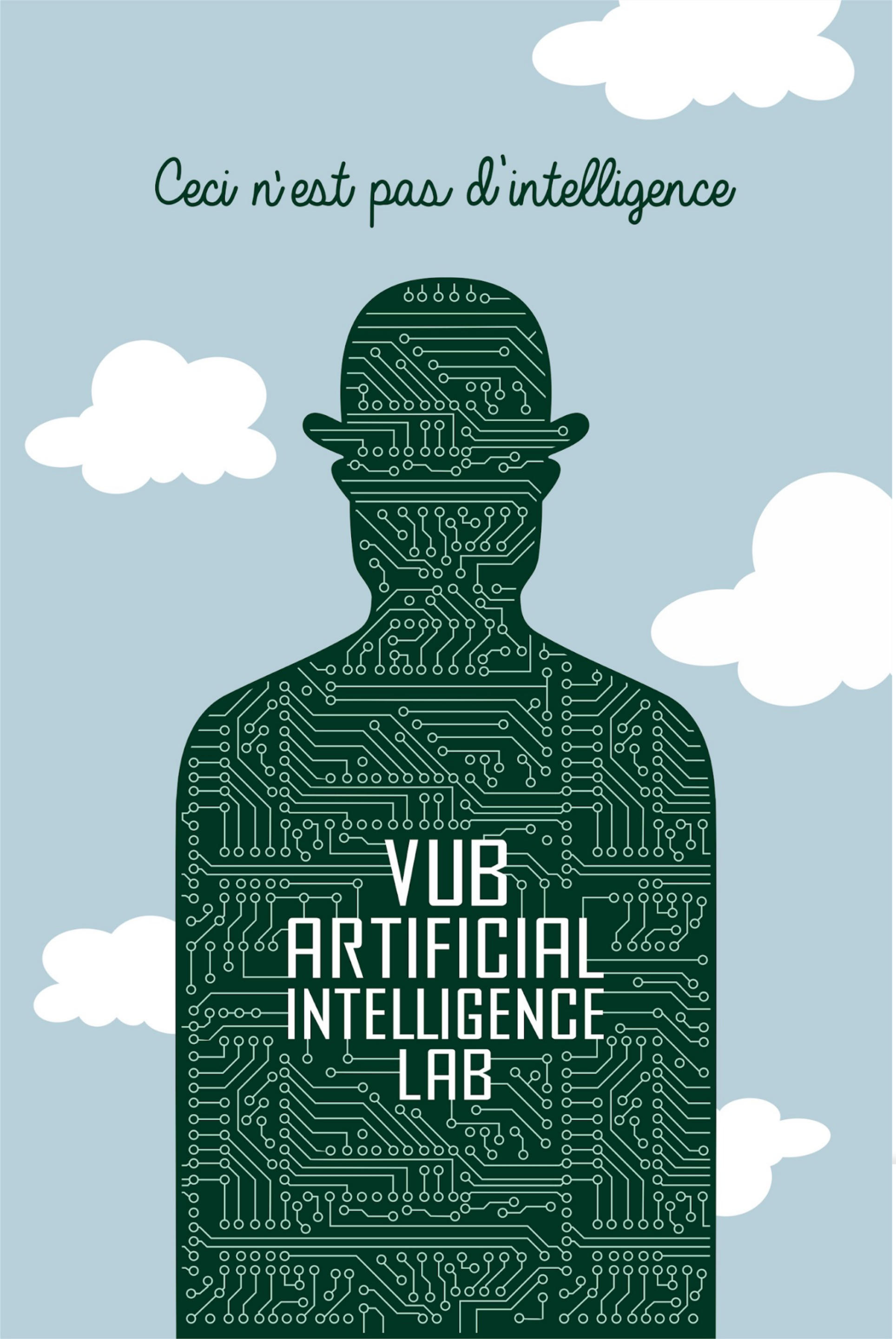

Logica en formele systemen

Lambda Calculus

Rekenen met Lambda Expressies

Prof. dr. Marjon Blondeel

Academiejaar 2024-2025

VUB ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Inhoud lambda calculus

- Inleiding
- Basisbegrippen
- Rekenen met lambda expressies
- Fixpunten en recursie

Beta-gelijkheid

De relatie $=_\beta \subseteq \Lambda \times \Lambda$ wordt gedefinieerd door de axioma's:

