# Unifying Lazy and Strict Computations

Walter Guttmann, Universität Ulm

2012.04.17

**theory** *ULSC*

**imports** *Main*

**begin**

**class** *mult* = *times*

**begin**

**notation**
  *times* (**infixl** · *70*) **and**
  *times* (**infixl** ; *70*)

**end**

**class** *neg* = *uminus*

**begin**

**no-notation**
  *uminus* (− - [*81*] *80*)

**notation**
  *uminus* (− - [*80*] *80*)

**end**

**context** *order*

**begin**

**definition** *isotone* :: $('a \Rightarrow 'a) \Rightarrow bool$
  **where** $isotone\ f \leftrightarrow (\forall x\ y\ .\ x \leq y \longrightarrow f(x) \leq f(y))$

**definition** *is-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** $is\text{-}fixpoint\ f\ x \leftrightarrow f(x) = x$
**definition** *is-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** $is\text{-}prefixpoint\ f\ x \leftrightarrow f(x) \leq x$
**definition** *is-postfixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** $is\text{-}postfixpoint\ f\ x \leftrightarrow f(x) \geq x$
**definition** *is-least-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** $is\text{-}least\text{-}fixpoint\ f\ x \leftrightarrow f(x) = x \wedge (\forall y\ .\ f(y) = y \longrightarrow x \leq y)$
**definition** *is-greatest-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** $is\text{-}greatest\text{-}fixpoint\ f\ x \leftrightarrow f(x) = x \wedge (\forall y\ .\ f(y) = y \longrightarrow x \geq y)$
**definition** *is-least-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** $is\text{-}least\text{-}prefixpoint\ f\ x \leftrightarrow f(x) \leq x \wedge (\forall y\ .\ f(y) \leq y \longrightarrow x \leq y)$
**definition** *is-greatest-postfixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** $is\text{-}greatest\text{-}postfixpoint\ f\ x \leftrightarrow f(x) \geq x \wedge (\forall y\ .\ f(y) \geq y \longrightarrow x \geq y)$
**definition** *has-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** $has\text{-}fixpoint\ f \leftrightarrow (\exists x\ .\ is\text{-}fixpoint\ f\ x)$
**definition** *has-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** $has\text{-}prefixpoint\ f \leftrightarrow (\exists x\ .\ is\text{-}prefixpoint\ f\ x)$
**definition** *has-postfixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** $has\text{-}postfixpoint\ f \leftrightarrow (\exists x\ .\ is\text{-}postfixpoint\ f\ x)$
**definition** *has-least-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** $has\text{-}least\text{-}fixpoint\ f \leftrightarrow (\exists x\ .\ is\text{-}least\text{-}fixpoint\ f\ x)$
**definition** *has-greatest-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** $has\text{-}greatest\text{-}fixpoint\ f \leftrightarrow (\exists x\ .\ is\text{-}greatest\text{-}fixpoint\ f\ x)$
**definition** *has-least-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** $has\text{-}least\text{-}prefixpoint\ f \leftrightarrow (\exists x\ .\ is\text{-}least\text{-}prefixpoint\ f\ x)$
**definition** *has-greatest-postfixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** $has\text{-}greatest\text{-}postfixpoint\ f \leftrightarrow (\exists x\ .\ is\text{-}greatest\text{-}postfixpoint\ f\ x)$
**definition** *the-least-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a\ (\mu\ -\ [201]\ 200)$ **where** $\mu\ f = (THE\ x\ .\ is\text{-}least\text{-}fixpoint\ f\ x)$
**definition** *the-greatest-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a\ (\nu\ -\ [201]\ 200)$ **where** $\nu\ f = (THE\ x\ .\ is\text{-}greatest\text{-}fixpoint\ f\ x)$
**definition** *the-least-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a\ (p\mu\ -\ [201]\ 200)$ **where** $p\mu\ f = (THE\ x\ .\ is\text{-}least\text{-}prefixpoint\ f\ x)$
**definition** *the-greatest-postfixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a\ (p\nu\ -\ [201]\ 200)$ **where** $p\nu\ f = (THE\ x\ .\ is\text{-}greatest\text{-}postfixpoint\ f\ x)$

**lemma** *least-fixpoint-unique*: $has\text{-}least\text{-}fixpoint\ f \longrightarrow (\exists!x\ .\ is\text{-}least\text{-}fixpoint\ f\ x)$
  **by** (*smt antisym has-least-fixpoint-def is-least-fixpoint-def*)

**lemma** *greatest-fixpoint-unique*: $has\text{-}greatest\text{-}fixpoint\ f \longrightarrow (\exists!x\ .\ is\text{-}greatest\text{-}fixpoint\ f\ x)$
  **by** (*smt antisym has-greatest-fixpoint-def is-greatest-fixpoint-def*)

**lemma** *least-prefixpoint-unique*: $has\text{-}least\text{-}prefixpoint\ f \longrightarrow (\exists!x\ .\ is\text{-}least\text{-}prefixpoint\ f\ x)$
  **by** (*smt antisym has-least-prefixpoint-def is-least-prefixpoint-def*)

**lemma** *greatest-postfixpoint-unique*: $has\text{-}greatest\text{-}postfixpoint\ f \longrightarrow (\exists!x\ .\ is\text{-}greatest\text{-}postfixpoint\ f\ x)$
  **by** (*smt antisym has-greatest-postfixpoint-def is-greatest-postfixpoint-def*)

**lemma** *least-fixpoint*: $has\text{-}least\text{-}fixpoint\ f \longrightarrow is\text{-}least\text{-}fixpoint\ f\ (\mu\ f)$
**proof**
  **assume** *has-least-fixpoint f*
  **hence** $is\text{-}least\text{-}fixpoint\ f\ (THE\ x\ .\ is\text{-}least\text{-}fixpoint\ f\ x)$
    **by** (*smt least-fixpoint-unique theI'*)
  **thus** $is\text{-}least\text{-}fixpoint\ f\ (\mu\ f)$
    **by** (*simp add*: *is-least-fixpoint-def the-least-fixpoint-def*)
**qed**

**lemma** *greatest-fixpoint*: *has-greatest-fixpoint f* $\longrightarrow$ *is-greatest-fixpoint f* ($\nu$ *f*)
**proof**
  **assume** *has-greatest-fixpoint f*
  **hence** *is-greatest-fixpoint f* (*THE x . is-greatest-fixpoint f x*)
    **by** (*smt greatest-fixpoint-unique theI'*)
  **thus** *is-greatest-fixpoint f* ($\nu$ *f*)
    **by** (*simp add*: *is-greatest-fixpoint-def the-greatest-fixpoint-def*)
**qed**

**lemma** *least-prefixpoint*: *has-least-prefixpoint f* $\longrightarrow$ *is-least-prefixpoint f* ($p\mu$ *f*)
**proof**
  **assume** *has-least-prefixpoint f*
  **hence** *is-least-prefixpoint f* (*THE x . is-least-prefixpoint f x*)
    **by** (*smt least-prefixpoint-unique theI'*)
  **thus** *is-least-prefixpoint f* ($p\mu$ *f*)
    **by** (*simp add*: *is-least-prefixpoint-def the-least-prefixpoint-def*)
**qed**

**lemma** *greatest-postfixpoint*: *has-greatest-postfixpoint f* $\longrightarrow$ *is-greatest-postfixpoint f* ($p\nu$ *f*)
**proof**
  **assume** *has-greatest-postfixpoint f*
  **hence** *is-greatest-postfixpoint f* (*THE x . is-greatest-postfixpoint f x*)
    **by** (*smt greatest-postfixpoint-unique theI'*)
  **thus** *is-greatest-postfixpoint f* ($p\nu$ *f*)
    **by** (*simp add*: *is-greatest-postfixpoint-def the-greatest-postfixpoint-def*)
**qed**

**lemma** *least-fixpoint-same*: *is-least-fixpoint f x* $\longrightarrow$ $x = \mu$ *f*
  **by** (*metis least-fixpoint least-fixpoint-unique has-least-fixpoint-def*)

**lemma** *greatest-fixpoint-same*: *is-greatest-fixpoint f x* $\longrightarrow$ $x = \nu$ *f*
  **by** (*metis greatest-fixpoint greatest-fixpoint-unique has-greatest-fixpoint-def*)

**lemma** *least-prefixpoint-same*: *is-least-prefixpoint f x* $\longrightarrow$ $x = p\mu$ *f*
  **by** (*metis least-prefixpoint least-prefixpoint-unique has-least-prefixpoint-def*)

**lemma** *greatest-postfixpoint-same*: *is-greatest-postfixpoint f x* $\longrightarrow$ $x = p\nu$ *f*
  **by** (*metis greatest-postfixpoint greatest-postfixpoint-unique has-greatest-postfixpoint-def*)

**lemma** *least-fixpoint-char*: *is-least-fixpoint f x* $\longleftrightarrow$ *has-least-fixpoint f* $\wedge$ $x = \mu$ *f*
  **by** (*metis least-fixpoint-same has-least-fixpoint-def*)

**lemma** *least-prefixpoint-char*: *is-least-prefixpoint f x* $\longleftrightarrow$ *has-least-prefixpoint f* $\wedge$ $x = p\mu$ *f*
  **by** (*metis least-prefixpoint-same has-least-prefixpoint-def*)

**lemma** *greatest-fixpoint-char*: *is-greatest-fixpoint f x* $\longleftrightarrow$ *has-greatest-fixpoint f* $\wedge$ $x = \nu$ *f*
  **by** (*metis greatest-fixpoint-same has-greatest-fixpoint-def*)

**lemma** *greatest-postfixpoint-char*: *is-greatest-postfixpoint f x* $\leftrightarrow$ *has-greatest-postfixpoint f* $\wedge$ $x = p\nu\ f$
**by** (*metis greatest-postfixpoint-same has-greatest-postfixpoint-def*)

**lemma** *least-prefixpoint-fixpoint*: *has-least-prefixpoint f* $\wedge$ *isotone f* $\rightarrow$ *is-least-fixpoint f* $(p\mu\ f)$
**by** (*smt eq-iff is-least-fixpoint-def is-least-prefixpoint-def isotone-def least-prefixpoint*)

**lemma** *pmu-mu*: *has-least-prefixpoint f* $\wedge$ *isotone f* $\rightarrow$ $p\mu\ f = \mu\ f$
**by** (*smt has-least-fixpoint-def is-least-fixpoint-def least-fixpoint-unique least-prefixpoint-fixpoint least-fixpoint*)

**definition** *lifted-less-eq* :: $('a \Rightarrow 'a) \Rightarrow ('a \Rightarrow 'a) \Rightarrow$ *bool* ((- $\leq\leq$ -) [*51*, *51*] *50*)
**where** $f \leq\leq g \leftrightarrow (\forall x\ .\ f(x) \leq g(x))$

**lemma** *lifted-reflexive*: $f = g \rightarrow f \leq\leq g$
**by** (*metis lifted-less-eq-def order-refl*)

**lemma** *lifted-transitive*: $f \leq\leq g \wedge g \leq\leq h \rightarrow f \leq\leq h$
**by** (*smt lifted-less-eq-def order-trans*)

**lemma** *lifted-antisymmetric*: $f \leq\leq g \wedge g \leq\leq f \rightarrow f = g$
**by** (*metis antisym ext lifted-less-eq-def*)

**lemma** *pmu-isotone*: *has-least-prefixpoint f* $\wedge$ *has-least-prefixpoint g* $\wedge$ $f \leq\leq g \rightarrow p\mu\ f \leq p\mu\ g$
**by** (*smt is-least-prefixpoint-def least-prefixpoint lifted-less-eq-def order-trans*)

**lemma** *mu-isotone*: *has-least-prefixpoint f* $\wedge$ *has-least-prefixpoint g* $\wedge$ *isotone f* $\wedge$ *isotone g* $\wedge$ $f \leq\leq g \rightarrow \mu\ f \leq \mu\ g$
**by** (*metis pmu-isotone pmu-mu*)

**lemma** *greatest-postfixpoint-fixpoint*: *has-greatest-postfixpoint f* $\wedge$ *isotone f* $\rightarrow$ *is-greatest-fixpoint f* $(p\nu\ f)$
**by** (*smt eq-iff is-greatest-fixpoint-def is-greatest-postfixpoint-def isotone-def greatest-postfixpoint*)

**lemma** *pnu-nu*: *has-greatest-postfixpoint f* $\wedge$ *isotone f* $\rightarrow$ $p\nu\ f = \nu\ f$
**by** (*smt has-greatest-fixpoint-def is-greatest-fixpoint-def greatest-fixpoint-unique greatest-postfixpoint-fixpoint greatest-fixpoint*)

**lemma** *pnu-isotone*: *has-greatest-postfixpoint f* $\wedge$ *has-greatest-postfixpoint g* $\wedge$ $f \leq\leq g \rightarrow p\nu\ f \leq p\nu\ g$
**by** (*smt is-greatest-postfixpoint-def lifted-less-eq-def order-trans greatest-postfixpoint*)

**lemma** *nu-isotone*: *has-greatest-postfixpoint f* $\wedge$ *has-greatest-postfixpoint g* $\wedge$ *isotone f* $\wedge$ *isotone g* $\wedge$ $f \leq\leq g \rightarrow \nu\ f \leq \nu\ g$
**by** (*metis pnu-isotone pnu-nu*)

**lemma** *mu-square*: *isotone f* $\wedge$ *has-least-fixpoint f* $\wedge$ *has-least-fixpoint* $(f \circ f) \rightarrow \mu\ f = \mu\ (f \circ f)$
**by** (*metis antisym is-least-fixpoint-def isotone-def least-fixpoint-char least-fixpoint-unique o-apply*)

**lemma** *nu-square*: *isotone f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *has-greatest-fixpoint* $(f \circ f) \rightarrow \nu\ f = \nu\ (f \circ f)$
**by** (*metis antisym is-greatest-fixpoint-def isotone-def greatest-fixpoint-char greatest-fixpoint-unique o-apply*)

**lemma** *mu-roll*: *isotone g* $\wedge$ *has-least-fixpoint* $(f \circ g) \wedge$ *has-least-fixpoint* $(g \circ f) \rightarrow \mu\ (g \circ f) = g(\mu\ (f \circ g))$

**apply** (*rule impI*, *rule antisym*)
**apply** (*smt is-least-fixpoint-def least-fixpoint o-apply*)
**by** (*smt is-least-fixpoint-def isotone-def least-fixpoint o-apply*)

**lemma** *nu-roll*: *isotone g* ∧ *has-greatest-fixpoint* (*f* ∘ *g*) ∧ *has-greatest-fixpoint* (*g* ∘ *f*) → *ν* (*g* ∘ *f*) = *g*(*ν* (*f* ∘ *g*))
**apply** (*rule impI*, *rule antisym*)
**apply** (*smt is-greatest-fixpoint-def greatest-fixpoint isotone-def o-apply*)
**by** (*smt is-greatest-fixpoint-def greatest-fixpoint o-apply*)

**lemma** *mu-below-nu*: *has-least-fixpoint f* ∧ *has-greatest-fixpoint f* → *μ f* ≤ *ν f*
**by** (*metis is-greatest-fixpoint-def is-least-fixpoint-def least-fixpoint greatest-fixpoint*)

**lemma** *pmu-below-pnu-fix*: *has-fixpoint f* ∧ *has-least-prefixpoint f* ∧ *has-greatest-postfixpoint f* → *pμ f* ≤ *pν f*
**by** (*smt has-fixpoint-def is-fixpoint-def is-greatest-postfixpoint-def is-least-prefixpoint-def le-less order-trans least-prefixpoint greatest-postfixpoint*)

**lemma** *pmu-below-pnu-iso*: *isotone f* ∧ *has-least-prefixpoint f* ∧ *has-greatest-postfixpoint f* → *pμ f* ≤ *pν f*
**by** (*metis has-fixpoint-def is-fixpoint-def is-least-fixpoint-def least-prefixpoint-fixpoint pmu-below-pnu-fix*)

**definition** *galois* :: (*'a* ⇒ *'a*) ⇒ (*'a* ⇒ *'a*) ⇒ *bool*
**where** *galois l u* ↔ (∀ *x y* . *l*(*x*) ≤ *y* ↔ *x* ≤ *u*(*y*))

**lemma** *galois-char*: *galois l u* ↔ (∀ *x* . *x* ≤ *u*(*l*(*x*))) ∧ (∀ *x* . *l*(*u*(*x*)) ≤ *x*) ∧ *isotone l* ∧ *isotone u*
**apply** (*rule iffI*)
**apply** (*smt galois-def isotone-def order-refl order-trans*)
**by** (*metis galois-def isotone-def order-trans*)

**lemma** *galois-closure*: *galois l u* → *l*(*x*) = *l*(*u*(*l*(*x*))) ∧ *u*(*x*) = *u*(*l*(*u*(*x*)))
**by** (*smt antisym galois-char isotone-def*)

**lemma** *mu-fusion-1*: *galois l u* ∧ *isotone h* ∧ *has-least-prefixpoint g* ∧ *has-least-fixpoint h* ∧ *l*(*g*(*u*(*μ h*))) ≤ *h*(*l*(*u*(*μ h*))) → *l*(*pμ g*) ≤ *μ h*
**proof**
**assume** *1*: *galois l u* ∧ *isotone h* ∧ *has-least-prefixpoint g* ∧ *has-least-fixpoint h* ∧ *l*(*g*(*u*(*μ h*))) ≤ *h*(*l*(*u*(*μ h*)))
**hence** *l*(*u*(*μ h*)) ≤ *μ h*
**by** (*metis galois-char*)
**hence** *g*(*u*(*μ h*)) ≤ *u*(*μ h*) **using** *1*
**by** (*smt galois-def least-fixpoint-same least-fixpoint-unique is-least-fixpoint-def isotone-def order-trans*)
**thus** *l*(*pμ g*) ≤ *μ h* **using** *1*
**by** (*metis galois-def least-prefixpoint is-least-prefixpoint-def*)
**qed**

**lemma** *mu-fusion-2*: *galois l u* ∧ *isotone h* ∧ *has-least-prefixpoint g* ∧ *has-least-fixpoint h* ∧ *l* ∘ *g* ≤≤ *h* ∘ *l* → *l*(*pμ g*) ≤ *μ h*
**by** (*metis lifted-less-eq-def mu-fusion-1 o-apply*)

**lemma** *mu-fusion-equal-1*: *galois l u* ∧ *isotone g* ∧ *isotone h* ∧ *has-least-prefixpoint g* ∧ *has-least-fixpoint h* ∧ *l*(*g*(*u*(*μ h*))) ≤ *h*(*l*(*u*(*μ h*))) ∧ *l*(*g*(*pμ g*)) = *h*(*l*(*pμ g*)) → *μ h* = *l*(*pμ g*) ∧ *μ h* = *l*(*μ g*)
**by** (*metis antisym least-fixpoint least-prefixpoint-fixpoint is-least-fixpoint-def mu-fusion-1 pmu-mu*)

**lemma** *mu-fusion-equal-2*: *galois l u* $\wedge$ *isotone h* $\wedge$ *has-least-prefixpoint g* $\wedge$ *has-least-prefixpoint h* $\wedge$ $l(g(u(\mu\ h))) \leq h(l(u(\mu\ h)))$ $\wedge$ $l(g(p\mu\ g)) = h(l(p\mu\ g)) \rightarrow p\mu\ h = l(p\mu\ g) \wedge \mu\ h = l(p\mu\ g)$
**by** (*smt antisym galois-char least-fixpoint-char least-prefixpoint least-prefixpoint-fixpoint is-least-prefixpoint-def isotone-def mu-fusion-1*)

**lemma** *mu-fusion-equal-3*: *galois l u* $\wedge$ *isotone g* $\wedge$ *isotone h* $\wedge$ *has-least-prefixpoint g* $\wedge$ *has-least-fixpoint h* $\wedge$ $l \circ g = h \circ l \rightarrow \mu\ h = l(p\mu\ g) \wedge \mu\ h = l(\mu\ g)$
**by** (*smt mu-fusion-equal-1 o-apply order-refl*)

**lemma** *mu-fusion-equal-4*: *galois l u* $\wedge$ *isotone h* $\wedge$ *has-least-prefixpoint g* $\wedge$ *has-least-prefixpoint h* $\wedge$ $l \circ g = h \circ l \rightarrow p\mu\ h = l(p\mu\ g) \wedge \mu\ h = l(p\mu\ g)$
**by** (*smt mu-fusion-equal-2 o-apply order-refl*)

**lemma** *nu-fusion-1*: *galois l u* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint g* $\wedge$ *has-greatest-fixpoint h* $\wedge$ $h(u(l(\nu\ h))) \leq u(g(l(\nu\ h))) \rightarrow \nu\ h \leq u(p\nu\ g)$
**proof**
**assume** *1*: *galois l u* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint g* $\wedge$ *has-greatest-fixpoint h* $\wedge$ $h(u(l(\nu\ h))) \leq u(g(l(\nu\ h)))$
**hence** $\nu\ h \leq u(l(\nu\ h))$
**by** (*metis galois-char*)
**hence** $l(\nu\ h) \leq g(l(\nu\ h))$ **using** *1*
**by** (*smt galois-def greatest-fixpoint-same greatest-fixpoint-unique is-greatest-fixpoint-def isotone-def order-trans*)
**thus** $\nu\ h \leq u(p\nu\ g)$ **using** *1*
**by** (*metis galois-def greatest-postfixpoint is-greatest-postfixpoint-def*)
**qed**

**lemma** *nu-fusion-2*: *galois l u* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint g* $\wedge$ *has-greatest-fixpoint h* $\wedge$ $h \circ u \leq\leq u \circ g \rightarrow \nu\ h \leq u(p\nu\ g)$
**by** (*metis lifted-less-eq-def nu-fusion-1 o-apply*)

**lemma** *nu-fusion-equal-1*: *galois l u* $\wedge$ *isotone g* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint g* $\wedge$ *has-greatest-fixpoint h* $\wedge$ $h(u(l(\nu\ h))) \leq u(g(l(\nu\ h)))$ $\wedge$ $h(u(p\nu\ g)) = u(g(p\nu\ g)) \rightarrow \nu\ h = u(p\nu\ g) \wedge \nu\ h = u(\nu\ g)$
**by** (*metis antisym greatest-fixpoint greatest-postfixpoint-fixpoint is-greatest-fixpoint-def nu-fusion-1 pnu-nu*)

**lemma** *nu-fusion-equal-2*: *galois l u* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint g* $\wedge$ *has-greatest-postfixpoint h* $\wedge$ $h(u(l(\nu\ h))) \leq u(g(l(\nu\ h)))$ $\wedge$ $h(u(p\nu\ g)) = u(g(p\nu\ g)) \rightarrow p\nu\ h = u(p\nu\ g) \wedge \nu\ h = u(p\nu\ g)$
**by** (*smt antisym galois-char greatest-fixpoint-char greatest-postfixpoint greatest-postfixpoint-fixpoint is-greatest-postfixpoint-def isotone-def nu-fusion-1*)

**lemma** *nu-fusion-equal-3*: *galois l u* $\wedge$ *isotone g* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint g* $\wedge$ *has-greatest-fixpoint h* $\wedge$ $h \circ u = u \circ g \rightarrow \nu\ h = u(p\nu\ g) \wedge \nu\ h = u(\nu\ g)$
**by** (*metis nu-fusion-equal-1 o-apply order-refl*)

**lemma** *nu-fusion-equal-4*: *galois l u* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint g* $\wedge$ *has-greatest-postfixpoint h* $\wedge$ $h \circ u = u \circ g \rightarrow p\nu\ h = u(p\nu\ g) \wedge \nu\ h = u(p\nu\ g)$
**by** (*smt nu-fusion-equal-2 o-apply order-refl*)

**lemma** *mu-exchange-1*: *galois l u* $\wedge$ *isotone g* $\wedge$ *isotone h* $\wedge$ *has-least-prefixpoint* $(l \circ h)$ $\wedge$ *has-least-prefixpoint* $(h \circ g)$ $\wedge$ *has-least-fixpoint* $(g \circ h)$ $\wedge$ $l \circ h \circ g \leq\leq g \circ h \circ l \rightarrow \mu(l \circ h) \leq \mu(g \circ h)$
**by** (*smt galois-char is-least-prefixpoint-def isotone-def least-fixpoint-char least-prefixpoint least-prefixpoint-fixpoint mu-fusion-2 mu-roll o-apply o-assoc*)

**lemma** *mu-exchange-2*: *galois l u* $\wedge$ *isotone g* $\wedge$ *isotone h* $\wedge$ *has-least-prefixpoint* $(l \circ h)$ $\wedge$ *has-least-prefixpoint* $(h \circ l)$ $\wedge$ *has-least-prefixpoint* $(h \circ g)$ $\wedge$ *has-least-fixpoint* $(g \circ h)$ $\wedge$ *has-least-fixpoint* $(h \circ g)$ $\wedge$ $l \circ h \circ g \leq\leq g \circ h \circ l \rightarrow \mu(h \circ l) \leq \mu(h \circ g)$
**by** (*smt galois-char isotone-def least-fixpoint-char least-prefixpoint-fixpoint mu-exchange-1 mu-roll o-apply*)

**lemma** *mu-exchange-equal*: *galois l u* $\wedge$ *galois k t* $\wedge$ *isotone h* $\wedge$ *has-least-prefixpoint* $(l \circ h)$ $\wedge$ *has-least-prefixpoint* $(h \circ l)$ $\wedge$ *has-least-prefixpoint* $(k \circ h)$ $\wedge$ *has-least-prefixpoint* $(h \circ k)$ $\wedge$

$l \circ h \circ k = k \circ h \circ l \rightarrow \mu(l \circ h) = \mu(k \circ h) \wedge \mu(h \circ l) = \mu(h \circ k)$
**by** (*smt antisym galois-char isotone-def least-fixpoint-char least-prefixpoint-fixpoint lifted-reflexive mu-exchange-1 mu-exchange-2 o-apply*)

**lemma** *nu-exchange-1*: *galois l u* $\wedge$ *isotone g* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint* $(u \circ h)$ $\wedge$ *has-greatest-postfixpoint* $(h \circ g)$ $\wedge$ *has-greatest-fixpoint* $(g \circ h)$ $\wedge$ $g \circ h \circ u \leq\leq u \circ h \circ g \rightarrow \nu(g \circ h) \leq \nu(u \circ h)$
**by** (*smt galois-char is-greatest-postfixpoint-def isotone-def greatest-fixpoint-char greatest-postfixpoint greatest-postfixpoint-fixpoint nu-fusion-2 nu-roll o-apply o-assoc*)

**lemma** *nu-exchange-2*: *galois l u* $\wedge$ *isotone g* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint* $(u \circ h)$ $\wedge$ *has-greatest-postfixpoint* $(h \circ u)$ $\wedge$ *has-greatest-postfixpoint* $(h \circ g)$ $\wedge$ *has-greatest-fixpoint* $(g \circ h)$ $\wedge$ *has-greatest-fixpoint* $(h \circ g)$ $\wedge$ $g \circ h \circ u \leq\leq u \circ h \circ g \rightarrow \nu(h \circ g) \leq \nu(h \circ u)$
**by** (*smt galois-char isotone-def greatest-fixpoint-char greatest-postfixpoint-fixpoint nu-exchange-1 nu-roll o-apply*)

**lemma** *nu-exchange-equal*: *galois l u* $\wedge$ *galois k t* $\wedge$ *isotone h* $\wedge$ *has-greatest-postfixpoint* $(u \circ h)$ $\wedge$ *has-greatest-postfixpoint* $(h \circ u)$ $\wedge$ *has-greatest-postfixpoint* $(t \circ h)$ $\wedge$ *has-greatest-postfixpoint* $(h \circ t)$ $\wedge$ $u \circ h \circ t = t \circ h \circ u \rightarrow \nu(u \circ h) = \nu(t \circ h) \wedge \nu(h \circ u) = \nu(h \circ t)$
**by** (*smt antisym galois-char isotone-def greatest-fixpoint-char greatest-postfixpoint-fixpoint lifted-reflexive nu-exchange-1 nu-exchange-2 o-apply*)

**lemma** *mu-commute-fixpoint-1*: *isotone f* $\wedge$ *has-least-fixpoint* $(f \circ g)$ $\wedge$ $f \circ g = g \circ f \rightarrow$ *is-fixpoint f* $(\mu(f \circ g))$
**by** (*metis is-fixpoint-def mu-roll*)

**lemma** *mu-commute-fixpoint-2*: *isotone g* $\wedge$ *has-least-fixpoint* $(f \circ g)$ $\wedge$ $f \circ g = g \circ f \rightarrow$ *is-fixpoint g* $(\mu(f \circ g))$
**by** (*metis is-fixpoint-def mu-roll*)

**lemma** *mu-commute-least-fixpoint*: *isotone f* $\wedge$ *isotone g* $\wedge$ *has-least-fixpoint f* $\wedge$ *has-least-fixpoint g* $\wedge$ *has-least-fixpoint* $(f \circ g)$ $\wedge$ $f \circ g = g \circ f \rightarrow (\mu(f \circ g) = \mu\ f \rightarrow \mu\ g \leq \mu\ f)$
**by** (*metis is-least-fixpoint-def least-fixpoint-same least-fixpoint-unique mu-roll*)

**lemma** *nu-commute-fixpoint-1*: *isotone f* $\wedge$ *has-greatest-fixpoint* $(f \circ g)$ $\wedge$ $f \circ g = g \circ f \rightarrow$ *is-fixpoint f* $(\nu(f \circ g))$
**by** (*metis is-fixpoint-def nu-roll*)

**lemma** *nu-commute-fixpoint-2*: *isotone g* $\wedge$ *has-greatest-fixpoint* $(f \circ g)$ $\wedge$ $f \circ g = g \circ f \rightarrow$ *is-fixpoint g* $(\nu(f \circ g))$
**by** (*metis is-fixpoint-def nu-roll*)

**lemma** *nu-commute-greatest-fixpoint*: *isotone f* $\wedge$ *isotone g* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *has-greatest-fixpoint g* $\wedge$ *has-greatest-fixpoint* $(f \circ g)$ $\wedge$ $f \circ g = g \circ f \rightarrow (\nu(f \circ g) = \nu\ f \rightarrow \nu\ f \leq \nu\ g)$
**by** (*smt is-greatest-fixpoint-def greatest-fixpoint-same greatest-fixpoint-unique nu-roll*)

**lemma** *mu-diagonal-1*: *isotone* $(\lambda x\ .\ f\ x\ x)$ $\wedge$ $(\forall x\ .$ *isotone* $(\lambda y\ .\ f\ x\ y))$ $\wedge$ *isotone* $(\lambda x\ .\ \mu(\lambda y\ .\ f\ x\ y))$ $\wedge$ $(\forall x\ .$ *has-least-fixpoint* $(\lambda y\ .\ f\ x\ y))$ $\wedge$ *has-least-prefixpoint* $(\lambda x\ .\ \mu(\lambda y\ .\ f\ x\ y)) \rightarrow \mu(\lambda x\ .\ f\ x\ x) = \mu(\lambda x\ .\ \mu(\lambda y\ .\ f\ x\ y))$
**by** (*smt is-least-fixpoint-def is-least-prefixpoint-def least-fixpoint-same least-fixpoint-unique least-prefixpoint least-prefixpoint-fixpoint*)

**lemma** *mu-diagonal-2*: $(\forall x\ .$ *isotone* $(\lambda y\ .\ f\ x\ y)$ $\wedge$ *isotone* $(\lambda y\ .\ f\ y\ x)$ $\wedge$ *has-least-prefixpoint* $(\lambda y\ .\ f\ x\ y))$ $\wedge$ *has-least-prefixpoint* $(\lambda x\ .\ \mu(\lambda y\ .\ f\ x\ y)) \rightarrow \mu(\lambda x\ .\ f\ x\ x) = \mu(\lambda x\ .\ \mu(\lambda y\ .\ f\ x\ y))$
**by** (*smt is-least-fixpoint-def is-least-prefixpoint-def isotone-def least-fixpoint-same least-prefixpoint least-prefixpoint-fixpoint*)

**lemma** *nu-diagonal-1*: *isotone* $(\lambda x\ .\ f\ x\ x)$ $\wedge$ $(\forall x\ .$ *isotone* $(\lambda y\ .\ f\ x\ y))$ $\wedge$ *isotone* $(\lambda x\ .\ \nu(\lambda y\ .\ f\ x\ y))$ $\wedge$ $(\forall x\ .$ *has-greatest-fixpoint* $(\lambda y\ .\ f\ x\ y))$ $\wedge$ *has-greatest-postfixpoint* $(\lambda x\ .\ \nu(\lambda y\ .\ f\ x\ y)) \rightarrow \nu(\lambda x\ .\ f\ x\ x) = \nu(\lambda x\ .\ \nu(\lambda y\ .\ f\ x\ y))$
**by** (*smt is-greatest-fixpoint-def is-greatest-postfixpoint-def greatest-fixpoint-same greatest-fixpoint-unique greatest-postfixpoint greatest-postfixpoint-fixpoint*)

**lemma** *nu-diagonal-2*: $(\forall x\ .\ isotone\ (\lambda y\ .\ f\ x\ y) \wedge isotone\ (\lambda y\ .\ f\ y\ x) \wedge has\text{-}greatest\text{-}postfixpoint\ (\lambda y\ .\ f\ x\ y)) \wedge has\text{-}greatest\text{-}postfixpoint\ (\lambda x\ .\ \nu(\lambda y\ .\ f\ x\ y)) \rightarrow \nu(\lambda x\ .\ f\ x\ x) = \nu(\lambda x\ .\ \nu(\lambda y\ .\ f\ x\ y))$
**using** [[ *smt-timeout* = *120* ]] **by** (*smt is-greatest-fixpoint-def is-greatest-postfixpoint-def isotone-def greatest-fixpoint-same greatest-postfixpoint greatest-postfixpoint-fixpoint*)

**end**

**class** *semiring* = *mult* + *one* + *ord* + *plus* + *zero* +
**assumes** *add-associative* : $(x + y) + z = x + (y + z)$
**assumes** *add-commutative* : $x + y = y + x$
**assumes** *add-idempotent* : $x + x = x$
**assumes** *add-left-zero* : $0 + x = x$
**assumes** *mult-associative* : $(x\ ;\ y)\ ;\ z = x\ ;\ (y\ ;\ z)$
**assumes** *mult-left-one* : $1\ ;\ x = x$
**assumes** *mult-right-one* : $x\ ;\ 1 = x$
**assumes** *mult-left-dist-add* : $x\ ;\ (y + z) = x\ ;\ y + x\ ;\ z$
**assumes** *mult-right-dist-add*: $(x + y)\ ;\ z = x\ ;\ z + y\ ;\ z$
**assumes** *mult-left-zero* : $0\ ;\ x = 0$
**assumes** *less-eq-def* : $x \leq y \leftrightarrow x + y = y$
**assumes** *less-def* : $x < y \leftrightarrow x \leq y \wedge \neg\ (y \leq x)$

**begin**

**subclass** *order*
**by** *unfold-locales* (*metis less-def*, *metis add-idempotent less-eq-def*, *metis add-associative less-eq-def*, *metis add-commutative less-eq-def*)

**lemma** *add-left-isotone*: $x \leq y \rightarrow x + z \leq y + z$
**by** (*smt add-associative add-commutative add-idempotent less-eq-def*)

**lemma** *add-right-isotone*: $x \leq y \rightarrow z + x \leq z + y$
**by** (*metis add-commutative add-left-isotone*)

**lemma** *add-isotone*: $w \leq y \wedge x \leq z \rightarrow w + x \leq y + z$
**by** (*smt add-associative add-commutative less-eq-def*)

**lemma** *add-left-upper-bound*: $x \leq x + y$
**by** (*metis add-associative add-idempotent less-eq-def*)

**lemma** *add-right-upper-bound*: $y \leq x + y$
**by** (*metis add-commutative add-left-upper-bound*)

**lemma** *add-least-upper-bound*: $x \leq z \wedge y \leq z \leftrightarrow x + y \leq z$
**by** (*smt add-associative add-commutative add-left-upper-bound less-eq-def*)

**lemma** *add-left-divisibility*: $x \leq y \leftrightarrow (\exists z\ .\ x + z = y)$

**by** (*metis add-left-upper-bound less-eq-def*)

**lemma** *add-right-divisibility*: $x \leq y \leftrightarrow (\exists z \,.\, z + x = y)$
**by** (*metis add-commutative add-left-divisibility*)

**lemma** *add-right-zero*: $x + 0 = x$
**by** (*metis add-commutative add-left-zero*)

**lemma** *zero-least*: $0 \leq x$
**by** (*metis add-left-upper-bound add-left-zero*)

**lemma** *mult-left-isotone*: $x \leq y \rightarrow x \,;\, z \leq y \,;\, z$
**by** (*metis less-eq-def mult-right-dist-add*)

**lemma** *mult-right-isotone*: $x \leq y \rightarrow z \,;\, x \leq z \,;\, y$
**by** (*metis less-eq-def mult-left-dist-add*)

**lemma** *mult-isotone*: $w \leq y \wedge x \leq z \rightarrow w \,;\, x \leq y \,;\, z$
**by** (*smt mult-left-isotone mult-right-isotone order-trans*)

**lemma** *mult-left-subdist-add-left*: $x \,;\, y \leq x \,;\, (y + z)$
**by** (*metis add-left-upper-bound mult-left-dist-add*)

