

Termherschrijven

Jan van Eijck

CWI

jve@cwi.nl

Achtergrondcollege Software Evolution, 22 september 2005

Samenvatting

Samenvatting

- Wat zijn termen?

Samenvatting

- Wat zijn termen?
- Wat zijn regels voor vereenvoudigen van termen?

Samenvatting

- Wat zijn termen?
- Wat zijn regels voor vereenvoudigen van termen?
- Het begrip 'normaalvorm'.

Samenvatting

- Wat zijn termen?
- Wat zijn regels voor vereenvoudigen van termen?
- Het begrip 'normaalvorm'.
- Terminatie en confluente.

Samenvatting

- Wat zijn termen?
- Wat zijn regels voor vereenvoudigen van termen?
- Het begrip 'normaalvorm'.
- Terminatie en confluente.
- Precieze definities: termen, algebra's, substituties, termherschrijfsystemen.

Samenvatting

- Wat zijn termen?
- Wat zijn regels voor vereenvoudigen van termen?
- Het begrip 'normaalvorm'.
- Terminatie en confluentie.
- Precieze definities: termen, algebra's, substituties, termherschrijfsystemen.
- De herschrijfrelatie; redexen.

Samenvatting

- Wat zijn termen?
- Wat zijn regels voor vereenvoudigen van termen?
- Het begrip 'normaalvorm'.
- Terminatie en confluentie.
- Precieze definities: termen, algebra's, substituties, termherschrijfsystemen.
- De herschrijfrelatie; redexen.
- Wat zijn bruikbare strategieën?

Samenvatting

- Wat zijn termen?
- Wat zijn regels voor vereenvoudigen van termen?
- Het begrip 'normaalvorm'.
- Terminatie en confluentie.
- Precieze definities: termen, algebra's, substituties, termherschrijfsystemen.
- De herschrijfrelatie; redexen.
- Wat zijn bruikbare strategieën?
- Waar is het goed voor?

Herschrijven: rekenen

Voorbeeld van termen: rekenkundige expressies. Grammatica hiervoor:

$$n ::= 0 \mid 1 \mid 2 \mid \dots$$

$$a ::= n \mid (a_1 + a_2) \mid (a_1 \cdot a_2)$$

Herschrijven: rekenen

Voorbeeld van termen: rekenkundige expressies. Grammatica hiervoor:

$$n ::= 0 \mid 1 \mid 2 \mid \dots$$

$$a ::= n \mid (a_1 + a_2) \mid (a_1 \cdot a_2)$$

Herschrijfregels zijn de rekenregels:

$$(2 + 5) \longrightarrow 7$$

$$(3 \cdot 5) \longrightarrow 15$$

$$(x + y) \cdot z \longrightarrow (x \cdot z) + (y \cdot z)$$

Voorbeeld herschrijving:

$$\begin{aligned}((3 + 5) \cdot (2 + 8)) &\rightarrow ((3 \cdot (2 + 8)) + (5 \cdot (2 + 8))) \\ &\rightarrow ((3 \cdot 10) + (5 \cdot (2 + 8))) \\ &\rightarrow ((3 \cdot 10) + (5 \cdot 10)) \\ &\rightarrow (30 + (5 \cdot 10)) \\ &\rightarrow (30 + 50) \\ &\rightarrow 80.\end{aligned}$$

Herschrijven: tellen

Termen: (namen voor) natuurlijke getallen.

Herschrijfregel is de regel 'plus één'.

Voorbeeld herschrijving:

$$4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow \dots$$

Herschrijven: terugtellen

Termen: (namen voor) natuurlijke getallen.

Herschrijfregel is de regel 'min één'.

Voorbeeld herschrijving:

$$4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0.$$

Herschrijven: Collatz probleem (of: Syracuse probleem)

Termen: natuurlijke getallen.

Herschrijfregels:

Herschrijven: Collatz probleem (of: Syracuse probleem)

Termen: natuurlijke getallen.

Herschrijfregels:

- als $n > 1$ en even dan $n \rightarrow m$ waarbij $m = n/2$,

Herschrijven: Collatz probleem (of: Syracuse probleem)

Termen: natuurlijke getallen.

Herschrijfgeregels:

- als $n > 1$ en even dan $n \rightarrow m$ waarbij $m = n/2$,
- als $n > 1$ en oneven dan $n \rightarrow m$ waarbij $m = 3n + 1$.

Voorbeeld herschrijving:

$$\begin{aligned} 11 &\rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \\ &\rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1. \end{aligned}$$

Implementatie als functioneel programma:

```
run :: Integer -> [Integer]
run n | n < 1 = error "argument not positive"
      | n == 1 = [1]
      | even n = n: run (div n 2)
      | odd n  = n: run (3*n+1)
```

Herschrijven: Optellen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2)$$

Herschrijfregels voor plus:

Herschrijven: Optellen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2)$$

Herschrijfgeregels voor plus:

Herschrijven: Optellen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x, P(0, x) \rightarrow x.$

Herschrijven: Optellen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x, P(0, x) \rightarrow x.$
- $P(x, Sy) \rightarrow SP(x, y), P(Sx, y) \rightarrow SP(x, y).$

Voorbeeld herschrijving:

$$\begin{aligned} & P(SS0, P(S0, SS0)) \\ \rightarrow & SP(S0, P(S0, SS0)) \\ \rightarrow & SSP(0, P(S0, SS0)) \\ \rightarrow & SSP(S0, SS0) \\ \rightarrow & SSSP(0, SS0) \\ \rightarrow & SSSSS0 \end{aligned}$$

Herschrijven: Optellen en Vermenigvuldigen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2) \mid M(t_1, t_2)$$

Herschrijfregels voor plus:

Herschrijven: Optellen en Vermenigvuldigen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2) \mid M(t_1, t_2)$$

Herschrijfregels voor plus:

Herschrijven: Optellen en Vermenigvuldigen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2) \mid M(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x$.

Herschrijven: Optellen en Vermenigvuldigen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2) \mid M(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x$.
- $P(x, Sy) \rightarrow SP(x, y)$.

Herschrijven: Optellen en Vermenigvuldigen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2) \mid M(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x$.
- $P(x, Sy) \rightarrow SP(x, y)$.

Herschrijfregels voor maal:

Herschrijven: Optellen en Vermenigvuldigen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2) \mid M(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x$.
- $P(x, Sy) \rightarrow SP(x, y)$.

Herschrijfregels voor maal:

Herschrijven: Optellen en Vermenigvuldigen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2) \mid M(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x$.
- $P(x, Sy) \rightarrow SP(x, y)$.

Herschrijfregels voor maal:

- $M(x, 0) \rightarrow 0$.

Herschrijven: Optellen en Vermenigvuldigen

Termen:

$$t ::= 0 \mid St \mid P(t_1, t_2) \mid M(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x$.
- $P(x, Sy) \rightarrow SP(x, y)$.

Herschrijfregels voor maal:

- $M(x, 0) \rightarrow 0$.
- $M(x, Sy) \rightarrow P(M(x, y), x)$.

Dit geeft, bij voorbeeld:

$$P(SSS0, SS0)$$

$$\rightarrow SP(SSS0, S0)$$

$$\rightarrow SSP(SSS0, 0)$$

$$\rightarrow SSSSS0$$

$M(SSS0, SS0)$
→ $P(M(SSS0, S0), SSS0)$
→ $P(P(M(SSS0, 0), SSS0), SSS0)$
→ $P(P(0, SSS0), SSS0)$
→ $P(SP(0, SS0), SSS0)$
→ $P(SSP(0, S0), SSS0)$
→ $P(SSSP(0, 0), SSS0)$
→ $P(SSS0, SSS0)$
→ $SP(SSS0, SS0)$
→ $SSP(SSS0, S0)$
→ $SSSP(SSS0, 0)$
→ $SSSSSS0.$

Implementatie als Functioneel Programma

```
step Zero = Stop
step (S t) = case step t of
                Stop -> Stop
                t'   -> S t'

step (P t Zero)      = t
step (P t1 (S t2))  = S (P t1 t2)
step (P t1 t2)      = P t1 (step t2)
step (M t Zero)     = Zero
step (M t1 (S t2))  = P (M t1 t2) t1
step (M t1 t2)      = M t1 (step t2)

compute :: Term -> [Term]
compute t = case step t of
    Stop      -> [t]
    t'        -> t : compute t'
```

```
Main> run (M (S (S (S Zero))) (S (S Zero)))
M(SSSO,SSO)
P(M(SSSO,S0),SSSO)
SP(M(SSSO,S0),SSO)
SSP(M(SSSO,S0),S0)
SSSP(M(SSSO,S0),0)
SSSM(SSSO,S0)
SSSP(M(SSSO,0),SSSO)
SSSSP(M(SSSO,0),SSO)
SSSSSP(M(SSSO,0),S0)
SSSSSSP(M(SSSO,0),0)
SSSSSSM(SSSO,0)
SSSSSSO
```

Herschrijven: Deling met Rest

Termen:

$$t ::= 0 \mid St \mid V(t_1, t_2) \mid Q(t_1, t_2) \mid R(t_1, t_2)$$

Herschrijfgeregels voor **verschil**:

Herschrijven: Deling met Rest

Termen:

$$t ::= 0 \mid St \mid V(t_1, t_2) \mid Q(t_1, t_2) \mid R(t_1, t_2)$$

Herschrijfregels voor **verschil**:

Herschrijven: Deling met Rest

Termen:

$$t ::= 0 \mid St \mid V(t_1, t_2) \mid Q(t_1, t_2) \mid R(t_1, t_2)$$

Herschrijfregels voor **verschil**:

- $V(x, 0) \rightarrow x$.