$(\beta) (\lambda x.M)P =_\beta [P/x]\mathrm{M}$

(α) $\lambda x.M =_\beta \lambda z.[z/x]M$ als $z \notin VV(M)$

(reflexief) $M =_\beta M$

(symmetrisch) $M =_\beta N$ dan $N =_\beta M$

(transitief) $M =_\beta N$ en $\mathrm{N} =_\beta L$, dan $M =_\beta L$

(congruent) $M =_\beta M'$ en $\mathrm{P} =_\beta P'$, dan $(M)P =_\beta (M')P'$

(congruent) $M =_\beta M'$ dan $\lambda x.M =_\beta \lambda x.M'$

Definitie

Beta-gelijkheid: intuitie (1/3)

$(\beta)(\lambda x.M)P =_\beta [P/x]\mathrm{M}$

formele parameter vervangen door de actuele parameter (substitutie)

$(\alpha)\ \lambda x.M =_\beta\ \lambda z.[z/x]M$ als $z\ \notin VV(M)$

de naam van een formele parameter mag men vervangen door een andere zolang deze niet vrij voorkomt in M

> $\lambda x.z$: hier mag men de formele parameter x niet vervangen door z

Beta-gelijkheid: intuitie (2/3)

(reflexief) $M =_\beta M$

(symmetrisch) $M =_\beta N$ dan $N =_\beta M$

(transitief) $M =_\beta N$ en $\mathrm{N} =_\beta L$, dan $M =_\beta L$

we willen dat de beta-gelijkheid een equivalentierelatie is

Beta-gelijkheid: intuitie (3/3)

(congruent) $M =_\beta M'$ en $\mathrm{P} =_\beta P'$, dan $(M)P =_\beta (M')P'$

het maakt niet uit of we eerst M of eerst P reduceren

(congruent) $M =_\beta M'$ dan $\lambda x. M =_\beta \lambda x. M'$

we mogen λx laten staan en eerst M reduceren

Beta-gelijkheid: voorbeeld

$$\big((\lambda x.\lambda y.\lambda z.((x)z)(y)z)\lambda x.x\big)\lambda x.x$$

$$=_\beta \big((\lambda x.\lambda y.\lambda z.((x)z)(y)z)\lambda x.x\big)\lambda u.u \qquad (\alpha)$$

$$=_\beta \big(\underline{(\lambda x.\lambda y.\lambda z.((x)z)(y)z)\lambda v.v}\big)\lambda u.u \qquad (\alpha)$$

$$=_\beta (\lambda y.\lambda z.(\underline{(\lambda v.v)z})(y)z)\lambda u.u \qquad (\beta)$$

$$=_\beta \underline{(\lambda y.\lambda z.(z)(y)z)\lambda u.u} \qquad (\beta)$$

$$=_\beta \lambda z.(z)\underline{(\lambda u.u)z} \qquad (\beta)$$

$$=_\beta \lambda z.(z)z \qquad (\beta)$$

Church getallen: inleiding

Lambda calculus kent geen constanten. Toch kunnen we natuurlijke getallen en rekenkundige functies voorstellen.

Overzicht

- natuurlijke getallen en λ-definieerbaarheid
- opvolger functie
- optelling
- vermenigvuldiging
- true, false, if, …
- aftrekking

Hulpdefinitie

Beschouw λ-expressies F en M. De expressie $(F)^n M$ wordt inductief gedefinieerd als

- $(F)^0 M \equiv M$
- $(F)^{1+n} M \equiv (F)(F)^n M$

$(F)^n M$ zit niet in de verzameling van λ-expressies. Het is een kortere notatie voor de λ-expressie $(F)(F) \dots (F)M$.

Lemma (“+”)

Beschouw λ-expressies F en M. Voor alle $m, n \in \mathbb{N}$ geldt

$$(F)^{n+m}M \equiv (F)^n(F)^mM$$

Lemma

Lemma ("+"): bewijs

We bewijzen dit via inductie op n.

Als $n = 0$. Dan $(F)^{n+m}M \equiv (F)^m M \equiv_{DEF} (F)^0 (F)^m M \equiv (F)^n (F)^m M$

Inductiehypothese: bewering geldt voor $\mathrm{n} = k$ $(F)^{k+m}M \equiv (F)^k (F)^m M$.

