\texttt{pgla2pga}(X)|$.

5.4 Conventionele terminatie in PGLC en PGLD

PGLC is een variatie op PGLB: in PGLC is de terminatie-instructie $!$ weggelaten.

- $+a$ gedraagt zich als $+a;!;!$
- $a;\#10$ en $a;\backslash\#10$ gedragen zich als $a;!$

5.4 Conventionele terminatie in PGLC en PGLD

De projectie `pglc2pglb` : $\mathrm{PGLC} \to \mathrm{PGLB}$ is als volgt gedefinieerd:

$$\texttt{pglc2pglb}(u_1; \ldots; u_k) = \psi_1(u_1); \ldots; \psi_k(u_k); !; !$$

met voor $j = 1, \ldots, k$,

$$\begin{aligned} \psi_j(\#l) &= ! \text{ als } j + l > k, \\ \psi_j(\backslash\#l) &= ! \text{ als } l \geq j, \\ \psi_j(u) &= u \text{ anders.} \end{aligned}$$

5.4 Conventionele terminatie in PGLC en PGLD

Merk op: de uitbreiding van $\psi_1(u_1); \ldots ; \psi_k(u_k)$ met $!;!$ behoudt terminatiegedrag.

Voorbeelden:

$$\texttt{pglc2pglb}(+b) = +b;!;!,$$

$$\texttt{pglc2pglb}(+c;\#10;\backslash\#1;-c;\#2;+b) = +c;!;\backslash\#1;-c;!;+b;!;!.$$

$$|X|_{pglc} = |\texttt{pglc2pglb}(X)|_{pglb}$$

5.4 Conventionele terminatie in PGLC en PGLD

PGLD is een variatie op PGLC: in plaats van voorwaartse en achterwaartse sprongen kent PGLD de *absolute sprongopdracht* $\#\#n$ voor $n \in \mathbb{N}$.

- $a; b; \#\#0$ en $a; b; \#\#4$ gedragen zich als $a; b; !$
- $a; b; \#\#1$ gedraagt zich als $(a; b)^\omega$

Dus $\#\#n$ schrijft een sprong voor naar de n-de instructie als die er is, en anders terminatie.

5.4 Conventionele terminatie in PGLC en PGLD

De projectie `pgld2pglc` : $\mathrm{PGLD} \to \mathrm{PGLC}$ is als volgt gedefinieerd:

$$\texttt{pgld2pglc}(u_1; \ldots; u_k) = \psi_1(u_1); \ldots; \psi_k(u_k),$$

met voor $j = 1, \ldots, k$,

$$\begin{aligned} \psi_j(\#\#l) &= \#l - j \ \text{ als } l \geq j, \\ \psi_j(\#\#l) &= \backslash\#j - l \ \text{ als } l < j, \\ \psi_j(u) &= u \ \text{ anders.} \end{aligned}$$

5.4 Conventionele terminatie in PGLC en PGLD

Voorbeeld:

$$\texttt{pgld2pglc}(a; +b; \#\#1; \#\#8; c; \#\#5; d) = a; +b; \backslash\#2; \#4; c; \backslash\#1; d.$$

$$|X|_{pgld} = |\texttt{pgld2pglc}(X)|_{pglc}$$

5.5 Labels en goto's in PGLDg en PGLE

PGLDg is een variatie op PGLD: in plaats van absolute sprongen kent PGLDg *goto*'s $\#\#\pounds n$ en *labels* $\pounds n$ voor $n \in \mathbb{N}$; bovendien is $!$ weer toegevoegd.