Herschrijven: Deling met Rest

Termen:

$$t ::= 0 \mid St \mid V(t_1, t_2) \mid Q(t_1, t_2) \mid R(t_1, t_2)$$

Herschrijfregels voor **verschil**:

- $V(x, 0) \rightarrow x$.
- $V(Sx, Sy) \rightarrow V(x, y)$.

Herschrijven: Deling met Rest

Termen:

$$t ::= 0 \mid St \mid V(t_1, t_2) \mid Q(t_1, t_2) \mid R(t_1, t_2)$$

Herschrijfregels voor **verschil**:

- $V(x, 0) \rightarrow x$.
- $V(Sx, Sy) \rightarrow V(x, y)$.

Herschrijfregels voor **quotiënt** bij deling met rest:

Herschrijven: Deling met Rest

Termen:

$$t ::= 0 \mid St \mid V(t_1, t_2) \mid Q(t_1, t_2) \mid R(t_1, t_2)$$

Herschrijfregels voor **verschil**:

- $V(x, 0) \rightarrow x$.
- $V(Sx, Sy) \rightarrow V(x, y)$.

Herschrijfregels voor **quotiënt** bij deling met rest:

Herschrijven: Deling met Rest

Termen:

$$t ::= 0 \mid St \mid V(t_1, t_2) \mid Q(t_1, t_2) \mid R(t_1, t_2)$$

Herschrijfregels voor **verschil**:

- $V(x, 0) \rightarrow x$.
- $V(Sx, Sy) \rightarrow V(x, y)$.

Herschrijfregels voor **quotiënt** bij deling met rest:

- Als $x < y$ dan $Q(x, y) \rightarrow 0$.

Herschrijven: Deling met Rest

Termen:

$$t ::= 0 \mid St \mid V(t_1, t_2) \mid Q(t_1, t_2) \mid R(t_1, t_2)$$

Herschrijfgeregels voor **verschil**:

- $V(x, 0) \rightarrow x$.
- $V(Sx, Sy) \rightarrow V(x, y)$.

Herschrijfgeregels voor **quotiënt** bij deling met rest:

- Als $x < y$ dan $Q(x, y) \rightarrow 0$.
- Als $x \geq y$ dan $Q(x, y) \rightarrow SQ(V(x, y), y)$.

Herschrijfgeregels voor **rest**:

Herschrijfgeregels voor **rest**:

Herschrijfregels voor **rest**:

- Als $x < y$ dan $R(x, y) \rightarrow x$.

Herschrijfregels voor **rest**:

- Als $x < y$ dan $R(x, y) \rightarrow x$.
- Als $x \geq y$ dan $R(x, y) \rightarrow R(V(x, y), y)$.

Herschrijfregels voor **rest**:

- Als $x < y$ dan $R(x, y) \rightarrow x$.
- Als $x \geq y$ dan $R(x, y) \rightarrow R(V(x, y), y)$.

Voorbeeld berekening:

$$\begin{aligned} & Q(SSS0, SS0) \\ \rightarrow & SQ(V(SSS0, SS0), SS0) \\ \rightarrow & SQ(V(SS0, S0), SS0) \\ \rightarrow & SQ(V(S0, 0), SS0) \\ \rightarrow & SQ(S0, SS0) \\ \rightarrow & S0 \end{aligned}$$

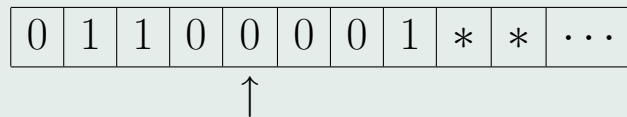
$$\begin{aligned} & R(SSS0, SS0) \\ \rightarrow & R(V(SSS0, SS0), SS0) \\ \rightarrow & R(V(SS0, S0), SS0) \\ \rightarrow & R(V(S0, 0), SS0) \\ \rightarrow & R(S0, SS0) \\ \rightarrow & S0 \end{aligned}$$

Quotient en rest berekening met een functioneel programma:

```
qr :: (Integer,Integer) -> (Integer,Integer)
qr (n,d) | d > n      = (0,n)
          | otherwise = (succ q, r)
  where (q,r) = qr (n-d,d)
```

Herschrijven: rekenen met een Turing machine

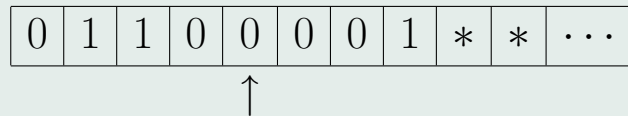
Een Turing machine bestaat uit een eindige verzameling toestanden $\{q_1, \dots, q_n\}$, plus een naar rechts oneindig doorlopende band met een beginstuk beschreven met nullen en enen, en een lees/schrijf kop die wijst naar één van de vakjes (* staat voor een leeg vakje):



Bij elke toestand q hoort (hoogstens) een drietal instructies, bij voorbeeld:

Herschrijven: rekenen met een Turing machine

Een Turing machine bestaat uit een eindige verzameling toestanden $\{q_1, \dots, q_n\}$, plus een naar rechts oneindig doorlopende band met een beginstuk beschreven met nullen en enen, en een lees/schrijf kop die wijst naar één van de vakjes (* staat voor een leeg vakje):

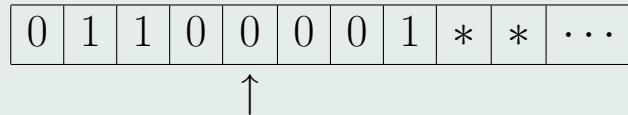


Bij elke toestand q hoort (hoogstens) een drietal instructies, bij voorbeeld:

- $I(q, 0) = (r, 1, \leftarrow)$. (Als 0 wordt gelezen, vervang dan de 0 door een 1, ga naar toestand r , en beweeg de kop naar links.)

Herschrijven: rekenen met een Turing machine

Een Turing machine bestaat uit een eindige verzameling toestanden $\{q_1, \dots, q_n\}$, plus een naar rechts oneindig doorlopende band met een beginstuk beschreven met nullen en enen, en een lees/schrijf kop die wijst naar één van de vakjes (* staat voor een leeg vakje):



Bij elke toestand q hoort (hoogstens) een drietal instructies, bij voorbeeld:

- $I(q, 0) = (r, 1, \leftarrow)$. (Als 0 wordt gelezen, vervang dan de 0 door een 1, ga naar toestand r , en beweeg de kop naar links.)
- $I(q, *) = (s, *, \rightarrow)$. (Als de kop wijst naar een leeg vakje, laat dan dat vakje leeg, ga naar toestand s , en beweeg de kop naar rechts.)

Turing Machine: Voorbeeld 1

l	$*$	0	1
q	$q, *, \rightarrow$	$r, 1, \rightarrow$	$r, 0, \rightarrow$
r	$s, *, \leftarrow$	$r, 1, \rightarrow$	$r, 0, \rightarrow$
s		$s, 0, \leftarrow$	$s, 1, \leftarrow$

Wat doet deze machine?