We tonen voor $\mathrm{n} = k + 1$

$(F)^{(k+1)+m}M \equiv (F)^{1+(k+m)}M \equiv_{DEF} (F)(F)^{k+m}M \equiv_{IH} (F)(F)^k(F)^m M$

$\equiv_{DEF} (F)^{1+k}(F)^m M \equiv (F)^{k+1}(F)^m M$

Bewijs

Church getallen: definitie

De zogenaamde Church getallen c_n $(n \in \mathbb{N})$ worden gedefinieerd door

$$c_n \equiv \lambda f.\lambda x.(f)^n x$$

Intuïtief: gegeven een functie f en een parameter x, passen we f n keer toe op x.

Definitie

Church getallen: natuurlijke getallen

Door gebruik te maken van Church getallen kunnen we de natuurlijke getallen voorstellen:

$\lambda f.\lambda x.x$ komt overeen met 0

$\lambda f.\lambda x.(f)x$ komt overeen met 1

$\lambda f.\lambda x.(f)(f)x$ komt overeen met 2

$\lambda f.\lambda x.(f)(f)(f)x$ komt overeen met 3

Volgende stap: hoe bewerkingen voorstellen? Eerst formeel opschrijven waaraan de operaties moeten voldoen.

λ-definieerbaar

Een numerieke functie $f: \mathbb{N}^p \to \mathbb{N}$ is λ-definieerbaar als er een combinator F bestaat zodat

$$((((F)c_{n_1})c_{n_2}) \dots)c_{n_p} =_\beta c_{f(n_1,\dots,n_p)}$$

voor $n_1, \dots, n_p \in \mathbb{N}$

Intuïtief: er bestaat een combinator waarvan het effect op de Church getallen hetzelfde is als het effect op de natuurlijke getallen

Herinner: combinator is λ-expressie zonder vrije variabelen

Definitie

λ-definieerbaar: voorbeeld

$+: \mathbb{N}^2 \to \mathbb{N}$ is λ-definieerbaar indien er een λ-expressie $plus$ bestaat zodat

$$((plus)c_{n_1})c_{n_2} =_\beta c_{n_1+n_2}$$

bijvoorbeeld: $((plus)c_2)c_3 =_\beta c_5$

We zullen $plus$ later definiëren.

Overzicht

- natuurlijke getallen en λ-definieerbaarheid
- **opvolger functie**
- optelling
- vermenigvuldiging
- true, false, if, …
- aftrekking

Opvolger functie is λ-definieerbaar (1/4)

opvolger functie: $\mathbf{opvolger}(n) = n + 1$

We tonen dat $\mathbf{opvolger}$ λ-definieerbaar is. We zijn dus op zoek naar een λ-expressie $succ$ waarvoor het effect op de church getallen hetzelfde is als van opvolger:

$$(succ)c_n =_\beta c_{n+1}$$

ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Opvolger functie is λ-definieerbaar (2/4)

we willen $(succ)c_n =_\beta c_{n+1}$, waar $c_n \equiv \lambda f.\lambda x.(f)^n x$

$n = 0$: we willen $(succ)c_0 =_\beta c_1$

- $(succ)c_0 \equiv (succ)\lambda f.\lambda x.(f)^0 x \equiv (succ)\ \lambda f.\lambda x.x$
- $c_1 \equiv \lambda f.\lambda x.(f)^1 x$

$n = 1$: we willen $(succ)c_1 =_\beta c_2$

- $(succ)c_1 \equiv (succ)\lambda f.\lambda x.(f)^1 x$
- $c_2 \equiv \lambda f.\lambda x.(f)^2 x$

patroon: $(succ)c_n$ vs $\lambda f.\lambda x.(f)^{n+1} x \equiv \lambda f.\lambda x.(f)(f)^n x$

Opvolger functie is λ-definieerbaar (3/4)

patroon: $(succ)c_n$ vs $\lambda f.\lambda x.(f)^{n+1}x \equiv \lambda f.\lambda x.(f)(f)^n x$

waar $c_n \equiv \lambda f.\lambda x.(f)^n x$

Merk op:

$$
\begin{aligned}
(f)^n x &=_\beta (\lambda x.(f)^n x)x && (\beta)\\
&=_\beta \big((\lambda f.\lambda x.(f)^n x)f\big)x && (\beta)\\
&\equiv \big((c_n)f\big)x
\end{aligned}
$$