**lemma** *mult-left-subdist-add-right*: $x \,;\, z \leq x \,;\, (y + z)$
**by** (*metis add-right-upper-bound mult-left-dist-add*)

**lemma** *mult-right-subdist-add-left*: $x \,;\, z \leq (x + y) \,;\, z$
**by** (*metis add-left-upper-bound mult-right-dist-add*)

**lemma** *mult-right-subdist-add-right*: $y \,;\, z \leq (x + y) \,;\, z$
**by** (*metis add-right-upper-bound mult-right-dist-add*)

**lemma** *case-split-left*: $1 \leq w + z \wedge w \,;\, x \leq y \wedge z \,;\, x \leq y \rightarrow x \leq y$
**by** (*smt add-least-upper-bound mult-left-isotone mult-left-one mult-right-dist-add order-trans*)

**lemma** *case-split-left-equal*: $w + z = 1 \wedge w \,;\, x = w \,;\, y \wedge z \,;\, x = z \,;\, y \rightarrow x = y$
**by** (*metis mult-left-one mult-right-dist-add*)

**lemma** *case-split-right*: $1 \leq w + z \wedge x \,;\, w \leq y \wedge x \,;\, z \leq y \rightarrow x \leq y$
**by** (*smt add-least-upper-bound mult-right-isotone mult-right-one mult-left-dist-add order-trans*)

**lemma** *case-split-right-equal*: $w + z = 1 \wedge x \,;\, w = y \,;\, w \wedge x \,;\, z = y \,;\, z \rightarrow x = y$
**by** (*metis mult-left-dist-add mult-right-one*)

**lemma** *zero-right-mult-decreasing*: $x \,;\, 0 \leq x$
**by** (*metis add-right-zero mult-left-subdist-add-right mult-right-one*)

**lemma** *add-same-context*:$x \leq y + z \wedge y \leq x + z \rightarrow x + z = y + z$
**by** (*smt add-associative add-commutative less-eq-def*)

**end**

**class** *semiring-T* = *semiring* +
**fixes** $T :: 'a$ ($\top$)
**assumes** *add-left-top*: $T + x = T$

**begin**

**lemma** *add-right-top*: $x + T = T$
**by** (*metis add-commutative add-left-top*)

**lemma** *top-greatest*: $x \leq T$
**by** (*metis add-left-top add-right-upper-bound*)

**lemma** *top-left-mult-increasing*: $x \leq T \ ; \ x$
**by** (*metis mult-left-isotone mult-left-one top-greatest*)

**lemma** *top-right-mult-increasing*: $x \leq x \ ; \ T$
**by** (*metis mult-right-isotone mult-right-one top-greatest*)

**lemma** *top-mult-top*: $T \ ; \ T = T$
**by** (*metis add-right-divisibility add-right-top top-right-mult-increasing*)

**definition** *vector* :: $'a \Rightarrow$ *bool*
**where** *vector* $x \leftrightarrow x = x \ ; \ T$

**lemma** *vector-zero*: *vector 0*
**by** (*metis mult-left-zero vector-def*)

**lemma** *vector-top*: *vector T*
**by** (*metis top-mult-top vector-def*)

**lemma** *vector-add-closed*: *vector* $x \wedge$ *vector* $y \rightarrow$ *vector* $(x + y)$
**by** (*metis mult-right-dist-add vector-def*)

**lemma** *vector-left-mult-closed*: *vector* $y \rightarrow$ *vector* $(x \ ; \ y)$
**by** (*metis mult-associative vector-def*)

**end**

**class** *itering-0* = *semiring* +
**fixes** *circ* :: $'a \Rightarrow 'a$ ($-^\circ$ [*100*] *100*)

**assumes** *circ-mult*: $(x\ ;\ y)^\circ = 1 + x\ ;\ (y\ ;\ x)^\circ\ ;\ y$
**assumes** *circ-add*: $(x + y)^\circ = (x^\circ\ ;\ y)^\circ\ ;\ x^\circ$

**begin**

**lemma** *circ-isotone*: $x \leq y \rightarrow x^\circ \leq y^\circ$
**by** (*metis add-left-divisibility circ-add circ-mult mult-left-one mult-right-subdist-add-left*)

**lemma** *circ-slide*: $x\ ;\ (y\ ;\ x)^\circ = (x\ ;\ y)^\circ\ ;\ x$
**by** (*smt circ-mult mult-associative mult-left-dist-add mult-left-one mult-right-dist-add mult-right-one*)

**lemma** *circ-right-unfold*: $1 + x^\circ\ ;\ x = x^\circ$
**by** (*metis circ-mult mult-left-one mult-right-one*)

**lemma** *circ-left-unfold*: $1 + x\ ;\ x^\circ = x^\circ$
**by** (*metis circ-mult circ-slide mult-associative mult-right-one*)

**lemma** *circ-zero*: $0^\circ = 1$
**by** (*metis add-right-zero circ-left-unfold mult-left-zero*)

**lemma** *circ-increasing*: $x \leq x^\circ$
**by** (*metis add-right-upper-bound circ-right-unfold mult-left-one mult-right-subdist-add-left order-trans*)

**lemma** *circ-transitive-equal*: $x^\circ\ ;\ x^\circ = x^\circ$
**by** (*metis add-idempotent circ-add circ-mult mult-associative mult-left-one mult-right-one*)

**lemma** *circ-circ-circ*: $x^{\circ\circ\circ} = x^{\circ\circ}$
**by** (*metis add-idempotent circ-add circ-increasing circ-transitive-equal less-eq-def*)

**lemma** *circ-one*: $1^\circ = 1^{\circ\circ}$
**by** (*metis circ-circ-circ circ-zero*)

**lemma** *circ-add-1*: $(x + y)^\circ = x^\circ\ ;\ (y\ ;\ x^\circ)^\circ$
**by** (*metis circ-add circ-slide*)

**lemma** *circ-reflexive*: $1 \leq x^\circ$
**by** (*metis add-left-divisibility circ-right-unfold*)

**lemma** *circ-add-2*: $(x + y)^\circ \leq (x^\circ\ ;\ y^\circ)^\circ$
**by** (*metis add-least-upper-bound circ-increasing circ-isotone circ-reflexive mult-isotone mult-left-one mult-right-one*)

**lemma** *mult-zero-circ*: $(x\ ;\ 0)^\circ = 1 + x\ ;\ 0$
**by** (*metis circ-mult mult-associative mult-left-zero*)

**lemma** *mult-zero-add-circ*: $(x + y\ ;\ 0)^\circ = x^\circ\ ;\ (y\ ;\ 0)^\circ$
**by** (*metis circ-add-1 mult-associative mult-left-zero*)

**lemma** *mult-zero-add-circ-2*: $(x + y\ ;\ 0)^\circ = x^\circ + x^\circ\ ;\ y\ ;\ 0$
**by** (*metis mult-associative mult-left-dist-add mult-right-one mult-zero-add-circ mult-zero-circ*)

**lemma** *circ-plus-same*: $x^\circ\ ;\ x = x\ ;\ x^\circ$
**by** (*metis circ-slide mult-left-one mult-right-one*)

**lemma** *circ-plus-one*: $x^\circ = 1 + x^\circ$
**by** (*metis less-eq-def circ-reflexive*)

**lemma** *circ-rtc-2*: $1 + x + x^\circ\ ;\ x^\circ = x^\circ$
**by** (*metis add-associative circ-increasing circ-plus-one circ-transitive-equal less-eq-def*)

**lemma** *circ-unfold-sum*: $(x + y)^\circ = x^\circ + x^\circ\ ;\ y\ ;\ (x + y)^\circ$
**by** (*smt circ-add-1 circ-mult mult-associative mult-left-dist-add mult-right-one*)

**lemma** *circ-loop-fixpoint*: $y\ ;\ (y^\circ\ ;\ z) + z = y^\circ\ ;\ z$
**by** (*metis add-commutative circ-left-unfold mult-associative mult-left-one mult-right-dist-add*)

**lemma** *left-plus-below-circ*: $x\ ;\ x^\circ \leq x^\circ$
**by** (*metis add-right-upper-bound circ-left-unfold*)

**lemma** *right-plus-below-circ*: $x^\circ\ ;\ x \leq x^\circ$
**by** (*metis circ-plus-same left-plus-below-circ*)

**lemma** *circ-add-upper-bound*: $x \leq z^\circ \wedge y \leq z^\circ \rightarrow x + y \leq z^\circ$
**by** (*metis add-least-upper-bound*)

**lemma** *circ-mult-upper-bound*: $x \leq z^\circ \wedge y \leq z^\circ \rightarrow x\ ;\ y \leq z^\circ$
**by** (*metis mult-isotone circ-transitive-equal*)

**lemma** *circ-sub-dist*: $x^\circ \leq (x + y)^\circ$
**by** (*metis add-left-upper-bound circ-isotone*)

**lemma** *circ-sub-dist-1*: $x \leq (x + y)^\circ$
**by** (*metis add-least-upper-bound circ-increasing*)

**lemma** *circ-sub-dist-2*: $x\ ;\ y \leq (x + y)^\circ$
**by** (*metis add-commutative circ-mult-upper-bound circ-sub-dist-1*)

**lemma** *circ-sub-dist-3*: $x^\circ\ ;\ y^\circ \leq (x + y)^\circ$
**by** (*metis add-commutative circ-mult-upper-bound circ-sub-dist*)

**lemma** *circ-sup-one-left-unfold*: $1 \leq x \rightarrow x\ ;\ x^\circ = x^\circ$
**by** (*metis antisym less-eq-def mult-left-one mult-right-subdist-add-left left-plus-below-circ*)

**lemma** *circ-sup-one-right-unfold*: $1 \leq x \rightarrow x^\circ\ ;\ x = x^\circ$
**by** (*metis antisym less-eq-def mult-left-subdist-add-left mult-right-one right-plus-below-circ*)

**lemma** *circ-decompose-4*: $(x^\circ \ ; \ y^\circ)^\circ = x^\circ \ ; \ (y^\circ \ ; \ x^\circ)^\circ$
**by** (*smt circ-add circ-increasing circ-plus-one circ-slide circ-transitive-equal less-eq-def mult-associative mult-left-subdist-add-left mult-right-one*)

**lemma** *circ-decompose-5*: $(x^\circ \ ; \ y^\circ)^\circ = (y^\circ \ ; \ x^\circ)^\circ$
**by** (*metis add-associative add-commutative circ-add circ-decompose-4 circ-slide circ-zero mult-right-one*)

**lemma** *circ-decompose-6*: $x^\circ \ ; \ (y \ ; \ x^\circ)^\circ = y^\circ \ ; \ (x \ ; \ y^\circ)^\circ$
**by** (*metis add-commutative circ-add-1*)

**lemma** *circ-decompose-7*: $(x + y)^\circ = x^\circ \ ; \ y^\circ \ ; \ (x + y)^\circ$
**by** (*metis add-commutative circ-add circ-slide circ-transitive-equal mult-associative*)

**lemma** *circ-decompose-8*: $(x + y)^\circ = (x + y)^\circ \ ; \ x^\circ \ ; \ y^\circ$
**by** (*metis antisym eq-refl mult-associative mult-isotone mult-right-one circ-mult-upper-bound circ-reflexive circ-sub-dist-3*)

**lemma** *circ-decompose-9*: $(x^\circ \ ; \ y^\circ)^\circ = x^\circ \ ; \ y^\circ \ ; \ (x^\circ \ ; \ y^\circ)^\circ$
**by** (*metis circ-decompose-4 mult-associative*)

**lemma** *circ-decompose-10*: $(x^\circ \ ; \ y^\circ)^\circ = (x^\circ \ ; \ y^\circ)^\circ \ ; \ x^\circ \ ; \ y^\circ$
**by** (*metis circ-decompose-9 circ-plus-same mult-associative*)

**lemma** *circ-add-mult-zero*: $x^\circ \ ; \ y = (x + y \ ; \ 0)^\circ \ ; \ y$
**by** (*smt mult-associative mult-left-zero mult-right-one circ-add circ-slide circ-zero*)

**lemma** *circ-back-loop-fixpoint*: $(z \ ; \ y^\circ) \ ; \ y + z = z \ ; \ y^\circ$
**by** (*metis add-commutative mult-associative mult-left-dist-add mult-right-one circ-plus-same circ-left-unfold*)

**lemma** *circ-loop-is-fixpoint*: *is-fixpoint* $(\lambda x \ . \ y \ ; \ x + z) \ (y^\circ \ ; \ z)$
**by** (*metis circ-loop-fixpoint is-fixpoint-def*)

**lemma** *circ-back-loop-is-fixpoint*: *is-fixpoint* $(\lambda x \ . \ x \ ; \ y + z) \ (z \ ; \ y^\circ)$
**by** (*metis circ-back-loop-fixpoint is-fixpoint-def*)

**lemma** *circ-circ-add*: $(1 + x)^\circ = x^{\circ\circ}$
**by** (*metis add-commutative circ-add-1 circ-decompose-4 circ-zero mult-right-one*)

**lemma** *circ-circ-mult-sub*: $x^\circ \ ; \ 1^\circ \leq x^{\circ\circ}$
**by** (*metis circ-increasing circ-isotone circ-mult-upper-bound circ-reflexive*)

**lemma** *right-plus-circ*: $(x^\circ \ ; \ x)^\circ = x^\circ$
**by** (*metis add-idempotent circ-add circ-mult circ-right-unfold circ-slide*)

**lemma** *left-plus-circ*: $(x \ ; \ x^\circ)^\circ = x^\circ$
**by** (*metis circ-plus-same right-plus-circ*)

**lemma** *circ-add-sub-add-one*: $x \ ; \ x^\circ \ ; \ (x + y) \leq x \ ; \ x^\circ \ ; \ (1 + y)$

**by** (*smt add-least-upper-bound add-left-upper-bound add-right-upper-bound circ-plus-same mult-associative mult-isotone mult-left-dist-add mult-right-one right-plus-below-circ*)

**lemma** *circ-elimination*: $x\ ;\ y = 0 \rightarrow x\ ;\ y^\circ \leq x$
**by** (*metis add-left-zero circ-back-loop-fixpoint circ-plus-same le-less mult-associative mult-left-zero*)

**lemma** *circ-square*: $(x\ ;\ x)^\circ \leq x^\circ$
**by** (*metis circ-increasing circ-isotone left-plus-circ mult-right-isotone*)

**lemma** *circ-mult-sub-add*: $(x\ ;\ y)^\circ \leq (x + y)^\circ$
**by** (*metis add-left-upper-bound add-right-upper-bound circ-isotone circ-square mult-isotone order-trans*)

**end**

**class** *itering-1* = *itering-0* +
**assumes** *circ-simulate*: $z\ ;\ x \leq y\ ;\ z \rightarrow z\ ;\ x^\circ \leq y^\circ\ ;\ z$

**begin**

**lemma** *circ-circ-mult*: $1^\circ\ ;\ x^\circ = x^{\circ\circ}$
**by** (*metis antisym circ-circ-add circ-reflexive circ-simulate circ-sub-dist-3 circ-sup-one-left-unfold circ-transitive-equal mult-left-one order-refl*)

**lemma** *sub-mult-one-circ*: $x\ ;\ 1^\circ \leq 1^\circ\ ;\ x$
**by** (*metis circ-simulate mult-left-one mult-right-one order-refl*)

**end**

**class** *itering-2* = *itering-1* +
**assumes** *circ-simulate-right*: $z\ ;\ x \leq y\ ;\ z + w \rightarrow z\ ;\ x^\circ \leq y^\circ\ ;\ (z + w\ ;\ x^\circ)$
**assumes** *circ-simulate-left*: $x\ ;\ z \leq z\ ;\ y + w \rightarrow x^\circ\ ;\ z \leq (z + x^\circ\ ;\ w)\ ;\ y^\circ$

**begin**

**lemma** *circ-simulate-right-1*: $z\ ;\ x \leq y\ ;\ z \rightarrow z\ ;\ x^\circ \leq y^\circ\ ;\ z$
**by** (*metis add-right-zero circ-simulate-right mult-left-zero*)

**lemma** *circ-simulate-left-1*: $x\ ;\ z \leq z\ ;\ y \rightarrow x^\circ\ ;\ z \leq z\ ;\ y^\circ + x^\circ\ ;\ 0$
**by** (*smt add-right-zero circ-simulate-left mult-associative mult-left-zero mult-right-dist-add*)

**lemma** *circ-simulate-1*: $y\ ;\ x \leq x\ ;\ y \rightarrow y^\circ\ ;\ x^\circ \leq x^\circ\ ;\ y^\circ$
**by** (*smt add-associative add-right-zero circ-loop-fixpoint circ-simulate circ-simulate-left-1 mult-associative mult-left-zero mult-zero-add-circ-2*)

**lemma** *circ-separate-1*: $y\ ;\ x \leq x\ ;\ y \rightarrow (x + y)^\circ = x^\circ\ ;\ y^\circ$
**proof** −
**have** $y\ ;\ x \leq x\ ;\ y \rightarrow y^\circ\ ;\ x\ ;\ y^\circ \leq x\ ;\ y^\circ + y^\circ\ ;\ 0$
**by** (*smt circ-simulate-left-1 circ-transitive-equal mult-associative mult-left-isotone mult-left-zero mult-right-dist-add*)
**thus** *?thesis*
**by** (*smt add-commutative circ-add-1 circ-simulate-right circ-sub-dist-3 less-eq-def mult-associative mult-left-zero zero-right-mult-decreasing*)

**qed**

**lemma** *circ-circ-mult-1*: $x^\circ \ ; \ 1^\circ = x^{\circ\circ}$
**by** (*metis add-commutative circ-circ-add circ-separate-1 mult-left-one mult-right-one order-refl*)

**lemma** *atomicity-refinement*: $s = s \ ; \ q \wedge x = q \ ; \ x \wedge q \ ; \ b = 0 \wedge r \ ; \ b \leq b \ ; \ r \wedge r \ ; \ l \leq l \ ; \ r \wedge x \ ; \ l \leq l \ ; \ x \wedge b \ ; \ l \leq l \ ; \ b \wedge q \ ; \ l \leq l \ ; \ q \wedge r^\circ \ ; \ q \leq q \ ; \ r^\circ \wedge q \leq 1 \rightarrow s \ ; \ (x + b + r + l)^\circ \ ; \ q \leq s \ ; \ (x \ ; \ b^\circ \ ; \ q + r + l)^\circ$
**proof**
**assume** *1*: $s = s \ ; \ q \wedge x = q \ ; \ x \wedge q \ ; \ b = 0 \wedge r \ ; \ b \leq b \ ; \ r \wedge r \ ; \ l \leq l \ ; \ r \wedge x \ ; \ l \leq l \ ; \ x \wedge b \ ; \ l \leq l \ ; \ b \wedge q \ ; \ l \leq l \ ; \ q \wedge r^\circ \ ; \ q \leq q \ ; \ r^\circ \wedge q \leq 1$
**hence** $s \ ; \ (x + b + r + l)^\circ \ ; \ q = s \ ; \ l^\circ \ ; \ (x + b + r)^\circ \ ; \ q$
**using** [[ *smt-timeout = 120* ]] **by** (*smt add-commutative add-least-upper-bound circ-separate-1 mult-associative mult-left-subdist-add-right mult-right-dist-add order-trans*)
**also have** $... = s \ ; \ l^\circ \ ; \ b^\circ \ ; \ r^\circ \ ; \ q \ ; \ (x \ ; \ b^\circ \ ; \ r^\circ \ ; \ q)^\circ$ **using** *1*
**by** (*smt add-associative add-commutative circ-add-1 circ-separate-1 circ-slide mult-associative*)
**also have** $... \leq s \ ; \ l^\circ \ ; \ b^\circ \ ; \ r^\circ \ ; \ q \ ; \ (x \ ; \ b^\circ \ ; \ q \ ; \ r^\circ)^\circ$ **using** *1*
**by** (*metis circ-isotone mult-associative mult-right-isotone*)
**also have** $... \leq s \ ; \ q \ ; \ l^\circ \ ; \ b^\circ \ ; \ r^\circ \ ; \ (x \ ; \ b^\circ \ ; \ q \ ; \ r^\circ)^\circ$ **using** *1*
**by** (*metis mult-left-isotone mult-right-isotone mult-right-one*)
**also have** $... \leq s \ ; \ l^\circ \ ; \ q \ ; \ b^\circ \ ; \ r^\circ \ ; \ (x \ ; \ b^\circ \ ; \ q \ ; \ r^\circ)^\circ$ **using** *1*
**by** (*metis circ-simulate mult-associative mult-left-isotone mult-right-isotone*)
**also have** $... \leq s \ ; \ l^\circ \ ; \ r^\circ \ ; \ (x \ ; \ b^\circ \ ; \ q \ ; \ r^\circ)^\circ$ **using** *1*
**by** (*metis add-left-zero circ-back-loop-fixpoint circ-plus-same mult-associative mult-left-zero mult-left-isotone mult-right-isotone mult-right-one*)
**also have** $... \leq s \ ; \ (x \ ; \ b^\circ \ ; \ q + r + l)^\circ$ **using** *1*
**by** (*metis add-commutative circ-add-1 circ-sub-dist-3 mult-associative mult-right-isotone*)
**finally show** $s \ ; \ (x + b + r + l)^\circ \ ; \ q \leq s \ ; \ (x \ ; \ b^\circ \ ; \ q + r + l)^\circ$ **.**
**qed**

**end**

**class** *itering-3* = *itering-2* +
**assumes** *circ-simulate-right-plus*: $z \ ; \ x \leq y \ ; \ y^\circ \ ; \ z + w \rightarrow z \ ; \ x^\circ \leq y^\circ \ ; \ (z + w \ ; \ x^\circ)$
**assumes** *circ-simulate-left-plus*: $x \ ; \ z \leq z \ ; \ y^\circ + w \rightarrow x^\circ \ ; \ z \leq (z + x^\circ \ ; \ w) \ ; \ y^\circ$

**begin**

**lemma** *circ-simulate-right-plus-1*: $z \ ; \ x \leq y \ ; \ y^\circ \ ; \ z \rightarrow z \ ; \ x^\circ \leq y^\circ \ ; \ z$
**by** (*metis add-right-zero circ-simulate-right-plus mult-left-zero*)

**lemma** *circ-simulate-left-plus-1*: $x \ ; \ z \leq z \ ; \ y^\circ \rightarrow x^\circ \ ; \ z \leq z \ ; \ y^\circ + x^\circ \ ; \ 0$
**by** (*smt add-right-zero circ-simulate-left-plus mult-associative mult-left-zero mult-right-dist-add*)

**lemma** *circ-simulate-2*: $y \ ; \ x^\circ \leq x^\circ \ ; \ y^\circ \leftrightarrow y^\circ \ ; \ x^\circ \leq x^\circ \ ; \ y^\circ$
**by** (*rule iffI*,
*smt add-associative add-right-zero circ-loop-fixpoint circ-simulate-left-plus-1 mult-associative mult-left-zero mult-zero-add-circ-2*,
*metis circ-increasing mult-left-isotone order-trans*)

**lemma** *circ-simulate-absorb*: $y \ ; \ x \leq x \rightarrow y^\circ \ ; \ x \leq x + y^\circ \ ; \ 0$
**by** (*metis circ-simulate-left-plus-1 circ-zero mult-right-one*)

**lemma** *circ-simulate-3*: $y\ ;\ x^\circ \leq x^\circ \rightarrow y^\circ\ ;\ x^\circ \leq x^\circ\ ;\ y^\circ$
**by** (*metis add-least-upper-bound circ-reflexive circ-simulate-2 less-eq-def mult-right-isotone mult-right-one*)

**lemma** *circ-square-2*: $(x\ ;\ x)^\circ\ ;\ (x + 1) \leq x^\circ$
**by** (*metis add-least-upper-bound circ-increasing circ-mult-upper-bound circ-reflexive circ-simulate-right-plus-1 mult-left-one mult-right-isotone mult-right-one*)

**lemma** *circ-separate-mult-1*: $y\ ;\ x \leq x\ ;\ y \rightarrow (x\ ;\ y)^\circ \leq x^\circ\ ;\ y^\circ$
**by** (*metis circ-mult-sub-add circ-separate-1*)

**lemma** *circ-extra-circ*: $(y\ ;\ x^\circ)^\circ = (y\ ;\ y^\circ\ ;\ x^\circ)^\circ$
**proof** −
**have** $y\ ;\ y^\circ\ ;\ x^\circ \leq y\ ;\ x^\circ\ ;\ (y\ ;\ x^\circ)^\circ$
**by** (*metis add-commutative circ-add-1 circ-sub-dist-3 mult-associative mult-right-isotone*)
**hence** $(y\ ;\ y^\circ\ ;\ x^\circ)^\circ \leq (y\ ;\ x^\circ)^\circ$
**by** (*metis circ-simulate-right-plus-1 mult-left-one mult-right-one*)
**thus** *?thesis*
**by** (*metis antisym circ-back-loop-fixpoint circ-isotone mult-right-subdist-add-right*)
**qed**

**lemma** *circ-separate-unfold*: $(y\ ;\ x^\circ)^\circ = y^\circ + y^\circ\ ;\ y\ ;\ x\ ;\ x^\circ\ ;\ (y\ ;\ x^\circ)^\circ$
**by** (*smt add-commutative circ-add circ-left-unfold circ-loop-fixpoint mult-associative mult-left-dist-add mult-right-one*)

**lemma** *separation*: $y\ ;\ x \leq x\ ;\ y^\circ \rightarrow (x + y)^\circ = x^\circ\ ;\ y^\circ$
**proof** −
**have** $y\ ;\ x \leq x\ ;\ y^\circ \rightarrow y^\circ\ ;\ x\ ;\ y^\circ \leq x\ ;\ y^\circ + y^\circ\ ;\ 0$
**by** (*smt circ-simulate-left-plus-1 circ-transitive-equal mult-associative mult-left-isotone mult-left-zero mult-right-dist-add*)
**thus** *?thesis*
**by** (*smt add-commutative circ-add-1 circ-simulate-right circ-sub-dist-3 less-eq-def mult-associative mult-left-zero zero-right-mult-decreasing*)
**qed**

**lemma** *circ-simulate-4*: $y\ ;\ x \leq x\ ;\ x^\circ\ ;\ (1 + y) \rightarrow y^\circ\ ;\ x^\circ \leq x^\circ\ ;\ y^\circ$
**proof** −
**have** $y\ ;\ x \leq x\ ;\ x^\circ\ ;\ (1 + y) \rightarrow (1 + y)\ ;\ x \leq x\ ;\ x^\circ\ ;\ (1 + y)$
**by** (*smt add-associative add-commutative add-left-upper-bound circ-back-loop-fixpoint less-eq-def mult-left-dist-add mult-left-one mult-right-dist-add mult-right-one*)
**thus** *?thesis*
**by** (*smt add-least-upper-bound circ-increasing circ-reflexive circ-simulate-2 circ-simulate-right-plus-1 mult-right-isotone mult-right-subdist-add-right order-trans*)
**qed**

**lemma** *circ-simulate-5*: $y\ ;\ x \leq x\ ;\ x^\circ\ ;\ (x + y) \rightarrow y^\circ\ ;\ x^\circ \leq x^\circ\ ;\ y^\circ$
**by** (*metis circ-add-sub-add-one circ-simulate-4 order-trans*)

**lemma** *circ-simulate-6*: $y\ ;\ x \leq x\ ;\ (x + y) \rightarrow y^\circ\ ;\ x^\circ \leq x^\circ\ ;\ y^\circ$
**by** (*metis add-commutative circ-back-loop-fixpoint circ-simulate-5 mult-right-subdist-add-left order-trans*)

**lemma** *circ-separate-4*: $y\ ;\ x \leq x\ ;\ x^\circ\ ;\ (1 + y) \rightarrow (x + y)^\circ = x^\circ\ ;\ y^\circ$
**proof**

**assume** *1*: $y\ ;\ x \leq x\ ;\ x^\circ\ ;\ (1 + y)$
**hence** $y\ ;\ x\ ;\ x^\circ \leq x\ ;\ x^\circ + x\ ;\ x^\circ\ ;\ y\ ;\ x^\circ$
**by** (*smt circ-transitive-equal less-eq-def mult-associative mult-left-dist-add mult-right-dist-add mult-right-one*)
**also have** $... \leq x\ ;\ x^\circ + x\ ;\ x^\circ\ ;\ x^\circ\ ;\ y^\circ$ **using** *1*
**by** (*metis add-right-isotone circ-simulate-2 circ-simulate-4 mult-associative mult-right-isotone*)
**finally have** $y\ ;\ x\ ;\ x^\circ \leq x\ ;\ x^\circ\ ;\ y^\circ$
**by** (*metis circ-reflexive circ-transitive-equal less-eq-def mult-associative mult-right-isotone mult-right-one*)
**hence** $y^\circ\ ;\ (y^\circ\ ;\ x)^\circ \leq x^\circ\ ;\ (y^\circ + y^\circ\ ;\ 0\ ;\ (y^\circ\ ;\ x)^\circ)$
**by** (*smt add-right-upper-bound circ-back-loop-fixpoint circ-simulate-left-plus-1 circ-simulate-right-plus circ-transitive-equal mult-associative order-trans*)
**thus** $(x + y)^\circ = x^\circ\ ;\ y^\circ$
**by** (*smt add-commutative antisym circ-add-1 circ-slide circ-sub-dist-3 circ-transitive-equal less-eq-def mult-associative mult-left-zero mult-right-subdist-add-right zero-right-mult-decreasing*)
**qed**

**lemma** *circ-separate-5*: $y\ ;\ x \leq x\ ;\ x^\circ\ ;\ (x + y) \rightarrow (x + y)^\circ = x^\circ\ ;\ y^\circ$
**by** (*metis circ-add-sub-add-one circ-separate-4 order-trans*)

**lemma** *circ-separate-6*: $y\ ;\ x \leq x\ ;\ (x + y) \rightarrow (x + y)^\circ = x^\circ\ ;\ y^\circ$
**by** (*metis add-commutative circ-back-loop-fixpoint circ-separate-5 mult-right-subdist-add-left order-trans*)

**end**

**class** *itering-T* = *semiring-T* + *itering-3*

**begin**

**lemma** *circ-top*: $T^\circ = T$
**by** (*metis add-right-top antisym circ-left-unfold mult-left-subdist-add-left mult-right-one top-greatest*)

**lemma** *circ-right-top*: $x^\circ\ ;\ T = T$
**by** (*metis add-right-top circ-loop-fixpoint*)

**lemma** *circ-left-top*: $T\ ;\ x^\circ = T$
**by** (*metis add-right-top circ-add circ-right-top circ-top*)

**lemma** *mult-top-circ*: $(x\ ;\ T)^\circ = 1 + x\ ;\ T$
**by** (*metis circ-left-top circ-mult mult-associative*)

**end**

**class** *itering-L* = *itering-3* +
**fixes** $L :: 'a$
**assumes** *L-def*: $L = 1^\circ\ ;\ 0$

**begin**

**lemma** *one-circ-split*: $1^\circ = L + 1$
**by** (*metis L-def add-commutative antisym circ-add-upper-bound circ-reflexive circ-simulate-absorb mult-right-one order-refl zero-right-mult-decreasing*)

**lemma** *one-circ-mult-split*: $1^\circ \ ; \ x = L + x$
**by** (*metis L-def add-commutative circ-loop-fixpoint mult-associative mult-left-zero mult-zero-circ one-circ-split*)

**lemma** *one-circ-circ-split*: $1^{\circ\circ} = L + 1$
**by** (*metis circ-one one-circ-split*)

**lemma** *one-circ-mult-split-2*: $1^\circ \ ; \ x = x \ ; \ 1^\circ + L$
**by** (*smt L-def add-associative add-commutative circ-left-unfold less-eq-def mult-left-subdist-add-left mult-right-one one-circ-mult-split sub-mult-one-circ*)

**lemma** *sub-mult-one-circ-split*: $x \ ; \ 1^\circ \leq x + L$
**by** (*metis add-commutative one-circ-mult-split sub-mult-one-circ*)

**lemma** *sub-mult-one-circ-split-2*: $x \ ; \ 1^\circ \leq x + 1^\circ$
**by** (*metis L-def add-right-isotone order-trans sub-mult-one-circ-split zero-right-mult-decreasing*)

**lemma** *L-split*: $x \ ; \ L \leq x \ ; \ 0 + L$
**by** (*smt L-def mult-associative mult-left-isotone mult-right-dist-add sub-mult-one-circ-split-2*)

**lemma** *L-left-zero*: $L \ ; \ x = L$
**by** (*metis L-def mult-associative mult-left-zero*)

**lemma** *one-circ-L*: $1^\circ \ ; \ L = L$
**by** (*metis add-idempotent one-circ-mult-split*)

**lemma** *circ-circ-split*: $x^{\circ\circ} = L + x^\circ$
**by** (*metis circ-circ-mult one-circ-mult-split*)

**lemma** *circ-left-induct-mult-L*: $L \leq x \rightarrow x \ ; \ y \leq x \rightarrow x \ ; \ y^\circ \leq x$
**by** (*metis circ-one circ-simulate less-eq-def one-circ-mult-split*)

**lemma** *circ-left-induct-mult-iff-L*: $L \leq x \rightarrow x \ ; \ y \leq x \leftrightarrow x \ ; \ y^\circ \leq x$
**by** (*smt add-least-upper-bound circ-back-loop-fixpoint circ-left-induct-mult-L less-eq-def*)

**lemma** *circ-left-induct-L*: $L \leq x \rightarrow x \ ; \ y + z \leq x \rightarrow z \ ; \ y^\circ \leq x$
**by** (*metis add-least-upper-bound circ-left-induct-mult-L less-eq-def mult-right-dist-add*)

**lemma** *mult-L-circ*: $(x \ ; \ L)^\circ = 1 + x \ ; \ L$
**by** (*metis L-left-zero circ-mult mult-associative*)

**lemma** *mult-L-circ-mult*: $(x \ ; \ L)^\circ \ ; \ y = y + x \ ; \ L$
**by** (*metis L-left-zero mult-L-circ mult-associative mult-left-one mult-right-dist-add*)

**lemma** *circ-L*: $L^\circ = L + 1$
**by** (*metis L-left-zero add-commutative circ-left-unfold*)

**lemma** *L-below-one-circ*: $L \leq 1^\circ$
**by** (*metis add-left-upper-bound one-circ-split*)

**lemma** *circ-add-6*: $L + (x + y)^\circ = (x^\circ \ ; \ y^\circ)^\circ$
**by** (*metis add-associative add-commutative circ-add-1 circ-circ-add circ-circ-split circ-decompose-4*)

**end**

**class** *kleene-algebra* = *semiring* +
**fixes** *star* :: $'a \Rightarrow 'a$ (-* [*100*] *100*)
**assumes** *star-left-unfold* : $1 + y \ ; \ y^* \leq y^*$
**assumes** *star-left-induct* : $z + y \ ; \ x \leq x \rightarrow y^* \ ; \ z \leq x$
**assumes** *star-right-induct*: $z + x \ ; \ y \leq x \rightarrow z \ ; \ y^* \leq x$

**begin**

**lemma** *star-left-unfold-equal*: $1 + x \ ; \ x^* = x^*$
**by** (*smt add-least-upper-bound antisym-conv mult-right-isotone mult-right-one order-refl order-trans star-left-induct star-left-unfold*)

**lemma** *star-unfold-slide*: $(x \ ; \ y)^* = 1 + x \ ; \ (y \ ; \ x)^* \ ; \ y$
**by** (*smt antisym mult-associative mult-left-dist-add mult-left-one mult-right-dist-add mult-right-isotone mult-right-one order-refl star-left-induct star-left-unfold-equal*)

**lemma** *star-decompose*: $(x + y)^* = (x^* \ ; \ y)^* \ ; \ x^*$
**apply** (*rule antisym*)
**apply** (*smt add-least-upper-bound add-left-upper-bound add-right-upper-bound mult-associative mult-left-dist-add mult-left-one mult-right-dist-add mult-right-one star-left-induct star-left-unfold-equal*)
**by** (*smt add-least-upper-bound less-eq-def mult-left-subdist-add-left mult-right-one star-left-induct star-left-unfold star-right-induct*)

**lemma** *star-simulation-right*: $z \ ; \ x \leq y \ ; \ z \rightarrow z \ ; \ x^* \leq y^* \ ; \ z$
**by** (*smt add-least-upper-bound less-eq-def mult-associative mult-left-dist-add mult-left-one mult-right-subdist-add-left star-left-induct star-left-unfold star-right-induct*)

**end**

**sublocale** *kleene-algebra* < *star*!: *itering-1* **where** *circ* = *star*
**by** (*unfold-locales*, *metis star-unfold-slide*, *metis star-decompose*, *metis star-simulation-right*)

**context** *kleene-algebra*

**begin**

Most lemmas in this class are taken from Georg Struth's theories.