Je kunt de stappen die een Turing machine uitvoert zien als een herschrijfproces:

	q	*	*	0	1	0	1	*	*
→	*	q	*	0	1	0	1	*	*
→	*	*	q	0	1	0	1	*	*
→	*	*	1	r	1	0	1	*	*
→	*	*	1	0	r	0	1	*	*
→	*	*	1	0	1	r	1	*	*
→	*	*	1	0	1	0	r	*	*
→	*	*	1	0	1	s	0	*	*
→	*	*	1	0	s	1	0	*	*
→	*	*	1	s	0	1	0	*	*
→	*	*	s	1	0	1	0	*	*
→	*	s	*	1	0	1	0	*	*

Het herschrijven stopt hier, want er is geen instructie $I(s, *)$.

```
Main> run turing1
([*,*,0,1,0,1,*,*],0,Q)
([*,*,0,1,0,1,*,*],1,Q)
([*,*,0,1,0,1,*,*],2,Q)
([*,*,1,1,0,1,*,*],3,R)
([*,*,1,0,0,1,*,*],4,R)
([*,*,1,0,1,1,*,*],5,R)
([*,*,1,0,1,0,*,*],6,R)
([*,*,1,0,1,0,*,*],5,S)
([*,*,1,0,1,0,*,*],4,S)
([*,*,1,0,1,0,*,*],3,S)
([*,*,1,0,1,0,*,*],2,S)
([*,*,1,0,1,0,*,*],1,S)
```

Turing Machine: Voorbeeld 2

l	*	0	1
q	$q, *, \rightarrow$	$q, 0, \rightarrow$	r, q, \rightarrow
r		$s, 1, \leftarrow$	$r, 1, \rightarrow$
s	$t, *, \rightarrow$	$t, 0, \rightarrow$	$s, 1, \leftarrow$
t		$q, 0, \rightarrow$	$q, 0, \rightarrow$

Wat doet deze machine?

```
Main> run turing2
([*,0,1,0,0,*],0,Q)
([*,0,1,0,0,*],1,Q)
([*,0,1,0,0,*],2,Q)
([*,0,1,0,0,*],3,R)
([*,0,1,1,0,*],2,S)
([*,0,1,1,0,*],1,S)
([*,0,1,1,0,*],2,T)
([*,0,0,1,0,*],3,Q)
([*,0,0,1,0,*],4,R)
([*,0,0,1,1,*],3,S)
([*,0,0,1,1,*],2,S)
([*,0,0,1,1,*],3,T)
([*,0,0,0,1,*],4,Q)
([*,0,0,0,1,*],5,R)
```

Herschrijven = vereenvoudigen

Betekenis van een term = herschrijving van die term naar **normaalvorm**.

Een **normaalvorm** is een term die niet verder kan worden vereenvoudigd.

Unieke betekenis is gegarandeerd als herschrijfregels twee eigenschappen hebben:

Herschrijven = vereenvoudigen

Betekenis van een term = herschrijving van die term naar **normaalvorm**.
Een **normaalvorm** is een term die niet verder kan worden vereenvoudigd.
Unieke betekenis is gegarandeerd als herschrijfregels twee eigenschappen hebben:

- regels moeten terminerend (SN) zijn, d.w.z., er zijn geen oneindige herschrijvingen



Herschrijven = vereenvoudigen

Betekenis van een term = herschrijving van die term naar **normaalvorm**.
Een **normaalvorm** is een term die niet verder kan worden vereenvoudigd.
Unieke betekenis is gegarandeerd als herschrijfgeregels twee eigenschappen hebben:

- regels moeten terminerend (SN) zijn, d.w.z., er zijn geen oneindige herschrijvingen



- regels moeten confluent (CR) zijn., d.w.z.,



Eigenschappen van de voorbeelden

Eigenschappen van de voorbeelden

- Rekenen SN? Ja. Rekenen CR? Ja. Normaalvormen: natuurlijke getallen.

Eigenschappen van de voorbeelden

- Rekenen SN? Ja. Rekenen CR? Ja. Normaalvormen: natuurlijke getallen.
- Tellen SN? Nee. Tellen CR? Ja. Normaalvormen: geen.

Eigenschappen van de voorbeelden

- Rekenen SN? Ja. Rekenen CR? Ja. Normaalvormen: natuurlijke getallen.
- Tellen SN? Nee. Tellen CR? Ja. Normaalvormen: geen.
- Terugtellen SN? Ja. Terugtellen CR? Ja. Normaalvorm: 0.

Eigenschappen van de voorbeelden

- Rekenen SN? Ja. Rekenen CR? Ja. Normaalvormen: natuurlijke getallen.
- Tellen SN? Nee. Tellen CR? Ja. Normaalvormen: geen.
- Terugtellen SN? Ja. Terugtellen CR? Ja. Normaalvorm: 0.
- Collatz SN? Onbekend. Collatz CR? Ja. Normaalvorm: 1.

Eigenschappen van de voorbeelden

- Rekenen SN? Ja. Rekenen CR? Ja. Normaalvormen: natuurlijke getallen.
- Tellen SN? Nee. Tellen CR? Ja. Normaalvormen: geen.
- Terugtellen SN? Ja. Terugtellen CR? Ja. Normaalvorm: 0.
- Collatz SN? Onbekend. Collatz CR? Ja. Normaalvorm: 1.
- Plus en maal SN? Ja. Plus en maal CR? Ja. Normaalvormen: S^*0 .

Eigenschappen van de voorbeelden

- Rekenen SN? Ja. Rekenen CR? Ja. Normaalvormen: natuurlijke getallen.
- Tellen SN? Nee. Tellen CR? Ja. Normaalvormen: geen.
- Terugtellen SN? Ja. Terugtellen CR? Ja. Normaalvorm: 0.
- Collatz SN? Onbekend. Collatz CR? Ja. Normaalvorm: 1.
- Plus en maal SN? Ja. Plus en maal CR? Ja. Normaalvormen: S^*0 .
- Quotiënt en rest SN? Ja. Quotiënt en rest CR? Ja. Normaalvormen: S^*0 .

Eigenschappen van de voorbeelden

- Rekenen SN? Ja. Rekenen CR? Ja. Normaalvormen: natuurlijke getallen.
- Tellen SN? Nee. Tellen CR? Ja. Normaalvormen: geen.
- Terugtellen SN? Ja. Terugtellen CR? Ja. Normaalvorm: 0.
- Collatz SN? Onbekend. Collatz CR? Ja. Normaalvorm: 1.
- Plus en maal SN? Ja. Plus en maal CR? Ja. Normaalvormen: S^*0 .
- Quotiënt en rest SN? Ja. Quotiënt en rest CR? Ja. Normaalvormen: S^*0 .
- Turing rekenen SN? (i.h.a) Nee. Turing rekenen CR? Ja. Normaalvormen: configuraties zonder instructie.

Niet triviaal voorbeeld van CR Systeem

Beschouw nogmaals het herschrijfsysteem voor optellen, met termen:

$$t ::= 0 \mid St \mid P(t_1, t_2)$$

Herschrijfregels voor plus:

Niet triviaal voorbeeld van CR Systeem

Beschouw nogmaals het herschrijfsysteem voor optellen, met termen:

$$t ::= 0 \mid St \mid P(t_1, t_2)$$

Herschrijfregels voor plus:

Niet triviaal voorbeeld van CR Systeem

Beschouw nogmaals het herschrijfsysteem voor optellen, met termen:

$$t ::= 0 \mid St \mid P(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x, P(0, x) \rightarrow x.$

Niet triviaal voorbeeld van CR Systeem

Beschouw nogmaals het herschrijfsysteem voor optellen, met termen:

$$t ::= 0 \mid St \mid P(t_1, t_2)$$

Herschrijfregels voor plus:

- $P(x, 0) \rightarrow x, P(0, x) \rightarrow x.$
- $P(x, Sy) \rightarrow SP(x, y), P(Sx, y) \rightarrow SP(x, y).$

Dit geeft, bij voorbeeld:

$$\begin{array}{ccc} P(SS0, SS0) & \rightarrow & SP(S0, SS0) \\ \downarrow & & \downarrow \\ SP(SS0, S0) & \rightarrow & SSP(S0, S0) \end{array}$$

Herschrijven: consistentie checken met Herbrand stappen

Termen: propositionele formules:

$$P ::= p[0..9]^*$$

$$F ::= P \mid \neg F \mid (F_1 \wedge F_2) \mid (F_1 \vee F_2) \mid (F_1 \Rightarrow F_2) \mid (F_1 \Leftrightarrow F_2)$$

Voorbeeldformules:

Herschrijven: consistentie checken met Herbrand stappen

Termen: propositionele formules:

$$P ::= p[0..9]^*$$

$$F ::= P \mid \neg F \mid (F_1 \wedge F_2) \mid (F_1 \vee F_2) \mid (F_1 \Rightarrow F_2) \mid (F_1 \Leftrightarrow F_2)$$

Voorbeeldformules:

- $p \vee \neg p$

Herschrijven: consistentie checken met Herbrand stappen

Termen: propositionele formules:

$$P ::= p[0..9]^*$$

$$F ::= P \mid \neg F \mid (F_1 \wedge F_2) \mid (F_1 \vee F_2) \mid (F_1 \Rightarrow F_2) \mid (F_1 \Leftrightarrow F_2)$$

Voorbeeldformules:

- $p \vee \neg p$
- $p \wedge \neg p$.