Dus $\lambda f.\lambda x.(f)(f)^n x =_\beta \lambda f.\lambda x.(f)\big((c_n)f\big)x$

Opvolger functie is λ-definieerbaar (4/4)

$(succ)c_n$ vs $\lambda f.\lambda x.(f)((c_n)f)x$

We hebben afgeleid waaraan een toepassing van $succ$ op de Church getallen moet voldoen. We schrijven nu een definitie op met een formele parameter n.

$$succ \equiv \lambda n.\lambda f.\lambda x.(f)((n)f)x$$

Opvolger functie: oefening

$(succ)c_n =_\beta c_{n+1}$ voor alle $n \in \mathbb{N}$

we tonen hier voor $n = 0$ en $n = 1$

$(succ)c_0$	$\equiv (\lambda n.\lambda f.\lambda x.(f)\big((n)f\big)x)\ \lambda f.\lambda x.(f)^0 x$	(def)
	$\equiv \underline{(\lambda n.\lambda f.\lambda x.(f)\big((n)f\big)x)\ \lambda f.\lambda x.x}$	(def)
	$=_\beta \lambda f.\lambda x.(f)\big(\underline{(\lambda f.\lambda x.x)f}\big)x$	(β)
	$=_\beta \lambda f.\lambda x.(f)\underline{(\lambda x.x)x}$	(β)
	$=_\beta \lambda f.\lambda x.(f)\ x$	(β)
	$\equiv c_1$	(def)

Opvolger functie: oefening

$(succ)c_n =_\beta c_{n+1}$ voor alle $n \in \mathbb{N}$

we tonen hier voor $\boldsymbol{n = 0}$ en $\boldsymbol{n = 1}$

$$
\begin{aligned}
(succ)c_1 &\equiv (\lambda n.\lambda f.\lambda x.(f)\big((n)f\big)x)\ \lambda f.\lambda x.(f)^1 x && \text{(def)}\\
&\equiv \underline{(\lambda n.\lambda f.\lambda x.(f)\big((n)f\big)x)\ \lambda f.\lambda x.(f)x} && \text{(def)}\\
&=_\beta \lambda f.\lambda x.(f)\big(\underline{(\lambda f.\lambda x.(f)x)f}\big)x && (\beta)\\
&=_\beta \lambda f.\lambda x.(f)\underline{(\lambda x.(f)x)x} && (\beta)\\
&=_\beta \lambda f.\lambda x.(f)\ (f)x && (\beta)\\
&\equiv c_2 && \text{(def)}
\end{aligned}
$$

Overzicht

- natuurlijke getallen en λ-definieerbaarheid
- opvolger functie
- **optelling**
- vermenigvuldiging
- true, false, if, …
- aftrekking

Optelling: intuïtie

De opvolger functie is een speciaal geval van de optelling:

$$\text{opvolger}(n) = \text{n} + 1$$

De opvolger is λ-definieerbaar via

$$succ \equiv \lambda n.\lambda f.\lambda x.(f)\big((n)f\big)x$$

Je kan dit zien als n functieaanroepen en dan 1 functieaanroep.

Veralgemening idee naar $n + m$: n functieaanroepen en dan m functieaanroepen

Optelling: intuïtie

$+: \mathbb{N}^2 \rightarrow \mathbb{N}$ is λ-definieerbaar indien er een λ-expressie $plus$ bestaat zodat

$$((plus)c_n)c_m =_\beta c_{n+m}$$

Voorstel definitie $((plus)c_n)c_m$: $\lambda f.\lambda x.((c_m)f)\big(((c_n)f)x\big)$

Gebaseerd op: $(succ)\ c_n \equiv \lambda f.\lambda x.(f)\big(((c_n)f)x\big)$

Optelling

$$plus \equiv \lambda n.\lambda m.\lambda f.\lambda x.\big((n)f\big)\big((m)f\big)x$$

Dan geldt voor alle $m, n \in \mathbb{N}$: $\big((plus)c_n\big)c_m =_\beta c_{n+m}$

