Relatório Métodos Formais

Alunos: Kamilla Borges, Miguel Gheno e Pedro Riva

Provar:

$\forall xs \in List(\tau)$ (reverso(reverso(xs)) = xs)
Utilizando os seguintes lemas auxiliares:
(L1) $\forall xs, ys, zs \in List(\tau)$ (cat(xs, cat(ys, zs)) = cat(cat(xs, ys), zs))
(L2) $\forall xs \in List(\tau)$ (cat(xs, []) = xs)
(L3) $\forall xs, ys \in List(\tau)$ (reverso(cat(xs, ys)) = cat(reverso(ys), reverso(xs)))

Definições:
cat:
(EqC1) cat([], ys) = ys ≡ cateq1
(EqC2) cat(x:xs, ys) = x : cat(xs, ys) ≡ cateq2

reverso:
(EqR1) reverso([]) = [] ≡ reveq1
(EqR2) reverso(x:xs) = cat(reverso(xs), [x]) ≡ reveq2

Lema 1 -> Associatividade de cat
Provar $\forall xs, ys, zs \in List(\tau)$ (cat(xs, cat(ys, zs)) = cat(cat(xs, ys), zs))
Por indução em xs
Seja P(xs) ≡ (cat(xs, cat(ys, zs)) = cat(cat(xs, ys), zs))

Caso Base: P([ ]) (isto é-> xs = [])
Provar
$\forall$ ys, zs $\in List(\tau)$ (cat([], cat(ys, zs)) = cat(cat([], ys), zs))
Sejam ys e zs $\in List(\tau)$ arbitrários
cat ([ ], cat (ys, zs))
= cat (ys, zs) (por equação C1)
= cat (cat ([], ys), zs) (por equação C1)
q. e. d

Passo indutivo: P(xs) -> P(x:xs) //(Suponha P(xs) e prove P(x:xs))
Sejam $x \in \tau$, xs $\in List(\tau)$ arbitrários
Assumir HI $\forall$ ys, zs $\in List(\tau)$.(cat(xs, cat(ys, zs)) = cat(cat(xs, ys), zs))

Provar: $\forall$ ys, zs $\in List(\tau)$. (cat(x:xs, cat(ys,zs)) = cat (cat (x:xs,ys), zs))
Sejam ys,zs $\in List(\tau)$
cat (x:xs, cat (ys,zs))
= x: cat (xs, cat(ys,zs)) (por equação C2)
= x: cat(cat(xs,ys), zs) (por HI)

= cat (x:cat (xs,ys), zs)   (por equação C2)
= cat (cat (x:xs,ys), zs)   (por equação C2)
q.e.d

Lema 2 -> Neutro à direita de cat
Provar $\forall$xs $\in$ List(τ) (cat(xs, []) = xs)
Por indução em xs

Seja P (xs) $\equiv$ (cat(xs, []) = xs)

Caso Base: P([ ])   (isto é-> xs = [])
cat ([], [])
= []   (por equação C1)
q.e.d

Passo Indutivo: P(xs) -> P (x:xs)   //Suponha P(xs) e prove P(x:xs)
Sejam x $\in$ τ, xs $\in$ List(τ) arbitrários
Assumir HI: cat(xs, []) = xs
Provar: cat(x:xs, []) = x:xs
cat (x:xs, [])
= x: cat (xs, [])   (por equação C2)
= x: xs   (por HI)
q.e.d

Lema 3 – Reverso inverte concatenação
Provar $\forall$xs, ys $\in$ List(τ) (reverso(cat(xs, ys)) = cat(reverso(ys), reverso(xs)))
Por indução em xs
Seja P(xs) $\equiv$ ($\forall$ys $\in$ List(τ). reverso(cat(xs, ys)) = cat(reverso(ys), reverso(xs)))

Caso Base: P([])   (isto é -> xs = [])
Seja ys $\in$ List(τ) arbitrário
reverso(cat([], ys))
= reverso(ys)   (por equação C1)
= cat(reverso(ys), [])   (por Lema 2)
= cat(reverso(ys), reverso([]))   (por equação R1)
q.e.d

Passo indutivo: P(xs) -> P(x:xs)   // Suponha P(xs) e prove P(x:xs)
Sejam x $\in$ τ, xs, ys $\in$ List(τ) arbitrários
Assumir HI: reverso(cat(xs, ys)) = cat(reverso(ys), reverso(xs))
Provar: reverso(cat(x:xs, ys)) = cat(reverso(ys), reverso(x:xs))
reverso(cat(x:xs, ys))
= reverso(x : cat(xs, ys))   (por equação C2)
= cat(reverso(cat(xs, ys)), [x])   (por equação R2)
= cat(cat(reverso(ys), reverso(xs)), [x])   (por HI)
= cat(reverso(ys), cat(reverso(xs), [x]))   (por Lema 1)
= cat(reverso(ys), reverso(x:xs))   (por equação R2)
q.e.d

T1 – duplo reverso é a identidade

Provar $\forall xs \in List(\tau)$ (reverso(reverso(xs)) = xs)

Por indução em xs

Seja P(xs) $\equiv$ (reverso(reverso(xs)) = xs)

Caso Base: P([])

reverso(reverso([]))

= reverso([]) (por EqR1)

= [] (por EqR1)

q.e.d

Passo Indutivo: P(xs) -> P(x:xs)

Sejam $x \in \tau$, $xs \in List(\tau)$ arbitrários

Assumir HI: reverso(reverso(xs)) = xs

Provar: reverso(reverso(x:xs)) = x:xs

reverso(reverso(x:xs))

= reverso(cat(reverso(xs), [x])) (por EqR2)

= cat(reverso([x]), reverso(reverso(xs))) (por Lema 3)

= cat(reverso[x], xs) (por HI)

--

Simplificando reverso[x]:

= reverso (x:[ ]) (por Def. Lista unitária)

= cat (reverso ([]), [x]) (por EqR2)

= cat([], [x]) (por EqR1)

=[x] (por EqC1)

-- Utilizando a simplificação:

= cat ([x], xs) (por Simplificação acima)

= cat (x: [], xs) (por Def. Lista unitária)

= x: cat ([], xs) (por EqC2)

= x:xs (por EqC1)

q.e.d

Conclusão: por indução em xs, $\forall xs$. reverso(reverso(xs)) = xs.

Observação: Por convenção de notação como foi utilizado em aula, usamos o operador de cons como : no lugar de #.