**lemma** *star-sub-one*: $x \leq 1 \rightarrow x^* = 1$
**by** (*metis add-least-upper-bound eq-iff mult-left-one star.circ-reflexive star-right-induct*)

**lemma** *star-one*: $1^* = 1$

**by** (*metis eq-iff star-sub-one*)

**lemma** *star-left-induct-mult*: $x\ ;\ y \leq y \rightarrow x^*\ ;\ y \leq y$
**by** (*metis add-commutative less-eq-def order-refl star-left-induct*)

**lemma** *star-left-induct-mult-iff*: $x\ ;\ y \leq y \leftrightarrow x^*\ ;\ y \leq y$
**by** (*smt mult-associative mult-left-isotone mult-left-one mult-right-isotone order-trans star-left-induct-mult star.circ-reflexive star.left-plus-below-circ*)

**lemma** *star-involutive*: $x^* = x^{**}$
**by** (*smt antisym less-eq-def mult-left-subdist-add-left mult-right-one star-left-induct star.circ-plus-one star.left-plus-below-circ star.circ-transitive-equal*)

**lemma** *star-sup-one*: $(1 + x)^* = x^*$
**by** (*smt add-commutative less-eq-def mult-left-dist-add mult-left-one mult-right-dist-add mult-right-one star-involutive star.circ-plus-one star.circ-sub-dist star.left-plus-below-circ*)

**lemma** *star-left-induct-equal*: $z + x\ ;\ y = y \rightarrow x^*\ ;\ z \leq y$
**by** (*metis order-refl star-left-induct*)

**lemma** *star-left-induct-mult-equal*: $x\ ;\ y = y \rightarrow x^*\ ;\ y \leq y$
**by** (*metis order-refl star-left-induct-mult*)

**lemma** *star-star-upper-bound*: $x^* \leq z^* \rightarrow x^{**} \leq z^*$
**by** (*metis star-involutive*)

**lemma** *star-simulation-left*: $x\ ;\ z \leq z\ ;\ y \rightarrow x^*\ ;\ z \leq z\ ;\ y^*$
**by** (*smt add-commutative add-least-upper-bound mult-right-dist-add less-eq-def mult-associative mult-right-one star.left-plus-below-circ star.circ-increasing star-left-induct star-involutive star.circ-isotone star.circ-reflexive mult-left-subdist-add-left*)

**lemma** *quasicomm-1*: $y\ ;\ x \leq x\ ;\ (x + y)^* \leftrightarrow y^*\ ;\ x \leq x\ ;\ (x + y)^*$
**by** (*smt mult-isotone order-refl order-trans star.circ-increasing star-involutive star-simulation-left*)

**lemma** *star-rtc-3*: $1 + x + y\ ;\ y = y \rightarrow x^* \leq y$
**by** (*metis add-least-upper-bound less-eq-def mult-left-subdist-add-left mult-right-one star-left-induct-mult-iff star.circ-sub-dist*)

**lemma** *star-decompose-1*: $(x + y)^* = (x^*\ ;\ y^*)^*$
**by** (*smt add-least-upper-bound antisym mult-isotone mult-left-one mult-right-one star.circ-increasing star-involutive star.circ-isotone star.circ-reflexive star.circ-sub-dist-3*)

**lemma** *star-sum*: $(x + y)^* = (x^* + y^*)^*$
**by** (*metis star-decompose-1 star-involutive*)

**lemma** *star-decompose-3*: $(x^*\ ;\ y^*)^* = x^*\ ;\ (y\ ;\ x^*)^*$
**by** (*metis star-decompose-1 star.circ-add-1*)

**lemma** *star-right-induct-mult*: $y\ ;\ x \leq y \rightarrow y\ ;\ x^* \leq y$
**by** (*metis add-commutative less-eq-def order-refl star-right-induct*)

**lemma** *star-right-induct-mult-iff*: $y\ ;\ x \leq y \leftrightarrow y\ ;\ x^* \leq y$
**by** (*metis mult-right-isotone order-trans star.circ-increasing star-right-induct-mult*)

**lemma** *star-simulation-right-equal*: $z \ ; \ x = y \ ; \ z \rightarrow z \ ; \ x^* = y^* \ ; \ z$
**by** (*metis eq-iff star-simulation-left star-simulation-right*)

**lemma** *star-simulation-star*: $x \ ; \ y \leq y \ ; \ x \rightarrow x^* \ ; \ y^* \leq y^* \ ; \ x^*$
**by** (*metis star-simulation-left star-simulation-right*)

**lemma** *star-right-induct-equal*: $z + y \ ; \ x = y \rightarrow z \ ; \ x^* \leq y$
**by** (*metis order-refl star-right-induct*)

**lemma** *star-right-induct-mult-equal*: $y \ ; \ x = y \rightarrow y \ ; \ x^* \leq y$
**by** (*metis order-refl star-right-induct-mult*)

**lemma** *star-loop-least-fixpoint*: $y \ ; \ x + z = x \rightarrow y^* \ ; \ z \leq x$
**by** (*metis add-commutative star-left-induct-equal*)

**lemma** *star-back-loop-least-fixpoint*: $x \ ; \ y + z = x \rightarrow z \ ; \ y^* \leq x$
**by** (*metis add-commutative star-right-induct-equal*)

**lemma** *star-loop-is-least-fixpoint*: *is-least-fixpoint* $(\lambda x \ . \ y \ ; \ x + z) \ (y^* \ ; \ z)$
**by** (*smt is-least-fixpoint-def star.circ-loop-fixpoint star-loop-least-fixpoint*)

**lemma** *star-loop-mu*: $\mu \ (\lambda x \ . \ y \ ; \ x + z) = y^* \ ; \ z$
**by** (*metis least-fixpoint-same star-loop-is-least-fixpoint*)

**lemma** *star-back-loop-is-least-fixpoint*: *is-least-fixpoint* $(\lambda x \ . \ x \ ; \ y + z) \ (z \ ; \ y^*)$
**by** (*smt is-least-fixpoint-def star.circ-back-loop-fixpoint star-back-loop-least-fixpoint*)

**lemma** *star-back-loop-mu*: $\mu \ (\lambda x \ . \ x \ ; \ y + z) = z \ ; \ y^*$
**by** (*metis least-fixpoint-same star-back-loop-is-least-fixpoint*)

**lemma** *star-square*: $x^* = (1 + x) \ ; \ (x \ ; \ x)^*$
**proof** −
**let** *?f* $= \lambda y \ . \ y \ ; \ x + 1$
**have** *1*: *isotone ?f*
**by** (*smt add-left-isotone isotone-def mult-left-isotone*)
**have** *2*: $?f \circ ?f = (\lambda y \ . \ y \ ; \ (x \ ; \ x) + (1 + x))$
**by** (*simp add*: *add-associative add-commutative mult-associative mult-left-one mult-right-dist-add o-def*)
**thus** *?thesis* **using** *1*
**by** (*metis mu-square mult-left-one star-back-loop-mu has-least-fixpoint-def star-back-loop-is-least-fixpoint*)
**qed**

**lemma** *star-square-2*: $x^* = (x \ ; \ x)^* \ ; \ (x + 1)$
**by** (*smt add-commutative mult-left-dist-add mult-right-dist-add mult-right-one star.circ-decompose-5 star-involutive star-one star.circ-slide star-square*)

**lemma** *star-circ-simulate-right-plus*: $z \ ; \ x \leq y \ ; \ y^* \ ; \ z + w \rightarrow z \ ; \ x^* \leq y^* \ ; \ (z + w \ ; \ x^*)$
**proof**

**assume** *1*: $z\ ;\ x \leq y\ ;\ y^*\ ;\ z + w$
**have** $y^*\ ;\ (z + w\ ;\ x^*)\ ;\ x \leq y^*\ ;\ z\ ;\ x + y^*\ ;\ w\ ;\ x^*$
**by** (*smt add-left-upper-bound add-right-isotone mult-associative mult-left-dist-add mult-right-dist-add star.circ-back-loop-fixpoint*)
**also have** $\ldots \leq y^*\ ;\ y\ ;\ z + y^*\ ;\ w + y^*\ ;\ w\ ;\ x^*$ **using** *1*
**by** (*smt add-left-isotone less-eq-def mult-associative mult-left-dist-add star.circ-plus-same star.circ-transitive-equal*)
**also have** $\ldots \leq y^*\ ;\ (z + w\ ;\ x^*)$
**by** (*smt add-associative add-idempotent add-left-isotone mult-associative mult-left-dist-add mult-right-one mult-right-subdist-add-right star.circ-plus-one star.circ-plus-same star.circ-transitive-equal star.circ-unfold-sum*)
**finally show** $z\ ;\ x^* \leq y^*\ ;\ (z + w\ ;\ x^*)$
**by** (*smt add-least-upper-bound add-right-divisibility star.circ-loop-fixpoint star-right-induct*)
**qed**

**lemma** *star-circ-simulate-left-plus*: $x\ ;\ z \leq z\ ;\ y^* + w \rightarrow x^*\ ;\ z \leq (z + x^*\ ;\ w)\ ;\ y^*$
**proof**
**assume** *1*: $x\ ;\ z \leq z\ ;\ y^* + w$
**have** $x\ ;\ ((z + x^*\ ;\ w)\ ;\ y^*) \leq x\ ;\ z\ ;\ y^* + x^*\ ;\ w\ ;\ y^*$
**by** (*smt add-right-isotone mult-associative mult-left-dist-add mult-right-dist-add mult-right-subdist-add-left star.circ-loop-fixpoint*)
**also have** $\ldots \leq (z + w + x^*\ ;\ w)\ ;\ y^*$ **using** *1*
**by** (*smt add-left-divisibility add-left-isotone mult-associative mult-right-dist-add star.circ-transitive-equal*)
**also have** $\ldots = (z + x^*\ ;\ w)\ ;\ y^*$
**by** (*metis add-associative add-right-upper-bound less-eq-def star.circ-loop-fixpoint*)
**finally show** $x^*\ ;\ z \leq (z + x^*\ ;\ w)\ ;\ y^*$
**by** (*metis add-least-upper-bound mult-left-subdist-add-left mult-right-one star.circ-right-unfold star-left-induct*)
**qed**

**end**

**sublocale** *kleene-algebra* $<$ *star*!: *itering-3* **where** *circ* $=$ *star*
**apply** *unfold-locales*
**apply** (*smt add-associative add-commutative add-left-upper-bound mult-associative mult-left-dist-add order-trans star.circ-loop-fixpoint star-circ-simulate-right-plus*)
**apply** (*smt add-commutative add-right-isotone mult-right-isotone order-trans star.circ-increasing star-circ-simulate-left-plus*)
**apply** (*rule star-circ-simulate-right-plus*)
**by** (*rule star-circ-simulate-left-plus*)

**class** *kleene-itering* $=$ *kleene-algebra* $+$ *itering-3*

**begin**

**lemma** *star-below-circ*: $x^* \leq x^\circ$
**by** (*metis circ-right-unfold mult-left-one order-refl star-right-induct*)

**lemma** *star-zero-below-circ-mult*: $x^*\ ;\ 0 \leq x^\circ\ ;\ y$
**by** (*metis mult-isotone star-below-circ zero-least*)

**lemma** *star-mult-circ*: $x^*\ ;\ x^\circ = x^\circ$
**by** (*metis add-right-divisibility antisym circ-left-unfold star-left-induct-mult star.circ-loop-fixpoint*)

**lemma** *circ-mult-star*: $x^\circ \ ; \ x^* = x^\circ$
**by** (*metis circ-slide mult-left-one mult-right-one star-mult-circ star-simulation-right-equal*)

**lemma** *circ-star*: $x^{\circ *} = x^\circ$
**by** (*metis circ-reflexive circ-transitive-equal less-eq-def mult-left-one star-one star-simulation-right-equal star-square*)

**lemma** *star-circ*: $x^{* \circ} = x^{\circ \circ}$
**by** (*metis antisym circ-circ-add circ-sub-dist less-eq-def star.circ-rtc-2 star-below-circ*)

**lemma** *circ-add-3*: $(x^\circ \ ; \ y^\circ)^* \leq (x + y)^\circ$
**by** (*metis circ-add-1 circ-isotone circ-left-unfold circ-star mult-left-subdist-add-left mult-right-isotone mult-right-one star.circ-isotone*)

**lemma** *circ-isolate*: $x^\circ = x^\circ \ ; \ 0 + x^*$
**by** (*metis add-commutative antisym circ-add-upper-bound circ-mult-star circ-simulate-absorb star.left-plus-below-circ star-below-circ zero-right-mult-decreasing*)

**lemma** *circ-isolate-mult*: $x^\circ \ ; \ y = x^\circ \ ; \ 0 + x^* \ ; \ y$
**by** (*metis circ-isolate mult-associative mult-left-zero mult-right-dist-add*)

**lemma** *circ-isolate-mult-sub*: $x^\circ \ ; \ y \leq x^\circ + x^* \ ; \ y$
**by** (*metis add-left-isotone circ-isolate-mult zero-right-mult-decreasing*)

**lemma** *circ-sub-decompose*: $(x^\circ \ ; \ y)^\circ \leq (x^* \ ; \ y)^\circ \ ; \ x^\circ$
**by** (*smt add-commutative add-least-upper-bound add-right-upper-bound circ-back-loop-fixpoint circ-isolate-mult mult-zero-add-circ-2 zero-right-mult-decreasing*)

**lemma** *circ-add-4*: $(x + y)^\circ = (x^* \ ; \ y)^\circ \ ; \ x^\circ$
**by** (*smt antisym circ-add circ-isotone circ-sub-decompose circ-transitive-equal mult-associative mult-left-isotone star-below-circ*)

**lemma** *circ-add-5*: $(x^\circ \ ; \ y)^\circ \ ; \ x^\circ = (x^* \ ; \ y)^\circ \ ; \ x^\circ$
**by** (*metis circ-add circ-add-4*)

**lemma** *plus-circ*: $(x^* \ ; \ x)^\circ = x^\circ$
**by** (*smt add-idempotent circ-add-4 circ-decompose-7 circ-star star.circ-decompose-5 star.right-plus-circ*)

**end**

**class** *kleene-algebra-T* = *semiring-T* + *kleene-algebra*

**sublocale** *kleene-algebra-T* < *star*!: *itering-T* **where** *circ* = *star* **..**

**class** *omega-algebra-T* = *kleene-algebra-T* +
**fixes** *omega* :: $'a \Rightarrow 'a$ ($-^\omega$ [100] 100)
**assumes** *omega-unfold*: $y^\omega = y \ ; \ y^\omega$
**assumes** *omega-induct*: $x \leq z + y \ ; \ x \rightarrow x \leq y^\omega + y^* \ ; \ z$

**begin**

Most lemmas in this class are taken from Georg Struth's theories.

**lemma** *star-zero-below-omega*: $x^* \ ; \ 0 \leq x^\omega$
**by** (*metis add-left-zero omega-unfold star-left-induct-equal*)

**lemma** *star-zero-below-omega-zero*: $x^* \ ; \ 0 \leq x^\omega \ ; \ 0$
**by** (*metis add-left-zero mult-associative omega-unfold star-left-induct-equal*)

**lemma** *omega-induct-mult*: $y \leq x \ ; \ y \rightarrow y \leq x^\omega$
**by** (*metis add-commutative add-left-zero less-eq-def omega-induct star-zero-below-omega*)

**lemma** *omega-sub-dist*: $x^\omega \leq (x{+}y)^\omega$
**by** (*metis mult-right-subdist-add-left omega-induct-mult omega-unfold*)

**lemma** *omega-isotone*: $x \leq y \rightarrow x^\omega \leq y^\omega$
**by** (*metis less-eq-def omega-sub-dist*)

**lemma** *omega-induct-equal*: $y = z + x \ ; \ y \rightarrow y \leq x^\omega + x^* \ ; \ z$
**by** (*metis omega-induct order-refl*)

**lemma** *omega-zero*: $0^\omega = 0$
**by** (*metis mult-left-zero omega-unfold*)

**lemma** *omega-one-greatest*: $x \leq 1^\omega$
**by** (*metis mult-left-one omega-induct-mult order-refl*)

**lemma** *omega-one*: $1^\omega = T$
**by** (*metis add-left-top less-eq-def omega-one-greatest*)

**lemma** *star-mult-omega*: $x^* \ ; \ x^\omega \ = x^\omega$
**by** (*metis antisym-conv mult-isotone omega-unfold star.circ-increasing star-left-induct-mult-equal star-left-induct-mult-iff*)

**lemma** *star-omega-top*: $x^{*\omega} = T$
**by** (*metis add-left-top less-eq-def omega-one omega-sub-dist star.circ-plus-one*)

**lemma** *omega-sub-vector*: $x^\omega \ ; \ y \leq x^\omega$
**by** (*metis mult-associative omega-induct-mult omega-unfold order-refl*)

**lemma** *omega-vector*: $x^\omega \ ; \ T = x^\omega$
**by** (*metis add-commutative less-eq-def omega-sub-vector top-right-mult-increasing*)

**lemma** *omega-simulation*: $z \ ; \ x \leq y \ ; \ z \rightarrow z \ ; \ x^\omega \leq y^\omega$
**by** (*smt less-eq-def mult-associative mult-right-subdist-add-left omega-induct-mult omega-unfold*)

**lemma** *omega-omega*: $x^{\omega\omega} \leq x^\omega$
**by** (*metis omega-sub-vector omega-unfold*)

**lemma** *left-plus-omega*: $(x\ ;\ x^*)^\omega = x^\omega$
**by** (*metis antisym mult-associative mult-right-isotone mult-right-one omega-induct-mult omega-unfold star-mult-omega star-one star.circ-reflexive star.circ-right-unfold star.circ-plus-same star-sum star-sup-one*)

**lemma** *omega-slide*: $x\ ;\ (y\ ;\ x)^\omega = (x\ ;\ y)^\omega$
**by** (*metis antisym mult-associative mult-right-isotone omega-simulation omega-unfold order-refl*)

**lemma** *omega-simulation-2*: $y\ ;\ x \leq x\ ;\ y \longrightarrow (x\ ;\ y)^\omega \leq x^\omega$
**by** (*metis less-eq-def mult-right-isotone omega-induct-mult omega-slide omega-sub-dist*)

**lemma** *wagner*: $(x + y)^\omega = x\ ;\ (x + y)^\omega + z \longrightarrow (x + y)^\omega = x^\omega + x^*\ ;\ z$
**by** (*metis add-commutative add-least-upper-bound eq-iff omega-induct omega-sub-dist star-left-induct*)

**lemma** *right-plus-omega*: $(x^*\ ;\ x)^\omega = x^\omega$
**by** (*metis left-plus-omega star.circ-plus-same*)

**lemma** *omega-sub-dist-1*: $(x\ ;\ y^*)^\omega \leq (x + y)^\omega$
**by** (*metis add-least-upper-bound left-plus-omega mult-isotone mult-left-one mult-right-dist-add omega-isotone order-refl star-decompose-1 star.circ-increasing star.circ-plus-one*)

**lemma** *omega-sub-dist-2*: $(x^*\ ;\ y)^\omega \leq (x + y)^\omega$
**by** (*metis add-commutative mult-isotone omega-slide omega-sub-dist-1 star-mult-omega star.circ-sub-dist*)

**lemma** *omega-star*: $(x^\omega)^* = 1 + x^\omega$
**by** (*metis mult-associative omega-unfold omega-vector star.circ-plus-same star-left-unfold-equal star-omega-top*)

**lemma** *omega-mult-omega-star*: $x^\omega\ ;\ x^{\omega *} = x^\omega$
**by** (*metis mult-associative omega-unfold omega-vector star.circ-plus-same star-omega-top*)

**lemma** *omega-sum-unfold-1*: $(x + y)^\omega = x^\omega + x^*\ ;\ y\ ;\ (x + y)^\omega$
**by** (*metis mult-associative mult-right-dist-add omega-unfold wagner*)

**lemma** *omega-sum-unfold-2*: $(x + y)^\omega \leq (x^*\ ;\ y)^\omega + (x^*\ ;\ y)^*\ ;\ x^\omega$
**by** (*metis omega-induct-equal omega-sum-unfold-1*)

**lemma** *omega-sum-unfold-3*: $(x^*\ ;\ y)^*\ ;\ x^\omega \leq (x + y)^\omega$
**by** (*metis omega-sum-unfold-1 star-left-induct-equal*)

**lemma** *omega-decompose*: $(x + y)^\omega = (x^*\ ;\ y)^\omega + (x^*\ ;\ y)^*\ ;\ x^\omega$
**by** (*metis add-least-upper-bound antisym omega-sub-dist-2 omega-sum-unfold-2 omega-sum-unfold-3*)

**lemma** *omega-loop-fixpoint*: $y\ ;\ (y^\omega + y^*\ ;\ z) + z = y^\omega + y^*\ ;\ z$
**by** (*metis add-associative mult-left-dist-add omega-unfold star.circ-loop-fixpoint*)

**lemma** *omega-loop-greatest-fixpoint*: $y\ ;\ x + z = x \longrightarrow x \leq y^\omega + y^*\ ;\ z$
**by** (*metis add-commutative omega-induct-equal*)

**lemma** *omega-square*: $x^\omega = (x\ ;\ x)^\omega$

**by** (*smt antisym mult-associative omega-induct-mult omega-slide omega-sub-vector omega-unfold omega-vector top-mult-top*)

**lemma** *mult-top-omega*: $(x\ ;\ T)^\omega \leq x\ ;\ T$
**by** (*metis mult-right-isotone omega-slide top-greatest*)

**lemma** *mult-zero-omega*: $(x\ ;\ 0)^\omega = x\ ;\ 0$
**by** (*metis mult-left-zero omega-slide*)

**lemma** *mult-zero-add-omega*: $(x + y\ ;\ 0)^\omega = x^\omega + x^*\ ;\ y\ ;\ 0$
**by** (*smt add-associative add-commutative add-idempotent mult-associative mult-left-one mult-left-zero mult-right-dist-add mult-zero-omega star.mult-zero-circ omega-decompose*)

**lemma** *omega-mult-star*: $x^\omega\ ;\ x^* = x^\omega$
**by** (*metis mult-associative omega-vector star.circ-left-top*)

**lemma** *omega-loop-is-greatest-fixpoint*: *is-greatest-fixpoint* $(\lambda x\ .\ y\ ;\ x + z)\ (y^\omega + y^*\ ;\ z)$
**by** (*smt is-greatest-fixpoint-def omega-loop-fixpoint omega-loop-greatest-fixpoint*)

**lemma** *omega-loop-nu*: $\nu\ (\lambda x\ .\ y\ ;\ x + z) = y^\omega + y^*\ ;\ z$
**by** (*metis greatest-fixpoint-same omega-loop-is-greatest-fixpoint*)

**lemma** *omega-loop-zero-is-greatest-fixpoint*: *is-greatest-fixpoint* $(\lambda x\ .\ y\ ;\ x)\ (y^\omega)$
**by** (*metis is-greatest-fixpoint-def omega-induct-mult omega-unfold order-refl*)

**lemma** *omega-loop-zero-nu*: $\nu\ (\lambda x\ .\ y\ ;\ x) = y^\omega$
**by** (*metis greatest-fixpoint-same omega-loop-zero-is-greatest-fixpoint*)

**lemma** *omega-separate-unfold*: $(x^*\ ;\ y)^\omega = y^\omega + y^*\ ;\ x\ ;\ (x^*\ ;\ y)^\omega$
**by** (*metis add-commutative mult-associative omega-slide omega-sum-unfold-1 star.circ-loop-fixpoint*)

**lemma** *omega-circ-simulate-right-plus*: $z\ ;\ x \leq y\ ;\ (y^\omega\ ;\ 0 + y^*)\ ;\ z + w \rightarrow z\ ;\ (x^\omega\ ;\ 0 + x^*) \leq (y^\omega\ ;\ 0 + y^*)\ ;\ (z + w\ ;\ (x^\omega\ ;\ 0 + x^*))$
**proof**
**assume** $z\ ;\ x \leq y\ ;\ (y^\omega\ ;\ 0 + y^*)\ ;\ z + w$
**hence** *1*: $z\ ;\ x \leq y^\omega\ ;\ 0 + y\ ;\ y^*\ ;\ z + w$
**by** (*metis mult-associative mult-left-dist-add mult-left-zero mult-right-dist-add omega-unfold*)
**hence** $(y^\omega\ ;\ 0 + y^*\ ;\ z + y^*\ ;\ w\ ;\ x^\omega\ ;\ 0 + y^*\ ;\ w\ ;\ x^*)\ ;\ x \leq y^\omega\ ;\ 0 + y^*\ ;\ (y^\omega\ ;\ 0 + y\ ;\ y^*\ ;\ z + w) + y^*\ ;\ w\ ;\ x^\omega\ ;\ 0 + y^*\ ;\ w\ ;\ x^*$
**by** (*smt add-associative add-left-upper-bound add-right-upper-bound less-eq-def mult-associative mult-left-dist-add mult-left-zero mult-right-dist-add star.circ-back-loop-fixpoint*)
**also have** $\ldots = y^\omega\ ;\ 0 + y^*\ ;\ y\ ;\ y^*\ ;\ z + y^*\ ;\ w\ ;\ x^\omega\ ;\ 0 + y^*\ ;\ w\ ;\ x^*$
**by** (*smt add-associative add-right-upper-bound less-eq-def mult-associative mult-left-dist-add star.circ-back-loop-fixpoint star-mult-omega*)
**finally have** $z + (y^\omega\ ;\ 0 + y^*\ ;\ z + y^*\ ;\ w\ ;\ x^\omega\ ;\ 0 + y^*\ ;\ w\ ;\ x^*)\ ;\ x \leq y^\omega\ ;\ 0 + y^*\ ;\ z + y^*\ ;\ w\ ;\ x^\omega\ ;\ 0 + y^*\ ;\ w\ ;\ x^*$
**by** (*smt add-least-upper-bound add-left-isotone add-left-upper-bound mult-left-isotone order-trans star.circ-loop-fixpoint star.circ-plus-same star.circ-transitive-equal star.left-plus-below-circ*)
**hence** *2*: $z\ ;\ x^* \leq y^\omega\ ;\ 0 + y^*\ ;\ z + y^*\ ;\ w\ ;\ x^\omega\ ;\ 0 + y^*\ ;\ w\ ;\ x^*$
**by** (*metis star-right-induct*)
**have** $z\ ;\ x^\omega\ ;\ 0 \leq (y^\omega\ ;\ 0 + y\ ;\ y^*\ ;\ z + w)\ ;\ x^\omega\ ;\ 0$ **using** *1*
**by** (*smt add-left-divisibility mult-associative mult-right-subdist-add-left omega-unfold*)
**hence** $z\ ;\ x^\omega\ ;\ 0 \leq y^\omega + y^*\ ;\ (y^\omega\ ;\ 0 + w\ ;\ x^\omega\ ;\ 0)$
**by** (*smt add-associative add-commutative left-plus-omega mult-associative mult-left-zero mult-right-dist-add omega-induct star.left-plus-circ*)

**thus** $z \ ; \ (x^\omega \ ; \ 0 + x^*) \leq (y^\omega \ ; \ 0 + y^*) \ ; \ (z + w \ ; \ (x^\omega \ ; \ 0 + x^*))$ **using** *2*
**by** (*smt add-associative add-commutative less-eq-def mult-associative mult-left-dist-add mult-left-zero mult-right-dist-add omega-unfold omega-zero star-mult-omega zero-right-mult-decreasing*)
**qed**

**lemma** *omega-circ-simulate-left-plus*: $x \ ; \ z \leq z \ ; \ (y^\omega \ ; \ 0 + y^*) + w \rightarrow (x^\omega \ ; \ 0 + x^*) \ ; \ z \leq (z + (x^\omega \ ; \ 0 + x^*) \ ; \ w) \ ; \ (y^\omega \ ; \ 0 + y^*)$
**proof**
**assume** *1*: $x \ ; \ z \leq z \ ; \ (y^\omega \ ; \ 0 + y^*) + w$
**have** $x \ ; \ (z \ ; \ y^\omega \ ; \ 0 + z \ ; \ y^* + x^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^*) \leq x \ ; \ z \ ; \ y^\omega \ ; \ 0 + x \ ; \ z \ ; \ y^* + x^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^*$
**by** (*smt add-associative add-right-upper-bound less-eq-def mult-associative mult-left-dist-add omega-unfold star.circ-loop-fixpoint*)
**also have** $... \leq (z \ ; \ y^\omega \ ; \ 0 + z \ ; \ y^* + w) \ ; \ y^\omega \ ; \ 0 + (z \ ; \ y^\omega \ ; \ 0 + z \ ; \ y^* + w) \ ; \ y^* + x^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^*$ **using** *1*
**by** (*metis add-left-isotone mult-associative mult-left-dist-add mult-left-isotone*)
**also have** $... = z \ ; \ y^\omega \ ; \ 0 + z \ ; \ y^* + x^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^*$
**by** (*smt add-associative add-commutative add-idempotent mult-associative mult-left-zero mult-right-dist-add star.circ-loop-fixpoint star.circ-transitive-equal star-mult-omega*)
**finally have** $(x^\omega \ ; \ 0 + x^*) \ ; \ z \leq z \ ; \ y^\omega \ ; \ 0 + z \ ; \ y^* + x^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^\omega \ ; \ 0 + x^* \ ; \ w \ ; \ y^*$
**by** (*smt add-least-upper-bound add-left-upper-bound mult-associative mult-left-zero mult-right-dist-add star.circ-back-loop-fixpoint star-left-induct*)
**thus** $(x^\omega \ ; \ 0 + x^*) \ ; \ z \leq (z + (x^\omega \ ; \ 0 + x^*) \ ; \ w) \ ; \ (y^\omega \ ; \ 0 + y^*)$
**by** (*smt add-associative mult-associative mult-left-dist-add mult-left-zero mult-right-dist-add*)
**qed**

**end**

**sublocale** *omega-algebra-T* $<$ *comb0*!: *itering-T* **where** *circ* $= (\lambda x \ . \ x^\omega \ ; \ 0 + x^*)$
**apply** *unfold-locales*
**apply** (*smt add-associative add-commutative mult-associative mult-left-dist-add mult-left-zero mult-right-dist-add omega-slide star.circ-mult*)
**apply** (*smt add-associative add-commutative star.circ-add mult-associative mult-left-dist-add mult-left-zero mult-right-dist-add star.mult-zero-add-circ-2 mult-zero-add-omega omega-decompose*)
**apply** (*smt add-isotone star.circ-simulate mult-associative mult-left-dist-add mult-left-isotone mult-left-zero mult-right-dist-add omega-simulation*)
**apply** (*smt add-commutative add-left-isotone mult-left-isotone mult-left-subdist-add-left mult-right-one omega-circ-simulate-right-plus order-trans star.circ-plus-one*)
**apply** (*smt add-commutative add-right-isotone mult-left-subdist-add-left mult-right-isotone omega-circ-simulate-left-plus order-trans star.circ-increasing*)
**apply** (*metis omega-circ-simulate-right-plus*)
**by** (*metis omega-circ-simulate-left-plus*)

**class** *omega-itering* $=$ *omega-algebra-T* $+$ *itering-T*

**begin**

**subclass** *kleene-itering* **..**

**lemma** *circ-below-omega-star*: $x^\circ \leq x^\omega + x^*$
**by** (*metis circ-left-unfold mult-right-one omega-induct order-refl*)

**lemma** *omega-mult-circ*: $x^\omega \ ; \ x^\circ = x^\omega$
**by** (*smt add-commutative circ-reflexive less-eq-def mult-left-dist-add mult-right-one omega-sub-vector*)

**lemma** *circ-mult-omega*: $x^\circ \ ; \ x^\omega = x^\omega$
**by** (*metis antisym circ-left-unfold circ-slide le-less mult-left-one mult-right-one mult-right-subdist-add-left omega-simulation*)

**lemma** *circ-omega*: $x^{\circ\omega} = T$

**by** (*metis circ-star star-omega-top*)

**lemma** *omega-circ*: $x^{\omega\circ} = 1 + x^{\omega}$
**by** (*metis circ-left-unfold circ-star mult-associative omega-vector star.circ-left-top*)

**end**

**class** *Omega* =
**fixes** *Omega* :: $'a \Rightarrow 'a$ ($-^{\Omega}$ [*100*] *100*)

**class** *dra* = *kleene-algebra-T* + *Omega* +
**assumes** *Omega-unfold* : $y^{\Omega} = 1 + y \ ; \ y^{\Omega}$
**assumes** *Omega-isolate* : $y^{\Omega} = y^{\Omega} \ ; \ 0 + y^{*}$
**assumes** *Omega-induct* : $x \leq z + y \ ; \ x \rightarrow x \leq y^{\Omega} \ ; \ z$

**begin**

**lemma** *Omega-mult*: $(x \ ; \ y)^{\Omega} = 1 + x \ ; \ (y \ ; \ x)^{\Omega} \ ; \ y$
**by** (*smt Omega-induct Omega-unfold antisym mult-associative mult-left-dist-add mult-left-one mult-right-dist-add mult-right-isotone mult-right-one order-refl*)

**lemma** *Omega-add-1*: $(x + y)^{\Omega} = x^{\Omega} \ ; \ (y \ ; \ x^{\Omega})^{\Omega}$
**by** (*smt Omega-induct Omega-unfold add-associative add-commutative add-right-isotone antisym mult-associative mult-left-one mult-right-dist-add mult-right-isotone mult-right-one order-refl*)

**lemma** *Omega-add*: $(x + y)^{\Omega} = (x^{\Omega} \ ; \ y)^{\Omega} \ ; \ x^{\Omega}$
**by** (*smt Omega-add-1 Omega-mult mult-associative mult-left-dist-add mult-left-one mult-right-dist-add mult-right-one*)

**lemma** *Omega-simulate*: $z \ ; \ x \leq y \ ; \ z \rightarrow z \ ; \ x^{\Omega} \leq y^{\Omega} \ ; \ z$
**by** (*smt Omega-induct Omega-unfold add-right-isotone mult-associative mult-left-dist-add mult-left-isotone mult-right-one*)

**end**

**sublocale** *dra* < *Omega*!: *itering-1* **where** *circ* = *Omega*
**apply** *unfold-locales*
**apply** (*metis Omega-mult*)
**apply** (*metis Omega-add*)
**by** (*metis Omega-simulate*)

**context** *dra*

**begin**

**lemma** *star-below-Omega*: $x^{*} \leq x^{\Omega}$
**by** (*metis Omega-isolate add-commutative add-left-divisibility*)

**lemma** *Omega-sum-unfold-1*: $(x + y)^{\Omega} = y^{\Omega} + y^{*} \ ; \ x \ ; \ (x + y)^{\Omega}$
**by** (*smt Omega.circ-add Omega.circ-loop-fixpoint Omega-isolate add-associative add-commutative mult-associative mult-left-zero mult-right-dist-add*)

**lemma** *Omega-add-3*: $(x + y)^\Omega = (x^* \ ; y)^\Omega \ ; x^\Omega$
**by** (*smt Omega.circ-add Omega.circ-isotone Omega-induct Omega-sum-unfold-1 add-commutative antisym mult-left-isotone order-refl star-below-Omega*)

**lemma** *Omega-one*: $1^\Omega = T$
**by** (*metis Omega.circ-transitive-equal Omega-induct add-left-top add-right-upper-bound less-eq-def mult-left-one*)

**lemma** *top-left-zero*: $T \ ; x = T$
**by** (*metis Omega-induct Omega-one add-left-top add-right-upper-bound less-eq-def mult-left-one*)

**lemma** *star-mult-Omega*: $x^\Omega = x^* \ ; x^\Omega$
**by** (*metis Omega.circ-plus-one Omega.circ-transitive-equal Omega-isolate add-commutative less-eq-def order-refl star.circ-add-mult-zero star.circ-increasing star.circ-plus-same star-involutive star-left-induct star-square*)

**lemma** *Omega-separate-2*: $y \ ; x \leq x \ ; (x + y) \rightarrow (x + y)^\Omega = x^\Omega \ ; y^\Omega$
**by** (*smt Omega.circ-sub-dist-3 Omega-induct Omega-sum-unfold-1 add-right-isotone antisym mult-associative mult-left-isotone star-mult-Omega star-simulation-left*)

**lemma** *Omega-circ-simulate-right-plus*: $z \ ; x \leq y \ ; y^\Omega \ ; z + w \rightarrow z \ ; x^\Omega \leq y^\Omega \ ; (z + w \ ; x^\Omega)$
**by** (*smt Omega.circ-back-loop-fixpoint Omega.circ-plus-same Omega.left-plus-circ Omega-induct add-associative add-commutative add-right-isotone less-eq-def mult-associative mult-right-dist-add*)

**lemma** *Omega-circ-simulate-left-plus*: $x \ ; z \leq z \ ; y^\Omega + w \rightarrow x^\Omega \ ; z \leq (z + x^\Omega \ ; w) \ ; y^\Omega$
**proof**
**assume** *1*: $x \ ; z \leq z \ ; y^\Omega + w$
**hence** $x \ ; ((z + x^\Omega \ ; w) \ ; y^\Omega) \leq z \ ; y^\Omega \ ; y^\Omega + w \ ; y^\Omega + x^\Omega \ ; w \ ; y^\Omega$
**by** (*smt Omega.left-plus-below-circ add-left-isotone add-right-isotone mult-associative mult-left-dist-add mult-left-isotone mult-right-dist-add order-trans*)
**hence** $x \ ; ((z + x^\Omega \ ; w) \ ; y^\Omega) \leq (z + x^\Omega \ ; w) \ ; y^\Omega$
**by** (*metis Omega.circ-transitive-equal mult-associative Omega.circ-reflexive add-associative less-eq-def mult-left-one mult-right-dist-add*)
**thus** $x^\Omega \ ; z \leq (z + x^\Omega \ ; w) \ ; y^\Omega$
**by** (*smt Omega.circ-back-loop-fixpoint Omega-isolate add-least-upper-bound mult-associative mult-left-zero mult-right-dist-add mult-right-subdist-add-left mult-right-subdist-add-right star-left-induct*)
**qed**

**end**

**sublocale** *dra* $<$ *Omega*!: *itering-T* **where** *circ* $=$ *Omega*
**apply** *unfold-locales*
**apply** (*smt Omega.circ-loop-fixpoint Omega-circ-simulate-right-plus add-associative add-commutative add-left-upper-bound mult-associative mult-left-dist-add order-trans*)
**apply** (*smt Omega.circ-increasing Omega-circ-simulate-left-plus add-commutative add-right-isotone mult-right-isotone order-trans*)
**apply** (*metis Omega-circ-simulate-right-plus*)
**by** (*metis Omega-circ-simulate-left-plus*)

**class** *dra-omega* $=$ *omega-algebra-T* $+$ *Omega* $+$
**assumes** *top-left-zero*: $T \ ; x = T$
**assumes** *Omega-def*: $x^\Omega = x^\omega + x^*$

**begin**

**lemma** *omega-left-zero*: $x^\omega \ ; \ y = x^\omega$
**by** (*metis mult-associative omega-vector top-left-zero*)

**lemma** *Omega-mult*: $(x \ ; \ y)^\Omega = 1 + x \ ; \ (y \ ; \ x)^\Omega \ ; \ y$
**by** (*smt add-associative add-commutative Omega-def mult-left-dist-add mult-right-dist-add omega-left-zero omega-slide star-unfold-slide*)

**lemma** *Omega-add*: $(x + y)^\Omega = (x^\Omega \ ; \ y)^\Omega \ ; \ x^\Omega$
**proof** −
**have** $(x^\Omega \ ; \ y)^\Omega \ ; \ x^\Omega = (x^* \ ; \ y)^* \ ; \ x^\omega + (x^* \ ; \ y)^\omega + (x^* \ ; \ y)^* \ ; \ x^{\omega *} \ ; \ x^\Omega$
**by** (*smt add-commutative Omega-def mult-associative mult-right-dist-add mult-zero-add-omega omega-left-zero star.circ-add-1*)
**thus** *?thesis*
**by** (*smt add-associative add-commutative Omega-def mult-associative mult-left-dist-add omega-decompose omega-left-zero star.circ-add-1 star.circ-loop-fixpoint star.circ-slide*)
**qed**

**lemma** *Omega-simulate*: $z \ ; \ x \leq y \ ; \ z \rightarrow z \ ; \ x^\Omega \leq y^\Omega \ ; \ z$
**by** (*smt add-isotone Omega-def mult-left-dist-add mult-right-dist-add omega-left-zero omega-simulation star-simulation-right*)

**subclass** *dra*
**apply** *unfold-locales*
**apply** (*metis Omega-def comb0.circ-left-unfold omega-left-zero*)
**apply** (*smt Omega-def add-associative less-eq-def mult-right-dist-add omega-left-zero zero-right-mult-decreasing*)
**by** (*metis Omega-def mult-right-dist-add omega-induct omega-left-zero*)

**end**

**class** *dom* =
**fixes** $d :: 'a \Rightarrow 'a$

**class** *domain-semiring* = *semiring* + *dom* +
**assumes** *d-restrict*: $x + d(x) \ ; \ x = d(x) \ ; \ x$
**assumes** *d-mult-d* : $d(x \ ; \ y) = d(x \ ; \ d(y))$
**assumes** *d-plus-one*: $d(x) + 1 = 1$
**assumes** *d-zero* : $d(0) = 0$
**assumes** *d-dist-add*: $d(x + y) = d(x) + d(y)$

**begin**

Most lemmas in this class are taken from Georg Struth's theories.