Optelling

$$
\begin{aligned}
((plus)c_n)c_m &\equiv ((\lambda n.\lambda m.\lambda f.\lambda x.((n)f)((m)f)x)c_n)c_m && \text{(def)}\\
&=_\beta (\lambda m.\lambda f.\lambda x.((c_n)f)((m)f)x)c_m && (\beta)\\
&\equiv (\lambda m.\lambda f.\lambda x.((\lambda f.\lambda x.(f)^n x)f)((m)f)x)c_m && \text{(def)}\\
&=_\beta (\lambda m.\lambda f.\lambda x.(\lambda x.(f)^n x)((m)f)x)c_m && (\beta)\\
&=_\beta (\lambda m.\lambda f.\lambda x.(f)^n((m)f)x)c_m && (\beta)\\
&=_\beta \lambda f.\lambda x.(f)^n((c_m)f)x && (\beta)\\
&\equiv \lambda f.\lambda x.(f)^n((\lambda f.\lambda x.(f)^m x)f)x && \text{(def)}\\
&=_\beta \lambda f.\lambda x.(f)^n(\lambda x.(f)^m x)x && (\beta)\\
&=_\beta \lambda f.\lambda x.(f)^n(f)^m x && (\beta)\\
&\equiv \lambda f.\lambda x.(f)^{n+m} x && \text{(lemma)}\\
&\equiv c_{n+m}
\end{aligned}
$$

Bewijs

Overzicht

- natuurlijke getallen en λ-definieerbaarheid
- opvolger functie
- optelling
- vermenigvuldiging
- true, false, if, …
- aftrekking

Lemma ("×")

Voor alle $m, n \in \mathbb{N}$ geldt

$$\left((c_n)f\right)^m y =_\beta (f)^{n\times m} y$$

Intuïtief: een soort kopieerfunctie: m keer na elkaar $\left((c_n)f\right)$ toepassen

Lemma

Lemma ("×"): bewijs (1/2)

We bewijzen dit via inductie op m.

Als $\mathrm{m} = 0$.

$$
\begin{aligned}
\big((c_n)f\big)^m y &\equiv \big((c_n)f\big)^0 y \\
&\equiv y && \text{(def)} \\
&\equiv (f)^0 y && \text{(def)} \\
&\equiv (f)^{n\times 0} y \\
&\equiv (f)^{n\times m} y
\end{aligned}
$$

Bewijs

Lemma ("×"): bewijs (2/2)

Inductiehypothese: bewering geldt voor $\mathrm{m} = k$: $\left((c_n)f\right)^k y =_\beta (f)^{n\times k} y$

We tonen voor $\mathrm{m} = k + 1$

$$
\begin{aligned}
\left((c_n)f\right)^{k+1} y &\equiv \left((c_n)f\right)\left((c_n)f\right)^k y && \text{(def)}\\
&=_\beta \left((c_n)f\right)(f)^{n\times k} y && \text{(IH)}\\
&\equiv ((\lambda f.\lambda x.(f)^n x)f)(f)^{n\times k} y && \text{(def)}\\
&=_\beta (\lambda x.(f)^n x)(f)^{n\times k} y && (\beta)\\
&=_\beta (f)^n (f)^{n\times k} y && (\beta)\\
&\equiv (f)^{n+n\times k} y && \text{(lemma)}\\
&\equiv (f)^{n\times(1+k)} y &&\\
&\equiv (f)^{n\times(k+1)} y &&
\end{aligned}
$$

Bewijs

VUB ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Vermenigvuldiging

$$times \equiv \lambda n.\lambda m.\lambda f.(n)(m)f$$

Dan geldt voor alle $m, n \in \mathbb{N}$: $\big((times)c_n\big)c_m =_\beta c_{n\times m}$

Vermenigvuldiging

$$
\begin{aligned}
((times)c_n)c_m &\equiv ((\lambda n.\lambda m.\lambda f.(n)(m)f)\mathrm{c_n})\mathrm{c_m} && \text{(def)}\\
&=_\beta (\lambda m.\lambda f.(c_n)(m)f)\mathrm{c_m} && (\beta)\\
&\equiv (\lambda m.\lambda f.(\lambda f.\lambda x.(f)^n x)(m)f)\mathrm{c_m} && \text{(def)}\\
&=_\beta \left(\lambda m.\lambda f.\lambda x.\left((m)f\right)^{\mathrm{n}} x\right)\mathrm{c_m} && (\beta)\\
&=_\beta \lambda f.\lambda x.\left((c_m)f\right)^{\mathrm{n}} x && (\beta)\\
&=_\beta \lambda f.\lambda x.(f)^{m\times \mathrm{n}} x && \text{(lemma)}\\
&\equiv c_{m\times n} && \text{(def)}\\
&\equiv c_{n\times m}
\end{aligned}
$$