Herschrijven: consistentie checken met Herbrand stappen

Termen: propositionele formules:

$$P ::= p[0..9]^*$$

$$F ::= P \mid \neg F \mid (F_1 \wedge F_2) \mid (F_1 \vee F_2) \mid (F_1 \Rightarrow F_2) \mid (F_1 \Leftrightarrow F_2)$$

Voorbeeldformules:

- $p \vee \neg p$
- $p \wedge \neg p$.
- $p \Rightarrow p$.

Herschrijven: consistentie checken met Herbrand stappen

Termen: propositionele formules:

$$P ::= p[0..9]^*$$

$$F ::= P \mid \neg F \mid (F_1 \wedge F_2) \mid (F_1 \vee F_2) \mid (F_1 \Rightarrow F_2) \mid (F_1 \Leftrightarrow F_2)$$

Voorbeeldformules:

- $p \vee \neg p$
- $p \wedge \neg p$.
- $p \Rightarrow p$.
- $p_1 \Rightarrow (p_2 \Rightarrow p_1)$.

Herschrijfregels voor Herbrand reductie

$$\neg x \rightarrow x + 1$$

$$x \Rightarrow y \rightarrow x \cdot y + x + 1$$

$$x \Leftrightarrow y \rightarrow x + y + 1$$

$$x \wedge y \rightarrow x \cdot y$$

$$x \vee y \rightarrow x \cdot y + x + y$$

$$x \cdot (y + z) \rightarrow x \cdot y + x \cdot z$$

$$x \cdot 1 \rightarrow x$$

$$x \cdot 0 \rightarrow 0$$

$$x \cdot x \rightarrow x$$

$$x + 0 \rightarrow x$$

$$x + x \rightarrow 0$$

- \cdot staat voor **en**, $+$ voor **exclusief of**.

- \cdot staat voor **en**, $+$ voor **exclusief of**.
- \cdot en $+$ zijn **associatief en commutatief** (AC).

- \cdot staat voor **en**, $+$ voor **exclusief of**.
- \cdot en $+$ zijn **associatief en commutatief** (AC).
- We herschrijven modulo AC, d.w.z. we mogen $p \cdot q$ vervangen door $q \cdot p$, en $p + (p + 1)$ door $(p + p) + 1$.

- \cdot staat voor **en**, $+$ voor **exclusief of**.
- \cdot en $+$ zijn **associatief en commutatief** (AC).
- We herschrijven modulo AC, d.w.z. we mogen $p \cdot q$ vervangen door $q \cdot p$, en $p + (p + 1)$ door $(p + p) + 1$.
- We mogen de haakjes in $p + (p + 1)$ ook wel weglaten: $p + p + 1$.

- \cdot staat voor **en**, $+$ voor **exclusief of**.
- \cdot en $+$ zijn **associatief en commutatief** (AC).
- We herschrijven modulo AC, d.w.z. we mogen $p \cdot q$ vervangen door $q \cdot p$, en $p + (p + 1)$ door $(p + p) + 1$.
- We mogen de haakjes in $p + (p + 1)$ ook wel weglaten: $p + p + 1$.
- We nemen aan dat \cdot sterker bindt dan $+$, zodat we ook de haakjes in $(p \cdot p) + 1$ mogen weglaten: $p \cdot p + 1$.

Voorbeeld herschrijving

$$\begin{aligned} p \vee \neg p &\rightarrow p \vee (p + 1) \\ &\rightarrow p \cdot (p + 1) + p + p + 1 \\ &\rightarrow p \cdot p + p \cdot 1 + p + p + 1 \\ &\rightarrow p + p \cdot 1 + p + p + 1 \\ &\rightarrow p + p + p + p + 1 \\ &\rightarrow 0 + p + p + 1 \\ &\rightarrow 0 + 0 + 1 \\ &\rightarrow 0 + 1 \\ &\rightarrow 1 \end{aligned}$$

Herschrijf nu zelf de formules $p \wedge \neg p$ en $p \Rightarrow p$.

Nog een voorbeeld

$$\begin{aligned} p_1 \Rightarrow (p_2 \Rightarrow p_1) &\rightarrow p_1 \Rightarrow (p_2 \cdot p_1 + p_2 + 1) \\ &\rightarrow p_1 \cdot (p_2 \cdot p_1 + p_2 + 1) + p_1 + 1 \\ &\rightarrow p_1 \cdot p_2 \cdot p_1 + p_1 \cdot p_2 + p_1 \cdot 1 + p_1 + 1 \\ &\rightarrow p_1 \cdot p_1 \cdot p_2 + p_1 \cdot p_2 + p_1 \cdot 1 + p_1 + 1 \\ &\rightarrow p_1 \cdot p_2 + p_1 \cdot p_2 + p_1 \cdot 1 + p_1 + 1 \\ &\rightarrow 0 + p_1 \cdot 1 + p_1 + 1 \\ &\rightarrow 0 + p_1 + p_1 + 1 \\ &\rightarrow 0 + 0 + 1 \\ &\rightarrow 0 + 1 \\ &\rightarrow 1 \end{aligned}$$

Herbrand Reductie: Eigenschappen

Herbrand reductie SN? Ja.

Herbrand reductie CR? Ja.

Normaalvormen?

Herbrand Reductie: Eigenschappen

Herbrand reductie SN? Ja.

Herbrand reductie CR? Ja.

Normaalvormen?

- 0

Herbrand Reductie: Eigenschappen

Herbrand reductie SN? Ja.

Herbrand reductie CR? Ja.

Normaalvormen?

- 0
- 1

Herbrand Reductie: Eigenschappen

Herbrand reductie SN? Ja.

Herbrand reductie CR? Ja.

Normaalvormen?

- 0
- 1
- sommen van producten, bij voorbeeld $p_1 \cdot p_2 + p_1 \cdot p_3$.

Eigenschappen (volledigheid):

Herbrand Reductie: Eigenschappen

Herbrand reductie SN? Ja.

Herbrand reductie CR? Ja.

Normaalvormen?

- 0
- 1
- sommen van producten, bij voorbeeld $p_1 \cdot p_2 + p_1 \cdot p_3$.

Eigenschappen (volledigheid):

- Elke tautologie heeft 1 als normaalvorm.

Herbrand Reductie: Eigenschappen

Herbrand reductie SN? Ja.

Herbrand reductie CR? Ja.

Normaalvormen?

- 0
- 1
- sommen van producten, bij voorbeeld $p_1 \cdot p_2 + p_1 \cdot p_3$.

Eigenschappen (volledigheid):

- Elke tautologie heeft 1 als normaalvorm.
- Elke contradictie heeft 0 als normaalvorm.

lets preciezer: Symbolen en Termen

Iets preciezer: Symbolen en Termen

- Functie-symbolen: f, g, \dots Elk functie-symbool heeft een plaatsigheid (E: *arity*) die aangeeft hoeveel argumentplaatsen het heeft. f/n geeft aan dat f plaatsigheid n heeft.

Iets preciezer: Symbolen en Termen

- Functie-symbolen: f, g, \dots . Elk functie-symbool heeft een **plaatsigheid** (E: **arity**) die aangeeft hoeveel argumentplaatsen het heeft. f/n geeft aan dat f plaatsigheid n heeft.
- Constanten: c, d, \dots . Nulplaatsige functie-symbolen.

Iets preciezer: Symbolen en Termen

- Functie-symbolen: f, g, \dots Elk functie-symbool heeft een **plaatsigheid** (E: **arity**) die aangeeft hoeveel argumentplaatsen het heeft. f/n geeft aan dat f plaatsigheid n heeft.
- Constanten: c, d, \dots Nulplaatsige functie-symbolen.
- Variabelen: x, y, z, \dots

Als F een verzameling symbolen is en X een verzameling variabelen, dan is $\text{Ter}(F, X)$ als volgt gedefiniëerd:

Iets preciezer: Symbolen en Termen

- Functie-symbolen: f, g, \dots Elk functie-symbool heeft een **plaatsigheid** (E: **arity**) die aangeeft hoeveel argumentplaatsen het heeft. f/n geeft aan dat f plaatsigheid n heeft.
- Constanten: c, d, \dots Nulplaatsige functie-symbolen.
- Variabelen: x, y, z, \dots

Als F een verzameling symbolen is en X een verzameling variabelen, dan is $\text{Ter}(F, X)$ als volgt gedefiniëerd:

- als $x \in X$, dan $x \in \text{Ter}(F, X)$.

Iets preciezer: Symbolen en Termen

- Functie-symbolen: f, g, \dots Elk functie-symbool heeft een **plaatsigheid** (E: **arity**) die aangeeft hoeveel argumentplaatsen het heeft. f/n geeft aan dat f plaatsigheid n heeft.
- Constanten: c, d, \dots Nulplaatsige functie-symbolen.
- Variabelen: x, y, z, \dots

Als F een verzameling symbolen is en X een verzameling variabelen, dan is $\text{Ter}(F, X)$ als volgt gedefiniëerd:

- als $x \in X$, dan $x \in \text{Ter}(F, X)$.
- als c een constante in F , dan $c \in \text{Ter}(F, X)$.