**lemma** *d-restrict-equals*: $x = d(x) \ ; \ x$
**by** (*metis add-commutative d-plus-one d-restrict mult-left-one mult-right-dist-add*)

**lemma** *d-involutive*: $d(d(x)) = d(x)$
**by** (*metis d-mult-d mult-left-one*)

**lemma** *d-fixpoint*: $(\exists \, y \ . \ x = d(y)) \leftrightarrow x = d(x)$

**by** (*metis d-involutive*)

**lemma** *d-type*: $\forall P\ .\ (\forall x\ .\ x = d(x) \rightarrow P(x)) \leftrightarrow (\forall x\ .\ P(d(x)))$
**by** (*metis d-involutive*)

**lemma** *d-mult-sub*: $d(x\ ;\ y) \leq d(x)$
**by** (*metis d-dist-add d-mult-d d-plus-one less-eq-def mult-left-subdist-add-left mult-right-one*)

**lemma** *d-sub-one*: $x \leq 1 \rightarrow x \leq d(x)$
**by** (*metis d-restrict-equals mult-right-isotone mult-right-one*)

**lemma** *d-strict*: $d(x) = 0 \leftrightarrow x = 0$
**by** (*metis d-restrict-equals d-zero mult-left-zero*)

**lemma** *d-one*: $d(1) = 1$
**by** (*metis d-restrict-equals mult-right-one*)

**lemma** *d-below-one*: $d(x) \leq 1$
**by** (*metis d-plus-one less-eq-def*)

**lemma** *d-isotone*: $x \leq y \rightarrow d(x) \leq d(y)$
**by** (*metis d-dist-add less-eq-def*)

**lemma** *d-plus-left-upper-bound*: $d(x) \leq d(x + y)$
**by** (*metis add-left-upper-bound d-isotone*)

**lemma** *d-export*: $d(d(x)\ ;\ y) = d(x)\ ;\ d(y)$
**by** (*smt add-commutative antisym d-mult-d d-mult-sub d-plus-left-upper-bound d-plus-one d-restrict d-sub-one mult-isotone mult-left-one mult-left-subdist-add-right mult-right-dist-add mult-right-one*)

**lemma** *d-idempotent*: $d(x)\ ;\ d(x) = d(x)$
**by** (*metis d-export d-restrict-equals*)

**lemma** *d-commutative*: $d(x)\ ;\ d(y) = d(y)\ ;\ d(x)$
**by** (*smt antisym d-export d-mult-d d-mult-sub d-one d-restrict-equals mult-isotone mult-left-one*)

**lemma** *d-least-left-preserver*: $x \leq d(y)\ ;\ x \leftrightarrow d(x) \leq d(y)$
**by** (*metis d-below-one d-involutive d-mult-sub d-restrict-equals eq-iff mult-left-isotone mult-left-one*)

**lemma** *d-weak-locality*: $x\ ;\ y = 0 \leftrightarrow x\ ;\ d(y) = 0$
**by** (*metis d-mult-d d-strict*)

**lemma** *d-add-closed*: $d(d(x) + d(y)) = d(x) + d(y)$
**by** (*metis d-dist-add d-involutive*)

**lemma** *d-mult-closed*: $d(d(x)\ ;\ d(y)) = d(x)\ ;\ d(y)$
**by** (*metis d-export d-mult-d*)

**lemma** *d-mult-left-lower-bound*: $d(x)\ ;\ d(y) \leq d(x)$
**by** (*metis d-export d-involutive d-mult-sub*)

**lemma** *d-mult-greatest-lower-bound*: $d(x) \leq d(y)\ ;\ d(z) \leftrightarrow d(x) \leq d(y) \wedge d(x) \leq d(z)$
**by** (*metis d-commutative d-idempotent d-mult-left-lower-bound mult-isotone order-trans*)

**lemma** *d-mult-left-absorb-add*: $d(x)\ ;\ (d(x) + d(y)) = d(x)$
**by** (*metis add-commutative d-idempotent d-plus-one mult-left-dist-add mult-right-one*)

**lemma** *d-add-left-absorb-mult*: $d(x) + d(x)\ ;\ d(y) = d(x)$
**by** (*metis add-commutative d-mult-left-lower-bound less-eq-def*)

**lemma** *d-add-left-dist-mult*: $d(x) + d(y)\ ;\ d(z) = (d(x) + d(y))\ ;\ (d(x) + d(z))$
**by** (*smt add-associative d-commutative d-idempotent d-mult-left-absorb-add mult-left-dist-add mult-right-dist-add*)

**lemma** *d-order*: $d(x) \leq d(y) \leftrightarrow d(x) = d(x)\ ;\ d(y)$
**by** (*metis d-mult-greatest-lower-bound d-mult-left-absorb-add less-eq-def order-refl*)

**lemma** *d-mult-below*: $d(x)\ ;\ y \leq y$
**by** (*metis add-left-divisibility d-plus-one mult-left-one mult-right-dist-add*)

**lemma** *d-preserves-equation*: $d(y)\ ;\ x \leq x\ ;\ d(y) \leftrightarrow d(y)\ ;\ x = d(y)\ ;\ x\ ;\ d(y)$
**by** (*smt antisym d-below-one d-idempotent mult-associative mult-left-isotone mult-left-one mult-right-isotone mult-right-one*)

**end**

**class** *semiring-dL-below* = *domain-semiring* +
**fixes** $L :: 'a$
**assumes** *L-left-zero-below*: $L\ ;\ x \leq L$
**assumes** *mult-L-split*: $x\ ;\ L = x\ ;\ 0 + d(x)\ ;\ L$

**begin**

**lemma** *d-zero-mult-L*: $d(x\ ;\ 0)\ ;\ L \leq x$
**by** (*metis add-least-upper-bound mult-L-split mult-associative mult-left-zero zero-right-mult-decreasing*)

**lemma** *mult-L*: $x\ ;\ L \leq x\ ;\ 0 + L$
**by** (*metis add-right-isotone d-mult-below mult-L-split*)

**lemma** *d-mult-L*: $d(x)\ ;\ L \leq x\ ;\ L$
**by** (*metis add-right-divisibility mult-L-split*)

**lemma** *d-L-split*: $x \ ; \ d(y) \ ; \ L = x \ ; \ 0 + d(x \ ; \ y) \ ; \ L$
**by** (*metis d-commutative d-mult-d d-zero mult-L-split mult-associative mult-left-zero*)

**lemma** *d-mult-mult-L*: $d(x \ ; \ y) \ ; \ L \leq x \ ; \ d(y) \ ; \ L$
**by** (*metis add-right-divisibility d-L-split*)

**lemma** *L-L*: $L \ ; \ L = L$
**by** (*metis d-restrict-equals less-eq-def mult-L-split zero-right-mult-decreasing*)

**end**

**class** *ed-below* = *omega-algebra-T* + *semiring-dL-below* + *Omega* +
**assumes** *Omega-def*: $x^\Omega = d(x^\omega) \ ; \ L + x^*$

**begin**

**lemma** *Omega-isotone*: $x \leq y \rightarrow x^\Omega \leq y^\Omega$
**by** (*metis Omega-def add-isotone d-isotone mult-left-isotone omega-isotone star.circ-isotone*)

**lemma** *star-below-Omega*: $x^* \leq x^\Omega$
**by** (*metis Omega-def add-right-upper-bound*)

**lemma** *one-below-Omega*: $1 \leq x^\Omega$
**by** (*metis add-least-upper-bound star.circ-plus-one star-below-Omega*)

**lemma** *L-left-zero-star*: $L \ ; \ x^* = L$
**by** (*metis L-left-zero-below add-right-upper-bound antisym star.circ-back-loop-fixpoint*)

**lemma** *L-left-zero-Omega*: $L \ ; \ x^\Omega = L$
**by** (*metis L-left-zero-below L-left-zero-star Omega-def less-eq-def mult-left-dist-add*)

**lemma** *mult-L-star*: $(x \ ; \ L)^* = 1 + x \ ; \ L$
**by** (*metis L-left-zero-star mult-associative star.circ-left-unfold*)

**lemma** *mult-L-omega-below*: $(x \ ; \ L)^\omega \leq x \ ; \ L$
**by** (*metis L-left-zero-below mult-right-isotone omega-slide*)

**lemma** *mult-L-add-star*: $(x \ ; \ L + y)^* = y^* + y^* \ ; \ x \ ; \ L$
**by** (*metis L-left-zero-star add-commutative mult-associative star.circ-unfold-sum*)

**lemma** *mult-L-add-omega-below*: $(x \ ; \ L + y)^\omega \leq y^\omega + y^* \ ; \ x \ ; \ L$
**proof** −
**have** $(x \ ; \ L + y)^\omega = (y^* \ ; \ x \ ; \ L)^\omega + (y^* \ ; \ x \ ; \ L)^* \ ; \ y^\omega$
**by** (*metis add-commutative mult-associative omega-decompose*)
**also have** ... $\leq y^* \ ; \ x \ ; \ L + (y^* \ ; \ x \ ; \ L)^* \ ; \ y^\omega$
**by** (*metis add-left-isotone mult-L-omega-below*)
**also have** ... $= y^* \ ; \ x \ ; \ L + y^* \ ; \ x \ ; \ L \ ; \ y^\omega + y^\omega$

**by** (*smt L-left-zero-star add-associative add-commutative mult-associative star.circ-loop-fixpoint*)
**also have** ... $\leq y^\omega + y^* \,;\, x \,;\, L$
**by** (*metis L-left-zero-star add-commutative eq-refl mult-associative star.circ-back-loop-fixpoint*)
**finally show** *?thesis*
.
**qed**

**lemma** *mult-L-add-circ*: $(x \,;\, L + y)^\Omega = d(y^\omega) \,;\, L + y^* + y^* \,;\, x \,;\, L$
**proof** −
**have** $(x \,;\, L + y)^\Omega = d((x \,;\, L + y)^\omega) \,;\, L + (x \,;\, L + y)^*$
**by** (*metis Omega-def*)
**also have** ... $\leq d(y^\omega + y^* \,;\, x \,;\, L) \,;\, L + (x \,;\, L + y)^*$
**by** (*metis add-left-isotone d-isotone mult-L-add-omega-below mult-left-isotone*)
**also have** ... $= d(y^\omega) \,;\, L + d(y^* \,;\, x \,;\, L) \,;\, L + (x \,;\, L + y)^*$
**by** (*metis d-dist-add mult-right-dist-add*)
**also have** ... $\leq d(y^\omega) \,;\, L + y^* \,;\, x \,;\, L \,;\, L + (x \,;\, L + y)^*$
**by** (*metis add-left-isotone add-right-isotone d-mult-L*)
**also have** ... $= d(y^\omega) \,;\, L + y^* + y^* \,;\, x \,;\, L$
**by** (*smt L-L add-associative add-commutative less-eq-def mult-L-add-star mult-associative order-refl*)
**finally have** *1*: $(x \,;\, L + y)^\Omega \leq d(y^\omega) \,;\, L + y^* + y^* \,;\, x \,;\, L$
.
**have** *2*: $d(y^\omega) \,;\, L \leq (x \,;\, L + y)^\Omega$
**by** (*metis Omega-def add-left-upper-bound add-right-upper-bound d-isotone mult-left-isotone omega-isotone order-trans*)
**have** $y^* + y^* \,;\, x \,;\, L \leq (x \,;\, L + y)^\Omega$
**by** (*metis Omega-def add-right-upper-bound mult-L-add-star*)
**hence** $d(y^\omega) \,;\, L + y^* + y^* \,;\, x \,;\, L \leq (x \,;\, L + y)^\Omega$ **using** *2*
**by** (*metis Omega-def add-least-upper-bound add-right-upper-bound mult-L-add-star*)
**thus** *?thesis* **using** *1*
**by** (*metis antisym*)
**qed**

**lemma** *circ-add-d*: $(x^\Omega \,;\, y)^\Omega \,;\, x^\Omega = d((x^* \,;\, y)^\omega) \,;\, L + ((x^* \,;\, y)^* \,;\, x^* + (x^* \,;\, y)^* \,;\, d(x^\omega) \,;\, L)$
**proof** −
**have** $(x^\Omega \,;\, y)^\Omega \,;\, x^\Omega = ((d(x^\omega) \,;\, L + x^*) \,;\, y)^\Omega \,;\, x^\Omega$
**by** (*metis Omega-def*)
**also have** ... $= (d(x^\omega) \,;\, L \,;\, y + x^* \,;\, y)^\Omega \,;\, x^\Omega$
**by** (*metis mult-right-dist-add*)
**also have** ... $\leq (d(x^\omega) \,;\, L + x^* \,;\, y)^\Omega \,;\, x^\Omega$
**by** (*metis L-left-zero-below Omega-isotone add-left-isotone mult-associative mult-left-isotone mult-right-isotone*)
**also have** ... $= (d((x^* \,;\, y)^\omega) \,;\, L + (x^* \,;\, y)^* + (x^* \,;\, y)^* \,;\, d(x^\omega) \,;\, L) \,;\, x^\Omega$
**by** (*metis mult-L-add-circ*)
**also have** ... $= d((x^* \,;\, y)^\omega) \,;\, L \,;\, x^\Omega + (x^* \,;\, y)^* \,;\, x^\Omega + (x^* \,;\, y)^* \,;\, d(x^\omega) \,;\, L \,;\, x^\Omega$
**by** (*metis mult-right-dist-add*)
**also have** ... $= d((x^* \,;\, y)^\omega) \,;\, L + (x^* \,;\, y)^* \,;\, x^\Omega + (x^* \,;\, y)^* \,;\, d(x^\omega) \,;\, L$
**by** (*smt L-left-zero-Omega mult-associative*)
**also have** ... $= d((x^* \,;\, y)^\omega) \,;\, L + ((x^* \,;\, y)^* \,;\, x^* + (x^* \,;\, y)^* \,;\, d(x^\omega) \,;\, L)$
**by** (*smt Omega-def add-associative add-commutative add-idempotent mult-associative mult-left-dist-add*)

**finally have** *1*: $(x^\Omega \ ; \ y)^\Omega \ ; \ x^\Omega \leq d((x^* \ ; \ y)^\omega) \ ; \ L + ((x^* \ ; \ y)^* \ ; \ x^* + (x^* \ ; \ y)^* \ ; \ d(x^\omega) \ ; \ L)$
.
**have** $d((x^* \ ; \ y)^\omega) \ ; \ L \leq (x^\Omega \ ; \ y)^\Omega$
**by** (*metis Omega-def Omega-isotone add-commutative add-right-upper-bound mult-left-isotone order-trans*)
**also have** $... \leq (x^\Omega \ ; \ y)^\Omega \ ; \ x^\Omega$
**by** (*metis mult-right-isotone mult-right-one one-below-Omega*)
**finally have** *2*: $d((x^* \ ; \ y)^\omega) \ ; \ L \leq (x^\Omega \ ; \ y)^\Omega \ ; \ x^\Omega$
.
**have** *3*: $(x^* \ ; \ y)^* \ ; \ x^* \leq (x^\Omega \ ; \ y)^\Omega \ ; \ x^\Omega$
**by** (*smt Omega-isotone mult-left-isotone mult-right-isotone order-trans star-below-Omega*)
**have** $(x^* \ ; \ y)^* \ ; \ d(x^\omega) \ ; \ L \leq (x^* \ ; \ y)^* \ ; \ x^\Omega$
**by** (*metis Omega-def add-commutative mult-associative mult-left-subdist-add-right*)
**also have** $... \leq (x^\Omega \ ; \ y)^\Omega \ ; \ x^\Omega$
**by** (*metis Omega-isotone mult-left-isotone order-trans star-below-Omega*)
**finally have** $d((x^* \ ; \ y)^\omega) \ ; \ L + ((x^* \ ; \ y)^* \ ; \ x^* + (x^* \ ; \ y)^* \ ; \ d(x^\omega) \ ; \ L) \leq (x^\Omega \ ; \ y)^\Omega \ ; \ x^\Omega$ **using** *2 3*
**by** (*smt add-associative less-eq-def*)
**thus** *?thesis* **using** *1*
**by** (*metis antisym*)
**qed**

**end**

**class** *ed* = *ed-below* +
**assumes** *L-left-zero*: $L \ ; \ x = L$

**begin**

**lemma** *mult-L-omega*: $(x \ ; \ L)^\omega = x \ ; \ L$
**by** (*metis L-left-zero omega-slide*)

**lemma** *mult-L-add-omega*: $(x \ ; \ L + y)^\omega = y^\omega + y^* \ ; \ x \ ; \ L$
**by** (*smt L-left-zero add-commutative add-left-upper-bound less-eq-def mult-L-omega mult-L-star mult-associative mult-left-one mult-right-dist-add omega-decompose*)

**lemma** *d-Omega-circ-simulate-right-plus*: $z \ ; \ x \leq y \ ; \ y^\Omega \ ; \ z + w \rightarrow z \ ; \ x^\Omega \leq y^\Omega \ ; \ (z + w \ ; \ x^\Omega)$
**proof**
**assume** $z \ ; \ x \leq y \ ; \ y^\Omega \ ; \ z + w$
**hence** $z \ ; \ x \leq y \ ; \ d(y^\omega) \ ; \ L \ ; \ z + y \ ; \ y^* \ ; \ z + w$
**by** (*metis Omega-def mult-associative mult-left-dist-add mult-right-dist-add*)
**also have** $... \leq y \ ; \ d(y^\omega) \ ; \ L + y \ ; \ y^* \ ; \ z + w$
**by** (*metis L-left-zero-below add-commutative add-right-isotone mult-associative mult-right-isotone*)
**also have** $... = y \ ; \ 0 + d(y \ ; \ y^\omega) \ ; \ L + y \ ; \ y^* \ ; \ z + w$
**by** (*metis d-L-split*)
**also have** $... = d(y^\omega) \ ; \ L + y \ ; \ y^* \ ; \ z + w$
**by** (*smt add-associative add-commutative add-left-zero mult-associative mult-left-dist-add omega-unfold*)
**finally have** *1*: $z \ ; \ x \leq d(y^\omega) \ ; \ L + y \ ; \ y^* \ ; \ z + w$

.
**have** $(d(y^\omega)\ ;\ L + y^*\ ;\ z + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^*) ; x = d(y^\omega)\ ;\ L\ ;\ x + y^*\ ;\ z\ ;\ x + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L\ ;\ x + y^*\ ;\ w\ ;\ x^*\ ;\ x$
**by** (*metis mult-right-dist-add*)
**also have** $\ldots \leq d(y^\omega)\ ;\ L + y^*\ ;\ z\ ;\ x + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L\ ;\ x + y^*\ ;\ w\ ;\ x^*\ ;\ x$
**by** (*metis L-left-zero-below add-left-isotone mult-associative mult-right-isotone*)
**also have** $\ldots \leq d(y^\omega)\ ;\ L + y^*\ ;\ z\ ;\ x + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^*\ ;\ x$
**by** (*metis L-left-zero-below add-commutative add-left-isotone mult-associative mult-right-isotone*)
**also have** $\ldots \leq d(y^\omega)\ ;\ L + y^*\ ;\ z\ ;\ x + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^*$
**by** (*metis add-left-upper-bound add-right-isotone star.circ-back-loop-fixpoint*)
**also have** $\ldots \leq d(y^\omega)\ ;\ L + y^*\ ;\ (d(y^\omega)\ ;\ L + y\ ;\ y^*\ ;\ z + w) + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^*$ **using** *1*
**by** (*smt add-left-isotone add-right-isotone less-eq-def mult-associative mult-left-dist-add*)
**also have** $\ldots = d(y^\omega)\ ;\ L + y^*\ ;\ y\ ;\ y^*\ ;\ z + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^*$
**by** (*smt add-associative add-commutative add-idempotent mult-associative mult-left-dist-add d-L-split star.circ-back-loop-fixpoint star-mult-omega*)
**also have** $\ldots \leq d(y^\omega)\ ;\ L + y^*\ ;\ z + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^*$
**by** (*metis add-left-isotone add-right-isotone mult-left-isotone star.circ-plus-same star.circ-transitive-equal star.left-plus-below-circ*)
**finally have** *2*: $z\ ;\ x^* \leq d(y^\omega)\ ;\ L + y^*\ ;\ z + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^*$
**by** (*smt add-least-upper-bound add-left-upper-bound star.circ-loop-fixpoint star-right-induct*)
**have** $z\ ;\ x\ ;\ x^\omega \leq y\ ;\ y^*\ ;\ z\ ;\ x^\omega + d(y^\omega)\ ;\ L\ ;\ x^\omega + w\ ;\ x^\omega$ **using** *1*
**by** (*metis add-commutative mult-left-isotone mult-right-dist-add*)
**also have** $\ldots \leq y\ ;\ y^*\ ;\ z\ ;\ x^\omega + d(y^\omega)\ ;\ L + w\ ;\ x^\omega$
**by** (*metis L-left-zero-below add-commutative add-right-isotone mult-associative mult-right-isotone*)
**finally have** $z\ ;\ x^\omega \leq y^\omega + y^*\ ;\ d(y^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^\omega$
**by** (*smt add-associative add-commutative left-plus-omega mult-associative mult-left-dist-add omega-induct omega-unfold star.left-plus-circ*)
**hence** $z\ ;\ x^\omega \leq y^\omega + y^*\ ;\ d(y^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^\omega$
**by** (*smt add-associative add-commutative left-plus-omega mult-associative mult-left-dist-add omega-induct omega-unfold star.left-plus-circ*)
**hence** $z\ ;\ x^\omega \leq y^\omega + y^*\ ;\ w\ ;\ x^\omega$
**by** (*metis add-commutative d-mult-L less-eq-def mult-associative mult-right-isotone omega-sub-vector order-trans star-mult-omega*)
**hence** $d(z\ ;\ x^\omega)\ ;\ L \leq d(y^\omega)\ ;\ L + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L$
**by** (*smt add-associative add-commutative d-L-split d-dist-add less-eq-def mult-right-dist-add*)
**hence** $z\ ;\ d(x^\omega)\ ;\ L \leq z\ ;\ 0 + d(y^\omega)\ ;\ L + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L$
**by** (*metis add-associative add-right-isotone d-L-split*)
**hence** $z\ ;\ d(x^\omega)\ ;\ L \leq d(y^\omega)\ ;\ L + y^*\ ;\ z + y^*\ ;\ w\ ;\ d(x^\omega)\ ;\ L + y^*\ ;\ w\ ;\ x^*$
**by** (*smt add-commutative add-left-isotone add-left-upper-bound order-trans star.circ-loop-fixpoint zero-right-mult-decreasing*)
**thus** $z\ ;\ x^\Omega \leq y^\Omega\ ;\ (z + w\ ;\ x^\Omega)$ **using** *2*
**by** (*smt L-left-zero Omega-def add-associative less-eq-def mult-associative mult-left-dist-add mult-right-dist-add*)
**qed**

**lemma** *d-Omega-circ-simulate-left-plus*: $x\ ;\ z \leq z\ ;\ y^\Omega + w \rightarrow x^\Omega\ ;\ z \leq (z + x^\Omega\ ;\ w)\ ;\ y^\Omega$
**proof**
**assume** *1*: $x\ ;\ z \leq z\ ;\ y^\Omega + w$
**have** $x\ ;\ (z\ ;\ d(y^\omega)\ ;\ L + z\ ;\ y^* + d(x^\omega)\ ;\ L + x^*\ ;\ w\ ;\ d(y^\omega)\ ;\ L + x^*\ ;\ w\ ;\ y^*) = x\ ;\ z\ ;\ d(y^\omega)\ ;\ L + x\ ;\ z\ ;\ y^* + d(x^\omega)\ ;\ L + x\ ;\ x^*\ ;\ w\ ;\ d(y^\omega)\ ;\ L + x\ ;\ x^*\ ;\ w\ ;\ y^*$
**by** (*smt add-associative add-commutative mult-associative mult-left-dist-add d-L-split omega-unfold*)
**also have** $\ldots \leq (z\ ;\ d(y^\omega)\ ;\ L + z\ ;\ y^* + w)\ ;\ d(y^\omega)\ ;\ L + (z\ ;\ d(y^\omega)\ ;\ L + z\ ;\ y^* + w)\ ;\ y^* + d(x^\omega)\ ;\ L + x^*\ ;\ w\ ;\ d(y^\omega)\ ;\ L + x^*\ ;\ w\ ;\ y^*$ **using** *1*
**by** (*smt Omega-def add-associative add-right-upper-bound less-eq-def mult-associative mult-left-dist-add mult-right-dist-add star.circ-loop-fixpoint*)
**also have** $\ldots = z\ ;\ d(y^\omega)\ ;\ L + z\ ;\ y^*\ ;\ d(y^\omega)\ ;\ L + w\ ;\ d(y^\omega)\ ;\ L + z\ ;\ y^* + w\ ;\ y^* + d(x^\omega)\ ;\ L + x^*\ ;\ w\ ;\ d(y^\omega)\ ;\ L + x^*\ ;\ w\ ;\ y^*$
**by** (*smt L-left-zero add-associative add-commutative add-idempotent mult-associative mult-right-dist-add star.circ-transitive-equal*)
**also have** $\ldots = z\ ;\ d(y^\omega)\ ;\ L + w\ ;\ d(y^\omega)\ ;\ L + z\ ;\ y^* + w\ ;\ y^* + d(x^\omega)\ ;\ L + x^*\ ;\ w\ ;\ d(y^\omega)\ ;\ L + x^*\ ;\ w\ ;\ y^*$

**by** (*smt add-associative add-commutative add-idempotent less-eq-def mult-associative d-L-split star-mult-omega zero-right-mult-decreasing*)
**finally have** $x\ ;\ (z\ ;\ d(y^\omega)\ ;\ L + z\ ;\ y^* + d(x^\omega)\ ;\ L + x^*\ ;\ w\ ;\ d(y^\omega)\ ;\ L + x^*\ ;\ w\ ;\ y^*) \leq z\ ;\ d(y^\omega)\ ;\ L + z\ ;\ y^* + d(x^\omega)\ ;\ L + x^*\ ;\ w\ ;\ d(y^\omega)\ ;\ L + x^*\ ;\ w\ ;\ y^*$
**by** (*smt add-associative add-commutative add-idempotent mult-associative star.circ-loop-fixpoint*)
**thus** $x^\Omega\ ;\ z \leq (z + x^\Omega\ ;\ w)\ ;\ y^\Omega$
**by** (*smt L-left-zero Omega-def add-associative add-least-upper-bound add-left-upper-bound mult-associative mult-left-dist-add mult-right-dist-add star.circ-back-loop-fixpoint star-left-induct*)
**qed**

**end**

**sublocale** *ed* < *ed-omega*!: *omega-itering* **where** *circ* = *Omega*
**apply** *unfold-locales*
**apply** (*smt L-left-zero add-associative add-commutative add-left-zero Omega-def mult-associative mult-left-dist-add mult-right-dist-add d-L-split omega-slide star-unfold-slide*)
**apply** (*smt add-associative add-commutative add-left-zero circ-add-d Omega-def mult-left-dist-add mult-right-dist-add d-L-split d-dist-add omega-decompose star.circ-add-1 star.circ-slide*)
**apply** (*smt L-left-zero add-associative add-commutative add-isotone add-left-zero Omega-def mult-associative mult-left-dist-add mult-left-isotone mult-right-dist-add d-L-split d-isotone omega-simulation star-simulation-right*)
**apply** (*smt d-Omega-circ-simulate-right-plus Omega-def add-left-isotone add-right-upper-bound mult-left-isotone mult-right-isotone order-trans star.circ-back-loop-fixpoint*)
**apply** (*smt Omega-def add-commutative add-least-upper-bound add-right-isotone mult-right-isotone d-Omega-circ-simulate-left-plus order-trans star.circ-increasing*)
**apply** (*metis d-Omega-circ-simulate-right-plus*)
**by** (*metis d-Omega-circ-simulate-left-plus*)

**sublocale** *ed* < *ed-star*!: *omega-itering* **where** *circ* = *star* **..**

**class** *meet* =
**fixes** *meet* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixl** $\frown$ 65)

**class** *semiring-lattice* = *semiring-T* + *meet* +
**assumes** *meet-associative* : $(x \frown y) \frown z = x \frown (y \frown z)$
**assumes** *meet-commutative* : $x \frown y = y \frown x$
**assumes** *meet-idempotent* : $x \frown x = x$
**assumes** *meet-left-zero* : $0 \frown x = 0$
**assumes** *meet-left-top* : $T \frown x = x$
**assumes** *meet-left-dist-add* : $x \frown (y + z) = (x \frown y) + (x \frown z)$
**assumes** *add-left-dist-meet* : $x + (y \frown z) = (x + y) \frown (x + z)$
**assumes** *meet-absorb* : $x \frown (x + y) = x$
**assumes** *add-absorb* : $x + (x \frown y) = x$

**begin**

**lemma** *less-eq-meet*: $x \leq y \leftrightarrow x \frown y = x$
**by** (*metis add-absorb add-right-upper-bound less-eq-def meet-absorb meet-commutative*)

**lemma** *meet-left-isotone*: $x \leq y \rightarrow x \frown z \leq y \frown z$
**by** (*smt less-eq-def meet-commutative meet-left-dist-add*)

**lemma** *meet-right-isotone*: $x \leq y \rightarrow z \frown x \leq z \frown y$

**by** (*metis meet-commutative meet-left-isotone*)

**lemma** *meet-isotone*: $w \leq y \wedge x \leq z \rightarrow w \frown x \leq y \frown z$
**by** (*smt less-eq-meet meet-associative meet-commutative*)

**lemma** *meet-left-upper-bound*: $x \frown y \leq x$
**by** (*metis add-absorb add-right-divisibility*)

**lemma** *meet-right-upper-bound*: $x \frown y \leq y$
**by** (*metis meet-commutative meet-left-upper-bound*)

**lemma** *meet-least-upper-bound*: $z \leq x \wedge z \leq y \leftrightarrow z \leq x \frown y$
**by** (*metis less-eq-meet meet-associative meet-left-upper-bound*)

**lemma** *meet-left-divisibility*: $y \leq x \leftrightarrow (\exists z \,.\, x \frown z = y)$
**by** (*metis less-eq-meet meet-commutative meet-left-upper-bound*)

**lemma** *meet-right-divisibility*: $y \leq x \leftrightarrow (\exists z \,.\, z \frown x = y)$
**by** (*metis meet-commutative meet-left-divisibility*)

**lemma** *meet-right-zero*: $x \frown 0 = 0$
**by** (*metis meet-commutative meet-left-zero*)

**lemma** *mult-left-subdist-meet-left*: $x \,;\, (y \frown z) \leq x \,;\, y$
**by** (*metis meet-left-upper-bound mult-right-isotone*)

**lemma** *mult-left-subdist-meet-right*: $x \,;\, (y \frown z) \leq x \,;\, z$
**by** (*metis meet-commutative mult-left-subdist-meet-left*)

**lemma** *mult-right-subdist-meet-left*: $(x \frown y) \,;\, z \leq x \,;\, z$
**by** (*metis meet-left-upper-bound mult-left-isotone*)

**lemma** *mult-right-subdist-meet-right*: $(x \frown y) \,;\, z \leq y \,;\, z$
**by** (*metis meet-right-upper-bound mult-left-isotone*)

**lemma** *mult-same-context*:$x \leq y \frown z \wedge y \leq x \frown z \rightarrow x \frown z = y \frown z$
**by** (*metis eq-iff meet-least-upper-bound*)

**lemma** *mult-right-top*: $x \frown T = x$
**by** (*metis meet-commutative meet-left-top*)

**lemma** *vector-mult-closed*: *vector* $x \wedge$ *vector* $y \rightarrow$ *vector* $(x \frown y)$
**by** (*metis antisym meet-least-upper-bound mult-right-subdist-meet-left mult-right-subdist-meet-right top-right-mult-increasing vector-def*)

**lemma** *relative-equality*: $x + z = y + z \wedge x \frown z = y \frown z \rightarrow x = y$
**by** (*metis add-absorb add-commutative add-left-dist-meet*)

**end**

**class** *domain-semiring-lattice* = *domain-semiring* + *semiring-lattice*

**begin**

**lemma** *d-top*: $d(T) = 1$
**by** (*metis add-left-top d-dist-add d-one d-plus-one*)

**lemma** *mult-domain-top*: $x \ ; \ d(y) \ ; \ T \leq d(x \ ; \ y) \ ; \ T$
**by** (*smt d-mult-d d-restrict-equals mult-associative mult-right-isotone top-greatest*)

**lemma** *meet-domain*: $x \frown d(y) \ ; \ z = d(y) \ ; \ (x \frown z)$
**apply** (*rule antisym*)
**apply** (*smt add-commutative add-right-divisibility d-export d-isotone d-mult-left-lower-bound d-plus-one d-restrict-equals meet-right-isotone meet-right-upper-bound mult-left-isotone mult-left-one mult-right-isotone order-trans*)
**by** (*metis d-plus-one meet-least-upper-bound mult-left-one mult-left-subdist-meet-right mult-right-subdist-add-left*)

**lemma** *meet-intro-domain*: $x \frown y = d(y) \ ; \ x \frown y$
**by** (*metis d-restrict-equals meet-commutative meet-domain*)

**lemma** *meet-domain-top*: $x \frown d(y) \ ; \ T = d(y) \ ; \ x$
**by** (*metis meet-domain mult-right-top*)

**lemma** *d-galois*: $d(x) \leq d(y) \leftrightarrow x \leq d(y) \ ; \ T$
**by** (*metis d-isotone d-least-left-preserver meet-domain-top meet-least-upper-bound*)

**lemma** *vector-meet*: $x \ ; \ T \frown y \leq d(x) \ ; \ y$
**by** (*metis d-galois d-mult-sub meet-commutative meet-domain-top meet-right-isotone*)