- $a; \pounds 0; b; c; \#\#\pounds 0$ gedraagt zich als $a; (b; c)^\omega$
- $a; \pounds 0; b; c; \#\#\pounds 1$ gedraagt zich als $a; b; c; !$
- $a; \pounds 0; b; \pounds 0; c; \#\#\pounds 0$ gedraagt zich ook als $a; (b; c)^\omega$

5.5 Labels en goto's in PGLDg en PGLE

De projectie `pgldg2pgld` : $\mathrm{PGLDg} \to \mathrm{PGLD}$ is als volgt gedefinieerd:

$$\texttt{pgldg2pgld}(u_1; \ldots; u_k) = \psi_1(u_1); \ldots; \psi_k(u_k)$$

met voor $j = 1, \ldots, k$

$$\begin{aligned} \psi_j(!) &= \#\#0, \\ \psi_j(\#\#\pounds n) &= \#\#\texttt{labelpositie}(n), \\ \psi_j(\pounds n) &= \#\#j+1, \\ \psi_j(u) &= u \ \text{ anders} \end{aligned}$$

met `labelpositie`(n) het kleinste getal i waarvoor $u_i = \pounds n$ en 0 als er geen instructie $\pounds n$ is.

5.5 Labels en goto's in PGLDg en PGLE

Voorbeeld:

$$\texttt{pgldg2pgld}(\pounds 0; +a; \#\#\pounds 1; \#\#\pounds 0) = \#\#2; +a; \#\#0; \#\#1$$

$|X|_{pgldg} = |\texttt{pgldg2pgld}(X)|_{pgld}$.

5.5 Labels en goto's in PGLDg en PGLE

PGLE is een deeltaal van PGLDg: elke testinstructie moet worden gevolgd door een goto of door terminatie.

De projectie van PGLE naar PGLDg is de identiteit en

$$|X|_{pgle} = |X|_{pgldg}$$

voor elk PGLE-programma X.

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

Conditionele constructie. In `if` c `then` X `else` Y wordt eerst de conditie c geëvalueerd. Als dit true oplevert, wordt X uitgevoerd, en anders Y.

While-loop. In `while` c `do` X wordt eerst de conditie c geëvalueerd. Als dit true oplevert, wordt X uitgevoerd waarna het geheel wordt herhaald, en anders is de uitvoering volbracht.

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

PGLEc is een uitbreiding van PGLE met

Positieve conditionele instructie : $+a\{$ voor elke $a \in A$.

Negatieve conditionele instructie : $-a\{$ voor elke $a \in A$.

Then/else-scheider : $\}\{$.

Sluitaccolade : $\}$.

Voorbeeld:

$$+a\{;b;c;\}\{;a;c;\}.$$

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

PGLEca is een uitbreiding van PGLE met geannoteerde intructies:

Positieve conditionele instructie : $+a\{$ voor elke $a \in A$.

Negatieve conditionele instructie : $-a\{$ voor elke $a \in A$.

Geannoteerde then/else-scheider : $\}n\{$ voor elke $n \in \mathbb{N}$.

Geannoteerde sluitaccolade : $\}n$ voor elk $n \in \mathbb{N}$.

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

De projectie `pglec2pgleca`$(u_1; \ldots; u_k)$ is als volgt gedefinieerd :

1. Vervang $\}\{$ door $\}n\{$ met n het instructienummer van de complementaire conditionele instructie; als die er niet is, vervang dan door $\}0\{$.

2. Vervang $\}$ door $\}n$ met n het instructienummer van de complementaire then/else-scheider; als die er niet is, vervang dan door $\}0$.

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

Voorbeelden:

$$\texttt{pglec2pgleca}(+a\{;b;c;\}\{;d;\};f) = +a\{;b;c;\}1\{;d;\}4;f,$$

$$\texttt{pglec2pgleca}(\}\{;a;\};\};b) = \ \}0\{;a;\}1;\}0;b.$$

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

$$\texttt{pgleca2pgle}(u_1;\ldots;u_k) = \psi_1(u_1);\ldots;\psi_k(u_k)$$

met voor $j = 1, \ldots, k$

$$\begin{aligned}
\psi_j(\#\#\pounds l) &= \#\#\pounds l + k + 1,\\
\psi_j(\pounds l) &= \pounds l + k + 1,\\
\psi_j(+a\{) &= -a; \#\#\pounds j,\\
\psi_j(-a\{) &= +a; \#\#\pounds j,\\
\psi_j(\}n) &= \pounds n,\\
\psi_j(\}n\{) &= \#\#\pounds j; \pounds n,\\
\psi_j(u) &= u \text{ anders.}
\end{aligned}$$