Iets preciezer: Symbolen en Termen

- Functie-symbolen: f, g, \dots Elk functie-symbool heeft een **plaatsigheid** (E: **arity**) die aangeeft hoeveel argumentplaatsen het heeft. f/n geeft aan dat f plaatsigheid n heeft.
- Constanten: c, d, \dots Nulplaatsige functie-symbolen.
- Variabelen: x, y, z, \dots

Als F een verzameling symbolen is en X een verzameling variabelen, dan is $\text{Ter}(F, X)$ als volgt gedefiniëerd:

- als $x \in X$, dan $x \in \text{Ter}(F, X)$.
- als c een constante in F , dan $c \in \text{Ter}(F, X)$.
- als $f/n \in F$, en $t_1, \dots, t_n \in \text{Ter}(F, X)$, dan $f(t_1, \dots, t_n) \in \text{Ter}(F, X)$.

Betekenis van Termen

De 'betekenis' van termen wordt gegeven door de constanten te interpreteren als elementen van een verzameling A en de functie-symbolen als functies op die verzameling A .

Een unaire functie op een eindige verzameling A kan worden gespecificeerd in de vorm van een tabel, als volgt:

	f
0	1
1	2
2	0

Een binaire functie op een eindige verzameling A kan worden gespecificeerd in de vorm van een tabel, als volgt:

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Een drieplaatsige functie op een eindige verzameling A moet helemaal worden uitgeschreven:

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Interpretaties en Algebra's

Een interpretatie van een verzameling constanten en functie-symbolen F in een niet-lege verzameling A is een functie $I : F \rightarrow A$ die voldoet aan:

Interpretaties en Algebra's

Een interpretatie van een verzameling constanten en functie-symbolen F in een niet-lege verzameling A is een functie $I : F \rightarrow A$ die voldoet aan:

- $I(c)$ is een element van A voor elke constante c in F .

Interpretaties en Algebra's

Een interpretatie van een verzameling constanten en functie-symbolen F in een niet-lege verzameling A is een functie $I : F \rightarrow A$ die voldoet aan:

- $I(c)$ is een element van A voor elke constante c in F .
- $I(f)$ is een n -plaatsige functie op A voor elk functie-symbool $f/n \in F$.

Een F -algebra is een paar (A, I) waar I een interpretatie is van F in A .

Voorbeelden van Algebra's

Voorbeelden van Algebra's

Natuurlijke getallen, met $F = \{+, \cdot, 0, 1\}$. Dit is de algebra

$$(\mathbb{N}, +, \cdot, 0, 1).$$

Voorbeelden van Algebra's

Natuurlijke getallen, met $F = \{+, \cdot, 0, 1\}$. Dit is de algebra

$$(\mathbb{N}, +, \cdot, 0, 1).$$

Gehele getallen modulo 3, met $F = \{+, \cdot, -, 0, 1, 2\}$. Dit is de algebra

$$(\mathbb{Z}_3, +, \cdot, -, 0, 1, 2).$$

Voorbeelden van Algebra's

Natuurlijke getallen, met $F = \{+, \cdot, 0, 1\}$. Dit is de algebra

$$(\mathbb{N}, +, \cdot, 0, 1).$$

Gehele getallen modulo 3, met $F = \{+, \cdot, -, 0, 1, 2\}$. Dit is de algebra

$$(\mathbb{Z}_3, +, \cdot, -, 0, 1, 2).$$

Waarheidswaarden 0 en 1, met $F = \{\vee, \wedge, \neg, \perp, \top\}$. Dit is de Boolese algebra:

$$(\{0, 1\}, \vee, \wedge, \neg, 0, 1).$$

waarbij $I(\perp) = 0$, $I(\top) = 1$, en de interpretatie van \vee, \wedge, \neg is gegeven door:

\vee	0	1	\wedge	0	1	\neg	
0	0	1	0	0	0	0	1
1	1	1	1	0	1	1	0

Laat U een verzameling zijn, en $\mathcal{P}(U)$ de verzameling van alle deelverzamelingen van U . Dan is

$$(\mathcal{P}(U), \cup, \cap, ', \emptyset, U)$$

de **deelverzamelingen algebra** van U .

Laat U een verzameling zijn, en $\mathcal{P}(U)$ de verzameling van alle deelverzamelingen van U . Dan is

$$(\mathcal{P}(U), \cup, \cap, ', \emptyset, U)$$

de **deelverzamelingen algebra** van U .

- \cup staat voor vereniging van verzamelingen,

Laat U een verzameling zijn, en $\mathcal{P}(U)$ de verzameling van alle deelverzamelingen van U . Dan is

$$(\mathcal{P}(U), \cup, \cap, ', \emptyset, U)$$

de **deelverzamelingen algebra** van U .

- \cup staat voor vereniging van verzamelingen,
- \cap staat voor doorsnede van verzamelingen,

Laat U een verzameling zijn, en $\mathcal{P}(U)$ de verzameling van alle deelverzamelingen van U . Dan is

$$(\mathcal{P}(U), \cup, \cap, ', \emptyset, U)$$

de **deelverzamelingen algebra** van U .

- \cup staat voor vereniging van verzamelingen,
- \cap staat voor doorsnede van verzamelingen,
- $'$ staat voor complement t.o.v. U ,

Laat U een verzameling zijn, en $\mathcal{P}(U)$ de verzameling van alle deelverzamelingen van U . Dan is

$$(\mathcal{P}(U), \cup, \cap, ', \emptyset, U)$$

de **deelverzamelingen algebra** van U .

- \cup staat voor vereniging van verzamelingen,
- \cap staat voor doorsnede van verzamelingen,
- $'$ staat voor complement t.o.v. U ,
- \emptyset staat voor de lege verzameling.

Opmerking: Als we U gelijk nemen aan $\{a\}$, dan krijgen we in feite een Boolese algebra, met $1 = \{a\}$, $0 = \emptyset$, $\vee = \cup$, $\wedge = \cap$, $\neg = '.$

Term Functies

Laat een algebra (A, I) voor het vocabulair F gegeven zijn. Dan legt elke term $t(x_1, \dots, x_n)$ uit $\text{Ter}(F, \{x_1, \dots, x_n\})$ een n -plaatsige functie $t^{(A,I)} : A^n \rightarrow A$ vast, die als volgt is gedefiniëerd:

Term Functies

Laat een algebra (A, I) voor het vocabulair F gegeven zijn. Dan legt elke term $t(x_1, \dots, x_n)$ uit $\text{Ter}(F, \{x_1, \dots, x_n\})$ een n -plaatsige functie $t^{(A,I)} : A^n \rightarrow A$ vast, die als volgt is gedefiniëerd:

- Als $t(x_1, \dots, x_n)$ de variabele x_i is, dan:
$$t^{(A,I)}(a_1, \dots, a_n) = a_i.$$

Term Functies

Laat een algebra (A, I) voor het vocabulair F gegeven zijn. Dan legt elke term $t(x_1, \dots, x_n)$ uit $\text{Ter}(F, \{x_1, \dots, x_n\})$ een n -plaatsige functie $t^{(A,I)} : A^n \rightarrow A$ vast, die als volgt is gedefiniëerd:

- Als $t(x_1, \dots, x_n)$ de variabele x_i is, dan:
$$t^{(A,I)}(a_1, \dots, a_n) = a_i.$$
- Als $t(x_1, \dots, x_n)$ de constante c is, dan:
$$t^{(A,I)}(a_1, \dots, a_n) = I(c).$$

Term Functies

Laat een algebra (A, I) voor het vocabulair F gegeven zijn. Dan legt elke term $t(x_1, \dots, x_n)$ uit $\text{Ter}(F, \{x_1, \dots, x_n\})$ een n -plaatsige functie $t^{(A,I)} : A^n \rightarrow A$ vast, die als volgt is gedefiniëerd:

- Als $t(x_1, \dots, x_n)$ de variabele x_i is, dan:
$$t^{(A,I)}(a_1, \dots, a_n) = a_i.$$
- Als $t(x_1, \dots, x_n)$ de constante c is, dan:
$$t^{(A,I)}(a_1, \dots, a_n) = I(c).$$
- Als $t \equiv f(t_1, \dots, t_n)$, dan:
$$t^{(A,I)}(a_1, \dots, a_n) = I(f)(t_1^{(A,I)}(a_1, \dots, a_n), \dots, t_n^{(A,I)}(a_1, \dots, a_n)).$$

Voorbeeld: interpretatie van reken-termen

Voorbeeld: interpretatie van reken-termen

Neem de taal van het rekenen met natuurlijke getallen.