**end**

**class** *domain-semiring-lattice-L* = *domain-semiring-lattice* +
**fixes** $L :: 'a$
**assumes** *l1*: $x \ ; \ L = x \ ; \ 0 + d(x) \ ; \ L$
**assumes** *l2*: $d(L) \ ; \ x \leq x \ ; \ d(L)$
**assumes** *l3*: $d(L) \ ; \ T \leq L + d(L \ ; \ 0) \ ; \ T$
**assumes** *l4*: $L \ ; \ T \leq L$
**assumes** *l5*: $x \ ; \ 0 \frown L \leq (x \frown L) \ ; \ 0$

**begin**

**lemma** *l8*: $(x \frown L) \ ; \ 0 \leq x \ ; \ 0 \frown L$

**by** (*metis meet-commutative meet-least-upper-bound mult-right-subdist-meet-right zero-right-mult-decreasing*)

**lemma** *l9*: $x \ ; \ 0 \frown L \leq d(x \ ; \ 0) \ ; \ L$
**by** (*metis d-restrict-equals meet-commutative meet-domain meet-right-upper-bound*)

**lemma** *l10*: $L \ ; \ L = L$
**by** (*metis d-restrict-equals l1 less-eq-def zero-right-mult-decreasing*)

**lemma** *l11*: $d(x) \ ; \ L \leq x \ ; \ L$
**by** (*metis add-right-upper-bound l1*)

**lemma** *l12*: $d(x \ ; \ 0) \ ; \ L \leq x \ ; \ 0$
**by** (*metis add-right-divisibility l1 mult-associative mult-left-zero*)

**lemma** *l13*: $d(x \ ; \ 0) \ ; \ L \leq x$
**by** (*metis l12 order-trans zero-right-mult-decreasing*)

**lemma** *l14*: $x \ ; \ L \leq x \ ; \ 0 + L$
**by** (*metis add-right-isotone l1 meet-domain-top meet-left-upper-bound*)

**lemma** *l15*: $x \ ; \ d(y) \ ; \ L = x \ ; \ 0 + d(x \ ; \ y) \ ; \ L$
**by** (*metis d-commutative d-mult-d d-zero l1 mult-associative mult-left-zero*)

**lemma** *l16*: $x \ ; \ T \frown L \leq x \ ; \ L$
**by** (*metis l11 order-trans vector-meet*)

**lemma** *l17*: $d(x) \ ; \ L \leq d(x \ ; \ L) \ ; \ L$
**by** (*smt d-restrict-equals l11 meet-domain-top meet-least-upper-bound meet-left-divisibility order-trans*)

**lemma** *l18*: $d(x) \ ; \ L = d(x \ ; \ L) \ ; \ L$
**by** (*metis antisym d-mult-sub l17 mult-left-isotone*)

**lemma** *l19*: $d(x \ ; \ T \ ; \ 0) \ ; \ L \leq d(x \ ; \ L) \ ; \ L$
**by** (*metis d-mult-sub l18 mult-associative mult-left-isotone*)

**lemma** *l20*: $x \leq y \leftrightarrow x \leq y + L \wedge x \leq y + d(y \ ; \ 0) \ ; \ T$
**apply** *rule*
**apply** (*metis add-absorb less-eq-meet meet-left-dist-add*)
**by** (*smt add-commutative add-left-dist-meet l13 less-eq-def meet-domain-top*)

**lemma** *l21*: $d(x \ ; \ 0) \ ; \ L \leq x \ ; \ 0 \frown L$
**by** (*metis l12 meet-domain-top meet-least-upper-bound meet-left-upper-bound*)

**lemma** *l22*: $x \ ; \ 0 \frown L = d(x \ ; \ 0) \ ; \ L$
**by** (*metis antisym l9 l21*)

**lemma** *l23*: $x \ ; \ T \frown L = d(x) \ ; \ L$

**apply** (*rule antisym*)
**apply** (*metis vector-meet*)
**by** (*metis add-least-upper-bound d-mult-below l1 meet-least-upper-bound mult-right-isotone top-greatest*)

**lemma** *l29*: $L\ ;\ d(L) = L$
**by** (*metis d-preserves-equation d-restrict-equals l2*)

**lemma** *l30*: $d(L)\ ;\ x \leq (x \frown L) + d(L\ ;\ 0)\ ;\ x$
**by** (*metis l3 less-eq-def meet-domain-top meet-left-dist-add*)

**lemma** *l31*: $d(L)\ ;\ x = (x \frown L) + d(L\ ;\ 0)\ ;\ x$
**by** (*smt add-least-upper-bound antisym d-mult-sub d-restrict-equals l30 meet-domain mult-left-isotone mult-left-subdist-meet-left*)

**lemma** *l40*: $L\ ;\ x \leq L$
**by** (*metis add-right-top l4 mult-left-subdist-add-left order-trans*)

**lemma** *l41*: $L\ ;\ T = L$
**by** (*metis antisym l4 top-right-mult-increasing*)

**lemma** *l50*: $x\ ;\ 0 \frown L = (x \frown L)\ ;\ 0$
**by** (*metis eq-iff l5 l8*)

**lemma** *l51*: $d(x\ ;\ 0)\ ;\ L = (x \frown L)\ ;\ 0$
**by** (*metis l22 l50*)

**lemma** *l90*: $L\ ;\ T\ ;\ L = L$
**by** (*metis add-commutative d-restrict-equals d-top l1 l51 l31 meet-absorb meet-commutative mult-associative mult-left-dist-add mult-left-zero mult-right-one*)

**lemma** *l91*: $x = x\ ;\ T \rightarrow d(L\ ;\ 0)\ ;\ x \leq d(x\ ;\ 0)\ ;\ T$
**proof** −
**have** $d(L\ ;\ 0)\ ;\ x \leq d(d(L\ ;\ 0)\ ;\ x)\ ;\ T$
**by** (*metis d-galois order-refl*)
**also have** $\ldots = d(d(L\ ;\ 0)\ ;\ d(x))\ ;\ T$
**by** (*metis d-mult-d*)
**also have** $\ldots = d(d(x)\ ;\ L\ ;\ 0)\ ;\ T$
**by** (*metis d-commutative d-mult-d mult-associative*)
**also have** $\ldots \leq d(x\ ;\ L\ ;\ 0)\ ;\ T$
**by** (*metis d-isotone l11 mult-left-isotone*)
**also have** $\ldots \leq d(x\ ;\ T\ ;\ 0)\ ;\ T$
**by** (*metis d-isotone mult-left-isotone mult-right-isotone top-greatest*)
**finally show** *?thesis*
**by** *metis*
**qed**

**lemma** *l92*: $x = x\ ;\ T \rightarrow d(L\ ;\ 0)\ ;\ x \leq d((x \frown L)\ ;\ 0)\ ;\ T$
**proof**
**assume** *1*: $x = x\ ;\ T$

**have** $d(L\ ;\ 0)\ ;\ x = d(L)\ ;\ d(L\ ;\ 0)\ ;\ x$
**by** (*metis d-commutative d-mult-sub d-order*)
**also have** $\ldots \leq d(L)\ ;\ d(x\ ;\ 0)\ ;\ T$ **using** *1*
**by** (*metis eq-iff l91 mult-associative mult-isotone*)
**also have** $\ldots = d(d(x\ ;\ 0)\ ;\ L)\ ;\ T$
**by** (*metis d-commutative d-export*)
**also have** $\ldots \leq d((x \frown L)\ ;\ 0)\ ;\ T$
**by** (*metis l51 order-refl*)
**finally show** $d(L\ ;\ 0)\ ;\ x \leq d((x \frown L)\ ;\ 0)\ ;\ T$
**by** *metis*
**qed**

**end**

**class** *domain-itering-lattice-L* = *itering-T* + *domain-semiring-lattice-L*

**begin**

**lemma** *mult-L-circ*: $(x\ ;\ L)^\circ = 1 + x\ ;\ L$
**by** (*metis circ-back-loop-fixpoint circ-mult l40 less-eq-def mult-associative*)

**lemma** *mult-L-circ-mult-below*: $(x\ ;\ L)^\circ\ ;\ y \leq y + x\ ;\ L$
**by** (*smt add-right-isotone l40 mult-L-circ mult-associative mult-left-one mult-right-dist-add mult-right-isotone*)

**lemma** *circ-L*: $L^\circ = L + 1$
**by** (*metis add-commutative l10 mult-L-circ*)

**lemma** *circ-d0-L*: $x^\circ\ ;\ d(x\ ;\ 0)\ ;\ L = x^\circ\ ;\ 0$
**by** (*metis add-right-zero circ-loop-fixpoint circ-plus-same d-zero l15 mult-associative mult-left-zero*)

**lemma** *d0-circ-left-unfold*: $d(x^\circ\ ;\ 0) = d(x\ ;\ x^\circ\ ;\ 0)$
**by** (*metis add-commutative add-left-zero circ-loop-fixpoint mult-associative*)

**lemma** *d-circ-import*: $d(y)\ ;\ x \leq x\ ;\ d(y) \rightarrow d(y)\ ;\ x^\circ = d(y)\ ;\ (d(y)\ ;\ x)^\circ$
**by** (*rule*, *rule antisym*,
*metis circ-simulate circ-slide mult-associative d-idempotent d-preserves-equation*,
*metis circ-isotone mult-left-isotone mult-left-one mult-right-isotone d-below-one*)

**end**

**class** *domain-omega-algebra-lattice-L* = *omega-algebra-T* + *domain-semiring-lattice-L*

**begin**

**lemma** *mult-L-star*: $(x\ ;\ L)^* = 1 + x\ ;\ L$

**by** (*metis l40 less-eq-def mult-associative star.circ-back-loop-fixpoint star.circ-mult*)

**lemma** *mult-L-omega*: $(x\ ;\ L)^\omega \leq x\ ;\ L$
**by** (*smt l41 less-eq-def mult-associative mult-left-dist-add omega-unfold top-greatest*)

**lemma** *mult-L-add-star*: $(x\ ;\ L + y)^* = y^* + y^*\ ;\ x\ ;\ L$
**proof** (*rule antisym*)
**have** $(x\ ;\ L + y)\ ;\ (y^* + y^*\ ;\ x\ ;\ L) = x\ ;\ L\ ;\ (y^* + y^*\ ;\ x\ ;\ L) + y\ ;\ (y^* + y^*\ ;\ x\ ;\ L)$
**by** (*metis mult-associative mult-right-dist-add*)
**also have** $\ldots \leq x\ ;\ L + y\ ;\ (y^* + y^*\ ;\ x\ ;\ L)$
**by** (*metis add-left-isotone l40 mult-associative mult-right-isotone*)
**also have** $\ldots \leq x\ ;\ L + y\ ;\ y^* + y^*\ ;\ x\ ;\ L$
**by** (*smt add-associative add-commutative add-right-upper-bound mult-associative mult-left-dist-add star.circ-loop-fixpoint*)
**also have** $\ldots \leq x\ ;\ L + y^* + y^*\ ;\ x\ ;\ L$
**by** (*metis add-left-isotone add-right-isotone star.left-plus-below-circ*)
**also have** $\ldots = y^* + y^*\ ;\ x\ ;\ L$
**by** (*metis add-associative add-commutative mult-associative star.circ-loop-fixpoint star.circ-reflexive star.circ-sup-one-right-unfold star-involutive*)
**finally have** $1 + (x\ ;\ L + y)\ ;\ (y^* + y^*\ ;\ x\ ;\ L) \leq y^* + y^*\ ;\ x\ ;\ L$
**by** (*metis add-commutative add-least-upper-bound add-right-divisibility star.circ-left-unfold*)
**thus** $(x\ ;\ L + y)^* \leq y^* + y^*\ ;\ x\ ;\ L$
**by** (*metis mult-right-one star-left-induct*)
**next**
**show** $y^* + y^*\ ;\ x\ ;\ L \leq (x\ ;\ L + y)^*$
**by** (*metis add-commutative add-least-upper-bound mult-associative star.circ-increasing star.circ-mult-upper-bound star.circ-sub-dist*)
**qed**

**lemma** *mult-L-add-omega*: $(x\ ;\ L + y)^\omega \leq y^\omega + y^*\ ;\ x\ ;\ L$
**proof** −
**have** *1*: $(y^*\ ;\ x\ ;\ L)^\omega \leq y^\omega + y^*\ ;\ x\ ;\ L$
**by** (*metis add-least-upper-bound add-right-isotone mult-L-omega*)
**have** $(y^*\ ;\ x\ ;\ L)^*\ ;\ y^\omega \leq y^\omega + y^*\ ;\ x\ ;\ L$
**by** (*metis add-right-isotone l40 mult-associative mult-right-isotone star-left-induct*)
**thus** *?thesis* **using** *1*
**by** (*smt add-associative add-commutative less-eq-def mult-associative omega-decompose*)
**qed**

**end**

**sublocale** *domain-omega-algebra-lattice-L* < *dL-star*!: *omega-itering* **where** *circ* = *star* **..**

**sublocale** *domain-omega-algebra-lattice-L* < *dL-star*!: *domain-itering-lattice-L* **where** *circ* = *star* **..**

**context** *domain-omega-algebra-lattice-L*

**begin**

**lemma** *d0-star-below-d0-omega*: $d(x^*\ ;\ 0) \leq d(x^\omega\ ;\ 0)$

**by** (*metis d-isotone star-zero-below-omega-zero*)

**lemma** *d0-below-d0-omega*: $d(x\ ;\ 0) \leq d(x^\omega\ ;\ 0)$
**by** (*metis d0-star-below-d0-omega d-isotone mult-left-isotone order-trans star.circ-increasing*)

**lemma** *star-d0-L*: $x^*\ ;\ d(x\ ;\ 0)\ ;\ L = x^*\ ;\ 0$
**by** (*metis dL-star.circ-d0-L*)

**lemma** *star-L-split*: $y \leq z \wedge x\ ;\ z\ ;\ L \leq x\ ;\ 0 + z\ ;\ L \rightarrow x^*\ ;\ y\ ;\ L \leq x^*\ ;\ 0 + z\ ;\ L$
**by** (*smt add-least-upper-bound add-left-isotone star.left-plus-below-circ mult-associative mult-left-dist-add mult-left-isotone order-trans star.circ-increasing star-left-induct*)

**lemma** *star-L-split-same*: $x\ ;\ y\ ;\ L \leq x\ ;\ 0 + y\ ;\ L \rightarrow x^*\ ;\ y\ ;\ L = x^*\ ;\ 0 + y\ ;\ L$
**by** (*smt add-associative add-left-zero antisym less-eq-def mult-associative mult-left-dist-add mult-left-one mult-right-subdist-add-left order-refl star-L-split star.circ-right-unfold*)

**lemma** *star-d-L-split-equal*: $d(x\ ;\ y) \leq d(y) \rightarrow x^*\ ;\ d(y)\ ;\ L = x^*\ ;\ 0 + d(y)\ ;\ L$
**by** (*metis add-right-isotone l15 less-eq-def mult-right-subdist-add-left star-L-split-same*)

**lemma** *d0-omega-mult*: $d(x^\omega\ ;\ y\ ;\ 0) = d(x^\omega\ ;\ 0)$
**by** (*smt antisym mult-associative mult-left-isotone mult-right-isotone omega-sub-vector zero-least*)

**lemma** *d-star-left-unfold*: $d(x^*\ ;\ 0) = d(x\ ;\ x^*\ ;\ 0)$
**by** (*metis dL-star.d0-circ-left-unfold*)

**lemma** *d-star-import*: $d(y)\ ;\ x \leq x\ ;\ d(y) \rightarrow d(y)\ ;\ x^* = d(y)\ ;\ (d(y)\ ;\ x)^*$
**by** (*metis dL-star.d-circ-import*)

**lemma** *d-omega-export*: $d(y)\ ;\ x \leq x\ ;\ d(y) \rightarrow d(y)\ ;\ x^\omega = (d(y)\ ;\ x)^\omega$
**apply** (*rule impI*, *rule antisym*)
**apply** (*metis d-preserves-equation omega-simulation order-refl*)
**by** (*smt less-eq-def mult-left-dist-add omega-simulation-2 omega-slide*)

**lemma** *d-omega-import*: $d(y)\ ;\ x \leq x\ ;\ d(y) \rightarrow d(y)\ ;\ x^\omega = d(y)\ ;\ (d(y)\ ;\ x)^\omega$
**by** (*metis d-idempotent d-omega-export mult-associative omega-slide*)

**lemma** *star-d-omega-top*: $x^*\ ;\ d(x^\omega)\ ;\ T = x^*\ ;\ 0 + d(x^\omega)\ ;\ T$
**apply** (*rule antisym*)
**apply** (*metis add-right-upper-bound dL-star.circ-mult-omega mult-domain-top order-trans*)
**by** (*metis add-commutative add-least-upper-bound add-left-divisibility dL-star.star-zero-below-circ-mult mult-associative star.circ-loop-fixpoint*)

**lemma** *omega-meet-L*: $x^\omega \frown L = d(x^\omega)\ ;\ L$
**by** (*metis l23 omega-vector*)

**end**

**class** *domain-semiring-lattice-apx* = *domain-semiring-lattice-L* +
  **fixes** *apx* :: $'a \Rightarrow 'a \Rightarrow bool$ (**infix** $\sqsubseteq$ *50*)
  **assumes** *apx-def*: $x \sqsubseteq y \leftrightarrow x \leq y + L \wedge d(L)\ ;\ y \leq x + d(x\ ;\ 0)\ ;\ T$

**begin**

**definition** *apx-isotone* :: $('a \Rightarrow 'a) \Rightarrow bool$
  **where** *apx-isotone* $f \leftrightarrow (\forall\, x\ y\ .\ x \sqsubseteq y \rightarrow f(x) \sqsubseteq f(y))$

**lemma** *apx-reflexive*: $x \sqsubseteq x$
  **by** (*metis add-least-upper-bound add-left-upper-bound apx-def d-plus-one mult-left-one mult-right-dist-add*)

**lemma** *apx-L-least*: $L \sqsubseteq x$
  **by** (*metis add-right-upper-bound apx-def l3 mult-right-isotone order-trans top-greatest*)

**lemma** *apx-transitive*: $x \sqsubseteq y \wedge y \sqsubseteq z \rightarrow x \sqsubseteq z$
**proof**
  **assume** *1*: $x \sqsubseteq y \wedge y \sqsubseteq z$
  **hence** *2*: $x \leq z + L$
    **by** (*smt add-associative add-commutative apx-def less-eq-def*)
  **have** $d(d(L)\ ;\ y\ ;\ 0)\ ;\ T \leq d((x + d(x\ ;\ 0)\ ;\ T)\ ;\ 0)\ ;\ T$ **using** *1*
    **by** (*metis apx-def d-isotone mult-left-isotone*)
  **also have** $\ldots \leq d(x\ ;\ 0)\ ;\ T$
    **by** (*metis add-least-upper-bound d-galois mult-left-isotone mult-right-dist-add order-refl zero-right-mult-decreasing*)
  **finally have** *3*: $d(d(L)\ ;\ y\ ;\ 0)\ ;\ T \leq d(x\ ;\ 0)\ ;\ T$
    **by** *metis*
  **have** $d(L)\ ;\ z = d(L)\ ;\ (d(L)\ ;\ z)$
    **by** (*metis d-idempotent mult-associative*)
  **also have** $\ldots \leq d(L)\ ;\ y + d(d(L)\ ;\ y\ ;\ 0)\ ;\ T$ **using** *1*
    **by** (*metis apx-def calculation d-export less-eq-meet meet-domain mult-associative mult-left-dist-add*)
  **also have** $\ldots \leq x + d(x\ ;\ 0)\ ;\ T$ **using** *1 3*
    **by** (*smt add-least-upper-bound add-right-upper-bound apx-def order-trans*)
  **finally show** $x \sqsubseteq z$ **using** *2*
    **by** (*metis apx-def*)
**qed**

**lemma** *apx-meet-L*: $y \sqsubseteq x \rightarrow x \frown L \leq y \frown L$
**proof**
  **assume** *1*: $y \sqsubseteq x$
  **have** $x \frown L = d(L)\ ;\ x \frown L$
    **by** (*metis d-restrict-equals meet-commutative meet-domain*)
  **also have** $\ldots \leq (y + d(y\ ;\ 0)\ ;\ T) \frown L$ **using** *1*
    **by** (*metis apx-def meet-left-isotone*)
  **also have** $\ldots \leq y$

**by** (*metis add-least-upper-bound l13 meet-commutative meet-domain meet-left-dist-add meet-right-upper-bound mult-right-top*)
**finally show** $x \frown L \leq y \frown L$
**by** (*metis meet-associative meet-idempotent meet-left-isotone*)
**qed**

**lemma** *apx-antisymmetric*: $x \sqsubseteq y \wedge y \sqsubseteq x \rightarrow x = y$
**by** (*metis add-same-context antisym apx-def apx-meet-L relative-equality*)

**lemma** *add-apx-left-isotone*: $x \sqsubseteq y \rightarrow x + z \sqsubseteq y + z$
**proof**
**assume** *1*: $x \sqsubseteq y$
**hence** *2*: $x + z \leq y + z + L$
**by** (*smt add-associative add-commutative add-left-isotone apx-def*)
**have** $d(L) \mathbin{;} (y + z) = d(L) \mathbin{;} y + d(L) \mathbin{;} z$
**by** (*metis mult-left-dist-add*)
**also have** $\ldots \leq d(L) \mathbin{;} y + z$
**by** (*metis add-commutative add-least-upper-bound add-right-upper-bound d-below-one mult-left-dist-add mult-left-isotone mult-left-one*)
**also have** $\ldots \leq x + d(x \mathbin{;} 0) \mathbin{;} T + z$ **using** *1*
**by** (*metis add-left-isotone apx-def*)
**also have** $\ldots \leq x + z + d((x + z) \mathbin{;} 0) \mathbin{;} T$
**by** (*smt add-associative add-commutative add-right-isotone d-isotone mult-left-isotone mult-right-subdist-add-left*)
**finally show** $x + z \sqsubseteq y + z$ **using** *2*
**by** (*metis apx-def*)
**qed**

**lemma** *add-apx-right-isotone*: $x \sqsubseteq y \rightarrow z + x \sqsubseteq z + y$
**by** (*metis add-commutative add-apx-left-isotone*)

**lemma** *add-apx-isotone*: $w \sqsubseteq y \wedge x \sqsubseteq z \rightarrow w + x \sqsubseteq y + z$
**by** (*metis add-apx-left-isotone add-apx-right-isotone apx-transitive*)

**lemma** *mult-apx-left-isotone*: $x \sqsubseteq y \rightarrow x \mathbin{;} z \sqsubseteq y \mathbin{;} z$
**proof**
**assume** *1*: $x \sqsubseteq y$
**hence** $x \mathbin{;} z \leq y \mathbin{;} z + L \mathbin{;} z$
**by** (*metis apx-def mult-left-isotone mult-right-dist-add*)
**hence** *2*: $x \mathbin{;} z \leq y \mathbin{;} z + L$
**by** (*metis add-commutative add-left-isotone l40 order-trans*)
**have** $d(L) \mathbin{;} y \mathbin{;} z \leq x \mathbin{;} z + d(x \mathbin{;} 0) \mathbin{;} T \mathbin{;} z$ **using** *1*
**by** (*metis apx-def mult-left-isotone mult-right-dist-add*)
**also have** $\ldots \leq x \mathbin{;} z + d(x \mathbin{;} z \mathbin{;} 0) \mathbin{;} T$
**by** (*metis add-right-isotone d-isotone mult-associative mult-isotone mult-right-isotone top-greatest zero-least*)
**finally show** $x \mathbin{;} z \sqsubseteq y \mathbin{;} z$ **using** *2*
**by** (*metis apx-def mult-associative*)
**qed**

**lemma** *mult-apx-right-isotone*: $x \sqsubseteq y \rightarrow z \mathbin{;} x \sqsubseteq z \mathbin{;} y$

**proof**
  **assume** *1*: $x \sqsubseteq y$
  **hence** $z\ ;\ x \leq z\ ;\ y + z\ ;\ L$
    **by** (*metis apx-def mult-left-dist-add mult-right-isotone*)
  **also have** $\ldots \leq z\ ;\ y + z\ ;\ 0 + L$
    **by** (*metis add-associative add-right-isotone l14*)
  **finally have** *2*: $z\ ;\ x \leq z\ ;\ y + L$
    **by** (*metis add-right-zero mult-left-dist-add*)
  **have** $d(L)\ ;\ z\ ;\ y \leq z\ ;\ d(L)\ ;\ y$
    **by** (*metis l2 mult-left-isotone*)
  **also have** $\ldots \leq z\ ;\ (x + d(x\ ;\ 0)\ ;\ T)$ **using** *1*
    **by** (*metis apx-def mult-associative mult-right-isotone*)
  **also have** $\ldots = z\ ;\ x + z\ ;\ d(x\ ;\ 0)\ ;\ T$
    **by** (*metis mult-associative mult-left-dist-add*)
  **also have** $\ldots \leq z\ ;\ x + d(z\ ;\ x\ ;\ 0)\ ;\ T$
    **by** (*metis add-right-isotone mult-associative mult-domain-top*)
  **finally show** $z\ ;\ x \sqsubseteq z\ ;\ y$ **using** *2*
    **by** (*metis apx-def mult-associative*)
**qed**

**lemma** *mult-apx-isotone*: $w \sqsubseteq y \wedge x \sqsubseteq z \rightarrow w\ ;\ x \sqsubseteq y\ ;\ z$
  **by** (*metis apx-transitive mult-apx-left-isotone mult-apx-right-isotone*)

**lemma** *meet-L-apx-isotone*: $x \sqsubseteq y \rightarrow x \frown L \sqsubseteq y \frown L$
  **by** (*smt add-absorb add-commutative apx-def apx-meet-L d-restrict-equals l20 meet-commutative meet-domain meet-left-upper-bound*)

**definition** *is-apx-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** *is-apx-prefixpoint* $f\ x \leftrightarrow f(x) \sqsubseteq x$
**definition** *is-apx-least-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** *is-apx-least-fixpoint* $f\ x \leftrightarrow f(x) = x \wedge (\forall\, y\ .\ f(y) = y \rightarrow x \sqsubseteq y)$
**definition** *is-apx-least-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a \Rightarrow bool$ **where** *is-apx-least-prefixpoint* $f\ x \leftrightarrow f(x) \sqsubseteq x \wedge (\forall\, y\ .\ f(y) \sqsubseteq y \rightarrow x \sqsubseteq y)$
**definition** *has-apx-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** *has-apx-prefixpoint* $f \leftrightarrow (\exists\, x\ .\ is\text{-}apx\text{-}prefixpoint\ f\ x)$
**definition** *has-apx-least-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** *has-apx-least-fixpoint* $f \leftrightarrow (\exists\, x\ .\ is\text{-}apx\text{-}least\text{-}fixpoint\ f\ x)$
**definition** *has-apx-least-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow bool$ **where** *has-apx-least-prefixpoint* $f \leftrightarrow (\exists\, x\ .\ is\text{-}apx\text{-}least\text{-}prefixpoint\ f\ x)$
**definition** *the-apx-least-fixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a\ (\xi\ \text{-}\ [201]\ 200)$ **where** $\xi\ \ f = (THE\ x\ .\ is\text{-}apx\text{-}least\text{-}fixpoint\ f\ x)$
**definition** *the-apx-least-prefixpoint* :: $('a \Rightarrow 'a) \Rightarrow 'a\ (p\xi\ \text{-}\ [201]\ 200)$ **where** $p\xi\ f = (THE\ x\ .\ is\text{-}apx\text{-}least\text{-}prefixpoint\ f\ x)$

**lemma** *apx-least-fixpoint-unique*: *has-apx-least-fixpoint* $f \rightarrow (\exists!\, x\ .\ is\text{-}apx\text{-}least\text{-}fixpoint\ f\ x)$
  **by** (*smt apx-antisymmetric has-apx-least-fixpoint-def is-apx-least-fixpoint-def*)

**lemma** *apx-least-prefixpoint-unique*: *has-apx-least-prefixpoint* $f \rightarrow (\exists!\, x\ .\ is\text{-}apx\text{-}least\text{-}prefixpoint\ f\ x)$
  **by** (*smt apx-antisymmetric has-apx-least-prefixpoint-def is-apx-least-prefixpoint-def*)

**lemma** *apx-least-fixpoint*: *has-apx-least-fixpoint* $f \rightarrow$ *is-apx-least-fixpoint* $f\ (\xi\ f)$
**proof**
  **assume** *has-apx-least-fixpoint f*
  **hence** *is-apx-least-fixpoint f* $(THE\ x\ .\ is\text{-}apx\text{-}least\text{-}fixpoint\ f\ x)$
    **by** (*smt apx-least-fixpoint-unique theI'*)
  **thus** *is-apx-least-fixpoint f* $(\xi\ f)$

**by** (*simp add*: *is-apx-least-fixpoint-def the-apx-least-fixpoint-def*)
**qed**

**lemma** *apx-least-prefixpoint*: *has-apx-least-prefixpoint f* ⟶ *is-apx-least-prefixpoint f* (*pξ f*)
**proof**
**assume** *has-apx-least-prefixpoint f*
**hence** *is-apx-least-prefixpoint f* (*THE x* . *is-apx-least-prefixpoint f x*)
**by** (*smt apx-least-prefixpoint-unique theI'*)
**thus** *is-apx-least-prefixpoint f* (*pξ f*)
**by** (*simp add*: *is-apx-least-prefixpoint-def the-apx-least-prefixpoint-def*)
**qed**

**lemma** *apx-least-fixpoint-same*: *is-apx-least-fixpoint f x* ⟶ *x* = *ξ f*
**by** (*metis apx-least-fixpoint apx-least-fixpoint-unique has-apx-least-fixpoint-def*)

**lemma** *apx-least-prefixpoint-same*: *is-apx-least-prefixpoint f x* ⟶ *x* = *pξ f*
**by** (*metis apx-least-prefixpoint apx-least-prefixpoint-unique has-apx-least-prefixpoint-def*)

**lemma** *apx-least-fixpoint-char*: *is-apx-least-fixpoint f x* ⟷ *has-apx-least-fixpoint f* ∧ *x* = *ξ f*
**by** (*metis apx-least-fixpoint-same has-apx-least-fixpoint-def*)

**lemma** *apx-least-prefixpoint-char*: *is-apx-least-prefixpoint f x* ⟷ *has-apx-least-prefixpoint f* ∧ *x* = *pξ f*
**by** (*metis apx-least-prefixpoint-same has-apx-least-prefixpoint-def*)

**lemma** *apx-least-prefixpoint-fixpoint*: *has-apx-least-prefixpoint f* ∧ *apx-isotone f* ⟶ *is-apx-least-fixpoint f* (*pξ f*)
**by** (*smt apx-antisymmetric apx-isotone-def apx-reflexive is-apx-least-fixpoint-def is-apx-least-prefixpoint-def apx-least-prefixpoint*)

**lemma** *pxi-xi*: *has-apx-least-prefixpoint f* ∧ *apx-isotone f* ⟶ *pξ f* = *ξ f*
**by** (*smt has-apx-least-fixpoint-def is-apx-least-fixpoint-def apx-least-fixpoint-unique apx-least-prefixpoint-fixpoint apx-least-fixpoint*)

**definition** *lifted-apx-less-eq* :: ('*a* ⇒ '*a*) ⇒ ('*a* ⇒ '*a*) ⇒ *bool* ((- ⊑⊑ -) [*51*, *51*] *50*)
**where** *f* ⊑⊑ *g* ⟷ (∀ *x* . *f*(*x*) ⊑ *g*(*x*))

**lemma** *lifted-reflexive*: *f* = *g* ⟶ *f* ⊑⊑ *g*
**by** (*metis lifted-apx-less-eq-def apx-reflexive*)

**lemma** *lifted-transitive*: *f* ⊑⊑ *g* ∧ *g* ⊑⊑ *h* ⟶ *f* ⊑⊑ *h*
**by** (*smt lifted-apx-less-eq-def apx-transitive*)

**lemma** *lifted-antisymmetric*: *f* ⊑⊑ *g* ∧ *g* ⊑⊑ *f* ⟶ *f* = *g*
**by** (*metis apx-antisymmetric ext lifted-apx-less-eq-def*)

**lemma** *pxi-isotone*: *has-apx-least-prefixpoint f* ∧ *has-apx-least-prefixpoint g* ∧ *f* ⊑⊑ *g* ⟶ *pξ f* ⊑ *pξ g*
**by** (*metis is-apx-least-prefixpoint-def apx-transitive apx-least-prefixpoint lifted-apx-less-eq-def*)

**lemma** *xi-isotone*: *has-apx-least-prefixpoint f* ∧ *has-apx-least-prefixpoint g* ∧ *apx-isotone f* ∧ *apx-isotone g* ∧ *f* ⊑⊑ *g* ⟶ *ξ f* ⊑ *ξ g*
**by** (*metis pxi-isotone pxi-xi*)

**lemma** *xi-square*: *apx-isotone f* $\wedge$ *has-apx-least-fixpoint f* $\wedge$ *has-apx-least-fixpoint* $(f \circ f) \rightarrow \xi\ f = \xi\ (f \circ f)$
**by** (*metis apx-antisymmetric is-apx-least-fixpoint-def apx-isotone-def apx-least-fixpoint-char apx-least-fixpoint-unique o-apply*)

**lemma** *mu-below-xi*: *has-least-fixpoint f* $\wedge$ *has-apx-least-fixpoint f* $\rightarrow \mu\ f \leq \xi\ f$
**by** (*metis apx-least-fixpoint is-apx-least-fixpoint-def is-least-fixpoint-def least-fixpoint*)

**lemma** *xi-below-nu*: *has-greatest-fixpoint f* $\wedge$ *has-apx-least-fixpoint f* $\rightarrow \xi\ f \leq \nu\ f$
**by** (*metis apx-least-fixpoint greatest-fixpoint is-apx-least-fixpoint-def is-greatest-fixpoint-def*)

**lemma** *xi-apx-below-mu*: *has-least-fixpoint f* $\wedge$ *has-apx-least-fixpoint f* $\rightarrow \xi\ f \sqsubseteq \mu\ f$
**by** (*metis apx-least-fixpoint is-apx-least-fixpoint-def is-least-fixpoint-def least-fixpoint*)

**lemma** *xi-apx-below-nu*: *has-greatest-fixpoint f* $\wedge$ *has-apx-least-fixpoint f* $\rightarrow \xi\ f \sqsubseteq \nu\ f$
**by** (*metis apx-least-fixpoint greatest-fixpoint is-apx-least-fixpoint-def is-greatest-fixpoint-def*)

**definition** *is-apx-meet* :: $'a \Rightarrow 'a \Rightarrow 'a \Rightarrow bool$ **where** *is-apx-meet x y z* $\leftrightarrow z \sqsubseteq x \wedge z \sqsubseteq y \wedge (\forall w\ .\ w \sqsubseteq x \wedge w \sqsubseteq y \rightarrow w \sqsubseteq z)$
**definition** *has-apx-meet* :: $'a \Rightarrow 'a \Rightarrow bool$ **where** *has-apx-meet x y* $\leftrightarrow (\exists z\ .$ *is-apx-meet x y z*$)$
**definition** *the-apx-meet* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixl** $\sqcap$ *66*) **where** $x \sqcap y = (THE\ z\ .$ *is-apx-meet x y z*$)$

**lemma** *apx-meet-unique*: *has-apx-meet x y* $\rightarrow (\exists !z\ .$ *is-apx-meet x y z*$)$
**by** (*smt apx-antisymmetric has-apx-meet-def is-apx-meet-def*)

**lemma** *apx-meet*: *has-apx-meet x y* $\rightarrow$ *is-apx-meet x y* $(x \sqcap y)$
**proof**
**assume** *has-apx-meet x y*
**hence** *is-apx-meet x y* $(THE\ z\ .$ *is-apx-meet x y z*$)$
**by** (*smt apx-meet-unique theI'*)
**thus** *is-apx-meet x y* $(x \sqcap y)$
**by** (*simp add*: *is-apx-meet-def the-apx-meet-def*)
**qed**

**lemma** *apx-greatest-lower-bound*: *has-apx-meet x y* $\rightarrow (w \sqsubseteq x \wedge w \sqsubseteq y \leftrightarrow w \sqsubseteq x \sqcap y)$
**by** (*smt apx-meet apx-transitive is-apx-meet-def*)

**lemma** *apx-meet-same*: *is-apx-meet x y z* $\rightarrow z = x \sqcap y$
**by** (*metis apx-meet apx-meet-unique has-apx-meet-def*)

**lemma** *apx-meet-char*: *is-apx-meet x y z* $\leftrightarrow$ *has-apx-meet x y* $\wedge\ z = x \sqcap y$
**by** (*metis apx-meet-same has-apx-meet-def*)

**definition** *xi-apx-meet* :: $('a \Rightarrow 'a) \Rightarrow bool$
**where** *xi-apx-meet f* $\leftrightarrow$ *has-apx-least-fixpoint f* $\wedge$ *has-apx-meet* $(\mu\ f)\ (\nu\ f) \wedge \xi\ f = \mu\ f \sqcap \nu\ f$

**definition** *xi-mu-nu* :: $('a \Rightarrow 'a) \Rightarrow bool$
**where** *xi-mu-nu f* $\leftrightarrow$ *has-apx-least-fixpoint f* $\wedge\ \xi\ f = \mu\ f + (\nu\ f \frown L)$