Bewijs

Overzicht

- natuurlijke getallen en λ-definieerbaarheid
- opvolger functie
- optelling
- vermenigvuldiging
- true, false, if, …
- aftrekking

Nuttige combinatoren

$true \equiv \lambda t. \lambda f. t$
$false \equiv \lambda t. \lambda f. f$
$if \equiv \lambda c. \lambda d. \lambda e. \big((c)d\big)e$
$iszero \equiv \lambda n. \big((n)\lambda x. false\big)true$
$cons \equiv \lambda a. \lambda d. \lambda z. \big((z)a\big)d$
$car \equiv \lambda a. \lambda d. a$
$cdr \equiv \lambda a. \lambda d. d$

zal enkel zoals verwacht werken als c naar $true/false$ reduceert

Nuttige combinatoren: intuïtie

$true \equiv \lambda t.\lambda f.t$
$false \equiv \lambda t.\lambda f.f$
$if \equiv \lambda c.\lambda d.\lambda e.\big((c)d\big)e$

$true$ en $false$ zijn zo geconstrueerd dat ze samenwerken met if: we passen de conditie c toe op 2 parameters d,e

c is $true$: eerste argument

c is $false$: tweede argument

Nuttige combinatoren: intuïtie

$iszero \equiv \lambda n.\big((n)\lambda x.false\big)true$

$c_n \equiv \lambda f.\lambda x.(f)^n x$: gegeven een functie f en een parameter x, wordt f n keer toegepast op de x

- hier is de functie $\lambda x.false$ (een λ-expressie die altijd $false$ teruggeeft) en de parameter $true$
- van zodra de functie toegepast wordt is het resultaat $false$
- enkel als $n = 0$ wordt de functie niet toegepast en krijgen we $true$

Nuttige combinatoren: intuïtie

$$cons \equiv \lambda a.\lambda d.\lambda z.\big((z)a\big)d$$
$$car \equiv \lambda a.\lambda d.a$$
$$cdr \equiv \lambda a.\lambda d.d$$

$cons$ combineert a en d en de resulterende λ-expressie verwacht 1 argument z (ofwel car of cdr), car geeft a en cdr geeft d

Nuttige combinatoren: voorbeelden (1/7)

$$((true)\ A)B) \equiv \underline{((\lambda t.\lambda f.t)}A)B$$

$$=_\beta \underline{(\lambda f.A)}B$$

$$=_\beta A$$

$$((false)\ A)B) \equiv \underline{((\lambda t.\lambda f.f)}A)B$$

$$=_\beta \underline{(\lambda f.f)}B$$

$$=_\beta B$$

ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Nuttige combinatoren: voorbeelden (2/7)

$$
\begin{aligned}
(if)true &\equiv \underline{\big(\lambda c.\lambda d.\lambda e.\big((c)d\big)e\big)true} \\
&=_\beta \lambda d.\lambda e.\big((true)d\big)e \\
&\equiv \lambda d.\lambda e.\big(\underline{(\lambda t.\lambda f.t)d}\big)e \\
&=_\beta \lambda d.\lambda e.\underline{(\lambda f.d)e} \\
&=_\beta \lambda d.\lambda e.d \\
&\equiv true
\end{aligned}
$$

ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Nuttige combinatoren: voorbeelden (3/7)

$$(true)false \equiv \underline{(\lambda t.\lambda f.t)\text{false}}$$
$$=_\beta \lambda f.false$$

$n > 0$:
$$(\lambda f.false)^n M \equiv \underline{(\lambda f.false)(\lambda f.false)^{n-1} M}$$
$$=_\beta false$$

ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Nuttige combinatoren: voorbeelden (4/7)

$$(iszero)c_0 \equiv \underline{(\lambda n.\big((n)\lambda x.false\big)true)c_0}$$

$$=_\beta \big((c_0)\lambda x.false\big)true$$

$$\equiv \big((\lambda f.\lambda x.(f)^0x)\lambda x.false\big)true$$

$$\equiv \big(\underline{(\lambda f.\lambda x.x)\lambda x.false}\big)true$$

$$=_\beta \underline{(\lambda x.x)true}$$

$$=_\beta true$$

ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Nuttige combinatoren: voorbeelden (5/7)