Beschouw de term $t(x, y, z) = x \cdot y + x \cdot z$.

Dan zal $t^{\mathbb{N}} : \mathbb{N}^3 \rightarrow \mathbb{N}$ het drietel $(2, 3, 1)$ afbeelden op $2 \cdot 3 + 2 \cdot 1 = 5$.

Voorbeeld: interpretatie van reken-termen

Neem de taal van het rekenen met natuurlijke getallen.

Beschouw de term $t(x, y, z) = x \cdot y + x \cdot z$.

Dan zal $t^{\mathbb{N}} : \mathbb{N}^3 \rightarrow \mathbb{N}$ het drietal $(2, 3, 1)$ afbeelden op $2 \cdot 3 + 2 \cdot 1 = 5$.

We gaan nog even na hoe dit volgt uit de definitie.

Uit de eerste regel van de definitie volgt:

- $x^{\mathbb{N}}(2, 3, 1) = 2$,
- $y^{\mathbb{N}}(2, 3, 1) = 3$,
- $z^{\mathbb{N}}(2, 3, 1) = 1$.

Immers, 3 staat hier voor a_1 , 2 voor a_2 en 1 voor a_3 .

Bij de rest moeten we bedenken dat we in het voorbeeld **infix** notatie hebben in plaats van de prefix notatie uit de definitie. We krijgen:

$$\begin{aligned}t^{\mathbb{N}}(2, 3, 1) &= (x \cdot y + x \cdot z)^{\mathbb{N}}(2, 3, 1) \\&= (x \cdot y)^{\mathbb{N}}(2, 3, 1) + (x \cdot z)^{\mathbb{N}}(2, 3, 1) \\&= (x^{\mathbb{N}}(2, 3, 1) \cdot y^{\mathbb{N}}(2, 3, 1)) + (x^{\mathbb{N}}(2, 3, 1) \cdot z^{\mathbb{N}}(2, 3, 1)) \\&= (2 \cdot 3) + (2 \cdot 1) \\&= 8\end{aligned}$$

Voorbeeld: waarheidstabellen voor formules van de propositie-logica

Beschouw de term $t(p, q, r) = (p \wedge q) \vee (p \wedge r)$. Dan wordt de functie $t^{\text{Prop}} : \{0, 1\}^3 \rightarrow \{0, 1\}$ gegeven door de volgende tabel:

p	q	r	$p \wedge q$	$p \wedge r$	$(p \wedge q) \vee (p \wedge r)$
1	1	1	1	1	1
0	1	1	0	0	0
1	0	1	0	1	1
0	0	1	0	0	0
1	1	0	1	0	1
0	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Om na te gaan dat dit klopt moeten we weer bedenken dat

Om na te gaan dat dit klopt moeten we weer bedenken dat

- $p^{\text{Bool}}(1, 0, 1) = 1$ (het eerste element van het rijtje argumenten),

Om na te gaan dat dit klopt moeten we weer bedenken dat

- $p^{\text{Bool}}(1, 0, 1) = 1$ (het eerste element van het rijtje argumenten),
- $q^{\text{Bool}}(1, 0, 1) = 0$ (het tweede element van het rijtje argumenten),

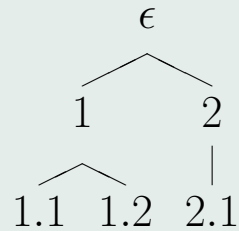
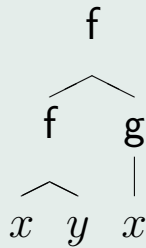
Om na te gaan dat dit klopt moeten we weer bedenken dat

- $p^{\text{Bool}}(1, 0, 1) = 1$ (het eerste element van het rijtje argumenten),
- $q^{\text{Bool}}(1, 0, 1) = 0$ (het tweede element van het rijtje argumenten),
- $r^{\text{Bool}}(1, 0, 1) = 1$ (het derde element van het rijtje argumenten).

Posities in een term

Posities in een term zijn het gemakkelijkst te beschrijven als we de term opvatten als een boom. Bij voorbeeld:

$$f (f (x, y), g (x))$$



De top positie heet ϵ . De plaats waar je terechtkomt door vanaf de top de dichtsbijzijnde knoop links onder de top te bezoeken heet 1. De plaats waar je terechtkomt door vanaf de top de dichtsbijzijnde knoop rechts onder de top te bezoeken heet 2. Vanaf positie 1 kun je zowel links als rechts, dus naar 1.1 en 1.2. Vanaf positie 2 kun je alleen recht naar beneden, dus naar 2.1.

Posities zijn dus rijtjes van positieve gehele getallen.

De rijtjes of lijsten van positieve gehele getallen die we gebruiken zijn gegeven door:

$$L ::= \epsilon \mid n.L$$

Hierbij staat ϵ voor de lege lijst. Omdat elke lijst eindigt op ϵ spreken we af dat we de ϵ aan het eind van een niet-lege lijst weglaten. We schrijven dus 2.1 in plaats van 2.1. ϵ .

Hier komt de formele definitie van de verzameling van alle posities in een term t :

$$O(t) = \begin{cases} \{\epsilon\} & \text{als } t \text{ een constante} \\ & \text{of variabele is} \\ \{\epsilon\} \cup \{i.p \mid 1 \leq i \leq n \text{ en } p \in O(t_i)\} & \text{als } t \equiv f(t_1, \dots, t_n) \end{cases}$$

Dus toegepast op het voorbeeld:

$$\begin{aligned} & O(f(f(x, y), g(x))) \\ = & \{\epsilon\} \\ & \cup \{1.p \mid p \in O(f(x, y))\} \\ & \cup \{2.p \mid p \in O(g(x))\} \\ = & \{\epsilon\} \\ & \cup \{1.\epsilon\} \cup \{1.1.p \mid p \in O(x)\} \cup \{1.2.p \mid p \in O(y)\} \\ & \cup \{2.\epsilon\} \cup \{2.1.p \mid p \in O(x)\} \\ = & \{\epsilon\} \\ & \cup \{1.\epsilon\} \cup \{1.1.\epsilon\} \cup \{1.2.\epsilon\} \\ & \cup \{2.\epsilon\} \cup \{2.1.\epsilon\} \\ = & \{\epsilon, 1, 1.1, 1.2, 2, 2.1\} \end{aligned}$$

Subtermen

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t|_p$ de subterm van t op positie p . Formeel:

$$t|_p = \begin{cases} t & \text{als } p = \epsilon, \\ (t_i)|_q & \text{als } t \equiv f(t_1, \dots, t_n) \text{ en } p = i.q. \end{cases}$$

Voorbeelden:

Subtermen

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t|_p$ de subterm van t op positie p . Formeel:

$$t|_p = \begin{cases} t & \text{als } p = \epsilon, \\ (t_i)|_q & \text{als } t \equiv f(t_1, \dots, t_n) \text{ en } p = i.q. \end{cases}$$

Voorbeelden:

- $f(f(x, y), g(x))|_\epsilon = f(f(x, y), g(x))$.

Subtermen

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t|_p$ de subterm van t op positie p . Formeel:

$$t|_p = \begin{cases} t & \text{als } p = \epsilon, \\ (t_i)|_q & \text{als } t \equiv f(t_1, \dots, t_n) \text{ en } p = i.q. \end{cases}$$

Voorbeelden:

- $f(f(x, y), g(x))|_\epsilon = f(f(x, y), g(x))$.
- $f(f(x, y), g(x))|_1 = f(x, y)$.

Subtermen

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t|_p$ de subterm van t op positie p . Formeel:

$$t|_p = \begin{cases} t & \text{als } p = \epsilon, \\ (t_i)|_q & \text{als } t \equiv f(t_1, \dots, t_n) \text{ en } p = i.q. \end{cases}$$

Voorbeelden:

- $f(f(x, y), g(x))|_\epsilon = f(f(x, y), g(x))$.
- $f(f(x, y), g(x))|_1 = f(x, y)$.
- $f(f(x, y), g(x))|_2 = g(x)$.

Subtermen

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t|_p$ de subterm van t op positie p . Formeel:

$$t|_p = \begin{cases} t & \text{als } p = \epsilon, \\ (t_i)|_q & \text{als } t \equiv f(t_1, \dots, t_n) \text{ en } p = i.q. \end{cases}$$

Voorbeelden:

- $f(f(x, y), g(x))|_\epsilon = f(f(x, y), g(x))$.
- $f(f(x, y), g(x))|_1 = f(x, y)$.
- $f(f(x, y), g(x))|_2 = g(x)$.
- $f(f(x, y), g(x))|_{1.1} = x$.