**definition** *nu-below-mu-nu* :: $('a \Rightarrow 'a) \Rightarrow bool$
  **where** *nu-below-mu-nu f* $\leftrightarrow$ $d(L)\ ;\ \nu\ f \leq \mu\ f + (\nu\ f \frown L) + d(\nu\ f\ ;\ 0)\ ;\ T$

**definition** *nu-below-mu-nu-2* :: $('a \Rightarrow 'a) \Rightarrow bool$
  **where** *nu-below-mu-nu-2 f* $\leftrightarrow$ $d(L)\ ;\ \nu\ f \leq \mu\ f + (\nu\ f \frown L) + d((\mu\ f + (\nu\ f \frown L))\ ;\ 0)\ ;\ T$

**definition** *mu-nu-apx-nu* :: $('a \Rightarrow 'a) \Rightarrow bool$
  **where** *mu-nu-apx-nu f* $\leftrightarrow$ $\mu\ f + (\nu\ f \frown L) \sqsubseteq \nu\ f$

**definition** *mu-nu-apx-meet* :: $('a \Rightarrow 'a) \Rightarrow bool$
  **where** *mu-nu-apx-meet f* $\leftrightarrow$ *has-apx-meet* $(\mu\ f)\ (\nu\ f) \wedge \mu\ f \sqcap \nu\ f = \mu\ f + (\nu\ f \frown L)$

**definition** *apx-meet-below-nu* :: $('a \Rightarrow 'a) \Rightarrow bool$
  **where** *apx-meet-below-nu f* $\leftrightarrow$ *has-apx-meet* $(\mu\ f)\ (\nu\ f) \wedge \mu\ f \sqcap \nu\ f \leq \nu\ f$

**lemma** *mu-below-l*: $\mu\ f \leq \mu\ f + (\nu\ f \frown L)$
  **by** (*metis add-left-upper-bound*)

**lemma** *l-below-nu*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\rightarrow$ $\mu\ f + (\nu\ f \frown L) \leq \nu\ f$
  **by** (*metis add-least-upper-bound meet-left-upper-bound mu-below-nu*)

**lemma** *n-l-nu*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\rightarrow$ $(\mu\ f + (\nu\ f \frown L)) \frown L = \nu\ f \frown L$
  **by** (*smt add-commutative add-left-dist-meet less-eq-def meet-absorb meet-associative meet-commutative mu-below-nu*)

**lemma** *l-apx-mu*: $\mu\ f + (\nu\ f \frown L) \sqsubseteq \mu\ f$
  **by** (*metis add-right-isotone apx-def meet-absorb meet-domain-top meet-least-upper-bound meet-left-upper-bound meet-right-upper-bound*)

**lemma** *nu-below-mu-nu-nu-below-mu-nu-2*: *nu-below-mu-nu f* $\rightarrow$ *nu-below-mu-nu-2 f*
**proof**
  **assume** *1*: *nu-below-mu-nu f*
  **have** $d(L)\ ;\ \nu\ f = d(L)\ ;\ (d(L)\ ;\ \nu\ f)$
    **by** (*metis d-idempotent mult-associative*)
  **also have** ... $\leq d(L)\ ;\ (\mu\ f + (\nu\ f \frown L) + d(\nu\ f\ ;\ 0)\ ;\ T)$ **using** *1*
    **by** (*metis mult-right-isotone nu-below-mu-nu-def*)
  **also have** ... $= d(L)\ ;\ (\mu\ f + (\nu\ f \frown L)) + d(L)\ ;\ d(\nu\ f\ ;\ 0)\ ;\ T$
    **by** (*metis mult-associative mult-left-dist-add*)
  **also have** ... $\leq \mu\ f + (\nu\ f \frown L) + d(L)\ ;\ d(\nu\ f\ ;\ 0)\ ;\ T$
    **by** (*metis add-left-isotone meet-domain-top meet-left-upper-bound*)
  **also have** ... $= \mu\ f + (\nu\ f \frown L) + d(d(\nu\ f\ ;\ 0)\ ;\ L)\ ;\ T$
    **by** (*smt d-commutative d-export*)
  **also have** ... $= \mu\ f + (\nu\ f \frown L) + d((\nu\ f \frown L)\ ;\ 0)\ ;\ T$
    **by** (*metis l51*)
  **also have** ... $\leq \mu\ f + (\nu\ f \frown L) + d((\mu\ f + (\nu\ f \frown L))\ ;\ 0)\ ;\ T$
    **by** (*metis add-right-isotone add-right-upper-bound d-dist-add mult-right-dist-add*)
  **finally show** *nu-below-mu-nu-2 f*
    **by** (*metis nu-below-mu-nu-2-def*)
**qed**

**lemma** *nu-below-mu-nu-2-nu-below-mu-nu*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *nu-below-mu-nu-2 f* $\longrightarrow$ *nu-below-mu-nu f*
**proof**
  **assume** *1*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *nu-below-mu-nu-2 f*
  **hence** $d(L)\ ;\ \nu\ f \leq \mu\ f + (\nu\ f \frown L) + d((\mu\ f + (\nu\ f \frown L))\ ;\ 0)\ ;\ T$
    **by** (*metis nu-below-mu-nu-2-def*)
  **also have** $\ldots \leq \mu\ f + (\nu\ f \frown L) + d(\nu\ f\ ;\ 0)\ ;\ T$ **using** *1*
    **by** (*smt add-absorb add-associative add-commutative d-dist-add l-below-nu less-eq-def meet-absorb mult-right-dist-add*)
  **finally show** *nu-below-mu-nu f*
    **by** (*metis nu-below-mu-nu-def*)
**qed**

**lemma** *nu-below-mu-nu-equivalent*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\longrightarrow$ (*nu-below-mu-nu f* $\leftrightarrow$ *nu-below-mu-nu-2 f*)
  **by** (*metis nu-below-mu-nu-2-nu-below-mu-nu nu-below-mu-nu-nu-below-mu-nu-2*)

**lemma** *nu-below-mu-nu-2-mu-nu-apx-nu*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *nu-below-mu-nu-2 f* $\longrightarrow$ *mu-nu-apx-nu f*
**proof**
  **assume** *1*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *nu-below-mu-nu-2 f*
  **hence** $\mu\ f + (\nu\ f \frown L) \leq \nu\ f + L$
    **by** (*metis add-commutative add-right-upper-bound l-below-nu order-trans*)
  **thus** *mu-nu-apx-nu f* **using** *1*
    **by** (*metis apx-def mu-nu-apx-nu-def nu-below-mu-nu-2-def*)
**qed**

**lemma** *mu-nu-apx-nu-mu-nu-apx-meet*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *mu-nu-apx-nu f* $\longrightarrow$ *mu-nu-apx-meet f*
**proof**
  **let** *?l* $= \mu\ f + (\nu\ f \frown L)$
  **assume** *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *mu-nu-apx-nu f*
  **hence** *is-apx-meet* $(\mu\ f)\ (\nu\ f)$ *?l*
    **by** (*smt add-apx-left-isotone add-commutative apx-meet-L is-apx-meet-def l-apx-mu less-eq-def meet-least-upper-bound mu-nu-apx-nu-def*)
  **thus** *mu-nu-apx-meet f*
    **by** (*smt apx-meet-char mu-nu-apx-meet-def*)
**qed**

**lemma** *mu-nu-apx-meet-apx-meet-below-nu*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *mu-nu-apx-meet f* $\longrightarrow$ *apx-meet-below-nu f*
  **by** (*metis apx-meet-below-nu-def l-below-nu mu-nu-apx-meet-def*)

**lemma** *apx-meet-below-nu-nu-below-mu-nu-2*: *apx-meet-below-nu f* $\longrightarrow$ *nu-below-mu-nu-2 f*
**proof** $-$
  **let** *?l* $= \mu\ f + (\nu\ f \frown L)$
  **have** $\forall\, m\ .\ m \sqsubseteq \mu\ f \wedge m \sqsubseteq \nu\ f \wedge m \leq \nu\ f \longrightarrow d(L)\ ;\ \nu\ f \leq$ *?l* $+ d($*?l* $;\ 0)\ ;\ T$
  **proof**
    **fix** $m$
    **show** $m \sqsubseteq \mu\ f \wedge m \sqsubseteq \nu\ f \wedge m \leq \nu\ f \longrightarrow d(L)\ ;\ \nu\ f \leq$ *?l* $+ d($*?l* $;\ 0)\ ;\ T$
    **proof**
      **assume** *1*: $m \sqsubseteq \mu\ f \wedge m \sqsubseteq \nu\ f \wedge m \leq \nu\ f$
      **hence** $m \leq$ *?l*

```
      by (smt add-commutative add-left-dist-meet add-left-upper-bound apx-def less-eq-meet meet-least-upper-bound)
    hence m + d(m ; 0) ; T ≤ ?l + d(?l ; 0) ; T
      by (metis add-isotone d-dist-add less-eq-def mult-right-dist-add)
    thus d(L) ; ν f ≤ ?l + d(?l ; 0) ; T using 1
      by (smt apx-def order-trans)
  qed
 qed
 thus ?thesis
  by (smt apx-meet-below-nu-def apx-meet-same apx-meet-unique is-apx-meet-def nu-below-mu-nu-2-def)
qed

lemma has-apx-least-fixpoint-xi-apx-meet: has-least-fixpoint f ∧ has-greatest-fixpoint f ∧ has-apx-least-fixpoint f → xi-apx-meet f
proof
 assume 1: has-least-fixpoint f ∧ has-greatest-fixpoint f ∧ has-apx-least-fixpoint f
 hence ∀ w . w ⊑ μ f ∧ w ⊑ ν f → w ⊑ ξ f
  by (smt add-left-isotone apx-def mu-below-xi mult-right-isotone order-trans xi-below-nu)
 thus xi-apx-meet f using 1
  by (smt apx-meet-char is-apx-meet-def xi-apx-below-mu xi-apx-below-nu xi-apx-meet-def)
qed

lemma xi-apx-meet-apx-meet-below-nu: has-greatest-fixpoint f ∧ xi-apx-meet f → apx-meet-below-nu f
 by (metis apx-meet-below-nu-def xi-apx-meet-def xi-below-nu)

lemma apx-meet-below-nu-xi-mu-nu: has-least-fixpoint f ∧ has-greatest-fixpoint f ∧ isotone f ∧ apx-isotone f ∧ apx-meet-below-nu f → xi-mu-nu f
proof
 let ?l = μ f + (ν f ⌢ L)
 let ?m = μ f ⊓ ν f
 assume 1: has-least-fixpoint f ∧ has-greatest-fixpoint f ∧ isotone f ∧ apx-isotone f ∧ apx-meet-below-nu f
 hence 2: ?m = ?l
  by (metis apx-meet-below-nu-nu-below-mu-nu-2 mu-nu-apx-meet-def mu-nu-apx-nu-mu-nu-apx-meet nu-below-mu-nu-2-mu-nu-apx-nu)
 have 3: ?l ≤ f(?l) + L
 proof −
  have ?l ≤ μ f + L
   by (metis add-right-isotone meet-right-upper-bound)
  also have ... = f(μ f) + L using 1
   by (metis is-least-fixpoint-def least-fixpoint)
  also have ... ≤ f(?l) + L using 1
   by (metis add-left-isotone add-left-upper-bound isotone-def)
  finally show ?l ≤ f(?l) + L
   by metis
 qed
 have d(L) ; f(?l) ≤ ?l + d(?l ; 0) ; T
 proof −
  have d(L) ; f(?l) ≤ d(L) ; f(ν f) using 1 2
   by (metis apx-meet-below-nu-def isotone-def mult-right-isotone)
  also have ... = d(L) ; ν f using 1
   by (metis greatest-fixpoint is-greatest-fixpoint-def)
```

```
    also have ... ≤ ?l + d(?l ; 0) ; T using 1
      by (metis apx-meet-below-nu-nu-below-mu-nu-2 nu-below-mu-nu-2-def)
    finally show d(L) ; f(?l) ≤ ?l + d(?l ; 0) ; T
      by metis
  qed
  hence 4: ?l ⊑ f(?l) using 3
    by (metis apx-def)
  have 5: f(?l) ⊑ μ f
  proof −
    have ?l ⊑ μ f
      by (metis l-apx-mu)
    thus f(?l) ⊑ μ f using 1
      by (metis apx-isotone-def is-least-fixpoint-def least-fixpoint)
  qed
  have 6: f(?l) ⊑ ν f
  proof −
    have ?l ⊑ ν f using 1 2
      by (metis apx-greatest-lower-bound apx-meet-below-nu-def apx-reflexive)
    thus f(?l) ⊑ ν f using 1
      by (metis apx-isotone-def greatest-fixpoint is-greatest-fixpoint-def)
  qed
  hence f(?l) ⊑ ?l using 1 2 5
    by (metis apx-greatest-lower-bound apx-meet-below-nu-def)
  hence 7: f(?l) = ?l using 4
    by (metis apx-antisymmetric)
  have ∀ y . f(y) = y → ?l ⊑ y
  proof
    fix y
    show f(y) = y → ?l ⊑ y
    proof
      assume 8: f(y) = y
      hence 9: ?l ≤ y + L using 1
        by (metis add-isotone is-least-fixpoint-def least-fixpoint meet-right-upper-bound)
      have y ≤ ν f using 1 8
        by (metis greatest-fixpoint is-greatest-fixpoint-def)
      hence d(L) ; y ≤ ?l + d(?l ; 0) ; T using 4 6
        by (smt apx-def apx-transitive mult-right-isotone order-trans)
      thus ?l ⊑ y using 9
        by (metis apx-def)
    qed
  qed
  thus xi-mu-nu f using 1 2 7
    by (smt apx-least-fixpoint-same has-apx-least-fixpoint-def is-apx-least-fixpoint-def xi-mu-nu-def)
qed

lemma xi-mu-nu-has-apx-least-fixpoint: xi-mu-nu f → has-apx-least-fixpoint f
  by (metis xi-mu-nu-def)
```

**lemma** *nu-below-mu-nu-xi-mu-nu*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *isotone f* $\wedge$ *apx-isotone f* $\wedge$ *nu-below-mu-nu f* $\longrightarrow$ *xi-mu-nu f*
**by** (*metis apx-meet-below-nu-xi-mu-nu mu-nu-apx-meet-apx-meet-below-nu mu-nu-apx-nu-mu-nu-apx-meet nu-below-mu-nu-nu-below-mu-nu-2 nu-below-mu-nu-2-mu-nu-apx-nu*)

**lemma** *xi-mu-nu-nu-below-mu-nu*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *xi-mu-nu f* $\longrightarrow$ *nu-below-mu-nu f*
**by** (*metis apx-meet-below-nu-nu-below-mu-nu-2 has-apx-least-fixpoint-xi-apx-meet nu-below-mu-nu-2-nu-below-mu-nu xi-apx-meet-apx-meet-below-nu xi-mu-nu-has-apx-least-fixpoint*)

**definition** *xi-mu-nu-L* :: $('a \Rightarrow 'a) \Rightarrow$ *bool*
**where** *xi-mu-nu-L f* $\leftrightarrow$ *has-apx-least-fixpoint f* $\wedge$ $\xi\ f = \mu\ f\ +\ d(\nu\ f\ ;\ 0)\ ;\ L$

**definition** *nu-below-mu-nu-L* :: $('a \Rightarrow 'a) \Rightarrow$ *bool*
**where** *nu-below-mu-nu-L f* $\leftrightarrow$ $d(L)\ ;\ \nu\ f \leq \mu\ f\ +\ d(\nu\ f\ ;\ 0)\ ;\ T$

**definition** *mu-nu-apx-nu-L* :: $('a \Rightarrow 'a) \Rightarrow$ *bool*
**where** *mu-nu-apx-nu-L f* $\leftrightarrow$ $\mu\ f\ +\ d(\nu\ f\ ;\ 0)\ ;\ L \sqsubseteq \nu\ f$

**definition** *mu-nu-apx-meet-L* :: $('a \Rightarrow 'a) \Rightarrow$ *bool*
**where** *mu-nu-apx-meet-L f* $\leftrightarrow$ *has-apx-meet* $(\mu\ f)\ (\nu\ f) \wedge \mu\ f \sqcap \nu\ f = \mu\ f\ +\ d(\nu\ f\ ;\ 0)\ ;\ L$

**lemma** *n-below-l*: $x\ +\ d(y\ ;\ 0)\ ;\ L \leq x\ +\ (y \frown L)$
**by** (*metis add-right-isotone d-mult-below l13 meet-least-upper-bound*)

**lemma** *n-equal-l*: *nu-below-mu-nu-L f* $\longrightarrow$ $\mu\ f\ +\ d(\nu\ f\ ;\ 0)\ ;\ L = \mu\ f\ +\ (\nu\ f \frown L)$
**proof**
**assume** *nu-below-mu-nu-L f*
**hence** $\nu\ f \frown L \leq (\mu\ f\ +\ d(\nu\ f\ ;\ 0)\ ;\ T) \frown L$
**by** (*smt meet-associative meet-intro-domain meet-right-divisibility nu-below-mu-nu-L-def*)
**also have** $\ldots \leq \mu\ f\ +\ d(\nu\ f\ ;\ 0)\ ;\ L$
**by** (*smt add-left-dist-meet add-right-divisibility meet-commutative meet-domain-top meet-left-isotone*)
**finally have** $\mu\ f\ +\ (\nu\ f \frown L) \leq \mu\ f\ +\ d(\nu\ f\ ;\ 0)\ ;\ L$
**by** (*metis add-least-upper-bound add-left-upper-bound*)
**thus** $\mu\ f\ +\ d(\nu\ f\ ;\ 0)\ ;\ L = \mu\ f\ +\ (\nu\ f \frown L)$
**by** (*metis antisym n-below-l*)
**qed**

**lemma** *nu-below-mu-nu-L-nu-below-mu-nu*: *nu-below-mu-nu-L f* $\longrightarrow$ *nu-below-mu-nu f*
**by** (*metis add-associative add-right-top mult-left-dist-add n-equal-l nu-below-mu-nu-L-def nu-below-mu-nu-def*)

**lemma** *nu-below-mu-nu-L-xi-mu-nu-L*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *isotone f* $\wedge$ *apx-isotone f* $\wedge$ *nu-below-mu-nu-L f* $\longrightarrow$ *xi-mu-nu-L f*
**by** (*metis n-equal-l nu-below-mu-nu-L-nu-below-mu-nu nu-below-mu-nu-xi-mu-nu xi-mu-nu-L-def xi-mu-nu-def*)

**lemma** *nu-below-mu-nu-L-mu-nu-apx-nu-L*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *nu-below-mu-nu-L f* $\longrightarrow$ *mu-nu-apx-nu-L f*
**by** (*metis mu-nu-apx-nu-L-def mu-nu-apx-nu-def n-equal-l nu-below-mu-nu-2-mu-nu-apx-nu nu-below-mu-nu-L-nu-below-mu-nu nu-below-mu-nu-nu-below-mu-nu-2*)

**lemma** *nu-below-mu-nu-L-mu-nu-apx-meet-L*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *nu-below-mu-nu-L f* $\longrightarrow$ *mu-nu-apx-meet-L f*
**by** (*metis mu-nu-apx-meet-L-def mu-nu-apx-meet-def mu-nu-apx-nu-mu-nu-apx-meet n-equal-l nu-below-mu-nu-2-mu-nu-apx-nu nu-below-mu-nu-L-nu-below-mu-nu nu-below-mu-nu-nu-below-mu-nu-2*)

**lemma** *mu-nu-apx-nu-L-nu-below-mu-nu-L*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *mu-nu-apx-nu-L f* $\rightarrow$ *nu-below-mu-nu-L f*
**proof**
**let** *?n* = $\mu\ f + d(\nu\ f\ ;\ 0)\ ;\ L$
**let** *?l* = $\mu\ f + (\nu\ f \frown L)$
**assume** *1*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *mu-nu-apx-nu-L f*
**hence** $d(L)\ ;\ \nu\ f \leq$ *?n* $+ d($*?n* $;\ 0)\ ;\ T$
**by** (*metis apx-def mu-nu-apx-nu-L-def*)
**also have** ... $\leq$ *?n* $+ d($*?l* $;\ 0)\ ;\ T$
**by** (*metis add-right-isotone d-isotone mult-left-isotone n-below-l*)
**also have** ... $\leq$ *?n* $+ d(\nu\ f\ ;\ 0)\ ;\ T$ **using** *1*
**by** (*metis add-right-isotone d-isotone l-below-nu mult-left-isotone*)
**finally show** *nu-below-mu-nu-L f*
**by** (*metis add-associative add-right-top mult-left-dist-add nu-below-mu-nu-L-def*)
**qed**

**lemma** *xi-mu-nu-L-mu-nu-apx-nu-L*: *has-greatest-fixpoint f* $\wedge$ *xi-mu-nu-L f* $\rightarrow$ *mu-nu-apx-nu-L f*
**by** (*metis mu-nu-apx-nu-L-def xi-apx-below-nu xi-mu-nu-L-def*)

**lemma** *mu-nu-apx-meet-L-mu-nu-apx-nu-L*: *mu-nu-apx-meet-L f* $\rightarrow$ *mu-nu-apx-nu-L f*
**by** (*metis apx-meet-same has-apx-meet-def is-apx-meet-def mu-nu-apx-meet-L-def mu-nu-apx-nu-L-def*)

**lemma** *xi-mu-nu-L-nu-below-mu-nu-L*: *has-least-fixpoint f* $\wedge$ *has-greatest-fixpoint f* $\wedge$ *xi-mu-nu-L f* $\rightarrow$ *nu-below-mu-nu-L f*
**by** (*metis mu-nu-apx-nu-L-nu-below-mu-nu-L xi-mu-nu-L-mu-nu-apx-nu-L*)

**end**

**class** *itering-apx* = *domain-itering-lattice-L* + *domain-semiring-lattice-apx*

**begin**

**lemma** *circ-apx-isotone*: $x \sqsubseteq y \rightarrow x^\circ \sqsubseteq y^\circ$
**proof**
**assume** $x \sqsubseteq y$
**hence** *1*: $x \leq y + L \wedge d(L)\ ;\ y \leq x + d(x\ ;\ 0)\ ;\ T$
**by** (*metis apx-def*)
**have** $d(L)\ ;\ y^\circ \leq (d(L)\ ;\ y)^\circ$
**by** (*smt circ-reflexive circ-transitive-equal d-below-one d-circ-import l2 mult-left-isotone order-trans*)
**also have** ... $\leq x^\circ\ ;\ (d(x\ ;\ 0)\ ;\ T\ ;\ x^\circ)^\circ$ **using** *1*
**by** (*metis circ-add-1 circ-isotone*)
**also have** ... $= x^\circ + x^\circ\ ;\ d(x\ ;\ 0)\ ;\ T$
**by** (*metis circ-left-top mult-associative mult-left-dist-add mult-right-one mult-top-circ*)
**also have** ... $\leq x^\circ + d(x^\circ\ ;\ x\ ;\ 0)\ ;\ T$
**by** (*metis add-right-isotone mult-associative mult-domain-top*)
**finally have** *2*: $d(L)\ ;\ y^\circ \leq x^\circ + d(x^\circ\ ;\ 0)\ ;\ T$
**by** (*metis circ-plus-same d0-circ-left-unfold*)
**have** $x^\circ \leq y^\circ\ ;\ L^\circ$ **using** *1*
**by** (*metis circ-add-1 circ-back-loop-fixpoint circ-isotone l40 less-eq-def mult-associative*)

**also have** ... = $y^\circ + y^\circ$ ; $L$
**by** (*metis add-commutative circ-L mult-left-dist-add mult-right-one*)
**also have** ... ≤ $y^\circ + y^\circ$ ; $0 + L$
**by** (*metis add-associative add-right-isotone l14*)
**finally have** $x^\circ \leq y^\circ + L$
**by** (*metis add-commutative less-eq-def zero-right-mult-decreasing*)
**thus** $x^\circ \sqsubseteq y^\circ$ **using** *2*
**by** (*metis apx-def*)
**qed**

**end**

**class** *omega-algebra-apx* = *domain-omega-algebra-lattice-L* + *domain-semiring-lattice-apx*

**sublocale** *omega-algebra-apx* < *star*!: *itering-apx* **where** *circ* = *star* **..**

**context** *omega-algebra-apx*

**begin**

**lemma** *omega-apx-isotone*: $x \sqsubseteq y \rightarrow x^\omega \sqsubseteq y^\omega$
**proof**
**assume** $x \sqsubseteq y$
**hence** *1*: $x \leq y + L \wedge d(L)$ ; $y \leq x + d(x$ ; $0)$ ; $T$
**by** (*metis apx-def*)
**have** $d(L)$ ; $y^\omega = (d(L)$ ; $y)^\omega$
**by** (*metis d-omega-export l2*)
**also have** ... ≤ $(x + d(x$ ; $0)$ ; $T)^\omega$ **using** *1*
**by** (*metis omega-isotone*)
**also have** ... = $(x^*$ ; $d(x$ ; $0)$ ; $T)^\omega + (x^*$ ; $d(x$ ; $0)$ ; $T)^*$ ; $x^\omega$
**by** (*metis mult-associative omega-decompose*)
**also have** ... ≤ $x^*$ ; $d(x$ ; $0)$ ; $T + (x^*$ ; $d(x$ ; $0)$ ; $T)^*$ ; $x^\omega$
**by** (*metis add-left-isotone mult-top-omega*)
**also have** ... = $x^*$ ; $d(x$ ; $0)$ ; $T + (1 + x^*$ ; $d(x$ ; $0)$ ; $T$ ; $(x^*$ ; $d(x$ ; $0)$ ; $T)^*)$ ; $x^\omega$
**by** (*metis mult-associative star.circ-left-top star.mult-top-circ*)
**also have** ... ≤ $x^\omega + x^*$ ; $d(x$ ; $0)$ ; $T$
**by** (*smt add-isotone add-least-upper-bound mult-associative mult-left-one mult-right-dist-add mult-right-isotone order-refl top-greatest*)
**also have** ... ≤ $x^\omega + d(x^*$ ; $x$ ; $0)$ ; $T$
**by** (*metis add-right-isotone mult-associative mult-domain-top*)
**also have** ... ≤ $x^\omega + d(x^*$ ; $0)$ ; $T$
**by** (*metis dL-star.d0-circ-left-unfold eq-refl star.circ-plus-same*)
**finally have** *2*: $d(L)$ ; $y^\omega \leq x^\omega + d(x^\omega$ ; $0)$ ; $T$
**by** (*smt add-right-isotone d0-star-below-d0-omega mult-left-isotone order-trans*)
**have** $x^\omega \leq (y + L)^\omega$ **using** *1*
**by** (*metis omega-isotone*)
**also have** ... = $(y^*$ ; $L)^\omega + (y^*$ ; $L)^*$ ; $y^\omega$
**by** (*metis omega-decompose*)

**also have** ... $= y^* \mathbin{;} L \mathbin{;} (y^* \mathbin{;} L)^\omega + (y^* \mathbin{;} L)^* \mathbin{;} y^\omega$
**by** (*metis omega-unfold*)
**also have** ... $\leq y^* \mathbin{;} L + (y^* \mathbin{;} L)^* \mathbin{;} y^\omega$
**by** (*metis add-left-isotone l40 mult-associative mult-right-isotone*)
**also have** ... $= y^* \mathbin{;} L + (1 + y^* \mathbin{;} L \mathbin{;} (y^* \mathbin{;} L)^*) \mathbin{;} y^\omega$
**by** (*metis star.circ-left-unfold*)
**also have** ... $\leq y^* \mathbin{;} L + y^\omega$
**by** (*metis add-commutative add-least-upper-bound add-right-upper-bound dL-star.mult-L-circ-mult-below mult-associative star.circ-mult star.circ-slide*)
**also have** ... $\leq y^* \mathbin{;} 0 + L + y^\omega$
**by** (*metis add-left-isotone l14*)
**finally have** $x^\omega \leq y^\omega + L$
**by** (*metis add-associative add-commutative less-eq-def star-zero-below-omega*)
**thus** $x^\omega \sqsubseteq y^\omega$ **using** *2*
**by** (*metis apx-def*)
**qed**

**lemma** *combined-apx-isotone*: $x \sqsubseteq y \rightarrow (x^\omega \frown L) + x^* \mathbin{;} z \sqsubseteq (y^\omega \frown L) + y^* \mathbin{;} z$
**by** (*metis add-apx-isotone mult-apx-left-isotone omega-apx-isotone star.circ-apx-isotone meet-L-apx-isotone*)

**lemma** *d-split-nu-mu*: $d(L) \mathbin{;} (y^\omega + y^* \mathbin{;} z) \leq y^* \mathbin{;} z + ((y^\omega + y^* \mathbin{;} z) \frown L) + d((y^\omega + y^* \mathbin{;} z) \mathbin{;} 0) \mathbin{;} T$
**proof** −
**have** $d(L) \mathbin{;} y^\omega \leq (y^\omega \frown L) + d(y^\omega \mathbin{;} 0) \mathbin{;} T$
**by** (*metis add-right-isotone l31 l91 omega-vector*)
**hence** $d(L) \mathbin{;} (y^\omega + y^* \mathbin{;} z) \leq y^* \mathbin{;} z + (y^\omega \frown L) + d(y^\omega \mathbin{;} 0) \mathbin{;} T$
**by** (*smt add-associative add-commutative add-isotone d-mult-below mult-left-dist-add*)
**thus** *?thesis*
**by** (*smt add-commutative add-isotone add-right-isotone add-right-upper-bound d-isotone meet-commutative meet-right-isotone mult-left-isotone order-trans*)
**qed**

**lemma** *loop-exists*: $d(L) \mathbin{;} \nu\ (\lambda x\ .\ y \mathbin{;} x + z) \leq \mu\ (\lambda x\ .\ y \mathbin{;} x + z) + (\nu\ (\lambda x\ .\ y \mathbin{;} x + z) \frown L) + d(\nu\ (\lambda x\ .\ y \mathbin{;} x + z) \mathbin{;} 0) \mathbin{;} T$
**by** (*metis d-split-nu-mu omega-loop-nu star-loop-mu*)

**lemma** *loop-isotone*: *isotone* $(\lambda x\ .\ y \mathbin{;} x + z)$
**by** (*smt add-commutative add-right-isotone isotone-def mult-right-isotone*)

**lemma** *loop-apx-isotone*: *apx-isotone* $(\lambda x\ .\ y \mathbin{;} x + z)$
**by** (*smt add-apx-left-isotone apx-isotone-def mult-apx-right-isotone*)

**lemma** *loop-has-least-fixpoint*: *has-least-fixpoint* $(\lambda x\ .\ y \mathbin{;} x + z)$
**by** (*metis has-least-fixpoint-def star-loop-is-least-fixpoint*)

**lemma** *loop-has-greatest-fixpoint*: *has-greatest-fixpoint* $(\lambda x\ .\ y \mathbin{;} x + z)$
**by** (*metis has-greatest-fixpoint-def omega-loop-is-greatest-fixpoint*)

**lemma** *loop-apx-least-fixpoint*: *is-apx-least-fixpoint* $(\lambda x\ .\ y \mathbin{;} x + z)$ $(\mu\ (\lambda x\ .\ y \mathbin{;} x + z) + (\nu\ (\lambda x\ .\ y \mathbin{;} x + z) \frown L))$
**by** (*metis apx-least-fixpoint-char loop-apx-isotone loop-exists loop-has-greatest-fixpoint loop-has-least-fixpoint loop-isotone nu-below-mu-nu-def nu-below-mu-nu-xi-mu-nu xi-mu-nu-def*)

**lemma** *loop-has-apx-least-fixpoint*: *has-apx-least-fixpoint* $(\lambda x\ .\ y\ ;\ x + z)$
  **by** (*metis has-apx-least-fixpoint-def loop-apx-least-fixpoint*)

**lemma** *loop-semantics*: $\xi\ (\lambda x\ .\ y\ ;\ x + z) = \mu\ (\lambda x\ .\ y\ ;\ x + z) + (\nu\ (\lambda x\ .\ y\ ;\ x + z) \frown L)$
  **by** (*metis apx-least-fixpoint-char loop-apx-least-fixpoint*)

**lemma** *loop-semantics-xi-mu-nu*: $\xi\ (\lambda x\ .\ y\ ;\ x + z) = (y^\omega \frown L) + y^*\ ;\ z$
**proof** −
  **have** $\xi\ (\lambda x\ .\ y\ ;\ x + z) = y^*\ ;\ z + ((y^\omega + y^*\ ;\ z) \frown L)$
    **by** (*metis loop-semantics omega-loop-nu star-loop-mu*)
  **thus** *?thesis*
    **by** (*smt add-absorb add-associative add-commutative add-left-dist-meet*)
**qed**

**lemma** *loop-semantics-xi-mu-nu-domain*: $\xi\ (\lambda x\ .\ y\ ;\ x + z) = d(y^\omega)\ ;\ L + y^*\ ;\ z$
  **by** (*metis loop-semantics-xi-mu-nu omega-meet-L*)

**lemma** *loop-semantics-apx-isotone*: $w \sqsubseteq y \rightarrow \xi\ (\lambda x\ .\ w\ ;\ x + z) \sqsubseteq \xi\ (\lambda x\ .\ y\ ;\ x + z)$
  **by** (*metis loop-semantics-xi-mu-nu combined-apx-isotone*)

**end**

**class** *while* =
  **fixes** *while* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixr** $*$ *60*)

**class** *binary-itering* = *semiring* + *while* +
  **assumes** *while-productstar*: $(x\ ;\ y) * z = z + x\ ;\ ((y\ ;\ x) * (y\ ;\ z))$
  **assumes** *while-sumstar*: $(x + y) * z = (x * y) * (x * z)$
  **assumes** *while-left-dist-add*: $x * (y + z) = (x * y) + (x * z)$
  **assumes** *while-sub-associative*: $(x * y)\ ;\ z \leq x * (y\ ;\ z)$
  **assumes** *while-simulate-left-plus*: $x\ ;\ z \leq z\ ;\ (y * 1) + w \rightarrow x * (z\ ;\ v) \leq z\ ;\ (y * v) + (x * (w\ ;\ (y * v)))$
  **assumes** *while-simulate-right-plus*: $z\ ;\ x \leq y\ ;\ (y * z) + w \rightarrow z\ ;\ (x * v) \leq y * (z\ ;\ v + w\ ;\ (x * v))$

**begin**

**lemma** *while-zero*: $0 * x = x$
  **by** (*metis add-right-zero mult-left-zero while-productstar*)

**lemma** *while-mult-increasing*: $x\ ;\ y \leq x * y$
  **by** (*metis add-least-upper-bound mult-left-one order-refl while-productstar*)

**lemma** *while-one-increasing*: $x \leq x * 1$
  **by** (*metis mult-right-one while-mult-increasing*)

**lemma** *while-increasing*: $y \leq x * y$

**by** (*metis add-left-divisibility mult-left-one while-productstar*)

**lemma** *while-right-isotone*: $y \leq z \rightarrow x * y \leq x * z$
**by** (*metis less-eq-def while-left-dist-add*)

**lemma** *while-left-isotone*: $x \leq y \rightarrow x * z \leq y * z$
**by** (*metis less-eq-def while-increasing while-sumstar*)

**lemma** *while-isotone*: $w \leq x \wedge y \leq z \rightarrow w * y \leq x * z$
**by** (*smt order-trans while-left-isotone while-right-isotone*)

**lemma** *while-left-unfold*: $x * y = y + x \ ; (x * y)$
**by** (*metis mult-left-one mult-right-one while-productstar*)

**lemma** *while-simulate-left-plus-1*: $x \ ; z \leq z \ ; (y * 1) \rightarrow x * (z \ ; w) \leq z \ ; (y * w) + (x * 0)$
**by** (*metis add-right-zero mult-left-zero while-simulate-left-plus*)

**lemma** *while-simulate-absorb*: $y \ ; x \leq x \rightarrow y * x \leq x + (y * 0)$
**by** (*metis while-simulate-left-plus-1 while-zero mult-right-one*)

**lemma** *while-transitive*: $x * (x * y) = x * y$
**by** (*metis add-right-upper-bound add-right-zero antisym while-increasing while-left-dist-add while-left-unfold while-simulate-absorb*)

**lemma** *while-slide*: $(x \ ; y) * (x \ ; z) = x \ ; ((y \ ; x) * z)$
**by** (*metis mult-associative mult-left-dist-add while-left-unfold while-productstar*)

**lemma** *while-zero-2*: $(x \ ; 0) * y = x \ ; 0 + y$
**by** (*metis add-commutative mult-associative mult-left-zero while-left-unfold*)

**lemma** *while-mult-star-exchange*: $x \ ; (x * y) = x * (x \ ; y)$
**by** (*metis mult-left-one while-slide*)

**lemma** *while-right-unfold*: $x * y = y + (x * (x \ ; y))$
**by** (*metis while-left-unfold while-mult-star-exchange*)

**lemma** *while-one-mult-below*: $(x * 1) \ ; y \leq x * y$
**by** (*metis mult-left-one while-sub-associative*)

**lemma** *while-plus-one*: $x * y = y + (x * y)$
**by** (*metis less-eq-def while-increasing*)

**lemma** *while-rtc-2*: $y + x \ ; y + (x * (x * y)) = x * y$
**by** (*metis add-associative less-eq-def while-mult-increasing while-plus-one while-transitive*)