$n > 0$:

$$(iszero)c_n \equiv \underline{\big(\lambda n.\big((n)\lambda x.false\big)true\big)c_n}$$

$$=_\beta \big((c_n)\lambda x.false\big)true$$

$$\equiv \big((\lambda f.\lambda x.(f)^n x)\lambda x.false\big)true$$

$$=_\alpha \underline{\big((\lambda f.\lambda x.(f)^n x)\lambda y.false\big)}true$$

$$=_\beta (\lambda x.\underline{(\lambda y.false)^n x})true$$

$$=_\beta \underline{(\lambda x.false)true} =_\beta false$$

ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Nuttige combinatoren: voorbeelden (6/7)

$$\big((cons)A\big)B \equiv \Big(\underline{\big(\lambda a.\lambda d.\lambda z.\big((z)a\big)d\big)\mathrm{A}}\Big)B$$

$$=_\beta \underline{\big(\lambda d.\lambda z.\big((z)A\big)d\big)B}$$

$$=_\beta \lambda z.\big((z)A\big)B$$

$$\Big(\big((cons)A\big)B\Big)car =_\beta \underline{\big(\lambda z.\big((z)A\big)B\big)car}$$

$$=_\beta \big((car)A\big)B$$

$$\equiv \big(\underline{(\lambda a.\lambda d.a)\mathrm{A}}\big)B$$

$$=_\beta (\lambda d.A)B$$

$$=_\beta A$$

Nuttige combinatoren: voorbeelden (7/7)

$$\big((cons)A\big)B =_\beta \lambda z.\big((z)A\big)B$$

$$\Big(\big((cons)A\big)B\Big)cdr =_\beta \underline{(\lambda z.\big((z)A\big)B)cdr}$$

$$=_\beta \big((cdr)A\big)B$$

$$\equiv \big(\underline{(\lambda a.\lambda d.d)A}\big)B$$

$$=_\beta \underline{(\lambda d.d)B}$$

$$=_\beta B$$

ARTIFICIAL INTELLIGENCE RESEARCH GROUP

Overzicht

- natuurlijke getallen en λ-definieerbaarheid
- optelling
- vermenigvuldiging
- true, false, if, …
- aftrekking

Aftrekking: inleiding

Voor de optelling gebruikten we opvolger. Voor de aftrekking zullen we voorganger gebruiken.

$voorganger(n) = n - 1$ als $n \in \mathbb{N}_0$

idee: paren maken $((cons\ c_n)c_{n-1})$

we kunnen dan cdr gebruiken om de voorganger te verkrijgen:

$\left(((cons\ c_n)c_{n-1})\right)cdr$

Voorganger functie: hulpfunctie

Hoe de paren definiëren? Eerst een hulpfunctie die het volgende paar teruggeeft. We willen dat

$$(nextp)((cons)c_n)c_{n-1} =_\beta ((cons)\ c_{n+1})c_n$$

We kunnen $nextp$ gemakkelijk definiëren:

$$nextp \equiv \lambda p.\big((cons)(succ)(p)car\big)(p)car$$

Voorganger functie: definitie

We starten met paar (c_0, c_0) en passen $nextp\ n$ keer toe

We kunnen gebruik maken van Church getallen:
$$\big((c_n)g\big)y \equiv \big((\lambda f.\lambda x.(f)^n x)g\big)y =_\beta (\lambda x.(g)^n x)y \ =_\beta (g)^n y$$
We kunnen (c_n, c_{n-1}) genereren door:

$$((c_n)nextp)((cons)c_0)c_0$$

$$pred \ \equiv \lambda n.\Big(\big((n)nextp\big)\big((cons)c_0\big)c_0\Big)\,cdr$$

Aftrekking: definitie

$$pred \equiv \lambda n.\Big(\big((n)nextp\big)\big((cons)c_0\big)c_0\Big)\,cdr$$

We kunnen nu de aftrekking definiëren:

$$minus \equiv \lambda m.\lambda n.\big((m)pred\big)n$$

$$\big((minus)c_n\big)c_m =_\beta c_{n-m}$$