Subtermen

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t|_p$ de subterm van t op positie p . Formeel:

$$t|_p = \begin{cases} t & \text{als } p = \epsilon, \\ (t_i)|_q & \text{als } t \equiv f(t_1, \dots, t_n) \text{ en } p = i.q. \end{cases}$$

Voorbeelden:

- $f(f(x, y), g(x))|_\epsilon = f(f(x, y), g(x))$.
- $f(f(x, y), g(x))|_1 = f(x, y)$.
- $f(f(x, y), g(x))|_2 = g(x)$.
- $f(f(x, y), g(x))|_{1.1} = x$.
- $f(f(x, y), g(x))|_{1.2} = y$.

Variabelen-posities

Een positie p in een term t is een variabelen-positie als $t|_p$ een variabele is.

Voorbeelden:

Variabelen-posities

Een positie p in een term t is een variabelen-positie als $t|_p$ een variabele is.

Voorbeelden:

- Positie 1.1 in term $f(f(x, y), g(x))$ is een variabelen-positie.

Variabelen-posities

Een positie p in een term t is een variabelen-positie als $t|_p$ een variabele is.

Voorbeelden:

- Positie 1.1 in term $f(f(x, y), g(x))$ is een variabelen-positie.
- Positie 1.2 in term $f(f(x, y), g(x))$ is een variabelen-positie.

Variabelen-posities

Een positie p in een term t is een variabelen-positie als $t|_p$ een variabele is.

Voorbeelden:

- Positie 1.1 in term $f(f(x, y), g(x))$ is een variabelen-positie.
- Positie 1.2 in term $f(f(x, y), g(x))$ is een variabelen-positie.
- Positie 2.1 in term $f(f(x, y), g(x))$ is een variabelen-positie.

Vervanging

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t[s]_p$ de term die ontstaat door de subterm van t op positie p te vervangen door de term s .

Vervanging

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t[s]_p$ de term die ontstaat door de subterm van t op positie p te vervangen door de term s .

Voorbeelden:

Vervanging

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t[s]_p$ de term die ontstaat door de subterm van t op positie p te vervangen door de term s .

Voorbeelden:

- $f(f(x, y), g(x))[g(z)]_\epsilon = g(z)$.

Vervanging

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t[s]_p$ de term die ontstaat door de subterm van t op positie p te vervangen door de term s .

Voorbeelden:

- $f(f(x, y), g(x))[g(z)]_\epsilon = g(z)$.
- $f(f(x, y), g(x))[g(z)]_1 = f(g(z), g(x))$.

Vervanging

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t[s]_p$ de term die ontstaat door de subterm van t op positie p te vervangen door de term s .

Voorbeelden:

- $f(f(x, y), g(x))[g(z)]_\epsilon = g(z)$.
- $f(f(x, y), g(x))[g(z)]_1 = f(g(z), g(x))$.
- $f(f(x, y), g(x))[g(z)]_2 = f(f(x, y), g(z))$.

Vervanging

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t[s]_p$ de term die ontstaat door de subterm van t op positie p te vervangen door de term s .

Voorbeelden:

- $f(f(x, y), g(x))[g(z)]_\epsilon = g(z)$.
- $f(f(x, y), g(x))[g(z)]_1 = f(g(z), g(x))$.
- $f(f(x, y), g(x))[g(z)]_2 = f(f(x, y), g(z))$.
- $f(f(x, y), g(x))[g(z)]_{1.1} = f(f(g(z), y), g(x))$.

Vervanging

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t[s]_p$ de term die ontstaat door de subterm van t op positie p te vervangen door de term s .

Voorbeelden:

- $f(f(x, y), g(x))[g(z)]_\epsilon = g(z)$.
- $f(f(x, y), g(x))[g(z)]_1 = f(g(z), g(x))$.
- $f(f(x, y), g(x))[g(z)]_2 = f(f(x, y), g(z))$.
- $f(f(x, y), g(x))[g(z)]_{1.1} = f(f(g(z), y), g(x))$.
- $f(f(x, y), g(x))[g(z)]_{1.2} = f(f(x, g(z)), g(x))$.

Vervanging

Als $p \in O(t)$, d.w.z, p is een positie in t , dan is $t[s]_p$ de term die ontstaat door de subterm van t op positie p te vervangen door de term s .

Voorbeelden:

- $f(f(x, y), g(x))[g(z)]_\epsilon = g(z)$.
- $f(f(x, y), g(x))[g(z)]_1 = f(g(z), g(x))$.
- $f(f(x, y), g(x))[g(z)]_2 = f(f(x, y), g(z))$.
- $f(f(x, y), g(x))[g(z)]_{1.1} = f(f(g(z), y), g(x))$.
- $f(f(x, y), g(x))[g(z)]_{1.2} = f(f(x, g(z)), g(x))$.
- $f(f(x, y), g(x))[g(z)]_{2.1} = f(f(x, y), g(g(z)))$.

Substitutie

Een substitutie is een toekenning van waarden aan variabelen. Formeel:

Een substitutie is een functie $\sigma : \text{Ter}(F, X) \rightarrow \text{Ter}(F, X)$ die voldoet aan:

$$\sigma(c) \equiv c$$

voor elke constante $c \in F$, en aan

$$\sigma(f(t_1, \dots, t_n)) \equiv f(\sigma(t_1), \dots, \sigma(t_n))$$

voor elke $f/n \in F$.

Substitutie

Een substitutie is een toekenning van waarden aan variabelen. Formeel:

Een substitutie is een functie $\sigma : \text{Ter}(F, X) \rightarrow \text{Ter}(F, X)$ die voldoet aan:

$$\sigma(c) \equiv c$$

voor elke constante $c \in F$, en aan

$$\sigma(f(t_1, \dots, t_n)) \equiv f(\sigma(t_1), \dots, \sigma(t_n))$$

voor elke $f/n \in F$.

Substituties worden volledig bepaald door de waarden die ze toekennen aan variabelen. We schrijven een substitutie op als:

$$\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$$

waarbij x_1, \dots, x_n de variabelen x zijn waarvoor $x \neq \sigma(x)$.

Voorbeelden

Voorbeelden

- Laat $\sigma = \{x \mapsto g(y), y \mapsto g(y)\}$.

Dan is $\sigma(f(f(x, y), g(x))) = f(f(g(y), g(y)), g(g(y)))$.

Voorbeelden

- Laat $\sigma = \{x \mapsto g(y), y \mapsto g(y)\}$.

Dan is $\sigma(f(f(x, y), g(x))) = f(f(g(y), g(y)), g(g(y)))$.

Voorbeelden

- Laat $\sigma = \{x \mapsto g(y), y \mapsto g(y)\}$.

Dan is $\sigma(f(f(x, y), g(x))) = f(f(g(y), g(y)), g(g(y)))$.

- Laat $\sigma = \{x \mapsto y, y \mapsto x\}$.

Dan is $\sigma(f(f(x, y), g(x))) = f(f(y, x), g(y))$.

Eindelijk: Termherschrijfsystemen

Een termherschrijfsysteem (E: term rewriting system, TRS) is een paar (F, R) waarbij F een verzameling functiesymbolen en constanten is, en $R \subseteq \text{Ter}(F, X) \times \text{Ter}(F, X)$ een verzameling **herschrijfregels**.

We schrijven een herschrijfregel (l, r) als $l \rightarrow r$.

Elke herschrijfregel $l \rightarrow r$ moet voldoen aan:

Eindelijk: Termherschrijfsystemen

Een termherschrijfsysteem (E: term rewriting system, TRS) is een paar (F, R) waarbij F een verzameling functiesymbolen en constanten is, en $R \subseteq \text{Ter}(F, X) \times \text{Ter}(F, X)$ een verzameling **herschrijfregels**.

We schrijven een herschrijfregel (l, r) als $l \rightarrow r$.

Elke herschrijfregel $l \rightarrow r$ moet voldoen aan:

- l is geen losse variabele.

Eindelijk: Termherschrijfsystemen

Een termherschrijfsysteem (E: term rewriting system, TRS) is een paar (F, R) waarbij F een verzameling functiesymbolen en constanten is, en $R \subseteq \text{Ter}(F, X) \times \text{Ter}(F, X)$ een verzameling **herschrijfregels**.

We schrijven een herschrijfregel (l, r) als $l \rightarrow r$.

Elke herschrijfregel $l \rightarrow r$ moet voldoen aan:

- l is geen losse variabele.
- in r komen alleen variabelen voor die ook in l voorkomen.