**lemma** *while-left-plus-below*: $x \ ; (x * y) \leq x * y$
**by** (*metis add-right-divisibility while-left-unfold*)

**lemma** *while-right-plus-below*: $x * (x \; ; \; y) \leq x * y$
**by** (*metis while-left-plus-below while-mult-star-exchange*)

**lemma** *while-right-plus-below-2*: $(x * x) \; ; \; y \leq x * y$
**by** (*smt order-trans while-right-plus-below while-sub-associative*)

**lemma** *while-mult-transitive*: $x \leq z * y \wedge y \leq z * w \rightarrow x \leq z * w$
**by** (*smt order-trans while-right-isotone while-transitive*)

**lemma** *while-mult-upper-bound*: $x \leq z * 1 \wedge y \leq z * w \rightarrow x \; ; \; y \leq z * w$
**by** (*metis less-eq-def mult-right-subdist-add-left order-trans while-mult-transitive while-one-mult-below*)

**lemma** *while-sub-dist*: $x * z \leq (x + y) * z$
**by** (*metis add-left-upper-bound while-left-isotone*)

**lemma** *while-sub-dist-1*: $x \; ; \; z \leq (x + y) * z$
**by** (*metis order-trans while-mult-increasing while-sub-dist*)

**lemma** *while-sub-dist-2*: $x \; ; \; y \; ; \; z \leq (x + y) * z$
**by** (*smt mult-associative mult-right-subdist-add-right order-trans while-mult-increasing while-mult-transitive while-sub-dist-1*)

**lemma** *while-sub-dist-3*: $x * (y * z) \leq (x + y) * z$
**by** (*metis add-right-upper-bound while-left-isotone while-mult-transitive while-sub-dist*)

**lemma** *while-absorb-2*: $x \leq y \rightarrow y * (x * z) = y * z$
**by** (*metis add-commutative less-eq-def while-left-dist-add while-plus-one while-sub-dist-3*)

**lemma** *while-simulate-right-plus-1*: $z \; ; \; x \leq y \; ; \; (y * z) \rightarrow z \; ; \; (x * w) \leq y * (z \; ; \; w)$
**by** (*metis add-right-zero mult-left-zero while-simulate-right-plus*)

**lemma** *while-sumstar-1-below*: $x * ((y \; ; \; (x * 1)) * z) \leq ((x * 1) \; ; \; y) * (x * z)$
**proof** −
**have** *1*: $x \; ; \; (((x * 1) \; ; \; y) * (x * z)) \leq ((x * 1) \; ; \; y) * (x * z)$
**by** (*smt add-isotone add-right-upper-bound mult-associative mult-left-dist-add mult-right-subdist-add-right while-left-unfold*)
**have** $x * ((y \; ; \; (x * 1)) * z) \leq (x * z) + (x * (y \; ; \; (((x * 1) \; ; \; y) * ((x * 1) \; ; \; z))))$
**by** (*metis eq-refl while-left-dist-add while-productstar*)
**also have** $... \leq (x * z) + (x * ((x * 1) \; ; \; y \; ; \; (((x * 1) \; ; \; y) * ((x * 1) \; ; \; z))))$
**by** (*metis add-right-isotone mult-associative mult-left-one mult-right-subdist-add-left while-left-unfold while-right-isotone*)
**also have** $... \leq (x * z) + (x * (((x * 1) \; ; \; y) * ((x * 1) \; ; \; z)))$
**by** (*metis add-right-isotone add-right-upper-bound while-left-unfold while-right-isotone*)
**also have** $... \leq x * (((x * 1) \; ; \; y) * (x * z))$
**by** (*smt add-associative add-left-upper-bound less-eq-def mult-left-one while-left-dist-add while-left-unfold while-sub-associative*)
**also have** $... \leq (((x * 1) \; ; \; y) * (x * z)) + (x * 0)$ **using** *1*
**by** (*metis while-simulate-absorb*)
**also have** $... = ((x * 1) \; ; \; y) * (x * z)$
**by** (*smt add-associative add-commutative add-left-zero while-left-dist-add while-left-unfold*)
**finally show** *?thesis*

.
**qed**

**lemma** *while-sumstar-2-below*: $((x * 1) \ ; \ y) * (x * z) \leq (x * y) * (x * z)$
**by** (*metis mult-left-one while-left-isotone while-sub-associative*)

**lemma** *while-add-1-below*: $x * ((y \ ; \ (x * 1)) * z) \leq (x + y) * z$
**proof** −
**have** $((x * 1) \ ; \ y) * ((x * 1) \ ; \ z) \leq (x + y) * z$
**by** (*metis while-isotone while-one-mult-below while-sumstar*)
**hence** $(y \ ; \ (x * 1)) * z \leq z + y \ ; \ ((x + y) * z)$
**by** (*metis add-right-isotone mult-right-isotone while-productstar*)
**also have** ... $\leq (x + y) * z$
**by** (*metis add-right-isotone add-right-upper-bound mult-left-isotone while-left-unfold*)
**finally show** *?thesis*
**by** (*metis add-commutative add-right-upper-bound while-isotone while-transitive*)
**qed**

**lemma** *while-while-while*: $((x * 1) * 1) * y = (x * 1) * y$
**by** (*smt add-commutative less-eq-def mult-right-one while-left-plus-below while-mult-star-exchange while-plus-one while-sumstar while-transitive*)

**lemma** *while-one*: $(1 * 1) * y = 1 * y$
**by** (*metis while-while-while while-zero*)

**lemma** *while-add-below*: $x + y \leq x * (y * 1)$
**by** (*smt add-commutative add-isotone case-split-right order-trans while-increasing while-left-plus-below while-mult-increasing while-plus-one*)

**lemma** *while-add-2*: $(x + y) * z \leq (x * (y * 1)) * z$
**by** (*metis while-add-below while-left-isotone*)

**lemma** *while-sup-one-left-unfold*: $1 \leq x \rightarrow x \ ; \ (x * y) = x * y$
**by** (*metis less-eq-def mult-left-one mult-right-dist-add while-mult-star-exchange while-right-unfold while-transitive*)

**lemma** *while-sup-one-right-unfold*: $1 \leq x \rightarrow x * (x \ ; \ y) = x * y$
**by** (*metis while-mult-star-exchange while-sup-one-left-unfold*)

**lemma** *while-decompose-7*: $(x + y) * z = x * (y * ((x + y) * z))$
**by** (*metis eq-iff order-trans while-increasing while-sub-dist-3 while-transitive*)

**lemma** *while-decompose-8*: $(x + y) * z = (x + y) * (x * (y * z))$
**by** (*metis add-commutative while-sumstar while-transitive*)

**lemma** *while-decompose-9*: $(x * (y * 1)) * z = x * (y * ((x * (y * 1)) * z))$
**by** (*smt add-commutative less-eq-def order-trans while-add-below while-increasing while-sub-dist-3*)

**lemma** *while-decompose-10*: $(x * (y * 1)) * z = (x * (y * 1)) * (x * (y * z))$
**proof** −

**have** *1*: $(x * (y * 1)) * z \leq (x * (y * 1)) * (x * (y * z))$
**by** (*metis add-associative less-eq-def while-left-dist-add while-plus-one*)
**have** $x + (y * 1) \leq x * (y * 1)$
**by** (*metis add-least-upper-bound while-add-below while-increasing*)
**hence** $(x * (y * 1)) * (x * (y * z)) \leq (x * (y * 1)) * z$
**by** (*smt antisym while-decompose-9 while-increasing while-mult-transitive while-right-isotone*)
**thus** *?thesis* **using** *1*
**by** (*metis antisym*)
**qed**

**lemma** *while-back-loop-fixpoint*: $z \ ; (y * (y \ ; x)) + z \ ; x = z \ ; (y * x)$
**by** (*metis add-commutative mult-left-dist-add while-right-unfold*)

**lemma** *while-back-loop-prefixpoint*: $z \ ; (y * 1) \ ; y + z \leq z \ ; (y * 1)$
**by** (*metis add-least-upper-bound mult-associative mult-right-isotone mult-right-one order-refl while-increasing while-mult-upper-bound while-one-increasing*)

**lemma** *while-loop-is-fixpoint*: *is-fixpoint* $(\lambda x \ . \ y \ ; x + z)$ $(y * z)$
**by** (*smt add-commutative is-fixpoint-def while-left-unfold*)

**lemma** *while-back-loop-is-prefixpoint*: *is-prefixpoint* $(\lambda x \ . \ x \ ; y + z)$ $(z \ ; (y * 1))$
**by** (*metis is-prefixpoint-def while-back-loop-prefixpoint*)

**lemma** *while-while-add*: $(1 + x) * y = (x * 1) * y$
**by** (*metis add-commutative while-decompose-10 while-sumstar while-zero*)

**lemma** *while-while-mult-sub*: $x * (1 * y) \leq (x * 1) * y$
**by** (*metis add-commutative while-sub-dist-3 while-while-add*)

**lemma** *while-right-plus*: $(x * x) * y = x * y$
**by** (*metis add-idempotent while-plus-one while-sumstar while-transitive*)

**lemma** *while-left-plus*: $(x \ ; (x * 1)) * y = x * y$
**by** (*metis mult-right-one while-mult-star-exchange while-right-plus*)

**lemma** *while-below-while-one*: $x * x \leq x * 1$
**by** (*metis while-one-increasing while-right-plus*)

**lemma** *while-below-while-one-mult*: $x \ ; (x * x) \leq x \ ; (x * 1)$
**by** (*metis mult-right-isotone while-below-while-one*)

**lemma** *while-add-sub-add-one*: $x * (x + y) \leq x * (1 + y)$
**by** (*metis add-left-isotone while-below-while-one while-left-dist-add*)

**lemma** *while-add-sub-add-one-mult*: $x \ ; (x * (x + y)) \leq x \ ; (x * (1 + y))$
**by** (*metis mult-right-isotone while-add-sub-add-one*)

**lemma** *while-elimination*: $x \ ; y = 0 \rightarrow x \ ; (y * z) = x \ ; z$

**by** (*metis add-right-zero mult-associative mult-left-dist-add mult-left-zero while-left-unfold*)

**lemma** *while-square*: $(x\ ;\ x) * y \leq x * y$
**by** (*metis while-left-isotone while-mult-increasing while-right-plus*)

**lemma** *while-mult-sub-add*: $(x\ ;\ y) * z \leq (x + y) * z$
**by** (*metis while-increasing while-isotone while-mult-increasing while-sumstar*)

**lemma** *while-absorb-1*: $x \leq y \rightarrow x * (y * z) = y * z$
**by** (*metis antisym less-eq-def while-increasing while-sub-dist-3*)

**lemma** *while-absorb-3*: $x \leq y \rightarrow x * (y * z) = y * (x * z)$
**by** (*metis while-absorb-1 while-absorb-2*)

**lemma** *while-square-2*: $(x\ ;\ x) * ((x + 1)\ ;\ y) \leq x * y$
**by** (*smt add-least-upper-bound while-increasing while-mult-transitive while-mult-upper-bound while-one-increasing while-square*)

**lemma** *while-separate-unfold-below*: $(y\ ;\ (x * 1)) * z \leq (y * z) + (y * (y\ ;\ x\ ;\ (x * ((y\ ;\ (x * 1)) * z))))$
**proof** −
**have** $(y\ ;\ (x * 1)) * z = (y * (y\ ;\ x\ ;\ (x * 1))) * (y * z)$
**by** (*metis mult-associative mult-left-dist-add mult-right-one while-left-unfold while-sumstar*)
**hence** $(y\ ;\ (x * 1)) * z = (y * z) + (y * (y\ ;\ x\ ;\ (x * 1)))\ ;\ ((y\ ;\ (x * 1)) * z)$
**by** (*metis while-left-unfold*)
**also have** ... $\leq (y * z) + (y * (y\ ;\ x\ ;\ (x * 1))\ ;\ ((y\ ;\ (x * 1)) * z))$
**by** (*metis add-right-isotone while-sub-associative*)
**also have** ... $\leq (y * z) + (y * (y\ ;\ x\ ;\ (x * ((y\ ;\ (x * 1)) * z))))$
**by** (*smt add-right-isotone mult-associative mult-right-isotone while-one-mult-below while-right-isotone*)
**finally show** *?thesis*
.
**qed**

**lemma** *while-mult-zero-add*: $(x + y\ ;\ 0) * z = x * ((y\ ;\ 0) * z)$
**proof** −
**have** $(x + y\ ;\ 0) * z = (x * (y\ ;\ 0)) * (x * z)$
**by** (*metis while-sumstar*)
**also have** ... $= (x * z) + (x * (y\ ;\ 0))\ ;\ ((x * (y\ ;\ 0)) * (x * z))$
**by** (*metis while-left-unfold*)
**also have** ... $\leq (x * z) + (x * (y\ ;\ 0))$
**by** (*metis add-right-isotone mult-associative mult-left-zero while-sub-associative*)
**also have** ... $= x * ((y\ ;\ 0) * z)$
**by** (*metis add-commutative while-left-dist-add while-zero-2*)
**finally show** *?thesis*
**by** (*metis le-neq-trans less-def while-sub-dist-3*)
**qed**

**lemma** *while-add-mult-zero*: $(x + y\ ;\ 0) * y = x * y$
**by** (*metis less-eq-def while-mult-zero-add while-zero-2 zero-right-mult-decreasing*)

**lemma** *while-mult-zero-add-2*: $(x + y \ ; \ 0) * z = (x * z) + (x * (y \ ; \ 0))$
**by** (*metis add-commutative while-left-dist-add while-mult-zero-add while-zero-2*)

**lemma** *while-add-zero-star*: $(x + y \ ; \ 0) * z = x * (y \ ; \ 0 + z)$
**by** (*metis while-mult-zero-add while-zero-2*)

**lemma** *while-unfold-sum*: $(x + y) * z = (x * z) + (x * (y \ ; \ ((x + y) * z)))$
**apply** (*rule antisym*)
**apply** (*smt add-associative less-eq-def while-absorb-1 while-increasing while-mult-star-exchange while-right-unfold while-sub-associative while-sumstar*)
**by** (*metis add-least-upper-bound while-decompose-7 while-mult-increasing while-right-isotone while-sub-dist*)

**lemma** *while-simulate-left*: $x \ ; \ z \leq z \ ; \ y + w \rightarrow x * (z \ ; \ v) \leq z \ ; \ (y * v) + (x * (w \ ; \ (y * v)))$
**by** (*smt add-commutative add-right-isotone mult-right-isotone order-trans while-one-increasing while-simulate-left-plus*)

**lemma** *while-simulate-right*: $z \ ; \ x \leq y \ ; \ z + w \rightarrow z \ ; \ (x * v) \leq y * (z \ ; \ v + w \ ; \ (x * v))$
**proof** −
**have** $y \ ; \ z + w \leq y \ ; \ (y * z) + w$
**by** (*metis add-left-isotone mult-right-isotone while-increasing*)
**thus** *?thesis*
**by** (*smt order-trans while-simulate-right-plus*)
**qed**

**lemma** *while-simulate*: $z \ ; \ x \leq y \ ; \ z \rightarrow z \ ; \ (x * v) \leq y * (z \ ; \ v)$
**by** (*metis add-right-zero mult-left-zero while-simulate-right*)

**lemma** *while-while-mult*: $1 * (x * y) = (x * 1) * y$
**proof** −
**have** $(x * 1) * y \leq (x * 1) \ ; \ ((x * 1) * y)$
**by** (*metis order-refl while-increasing while-sup-one-left-unfold*)
**also have** $... \leq 1 * ((x * 1) \ ; \ y)$
**by** (*metis mult-left-one order-refl while-mult-upper-bound while-simulate*)
**also have** $... \leq 1 * (x * y)$
**by** (*metis while-one-mult-below while-right-isotone*)
**finally show** *?thesis*
**by** (*metis antisym while-sub-dist-3 while-while-add*)
**qed**

**lemma** *while-simulate-left-1*: $x \ ; \ z \leq z \ ; \ y \rightarrow x * (z \ ; \ v) \leq z \ ; \ (y * v) + (x * 0)$
**by** (*metis add-right-zero mult-left-zero while-simulate-left*)

**lemma** *while-associative-1*: $1 \leq z \rightarrow x * (y \ ; \ z) = (x * y) \ ; \ z$
**proof**
**assume** *1*: $1 \leq z$
**have** $x * (y \ ; \ z) \leq x * ((x * y) \ ; \ z)$
**by** (*metis less-eq-def mult-right-dist-add while-plus-one while-right-isotone*)
**also have** $... \leq (x * y) \ ; \ (0 * z) + (x * 0)$

**by** (*metis mult-associative mult-right-subdist-add-right while-left-unfold while-simulate-absorb while-zero*)
**also have** ... $\leq (x * y)\ ;\ z + (x * 0)\ ;\ z$ **using** *1*
**by** (*metis add-least-upper-bound add-left-upper-bound add-right-upper-bound case-split-right while-plus-one while-zero*)
**also have** ... $= (x * y)\ ;\ z$
**by** (*metis add-right-zero mult-right-dist-add while-left-dist-add*)
**finally show** $x * (y\ ;\ z) = (x * y)\ ;\ z$
**by** (*metis antisym while-sub-associative*)
**qed**

**lemma** *while-associative-while-1*: $x * (y\ ;\ (z * 1)) = (x * y)\ ;\ (z * 1)$
**by** (*metis while-associative-1 while-increasing*)

**lemma** *while-one-while*: $(x * 1)\ ;\ (y * 1) = x * (y * 1)$
**by** (*metis mult-left-one while-associative-while-1*)

**lemma** *while-decompose-5-below*: $(x * (y * 1)) * z \leq (y * (x * 1)) * z$
**by** (*smt add-commutative mult-left-dist-add mult-right-one while-increasing while-left-unfold while-mult-star-exchange while-one-while while-plus-one while-sumstar*)

**lemma** *while-decompose-5*: $(x * (y * 1)) * z = (y * (x * 1)) * z$
**by** (*metis antisym while-decompose-5-below*)

**lemma** *while-decompose-4*: $(x * (y * 1)) * z = x * ((y * (x * 1)) * z)$
**by** (*metis while-decompose-5 while-decompose-9 while-transitive*)

**lemma** *while-simulate-2*: $y\ ;\ (x * 1) \leq x * (y * 1) \leftrightarrow y * (x * 1) \leq x * (y * 1)$
**proof** (*rule iffI*)
**assume** $y\ ;\ (x * 1) \leq x * (y * 1)$
**hence** $y\ ;\ (x * 1) \leq (x * 1)\ ;\ (y * 1)$
**by** (*metis while-one-while*)
**hence** $y * ((x * 1)\ ;\ 1) \leq (x * 1)\ ;\ (y * 1) + (y * 0)$
**by** (*metis while-simulate-left-plus-1*)
**hence** $y * (x * 1) \leq (x * (y * 1)) + (y * 0)$
**by** (*metis mult-right-one while-one-while*)
**also have** ... $= x * (y * 1)$
**by** (*metis add-commutative less-eq-def order-trans while-increasing while-right-isotone zero-least*)
**finally show** $y * (x * 1) \leq x * (y * 1)$
.
**next**
**assume** $y * (x * 1) \leq x * (y * 1)$
**thus** $y\ ;\ (x * 1) \leq x * (y * 1)$
**by** (*metis order-trans while-mult-increasing*)
**qed**

**lemma** *while-simulate-1*: $y\ ;\ x \leq x\ ;\ y \rightarrow y * (x * 1) \leq x * (y * 1)$
**by** (*smt mult-right-one order-trans while-one-increasing while-right-isotone while-simulate while-simulate-2*)

**lemma** *while-simulate-3*: $y\ ;\ (x * 1) \leq x * 1 \rightarrow y * (x * 1) \leq x * (y * 1)$

**by** (*metis add-idempotent case-split-right while-increasing while-mult-upper-bound while-simulate-2*)

**lemma** *while-extra-while*: $(y\ ;\ (x * 1)) * z = (y\ ;\ (y * (x * 1))) * z$
**proof** −
**have** $y\ ;\ (y * (x * 1)) \leq y\ ;\ (x * 1)\ ;\ (y\ ;\ (x * 1) * 1)$
**by** (*smt add-commutative add-left-upper-bound mult-right-one order-trans while-back-loop-prefixpoint while-left-isotone while-mult-star-exchange*)
**hence** *1*: $(y\ ;\ (y * (x * 1))) * z \leq (y\ ;\ (x * 1)) * z$
**by** (*metis while-simulate-right-plus-1 mult-left-one*)
**have** $(y\ ;\ (x * 1)) * z \leq (y\ ;\ (y * (x * 1))) * z$
**by** (*metis while-increasing while-left-isotone while-mult-star-exchange*)
**thus** *?thesis* **using** *1*
**by** (*metis antisym*)
**qed**

**lemma** *while-separate-4*: $y\ ;\ x \leq x\ ;\ (x * (1 + y)) \rightarrow (x + y) * z = x * (y * z)$
**proof**
**assume** *1*: $y\ ;\ x \leq x\ ;\ (x * (1 + y))$
**hence** $(1 + y)\ ;\ x \leq x\ ;\ (x * (1 + y))$
**by** (*smt add-associative add-least-upper-bound mult-left-one mult-left-subdist-add-left mult-right-dist-add mult-right-one while-left-unfold*)
**hence** *2*: $(1 + y)\ ;\ (x * 1) \leq x * (1 + y)$
**by** (*metis mult-right-one while-simulate-right-plus-1*)
**have** $y\ ;\ x\ ;\ (x * 1) \leq x\ ;\ (x * ((1 + y)\ ;\ (x * 1)))$ **using** *1*
**by** (*smt less-eq-def mult-associative mult-right-dist-add while-associative-1 while-increasing*)
**also have** $... \leq x\ ;\ (x * (1 + y))$ **using** *2*
**by** (*smt less-eq-def mult-left-dist-add order-refl while-absorb-2 while-left-dist-add*)
**also have** $... \leq x\ ;\ (x * 1)\ ;\ (y * 1)$
**by** (*metis add-least-upper-bound mult-associative mult-right-isotone while-increasing while-one-increasing while-one-while while-right-isotone*)
**finally have** $y * (x\ ;\ (x * 1)) \leq x\ ;\ (x * 1)\ ;\ (y * 1) + (y * 0)$
**by** (*metis mult-associative mult-right-one while-simulate-left-plus-1*)
**hence** $(y * 1)\ ;\ (y * x) \leq x\ ;\ (x * y * 1) + (y * 0)$
**by** (*smt less-eq-def mult-associative mult-right-one order-refl order-trans while-absorb-2 while-left-dist-add while-mult-star-exchange while-one-mult-below while-one-while while-plus-one*)
**hence** $(y * 1)\ ;\ ((y * x) * (y * z)) \leq x * ((y * 1)\ ;\ (y * z) + (y * 0)\ ;\ ((y * x) * (y * z)))$
**by** (*metis while-simulate-right-plus*)
**also have** $... \leq x * ((y * z) + (y * 0))$
**by** (*metis add-isotone mult-left-zero order-refl while-absorb-2 while-one-mult-below while-right-isotone while-sub-associative*)
**also have** $... = x * y * z$
**by** (*metis add-right-zero while-left-dist-add*)
**finally show** $(x + y) * z = x * (y * z)$
**by** (*smt add-commutative less-eq-def mult-left-one mult-right-dist-add while-plus-one while-sub-associative while-sumstar*)
**qed**

**lemma** *while-separate-5*: $y\ ;\ x \leq x\ ;\ (x * (x + y)) \rightarrow (x + y) * z = x * (y * z)$
**by** (*smt order-trans while-add-sub-add-one-mult while-separate-4*)

**lemma** *while-separate-6*: $y\ ;\ x \leq x\ ;\ (x + y) \rightarrow (x + y) * z = x * (y * z)$
**by** (*smt order-trans while-increasing while-mult-star-exchange while-separate-5*)

**lemma** *while-separate-1*: $y \ ; \ x \leq x \ ; \ y \rightarrow (x + y) * z = x * (y * z)$
**by** (*metis add-least-upper-bound less-eq-def mult-left-subdist-add-right while-separate-6*)

**lemma** *while-separate-mult-1*: $y \ ; \ x \leq x \ ; \ y \rightarrow (x \ ; \ y) * z \leq x * (y * z)$
**by** (*metis while-mult-sub-add while-separate-1*)

**lemma** *separation*: $y \ ; \ x \leq x \ ; \ (y * 1) \rightarrow (x + y) * z = x * (y * z)$
**proof**
**assume** $y \ ; \ x \leq x \ ; \ (y * 1)$
**hence** $y * x \leq x \ ; \ (y * 1) + (y * 0)$
**by** (*metis mult-right-one while-simulate-left-plus-1*)
**also have** $... \leq x \ ; \ (x * y * 1) + (y * 0)$
**by** (*metis add-left-isotone while-increasing while-mult-star-exchange*)
**finally have** $(y * 1) \ ; \ (y * x) \leq x \ ; \ (x * y * 1) + (y * 0)$
**by** (*metis order-refl order-trans while-absorb-2 while-one-mult-below*)
**hence** $(y * 1) \ ; \ ((y * x) * (y * z)) \leq x * ((y * 1) \ ; \ (y * z) + (y * 0) \ ; \ ((y * x) * (y * z)))$
**by** (*metis while-simulate-right-plus*)
**also have** $... \leq x * ((y * z) + (y * 0))$
**by** (*metis add-isotone mult-left-zero order-refl while-absorb-2 while-one-mult-below while-right-isotone while-sub-associative*)
**also have** $... = x * y * z$
**by** (*metis add-right-zero while-left-dist-add*)
**finally show** $(x + y) * z = x * (y * z)$
**by** (*smt add-commutative less-eq-def mult-left-one mult-right-dist-add while-plus-one while-sub-associative while-sumstar*)
**qed**

**lemma** *while-separate-left*: $y \ ; \ x \leq x \ ; \ (y * 1) \rightarrow y * (x * z) \leq x * (y * z)$
**by** (*metis add-commutative separation while-sub-dist-3*)

**lemma** *while-simulate-4*: $y \ ; \ x \leq x \ ; \ (x * (1 + y)) \rightarrow y * (x * z) \leq x * (y * z)$
**by** (*metis add-commutative while-separate-4 while-sub-dist-3*)

**lemma** *while-simulate-5*: $y \ ; \ x \leq x \ ; \ (x * (x + y)) \rightarrow y * (x * z) \leq x * (y * z)$
**by** (*smt order-trans while-add-sub-add-one-mult while-simulate-4*)

**lemma** *while-simulate-6*: $y \ ; \ x \leq x \ ; \ (x + y) \rightarrow y * (x * z) \leq x * (y * z)$
**by** (*smt order-trans while-increasing while-mult-star-exchange while-simulate-5*)

**lemma** *while-simulate-7*: $y \ ; \ x \leq x \ ; \ y \rightarrow y * (x * z) \leq x * (y * z)$
**by** (*metis add-commutative mult-left-subdist-add-left order-trans while-simulate-6*)

**lemma** *while-while-mult-1*: $x * (1 * y) = 1 * (x * y)$
**by** (*metis add-commutative mult-left-one mult-right-one order-refl while-separate-1*)

**lemma** *while-while-mult-2*: $x * (1 * y) = (x * 1) * y$
**by** (*metis while-while-mult while-while-mult-1*)

**lemma** *while-import*: $p \leq p \ ; \ p \wedge p \leq 1 \wedge p \ ; \ x \leq x \ ; \ p \rightarrow p \ ; \ (x * y) = p \ ; \ ((p \ ; \ x) * y)$

**proof**
**assume** *1*: $p \leq p\ ;\ p \wedge p \leq 1 \wedge p\ ;\ x \leq x\ ;\ p$
**hence** $p\ ;\ (x * y) \leq (p\ ;\ x) * (p\ ;\ y)$
**by** (*smt add-commutative less-eq-def mult-associative mult-left-dist-add mult-right-one while-simulate*)
**also have** $\ldots \leq (p\ ;\ x) * y$ **using** *1*
**by** (*metis less-eq-def mult-left-one mult-right-dist-add while-right-isotone*)
**finally have** *2*: $p\ ;\ (x * y) \leq p\ ;\ ((p\ ;\ x) * y)$ **using** *1*
**by** (*smt add-commutative less-eq-def mult-associative mult-left-dist-add mult-right-one*)
**have** $p\ ;\ ((p\ ;\ x) * y) \leq p\ ;\ (x * y)$ **using** *1*
**by** (*metis mult-left-isotone mult-left-one mult-right-isotone while-left-isotone*)
**thus** $p\ ;\ (x * y) = p\ ;\ ((p\ ;\ x) * y)$ **using** *2*
**by** (*metis antisym*)
**qed**

**lemma** *while-preserve*: $p \leq p\ ;\ p \wedge p \leq 1 \wedge p\ ;\ x \leq x\ ;\ p \rightarrow p\ ;\ (x * y) = p\ ;\ (x * (p\ ;\ y))$
**apply** (*rule*, *rule antisym*)
**apply** (*smt mult-associative mult-left-isotone mult-right-isotone order-trans while-simulate*)
**by** (*metis mult-left-isotone mult-left-one mult-right-isotone while-right-isotone*)

**lemma** *while-plus-below-while*: $(x * 1)\ ;\ x \leq x * 1$
**by** (*metis order-refl while-mult-upper-bound while-one-increasing*)

**lemma** *while-01*: $(w\ ;\ (x * 1)) * (y\ ;\ z) \leq (x * w) * ((x * y)\ ;\ z)$
**proof** −
**have** $(w\ ;\ (x * 1)) * (y\ ;\ z) = y\ ;\ z + w\ ;\ (((x * 1)\ ;\ w) * ((x * 1)\ ;\ y\ ;\ z))$
**by** (*metis mult-associative while-productstar*)
**also have** $\ldots \leq y\ ;\ z + w\ ;\ ((x * w) * ((x * y)\ ;\ z))$
**by** (*metis add-right-isotone mult-left-isotone mult-right-isotone while-isotone while-one-mult-below*)
**also have** $\ldots \leq (x * y)\ ;\ z + (x * w)\ ;\ ((x * w) * ((x * y)\ ;\ z))$
**by** (*metis add-isotone mult-right-subdist-add-left while-left-unfold*)
**finally show** *?thesis*
**by** (*metis while-left-unfold*)
**qed**

**lemma** *while-while-sub-associative*: $x * (y * z) \leq ((x * y) * z) + (x * z)$
**proof** −
**have** *1*: $x\ ;\ (x * 1) \leq (x * 1)\ ;\ ((x * y) * 1)$
**by** (*metis add-least-upper-bound order-trans while-back-loop-prefixpoint while-left-plus-below*)
**have** $x * (y * z) \leq x * ((x * 1)\ ;\ (y * z))$
**by** (*metis mult-left-isotone mult-left-one while-increasing while-right-isotone*)
**also have** $\ldots \leq (x * 1)\ ;\ ((x * y) * (y * z)) + (x * 0)$ **using** *1*
**by** (*metis while-simulate-left-plus-1*)
**also have** $\ldots \leq (x * 1)\ ;\ ((x * y) * z) + (x * z)$
**by** (*metis add-isotone order-refl while-absorb-2 while-increasing while-right-isotone zero-least*)
**also have** $\ldots = (x * 1)\ ;\ z + (x * 1)\ ;\ (x * y)\ ;\ ((x * y) * z) + (x * z)$
**by** (*metis mult-associative mult-left-dist-add while-left-unfold*)
**also have** $\ldots = (x * y)\ ;\ ((x * y) * z) + (x * z)$

**by** (*smt add-associative add-commutative less-eq-def mult-left-one mult-right-dist-add order-refl while-absorb-1 while-plus-one while-sub-associative*)
**also have** ... $\leq ((x * y) * z) + (x * z)$
**by** (*metis add-left-isotone while-left-plus-below*)
**finally show** *?thesis*
.
**qed**

**end**

**class** *binary-itering-T* = *semiring-T* + *binary-itering*

**begin**

**lemma** *while-right-top*: $x * T = T$
**by** (*metis add-left-top while-left-unfold*)

**lemma** *while-left-top*: $T \; ; (x * 1) = T$
**by** (*metis add-right-top antisym top-greatest while-back-loop-prefixpoint*)

**end**

**class** *binary-itering-1* = *binary-itering* +
**assumes** *while-denest-0*: $w \; ; (x * (y \; ; z)) \leq (w \; ; (x * y)) * (w \; ; (x * y) \; ; z)$

**begin**

**lemma** *while-denest-1*: $w \; ; (x * (y \; ; z)) \leq (w \; ; (x * y)) * z$
**by** (*metis order-trans while-denest-0 while-right-plus-below*)

**lemma** *while-mult-sub-while-while*: $x * (y \; ; z) \leq (x * y) * z$
**by** (*metis mult-left-one while-denest-1*)

**lemma** *while-zero-zero*: $(x * 0) * 0 = x * 0$
**by** (*smt less-eq-def mult-left-zero while-left-dist-add while-mult-star-exchange while-mult-sub-while-while while-mult-zero-add-2 while-plus-one while-sumstar*)

**lemma** *while-mult-zero-zero*: $(x \; ; (y * 0)) * 0 = x \; ; (y * 0)$
**apply** (*rule antisym*)
**apply** (*metis add-least-upper-bound add-right-zero mult-left-zero mult-right-isotone while-left-dist-add while-slide while-sub-associative*)
**by** (*metis mult-left-zero while-denest-1*)

**lemma** *while-denest-2*: $w \; ; ((x * (y \; ; w)) * z) = w \; ; (((x * y) \; ; w) * z)$
**apply** (*rule antisym*)
**apply** (*metis mult-associative while-denest-0 while-simulate-right-plus-1 while-slide*)

**by** (*metis mult-right-isotone while-left-isotone while-sub-associative*)

**lemma** *while-denest-3*: $(x * w) * (x * 0) = (x * w) * 0$
**by** (*metis while-absorb-2 while-right-isotone while-zero-zero zero-least*)

**lemma** *while-02*: $x * ((x * w) * ((x * y) \; ; z)) = (x * w) * ((x * y) \; ; z)$
**proof** −
**have** $x \; ; ((x * w) * ((x * y) \; ; z)) = x \; ; (x * y) \; ; z + x \; ; (x * w) \; ; ((x * w) * ((x * y) \; ; z))$
**by** (*metis mult-associative mult-left-dist-add while-left-unfold*)
**also have** ... $\leq (x * w) * ((x * y) \; ; z)$
**by** (*metis add-isotone mult-right-subdist-add-right while-left-unfold*)
**finally have** $x * ((x * w) * ((x * y) \; ; z)) \leq ((x * w) * ((x * y) \; ; z)) + (x * 0)$
**by** (*metis while-simulate-absorb*)
**hence** $x * ((x * w) * ((x * y) \; ; z)) \leq (x * w) * ((x * y) \; ; z)$
**by** (*smt add-least-upper-bound order-refl order-trans while-mult-sub-while-while while-right-isotone zero-least*)
**thus** *?thesis*
**by** (*metis antisym while-increasing*)
**qed**

**lemma** *while-sumstar-3-below*: $(x * y) * (x * z) \leq (x * y) * ((x * 1) \; ; z)$
**proof** −
**have** $(x * y) * (x * z) = (x * z) + ((x * y) * ((x * y) \; ; (x * z)))$
**by** (*metis while-right-unfold*)
**also have** ... $\leq (x * z) + ((x * y) * (x * (y \; ; (x * z))))$
**by** (*metis add-right-isotone while-right-isotone while-sub-associative*)
**also have** ... $\leq (x * z) + ((x * y) * (x * ((x * y) * ((x * 1) \; ; z)))))$
**by** (*smt add-right-isotone mult-left-isotone mult-left-one order-trans while-increasing while-mult-upper-bound while-one-increasing while-right-isotone while-sumstar while-transitive*)
**also have** ... $= (x * z) + ((x * y) * ((x * 1) \; ; z))$
**by** (*metis while-02 while-transitive*)
**also have** ... $= (x * y) * ((x * 1) \; ; z)$
**by** (*smt add-associative mult-left-one mult-right-dist-add while-02 while-left-dist-add while-plus-one*)
**finally show** *?thesis*
.
**qed**

**lemma** *while-sumstar-4-below*: $(x * y) * ((x * 1) \; ; z) \leq x * ((y \; ; (x * 1)) * z)$
**proof** −
**have** $(x * y) * ((x * 1) \; ; z) = (x * 1) \; ; z + (x * y) \; ; ((x * y) * ((x * 1) \; ; z))$
**by** (*metis while-left-unfold*)
**also have** ... $\leq (x * z) + (x * (y \; ; ((x * y) * ((x * 1) \; ; z))))$
**by** (*metis add-isotone while-one-mult-below while-sub-associative*)
**also have** ... $= (x * z) + (x * (y \; ; (((x * 1) \; ; y) * ((x * 1) \; ; z))))$
**by** (*metis mult-left-one while-denest-2*)
**also have** ... $= x * ((y \; ; (x * 1)) * z)$
**by** (*metis while-left-dist-add while-productstar*)
**finally show** *?thesis*
.