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

Voorbeelden:

$$\texttt{pgleca2pgle}(+a\{; b; c; \}1\{; d; \}4; e) = \\ -a; \#\#\pounds 1; b; c; \#\#\pounds 4; \pounds 1; d; \pounds 4; e,$$

$$\texttt{pgleca2pgle}(\}0\{; a; \}1; \}0; b) = \#\#\pounds 1; \pounds 0; a; \pounds 1; \pounds 0; b.$$

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

$\texttt{pglec2pgle}(X) = \texttt{pgleca2pgle}(\texttt{pglec2pgleca}(X))$

Voorbeeld:

$$\texttt{pglec2pgle}(+a\{;b;\}\{;c;\};\#\#\pounds 0;d;\pounds 0) =$$
$$-a;\#\#\pounds 1;b;\#\#\pounds 3;\pounds 1;c;\pounds 3;\#\#\pounds 9;d;\pounds 9.$$

$|X|_{pglec} = |\texttt{pglec2pgle}(X)|_{pgle}$

(De inbedding van PGLE naar PGLEc is de identiteit.)

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

PGLEcw is een uitbreiding van PGLEc met

Positieve while-header : $+a\{*$ voor elke $a \in A$.

Negatieve while-header : $-a\{*$ voor elke $a \in A$.

Niet-conditionele while-header : $\{*$.

While-sluiter : $*\}$.

Voorbeeld:

$$+a\{*; b; c; *\}; e; f.$$

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

PGLEcwa is de uitbreiding van PGLEca met instructies

- $*\}n$, waar $n \in \mathbb{N}$ de positie van de complementaire openingsaccolade aangeeft ($n = 0$ als die ontbreekt),
- $+a\{*n$ en $-a\{*n$, waar n de positie van de complementaire sluitaccolade aangeeft ($n = 0$ als die ontbreekt), en
- $\{*n$, de geannoteerde niet-conditionele while-header met while-sluiter op positie n.

De projectie `pglecw2pglecwa` voegt weer annotaties toe.

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

De projectie `pglecwa2pgle` : $\mathrm{PGLEcwa} \to \mathrm{PGLE}$ is als volgt gedefinieerd:

$$\texttt{pglecwa2pgle}(u_1; \ldots; u_k) = \psi_1(u_1); \ldots; \psi_k(u_k)$$

met $j = 1, \ldots, k$

$$\begin{aligned}
\psi_j(\{*n) &= \pounds j, \\
\psi_j(+a\{*n) &= \pounds j; -a; \#\#\pounds n, \\
\psi_j(-a\{*n) &= \pounds j; +a; \#\#\pounds n, \\
\psi_j(*\}n) &= \#\#\pounds n; \pounds j.
\end{aligned}$$

5.6 Klassieke programmeerconstructies: PGLEc en PGLEcw

Voorbeelden:

$$\texttt{pglecw2pglecwa}(a; +b\{*; c; d; *\}; e) = a; +b\{*5; c; d; *\}2; e,$$

$$\texttt{pglecwa2pgle}(a; +b\{*5; c; d; *\}2; e) \; = \; a; \pounds 2; -b; \#\#\pounds 5; c; d; \#\#\pounds 2; \pounds 5; e.$$

Definitie: $\texttt{pglecw2pgle}(X) = \texttt{pglecwa2pgle}(\texttt{pglecw2pglcwa}(X))$

$|X|_{pglecw} = |\texttt{pglecw2pgle}(X)|_{pgle}$

5.7 Alles op een rijtje

PGLEcw

`pglecw2pgle` $\downarrow\uparrow$ *identiteit*

PGLE

identiteit $\downarrow\uparrow$ `pgldg2pgle`

PGLDg

`pgldg2pgld` $\downarrow\uparrow$ `pgld2pgldg`

PGLD

`pgld2pglc` $\downarrow\uparrow$ `pglc2pgld`

PGLC

`pglc2pglb` $\downarrow\uparrow$ `pglb2pglc`

PGLB

`pglb2pgla` $\downarrow\uparrow$ `pgla2pglb`

PGLA

`pgla2pga` $\downarrow\uparrow$ `pga2pgla`

PGA