Meestal worden alleen de herschrijfregels gegeven. De volgende omschrijving legt het termherschrijfsysteem voor optellen volledig vast:

$$\{P(x, 0) \rightarrow x, P(x, Sy) \rightarrow SP(x, y)\}.$$

De Herschrijf Relatie

Laat R een verzameling herschrijfregels zijn. De bijbehorende herschrijfrelatie \rightarrow_R is als volgt gedefinieerd:

$s \rightarrow_R t$ geldt dan en slechts dan als

De Herschrijf Relatie

Laat R een verzameling herschrijfregels zijn. De bijbehorende herschrijfrelatie \rightarrow_R is als volgt gedefinieerd:

$s \rightarrow_R t$ geldt dan en slechts dan als

- er een herschrijfregel $l \rightarrow r$ in R is,

De Herschrijf Relatie

Laat R een verzameling herschrijfgeregels zijn. De bijbehorende herschrijfrelatie \rightarrow_R is als volgt gedefinieerd:

$s \rightarrow_R t$ geldt dan en slechts dan als

- er een herschrijfgregel $l \rightarrow r$ in R is,
- er een variabelen-positie $p \in O(s)$ is,

De Herschrijf Relatie

Laat R een verzameling herschrijfregels zijn. De bijbehorende herschrijfrelatie \rightarrow_R is als volgt gedefinieerd:

$s \rightarrow_R t$ geldt dan en slechts dan als

- er een herschrijfregel $l \rightarrow r$ in R is,
- er een variabelen-positie $p \in O(s)$ is,
- er een substitutie σ is,

dit alles zodanig dat $s|_p \equiv \sigma(l)$ en $t \equiv s[\sigma(r)]_p$.

De term $s|_p$ heet de **redex** (=reduceerbare **ex**pressie), en $s \rightarrow_R t$ heet de **herschrijfstep**.

$s \twoheadrightarrow_R t$ wordt gebruikt voor 's herschrijft in één of meer stappen tot t '.

De index R wordt soms weggelaten.

Nogmaals: optellen en vermenigvuldigen

Neem het volgende termherschrijfsysteem R :

$$\left\{ \begin{array}{l} x + 0 \quad \rightarrow \quad x, \\ x + Sy \quad \rightarrow \quad S(x + y), \\ x \cdot 0 \quad \rightarrow \quad 0, \\ x \cdot Sy \quad \rightarrow \quad x \cdot y + x \end{array} \right\}.$$

Dit is hetzelfde als het eerdere herschrijfsysteem voor optellen en vermenigvuldigen, alleen nu met de plus en maal functies in **infix** i.p.v. **prefix** notatie.

Redexen in rood:

$$SSS0 \cdot SS0$$

$$\rightarrow SSS0 \cdot S0 + SSS0$$

$$\rightarrow (SSS0 \cdot 0 + SSS0) + SSS0$$

$$\rightarrow (0 + SSS0) + SSS0$$

$$\rightarrow S(0 + SS0) + SSS0$$

$$\rightarrow SS(0 + S0) + SSS0$$

$$\rightarrow SSS(0 + 0) + SSS0$$

$$\rightarrow SSS0 + SSS0$$

$$\rightarrow S(SSS0 + SS0)$$

$$\rightarrow SS(SSS0 + S0)$$

$$\rightarrow SSS(SSS0 + 0)$$

$$\rightarrow SSSSSS0.$$

Strategieën

Strategische vraag: welke redex pakken we het eerst aan?

Strategieën

Strategische vraag: welke redex pakken we het eerst aan?

Antwoord: een strategie.

Strategieën

Strategische vraag: welke redex pakken we het eerst aan?

Antwoord: een strategie.

Bij voorbeeld:

Strategieën

Strategische vraag: welke redex pakken we het eerst aan?

Antwoord: een strategie.

Bij voorbeeld:

- Van buiten naar binnen, en van links naar rechts.

Strategieën

Strategische vraag: welke redex pakken we het eerst aan?

Antwoord: een strategie.

Bij voorbeeld:

- Van buiten naar binnen, en van links naar rechts.
- Van buiten naar binnen, en van rechts naar links.

Strategieën

Strategische vraag: welke redex pakken we het eerst aan?

Antwoord: een strategie.

Bij voorbeeld:

- Van buiten naar binnen, en van links naar rechts.
- Van buiten naar binnen, en van rechts naar links.
- Van binnen naar buiten, en van links naar rechts.

Strategieën

Strategische vraag: welke redex pakken we het eerst aan?

Antwoord: een strategie.

Bij voorbeeld:

- Van buiten naar binnen, en van links naar rechts.
- Van buiten naar binnen, en van rechts naar links.
- Van binnen naar buiten, en van links naar rechts.
- Van binnen naar buiten, en van rechts naar links.

Welke strategie is dit?

```
step Zero    = Stop
step (S t)   = case step t of
                Stop -> Stop
                t'   -> S t'

step (P t Zero)    = t
step (P t1 (S t2)) = S (P t1 t2)
step (P t1 t2)     = P t1 (step t2)
step (M t Zero)    = Zero
step (M t1 (S t2)) = P (M t1 t2) t1
step (M t1 t2)     = M t1 (step t2)
```

Welke strategie is dit?

```
step Zero    = Stop
step (S t)   = case step t of
                Stop -> Stop
                t'   -> S t'

step (P t Zero)    = t
step (P t1 (S t2)) = S (P t1 t2)
step (P t1 t2)     = P t1 (step t2)
step (M t Zero)    = Zero
step (M t1 (S t2)) = P (M t1 t2) t1
step (M t1 t2)     = M t1 (step t2)
```

Antwoord: van buiten naar binnen, en van rechts naar links.

De strategie in detail:

- Als de buitenste term van de vorm St is, vind dan de buitenste redex in t en herschrijf die.
- Als de buitenste term van de vorm $P(t_1, t_2)$ of $M(t_1, t_2)$ is, en het is een redex, herschrijf die dan.
- Als de buitenste term van de vorm $P(t_1, t_2)$ of $M(t_1, t_2)$ is, en het is geen redex, dan moet t_2 een redex bevatten. Vind die redex (gebruikmakend van **deze** strategie!) en herschrijf die.

```
Main> run (M (S (S (S Zero))) (P (S (S Zero)) (S (S Zero))))
M(SSSO,P(SSO,SSO))
M(SSSO,SP(SSO,S0))
P(M(SSSO,P(SSO,S0)),SSSO)
SP(M(SSSO,P(SSO,S0)),SSO)
SSP(M(SSSO,P(SSO,S0)),S0)
SSSP(M(SSSO,P(SSO,S0)),0)
SSSM(SSSO,P(SSO,S0))
SSSM(SSSO,SP(SSO,0))
SSSP(M(SSSO,P(SSO,0)),SSSO)
SSSSP(M(SSSO,P(SSO,0)),SSO)
SSSSSP(M(SSSO,P(SSO,0)),S0)
SSSSSP(M(SSSO,P(SSO,0)),0)
SSSSSM(SSSO,P(SSO,0))
SSSSSM(SSSO,SSO)
SSSSSP(M(SSSO,S0),SSSO)
```

SSSSSSSP(M(SSS0,S0),SS0)
SSSSSSSP(M(SSS0,S0),S0)
SSSSSSSSP(M(SSS0,S0),0)
SSSSSSSSM(SSS0,S0)
SSSSSSSSSP(M(SSS0,0),SSS0)
SSSSSSSSSP(M(SSS0,0),SS0)
SSSSSSSSSSP(M(SSS0,0),S0)
SSSSSSSSSSSP(M(SSS0,0),0)
SSSSSSSSSSM(SSS0,0)
SSSSSSSSSSS0

Conclusie

Elk rekenproces kan worden beschouwd als een herschrijfproces.

Conclusie

Elk rekenproces kan worden beschouwd als een herschrijfproces.

Geschikte machinerie voor herschrijven:

Conclusie

Elk rekenproces kan worden beschouwd als een herschrijfproces.

Geschikte machinerie voor herschrijven:

- $asf + sdf$

Conclusie

Elk rekenproces kan worden beschouwd als een herschrijfproces.

Geschikte machinerie voor herschrijven:

- asf + sdf
- talen gebaseerd op termherschrijven, b.v. Maude

Conclusie

Elk rekenproces kan worden beschouwd als een herschrijfproces.

Geschikte machinerie voor herschrijven:

- asf + sdf
- talen gebaseerd op termherschrijven, b.v. Maude
- functionele programmeertalen, b.v. Haskell

Conclusie

Elk rekenproces kan worden beschouwd als een herschrijfproces.

Geschikte machinerie voor herschrijven:

- asf + sdf
- talen gebaseerd op termherschrijven, b.v. Maude
- functionele programmeertalen, b.v. Haskell

Belangrijke toepassing:

Conclusie

Elk rekenproces kan worden beschouwd als een herschrijfproces.

Geschikte machinerie voor herschrijven:

- asf + sdf
- talen gebaseerd op termherschrijven, b.v. Maude
- functionele programmeertalen, b.v. Haskell

Belangrijke toepassing:

- Programma-transformaties m.b.v. termherschrijven.