**qed**

**lemma** *while-sumstar-1*: $(x + y) * z = (x * y) * ((x * 1) \ ; z)$
**by** (*smt eq-iff order-trans while-add-1-below while-sumstar while-sumstar-3-below while-sumstar-4-below*)

**lemma** *while-sumstar-2*: $(x + y) * z = x * ((y \ ; (x * 1)) * z)$
**by** (*metis eq-iff while-add-1-below while-sumstar-1 while-sumstar-4-below*)

**lemma** *while-sumstar-3*: $(x + y) * z = ((x * 1) \ ; y) * (x * z)$
**by** (*metis eq-iff while-sumstar while-sumstar-1-below while-sumstar-2 while-sumstar-2-below*)

**lemma** *while-decompose-6*: $x * ((y \ ; (x * 1)) * z) = y * ((x \ ; (y * 1)) * z)$
**by** (*metis add-commutative while-sumstar-2*)

**lemma** *while-denest-4*: $(x * w) * (x * (y \ ; z)) = (x * w) * ((x * y) \ ; z)$
**proof** −
**have** $(x * w) * (x * (y \ ; z)) = x * ((w \ ; (x * 1)) * (y \ ; z))$
**by** (*metis while-sumstar while-sumstar-2*)
**also have** $... \leq (x * w) * ((x * y) \ ; z)$
**by** (*smt antisym while-01 while-02 while-increasing while-right-isotone*)
**finally show** *?thesis*
**by** (*metis antisym while-right-isotone while-sub-associative*)
**qed**

**lemma** *while-denest-5*: $w \ ; ((x * (y \ ; w)) * (x * (y \ ; z))) = w \ ; (((x * y) \ ; w) * ((x * y) \ ; z))$
**by** (*metis while-denest-2 while-denest-4*)

**lemma** *while-denest-6*: $(w \ ; (x * y)) * z = z + w \ ; ((x + y \ ; w) * (y \ ; z))$
**by** (*metis while-denest-5 while-productstar while-sumstar*)

**lemma** *while-sum-below-one*: $y \ ; ((x + y) * z) \leq (y \ ; (x * 1)) * z$
**by** (*metis add-right-divisibility mult-left-one while-denest-6*)

**lemma** *while-separate-unfold*: $(y \ ; (x * 1)) * z = (y * z) + (y * (y \ ; x \ ; (x * ((y \ ; (x * 1)) * z))))$
**proof** −
**have** $y * (y \ ; x \ ; (x * ((y \ ; (x * 1)) * z))) \leq y * (y \ ; ((x + y) * z))$
**by** (*metis mult-associative mult-right-isotone while-sumstar-2 while-left-plus-below while-right-isotone*)
**also have** $... \leq (y \ ; (x * 1)) * z$
**by** (*metis add-commutative add-left-upper-bound while-absorb-1 while-mult-star-exchange while-sum-below-one*)
**finally have** $(y * z) + (y * (y \ ; x \ ; (x * ((y \ ; (x * 1)) * z)))) \leq (y \ ; (x * 1)) * z$
**by** (*metis add-least-upper-bound mult-left-subdist-add-left mult-right-one while-left-isotone while-left-unfold*)
**thus** *?thesis*
**by** (*metis antisym while-separate-unfold-below*)
**qed**

**lemma** *while-finite-associative*: $x * 0 = 0 \rightarrow (x * y) \ ; z = x * (y \ ; z)$
**by** (*metis while-denest-4 while-zero*)

**lemma** *atomicity-refinement*: $s = s \ ; \ q \wedge x = q \ ; \ x \wedge q \ ; \ b = 0 \wedge r \ ; \ b \leq b \ ; \ r \wedge r \ ; \ l \leq l \ ; \ r \wedge x \ ; \ l \leq l \ ; \ x \wedge b \ ; \ l \leq l \ ; \ b \wedge q \ ; \ l \leq l \ ; \ q \wedge r * q \leq q \ ; \ (r * 1) \wedge q \leq 1 \rightarrow s \ ; \ ((x + b + r + l) * (q \ ; z)) \leq s \ ; \ ((x \ ; (b * q) + r + l) * z)$
**proof**
**assume** *1*: $s = s \ ; \ q \wedge x = q \ ; \ x \wedge q \ ; \ b = 0 \wedge r \ ; \ b \leq b \ ; \ r \wedge r \ ; \ l \leq l \ ; \ r \wedge x \ ; \ l \leq l \ ; \ x \wedge b \ ; \ l \leq l \ ; \ b \wedge q \ ; \ l \leq l \ ; \ q \wedge r * q \leq q \ ; \ (r * 1) \wedge q \leq 1$
**hence** *2*: $(x + b + r) \ ; \ l \leq l \ ; \ (x + b + r)$
**by** (*smt add-commutative add-least-upper-bound mult-left-subdist-add-right mult-right-dist-add order-trans*)
**have** $q \ ; \ ((x \ ; (b * r * 1) \ ; q) * z) \leq (x \ ; (b * r * 1) \ ; q) * z$ **using** *1*
**by** (*smt eq-refl order-trans while-increasing while-mult-upper-bound*)
**also have** ... $\leq (x \ ; (b * ((r * 1) \ ; q))) * z$
**by** (*metis mult-associative mult-right-isotone while-left-isotone while-sub-associative*)
**also have** ... $\leq (x \ ; (b * r * q)) * z$
**by** (*metis mult-right-isotone while-left-isotone while-one-mult-below while-right-isotone*)
**also have** ... $\leq (x \ ; (b * (q \ ; (r * 1)))) * z$ **using** *1*
**by** (*metis mult-right-isotone while-left-isotone while-right-isotone*)
**finally have** *3*: $q \ ; \ ((x \ ; (b * r * 1) \ ; q) * z) \leq (x \ ; (b * q) \ ; (r * 1)) * z$
**by** (*metis mult-associative while-associative-while-1*)
**have** $s \ ; \ ((x + b + r + l) * (q \ ; z)) = s \ ; \ (l * (x + b + r) * (q \ ; z))$ **using** *2*
**by** (*metis add-commutative while-separate-1*)
**also have** ... $= s \ ; \ q \ ; \ (l * b * r * (q \ ; x \ ; (b * r * 1)) * (q \ ; z))$ **using** *1*
**by** (*smt add-associative add-commutative while-sumstar-2 while-separate-1*)
**also have** ... $= s \ ; \ q \ ; \ (l * b * r * (q \ ; ((x \ ; (b * r * 1) \ ; q) * z)))$
**by** (*smt mult-associative while-slide*)
**also have** ... $\leq s \ ; \ q \ ; \ (l * b * r * (x \ ; (b * q) \ ; (r * 1)) * z)$ **using** *3*
**by** (*metis mult-right-isotone while-right-isotone*)
**also have** ... $\leq s \ ; \ (l * q \ ; (b * r * (x \ ; (b * q) \ ; (r * 1)) * z))$ **using** *1*
**by** (*smt mult-associative mult-right-isotone while-simulate*)
**also have** ... $= s \ ; \ (l * q \ ; (r * (x \ ; (b * q) \ ; (r * 1)) * z))$ **using** *1*
**by** (*metis while-elimination*)
**also have** ... $\leq s \ ; \ (l * r * (x \ ; (b * q) \ ; (r * 1)) * z)$ **using** *1*
**by** (*smt add-left-divisibility mult-left-one mult-right-dist-add mult-right-isotone while-right-isotone*)
**also have** ... $= s \ ; \ (l * (r + x \ ; (b * q)) * z)$
**by** (*metis while-sumstar-2*)
**also have** ... $\leq s \ ; \ ((x \ ; (b * q) + r + l) * z)$
**by** (*metis add-commutative mult-right-isotone while-sub-dist-3*)
**finally show** $s \ ; \ ((x + b + r + l) * (q \ ; z)) \leq s \ ; \ ((x \ ; (b * q) + r + l) * z)$
.
**qed**

**end**

**class** *binary-itering-1-T* = *binary-itering-T* + *binary-itering-1*

**begin**

**end**

**class** *strict-itering* = *itering-3* + *while* +
  **assumes** *while-def*: $x * y = x^\circ \ ; \ y$

**begin**

**subclass** *binary-itering-1*
  **apply** *unfold-locales*
  **apply** (*metis add-commutative circ-loop-fixpoint circ-slide mult-associative while-def*)
  **apply** (*metis circ-add mult-associative while-def*)
  **apply** (*metis mult-left-dist-add while-def*)
  **apply** (*metis mult-associative order-refl while-def*)
  **apply** (*metis circ-simulate-left-plus mult-associative mult-left-isotone mult-right-dist-add mult-right-one while-def*)
  **apply** (*metis circ-simulate-right-plus mult-associative mult-left-isotone mult-right-dist-add while-def*)
  **by** (*metis add-right-divisibility circ-loop-fixpoint mult-associative while-def*)

**lemma** *while-associative*: $(x * y) \ ; \ z = x * (y \ ; \ z)$
  **by** (*metis mult-associative while-def*)

**lemma** *circ-import*: $p \leq p \ ; \ p \wedge p \leq 1 \wedge p \ ; \ x \leq x \ ; \ p \rightarrow p \ ; \ x^\circ = p \ ; \ (p \ ; \ x)^\circ$
**proof**
  **assume** *1*: $p \leq p \ ; \ p \wedge p \leq 1 \wedge p \ ; \ x \leq x \ ; \ p$
  **hence** $p \ ; \ x \leq p \ ; \ x \ ; \ p$
    **by** (*smt mult-associative mult-left-isotone mult-right-isotone order-trans*)
  **hence** $p \ ; \ x^\circ \leq (p \ ; \ x)^\circ$ **using** *1*
    **by** (*smt circ-mult-upper-bound circ-reflexive circ-simulate order-refl order-trans*)
  **hence** *2*: $p \ ; \ x^\circ \leq p \ ; \ (p \ ; \ x)^\circ$ **using** *1*
    **by** (*smt mult-associative mult-left-isotone mult-right-isotone order-trans*)
  **have** $p \ ; \ (p \ ; \ x)^\circ \leq p \ ; \ x^\circ$ **using** *1*
    **by** (*metis circ-isotone mult-left-isotone mult-left-one mult-right-isotone*)
  **thus** $p \ ; \ x^\circ = p \ ; \ (p \ ; \ x)^\circ$ **using** *2*
    **by** (*metis antisym*)
**qed**

**lemma** *while-one-mult*: $(x * 1) \ ; \ x = x * x$
  **by** (*metis mult-right-one while-def*)

**lemma** *while-back-loop-is-fixpoint*: *is-fixpoint* $(\lambda x \ . \ x \ ; \ y + z) \ (z \ ; \ (y * 1))$
  **by** (*metis circ-back-loop-is-fixpoint mult-right-one while-def*)

**lemma** $x \ ; \ z \leq z \wedge y \leq z \wedge x * 1 \leq z \rightarrow x * y \leq z$
  **by** (*metis add-commutative less-eq-def mult-left-one order-trans while-right-isotone while-simulate-absorb zero-right-mult-decreasing*)

**lemma** $(x + y) * z = ((x * 1) \ ; \ y) * ((x * 1) \ ; \ z)$

**by** (*metis mult-right-one while-def while-sumstar*)

**lemma** $(x * 1) \ ; \ y = x * y$
**by** (*metis mult-left-one while-associative*)

**end**

**class** *strict-itering-T* = *itering-T* + *strict-itering*

**begin**

**subclass** *binary-itering-1-T* **..**

**lemma** *while-top-2*: $T * z = T \ ; \ z$
**by** (*metis circ-top while-def*)

**lemma** *while-mult-top-2*: $(x \ ; \ T) * z = z + x \ ; \ T \ ; \ z$
**by** (*metis circ-left-top mult-associative while-def while-left-unfold*)

**end**

**class** *nonstrict-itering* = *omega-algebra-T* + *while* +
**assumes** *while-def*: $x * y = x^\omega + x^* \ ; \ y$

**begin**

**subclass** *binary-itering-T*
**proof** (*unfold-locales*)
**fix** *x y z*
**show** $(x \ ; \ y) * z = z + x \ ; \ ((y \ ; \ x) * (y \ ; \ z))$
**by** (*metis add-commutative mult-associative mult-left-dist-add omega-loop-fixpoint omega-slide star.circ-slide while-def*)
**next**
**fix** *x y z*
**show** $(x + y) * z = (x * y) * (x * z)$
**proof** −
**have** *1*: $(x + y) * z = (x^* \ ; \ y)^\omega + (x^* \ ; \ y)^* \ ; \ (x^\omega + x^* \ ; \ z)$
**by** (*smt add-associative mult-associative mult-left-dist-add omega-decompose star.circ-add while-def*)
**hence** *2*: $(x + y) * z \leq (x * y) * (x * z)$
**by** (*smt add-isotone add-right-upper-bound less-eq-def mult-left-isotone omega-sub-dist star.circ-sub-dist while-def*)
**let** *?rhs* = $x^* \ ; \ y \ ; \ ((x^\omega + x^* \ ; \ y)^\omega + (x^\omega + x^* \ ; \ y)^* \ ; \ (x^\omega + x^* \ ; \ z)) + (x^\omega + x^* \ ; \ z)$
**have** $x^\omega \ ; \ (x^\omega + x^* \ ; \ y)^\omega \leq x^\omega$

```
    by (metis omega-sub-vector)
  hence x^ω ; (x^ω + x^* ; y)^ω + x^* ; y ; (x^ω + x^* ; y)^ω ≤ ?rhs
    by (smt add-commutative add-isotone add-left-upper-bound mult-left-dist-add order-trans)
  hence 3: (x^ω + x^* ; y)^ω ≤ ?rhs
    by (metis mult-right-dist-add omega-unfold)
  have x^ω ; (x^ω + x^* ; y)^* ; (x^ω + x^* ; z) ≤ x^ω
    by (metis mult-associative omega-sub-vector)
  hence x^ω ; (x^ω + x^* ; y)^* ; (x^ω + x^* ; z) + x^* ; y ; (x^ω + x^* ; y)^* ; (x^ω + x^* ; z) ≤ ?rhs
    by (smt add-commutative add-isotone add-right-upper-bound mult-associative mult-left-dist-add order-trans)
  hence (x^ω + x^* ; y)^* ; (x^ω + x^* ; z) ≤ ?rhs
    by (smt add-associative add-right-upper-bound less-eq-def mult-associative mult-right-dist-add star.circ-loop-fixpoint)
  hence (x^ω + x^* ; y)^ω + (x^ω + x^* ; y)^* ; (x^ω + x^* ; z) ≤ ?rhs using 3
    by (metis add-least-upper-bound)
  hence (x^ω + x^* ; y)^ω + (x^ω + x^* ; y)^* ; (x^ω + x^* ; z) ≤ (x^* ; y)^ω + (x^* ; y)^* ; (x^ω + x^* ; z)
    by (metis add-commutative omega-induct)
  thus ?thesis using 1 2
    by (smt antisym while-def)
 qed
next
 fix x y z
 show x ∗ (y + z) = (x ∗ y) + (x ∗ z)
   by (smt add-associative add-commutative add-left-upper-bound less-eq-def mult-left-dist-add while-def)
next
 fix x y z
 show (x ∗ y) ; z ≤ x ∗ (y ; z)
   by (metis mult-associative mult-right-dist-add omega-loop-fixpoint omega-loop-greatest-fixpoint while-def)
next
 fix v w x y z
 show x ; z ≤ z ; (y ∗ 1) + w → x ∗ (z ; v) ≤ z ; (y ∗ v) + (x ∗ (w ; (y ∗ v)))
 proof
   assume x ; z ≤ z ; (y ∗ 1) + w
   hence 1: x ; z ≤ z ; y^ω + z ; y^* + w
     by (metis mult-left-dist-add mult-right-one while-def)
   let ?rhs = z ; (y^ω + y^* ; v) + x^ω + x^* ; w ; (y^ω + y^* ; v)
   have 2: z ; v ≤ ?rhs
     by (metis add-least-upper-bound add-left-upper-bound mult-left-dist-add omega-loop-fixpoint)
   have x ; z ; (y^ω + y^* ; v) ≤ ?rhs
   proof −
     have x ; z ; (y^ω + y^* ; v) ≤ (z ; y^ω + z ; y^* + w) ; (y^ω + y^* ; v) using 1
       by (metis mult-left-isotone)
     also have ... = z ; (y^ω ; (y^ω + y^* ; v) + y^* ; (y^ω + y^* ; v)) + w ; (y^ω + y^* ; v)
       by (smt mult-associative mult-left-dist-add mult-right-dist-add)
     also have ... = z ; (y^ω ; (y^ω + y^* ; v) + y^ω + y^* ; v) + w ; (y^ω + y^* ; v)
       by (smt add-associative mult-associative mult-left-dist-add star.circ-transitive-equal star-mult-omega)
     also have ... ≤ z ; (y^ω + y^* ; v) + x^* ; w ; (y^ω + y^* ; v)
       by (smt add-commutative add-isotone add-left-top mult-left-dist-add mult-left-one mult-right-dist-add mult-right-subdist-add-left omega-vector order-refl star.circ-plus-one)
     finally show ?thesis
```

```
      by (smt add-associative add-commutative less-eq-def)
    qed
    hence x ; ?rhs ≤ ?rhs
    by (smt add-associative add-commutative add-left-upper-bound less-eq-def mult-associative mult-left-dist-add mult-right-dist-add omega-unfold star.circ-increasing star.circ-transitive-equal)
    hence z ; v + x ; ?rhs ≤ ?rhs using 2
      by (metis add-least-upper-bound)
    hence x* ; z ; v ≤ ?rhs
      by (metis mult-associative star-left-induct)
    hence x^ω + x* ; z ; v ≤ ?rhs
      by (metis add-least-upper-bound add-left-upper-bound)
    thus x ∗ (z ; v) ≤ z ; (y ∗ v) + (x ∗ (w ; (y ∗ v)))
      by (smt add-associative mult-associative mult-left-dist-add while-def)
  qed
next
  fix v w x y z
  show z ; x ≤ y ; (y ∗ z) + w → z ; (x ∗ v) ≤ y ∗ (z ; v + w ; (x ∗ v))
  proof
    assume z ; x ≤ y ; (y ∗ z) + w
    hence z ; x ≤ y ; (y^ω + y* ; z) + w
      by (metis while-def)
    hence 1: z ; x ≤ y^ω + y ; y* ; z + w
      by (metis mult-associative mult-left-dist-add omega-unfold)
    let ?rhs = y^ω + y* ; z ; v + y* ; w ; (x^ω + x* ; v)
    have 2: z ; x^ω ≤ ?rhs
    proof −
      have z ; x^ω ≤ y ; y* ; z ; x^ω + y^ω ; x^ω + w ; x^ω using 1
        by (smt add-commutative less-eq-def mult-associative mult-right-dist-add omega-unfold)
      also have ... ≤ y ; y* ; z ; x^ω + y^ω + w ; x^ω
        by (metis add-left-isotone add-right-isotone omega-sub-vector)
      also have ... = y ; y* ; (z ; x^ω) + (y^ω + w ; x^ω)
        by (metis add-associative mult-associative)
      finally have z ; x^ω ≤ (y ; y*)^ω + (y ; y*)* ; (y^ω + w ; x^ω)
        by (metis add-commutative omega-induct)
      also have ... = y^ω + y* ; w ; x^ω
        by (metis left-plus-omega less-eq-def mult-associative mult-left-dist-add mult-left-subdist-add-left star.left-plus-circ star-mult-omega)
      also have ... ≤ ?rhs
        by (metis add-isotone add-left-upper-bound mult-left-subdist-add-left)
      finally show ?thesis
        by metis
    qed
    let ?rhs2 = y^ω + y* ; z + y* ; w ; (x^ω + x*)
    have ?rhs2 ; x ≤ ?rhs2
    proof −
      have 3: y^ω ; x ≤ ?rhs2
        by (metis add-associative less-eq-def omega-sub-vector)
      have y* ; z ; x ≤ y* ; (y^ω + y ; y* ; z + w) using 1
        by (metis mult-associative mult-right-isotone)
```

```
    also have ... = y^ω + y^* ; y ; y^* ; z + y^* ; w
      by (metis mult-associative mult-left-dist-add star-mult-omega)
    also have ... = y^ω + y ; y^* ; z + y^* ; w
      by (metis mult-associative star.circ-transitive-equal star-simulation-right-equal)
    also have ... ≤ y^ω + y^* ; z + y^* ; w
      by (metis add-left-isotone add-right-isotone mult-left-isotone star.left-plus-below-circ)
    also have ... ≤ y^ω + y^* ; z + y^* ; w ; x^*
      by (metis add-right-isotone add-right-upper-bound star.circ-back-loop-fixpoint)
    finally have 4: y^* ; z ; x ≤ ?rhs2
      by (smt add-associative add-commutative less-eq-def mult-left-dist-add)
    have (x^ω + x^*) ; x ≤ x^ω + x^*
      by (metis add-isotone mult-right-dist-add omega-sub-vector star.circ-plus-same star.left-plus-below-circ)
    hence y^* ; w ; (x^ω + x^*) ; x ≤ ?rhs2
      by (smt add-right-upper-bound mult-associative mult-right-isotone order-trans)
    thus ?thesis using 3 4
      by (smt add-associative less-eq-def mult-right-dist-add)
  qed
  hence z + ?rhs2 ; x ≤ ?rhs2
    by (smt add-commutative add-least-upper-bound add-right-divisibility while-def omega-loop-fixpoint)
  hence 5: z ; x^* ≤ ?rhs2
    by (metis star-right-induct)
  have z ; x^* ; v ≤ ?rhs
  proof -
    have z ; x^* ; v ≤ ?rhs2 ; v using 5
      by (metis mult-left-isotone)
    also have ... = y^ω ; v + y^* ; z ; v + y^* ; w ; (x^ω ; v + x^* ; v)
      by (metis mult-associative mult-right-dist-add)
    also have ... ≤ y^ω + y^* ; z ; v + y^* ; w ; (x^ω ; v + x^* ; v)
      by (metis add-left-isotone omega-sub-vector)
    also have ... ≤ ?rhs
      by (metis add-left-isotone add-right-isotone mult-right-isotone omega-sub-vector)
    finally show ?thesis
      by metis
  qed
  hence z ; (x^ω + x^* ; v) ≤ ?rhs using 2
    by (smt add-associative less-eq-def mult-associative mult-left-dist-add)
  thus z ; (x ∗ v) ≤ y ∗ (z ; v + w ; (x ∗ v))
    by (metis add-associative mult-associative mult-left-dist-add while-def)
 qed
qed

lemma while-top: T ∗ x = T
 by (metis add-left-top star.circ-top star-omega-top while-def)

lemma while-one-top: 1 ∗ x = T
 by (metis add-left-top omega-one while-def)
```

**lemma** *while-finite-associative*: $x^\omega = 0 \rightarrow (x * y) \ ; z = x * (y \ ; z)$
**by** (*metis add-left-zero mult-associative while-def*)

**lemma** *star-below-while*: $x^* \ ; y \leq x * y$
**by** (*metis add-right-upper-bound while-def*)

**lemma** *while-sub-mult-one*: $x \ ; (1 * y) \leq 1 * x$
**by** (*metis top-greatest while-one-top*)

**lemma** *while-while-one*: $y * (x * 1) = y^\omega + y^* \ ; x^\omega + y^* \ ; x^*$
**by** (*metis add-associative mult-left-dist-add mult-right-one while-def*)

**lemma** *omega-mult-star-2*: $x^\omega \ ; y^* = x^\omega$
**by** (*metis add-right-upper-bound antisym omega-sub-vector star.circ-back-loop-fixpoint*)

**lemma** *star-omega-absorb*: $y^* \ ; (y^* \ ; x)^* \ ; y^\omega = (y^* \ ; x)^* \ ; y^\omega$
**proof** −
**have** $y^* \ ; (y^* \ ; x)^* \ ; y^\omega = y^* \ ; y^* \ ; x \ ; (y^* \ ; x)^* \ ; y^\omega + y^* \ ; y^\omega$
**by** (*metis add-commutative mult-associative mult-right-dist-add star.circ-back-loop-fixpoint star.circ-plus-same*)
**thus** *?thesis*
**by** (*metis mult-associative star.circ-loop-fixpoint star.circ-transitive-equal star-mult-omega*)
**qed**

**lemma** *star-star-absorb*: $y^* \ ; (y^* \ ; x)^* \ ; y^* = (y^* \ ; x)^* \ ; y^*$
**by** (*metis add-commutative mult-associative star.circ-decompose-4 star.circ-slide star-decompose-1 star-decompose-3*)

**lemma** *while-simulate-4-plus*: $y \ ; x \leq x \ ; (x * (1 + y)) \rightarrow y \ ; x \ ; x^* \leq x \ ; (x * (1 + y))$
**proof**
**have** *1*: $x \ ; (x * (1 + y)) = x^\omega + x \ ; x^* + x \ ; x^* \ ; y$
**by** (*metis add-associative mult-associative mult-left-dist-add mult-right-one omega-unfold while-def*)
**assume** $y \ ; x \leq x \ ; (x * (1 + y))$
**hence** $y \ ; x \ ; x^* \leq (x^\omega + x \ ; x^* + x \ ; x^* \ ; y) \ ; x^*$ **using** *1*
**by** (*metis mult-left-isotone*)
**also have** $... = x^\omega \ ; x^* + x \ ; x^* \ ; x^* + x \ ; x^* \ ; y \ ; x^*$
**by** (*metis mult-right-dist-add*)
**also have** $... = x \ ; x^* \ ; (y \ ; x \ ; x^*) + x^\omega + x \ ; x^* + x \ ; x^* \ ; y$
**by** (*smt add-associative add-commutative mult-associative omega-mult-star-2 star.circ-back-loop-fixpoint star.circ-plus-same star.circ-transitive-equal*)
**finally have** $y \ ; x \ ; x^* \leq x \ ; x^* \ ; (y \ ; x \ ; x^*) + (x^\omega + x \ ; x^* + x \ ; x^* \ ; y)$
**by** (*metis add-associative*)
**hence** $y \ ; x \ ; x^* \leq (x \ ; x^*)^\omega + (x \ ; x^*)^* \ ; (x^\omega + x \ ; x^* + x \ ; x^* \ ; y)$
**by** (*metis add-commutative omega-induct*)
**also have** $... = x^\omega + x^* \ ; (x^\omega + x \ ; x^* + x \ ; x^* \ ; y)$
**by** (*metis left-plus-omega star.left-plus-circ*)
**finally show** $y \ ; x \ ; x^* \leq x \ ; (x * (1 + y))$ **using** *1*
**by** (*metis while-def while-mult-star-exchange while-transitive*)
**qed**

**lemma** *while-simulate-4-omega*: $y \ ; \ x \leq x \ ; \ (x * (1 + y)) \rightarrow y \ ; \ x^\omega \leq x^\omega$
**proof**
**have** *1*: $x \ ; \ (x * (1 + y)) = x^\omega + x \ ; \ x^* + x \ ; \ x^* \ ; \ y$
**by** (*metis add-associative mult-associative mult-left-dist-add mult-right-one omega-unfold while-def*)
**assume** $y \ ; \ x \leq x \ ; \ (x * (1 + y))$
**hence** $y \ ; \ x^\omega \leq (x^\omega + x \ ; \ x^* + x \ ; \ x^* \ ; \ y) \ ; \ x^\omega$ **using** *1*
**by** (*smt less-eq-def mult-associative mult-right-dist-add omega-unfold*)
**also have** $\ldots = x^\omega \ ; \ x^\omega + x \ ; \ x^* \ ; \ x^\omega + x \ ; \ x^* \ ; \ y \ ; \ x^\omega$
**by** (*metis mult-right-dist-add*)
**also have** $\ldots = x \ ; \ x^* \ ; \ (y \ ; \ x^\omega) + x^\omega$
**by** (*metis add-commutative less-eq-def mult-associative omega-sub-vector omega-unfold star-mult-omega*)
**finally have** $y \ ; \ x^\omega \leq x \ ; \ x^* \ ; \ (y \ ; \ x^\omega) + x^\omega$
**by** *metis*
**hence** $y \ ; \ x^\omega \leq (x \ ; \ x^*)^\omega + (x \ ; \ x^*)^* \ ; \ x^\omega$
**by** (*metis add-commutative omega-induct*)
**thus** $y \ ; \ x^\omega \leq x^\omega$
**by** (*metis add-idempotent left-plus-omega star-mult-omega*)
**qed**

**lemma** *while-unfold-below*: $x = z + y \ ; \ x \rightarrow x \leq y * z$
**by** (*metis omega-induct-equal while-def*)

**lemma** *while-unfold-below-1*: $x = y \ ; \ x \rightarrow x \leq y * 1$
**by** (*metis add-right-upper-bound omega-induct while-def*)

**lemma** *while-square-1*: $x * 1 = (x \ ; \ x) * (x + 1)$
**by** (*metis mult-right-one omega-square star-square-2 while-def*)

**lemma** *while-absorb-below-one*: $y \ ; \ x \leq x \rightarrow y * x \leq 1 * x$
**by** (*metis top-greatest while-one-top*)

**lemma** *omega-import*: $p \leq p \ ; \ p \wedge p \ ; \ x \leq x \ ; \ p \rightarrow p \ ; \ x^\omega = p \ ; \ (p \ ; \ x)^\omega$
**proof**
**assume** *1*: $p \leq p \ ; \ p \wedge p \ ; \ x \leq x \ ; \ p$
**hence** $p \ ; \ x^\omega \leq p \ ; \ (p \ ; \ x) \ ; \ x^\omega$
**by** (*metis mult-associative mult-left-isotone omega-unfold*)
**also have** $\ldots \leq p \ ; \ x \ ; \ p \ ; \ x^\omega$ **using** *1*
**by** (*metis mult-associative mult-left-isotone mult-right-isotone*)
**finally have** $p \ ; \ x^\omega \leq (p \ ; \ x)^\omega$
**by** (*metis mult-associative omega-induct-mult*)
**hence** $p \ ; \ x^\omega \leq p \ ; \ (p \ ; \ x)^\omega$ **using** *1*
**by** (*smt mult-associative mult-left-isotone mult-right-isotone order-trans*)
**thus** $p \ ; \ x^\omega = p \ ; \ (p \ ; \ x)^\omega$ **using** *1*
**by** (*metis add-left-divisibility antisym mult-right-isotone omega-induct-mult omega-slide omega-sub-dist*)
**qed**

**lemma** *star-import*: $p \leq p \ ; \ p \wedge p \leq 1 \wedge p \ ; \ x \leq x \ ; \ p \rightarrow p \ ; \ x^* = p \ ; \ (p \ ; \ x)^*$

**proof**
  **assume** *1*: $p \leq p \ ; \ p \wedge p \leq 1 \wedge p \ ; \ x \leq x \ ; \ p$
  **hence** $p \ ; \ (p \ ; \ x)^* \ ; \ x \leq p \ ; \ (x \ ; \ p)^* \ ; \ x$
    **by** (*smt add-right-divisibility mult-associative mult-left-isotone mult-left-subdist-add-right star.circ-simulate star.circ-slide*)
  **hence** $p \ ; \ (p \ ; \ x)^* \ ; \ x \leq (p \ ; \ x)^*$
    **by** (*metis add-right-divisibility order-trans star.circ-mult*)
  **hence** *2*: $p \ ; \ (p \ ; \ x)^* \ ; \ x \leq p \ ; \ (p \ ; \ x)^*$ **using** *1*
    **by** (*smt mult-associative mult-left-isotone mult-right-isotone order-trans*)
  **have** $p \leq p \ ; \ (p \ ; \ x)^*$
    **by** (*metis add-right-divisibility star.circ-back-loop-fixpoint*)
  **hence** $p \ ; \ x^* \leq p \ ; \ (p \ ; \ x)^*$ **using** *2*
    **by** (*metis add-least-upper-bound star-right-induct*)
  **thus** $p \ ; \ x^* = p \ ; \ (p \ ; \ x)^*$ **using** *1*
    **by** (*metis antisym mult-left-isotone mult-left-one mult-right-isotone star.circ-isotone*)
**qed**

**lemma** *while-loop-is-greatest-postfixpoint*: *is-greatest-postfixpoint* $(\lambda x \ . \ y \ ; \ x + z) \ (y * z)$
**proof** −
  **have** $(y * z) \leq (\lambda x \ . \ y \ ; \ x + z) \ (y * z)$
    **by** (*metis is-fixpoint-def order-refl while-loop-is-fixpoint*)
  **thus** *?thesis*
    **by** (*smt add-commutative is-greatest-postfixpoint-def omega-induct while-def*)
**qed**

**lemma** *while-loop-is-greatest-fixpoint*: *is-greatest-fixpoint* $(\lambda x \ . \ y \ ; \ x + z) \ (y * z)$
  **by** (*metis omega-loop-is-greatest-fixpoint while-def*)

**lemma** $x \ ; \ z \leq z \wedge y \leq z \wedge x * 1 \leq z \longrightarrow x * y \leq z$
  **by** (*metis add-commutative add-least-upper-bound add-left-zero less-eq-def while-right-isotone while-simulate-absorb*)

**end**

**class** *nonstrict-itering-zero* = *nonstrict-itering* +
  **assumes** *mult-right-zero*: $x \ ; \ 0 = 0$

**begin**

**lemma** *while-finite-associative-2*: $x * 0 = 0 \longrightarrow (x * y) \ ; \ z = x * (y \ ; \ z)$
  **by** (*metis add-left-zero add-right-zero mult-associative mult-right-zero while-def*)

**end**

**class** *nonstrict-itering-tarski* = *nonstrict-itering* +

**assumes** *tarski*: $x \leq x \ ; \ T \ ; \ x \ ; \ T$

**begin**

**lemma** *tarski-mult-top-idempotent*: $x \ ; \ T = x \ ; \ T \ ; \ x \ ; \ T$
**by** (*metis add-commutative less-eq-def mult-associative star.circ-back-loop-fixpoint star.circ-left-top tarski top-mult-top*)

**lemma** *tarski-top-omega-below*: $x \ ; \ T \leq (x \ ; \ T)^\omega$
**by** (*metis mult-associative omega-induct-mult order-refl tarski-mult-top-idempotent*)

**lemma** *tarski-top-omega*: $x \ ; \ T = (x \ ; \ T)^\omega$
**by** (*metis antisym mult-top-omega tarski-top-omega-below*)

**lemma** *tarski-below-top-omega*: $x \leq (x \ ; \ T)^\omega$
**by** (*metis tarski-top-omega top-right-mult-increasing*)

**lemma** *tarski-mult-omega-omega*: $(x \ ; \ y^\omega)^\omega = x \ ; \ y^\omega$
**by** (*metis mult-associative omega-vector tarski-top-omega*)

**lemma** *tarski-omega-idempotent*: $x^{\omega\omega} = x^\omega$
**by** (*metis omega-vector tarski-top-omega*)

**lemma** *while-denest-2a*: $w \ ; \ ((x * (y \ ; \ w)) * z) = w \ ; \ (((x * y) \ ; \ w) * z)$
**proof** −
**have** $(x^\omega + x^* \ ; \ y \ ; \ w)^\omega = (x^* \ ; \ y \ ; \ w)^* \ ; \ x^\omega \ ; \ (((x^* \ ; \ y \ ; \ w)^* \ ; \ x^\omega)^\omega + ((x^* \ ; \ y \ ; \ w)^* \ ; \ x^\omega)^* \ ; \ (x^* \ ; \ y \ ; \ w)^\omega) + (x^* \ ; \ y \ ; \ w)^\omega$
**by** (*metis add-commutative omega-decompose omega-loop-fixpoint*)
**also have** $... \leq (x^* \ ; \ y \ ; \ w)^* \ ; \ x^\omega + (x^* \ ; \ y \ ; \ w)^\omega$
**by** (*metis add-left-isotone mult-associative mult-right-isotone omega-sub-vector*)
**finally have** *1*: $w \ ; \ (x^\omega + x^* \ ; \ y \ ; \ w)^\omega \leq (w \ ; \ x^* \ ; \ y)^* \ ; \ w \ ; \ x^\omega + (w \ ; \ x^* \ ; \ y)^\omega$
**by** (*smt add-commutative less-eq-def mult-associative mult-left-dist-add while-def while-slide*)
**have** $(x^\omega + x^* \ ; \ y \ ; \ w)^* \ ; \ z = (x^* \ ; \ y \ ; \ w)^* \ ; \ x^\omega \ ; \ ((x^* \ ; \ y \ ; \ w)^* \ ; \ x^\omega)^* \ ; \ (x^* \ ; \ y \ ; \ w)^* \ ; \ z + (x^* \ ; \ y \ ; \ w)^* \ ; \ z$
**by** (*smt add-commutative mult-associative star.circ-add star.circ-loop-fixpoint*)
**also have** $... \leq (x^* \ ; \ y \ ; \ w)^* \ ; \ x^\omega + (x^* \ ; \ y \ ; \ w)^* \ ; \ z$
**by** (*smt add-commutative add-right-isotone mult-associative mult-right-isotone omega-sub-vector*)
**finally have** $w \ ; \ (x^\omega + x^* \ ; \ y \ ; \ w)^* \ ; \ z \leq (w \ ; \ x^* \ ; \ y)^* \ ; \ w \ ; \ x^\omega + (w \ ; \ x^* \ ; \ y)^* \ ; \ w \ ; \ z$
**by** (*metis mult-associative mult-left-dist-add mult-right-isotone star.circ-slide*)
**hence** $w \ ; \ (x^\omega + x^* \ ; \ y \ ; \ w)^\omega + w \ ; \ (x^\omega + x^* \ ; \ y \ ; \ w)^* \ ; \ z \leq (w \ ; \ x^* \ ; \ y)^* \ ; \ (w \ ; \ x^\omega)^\omega + (w \ ; \ x^* \ ; \ y)^\omega + (w \ ; \ x^* \ ; \ y)^* \ ; \ w \ ; \ z$ **using** *1*
**by** (*smt add-associative add-commutative less-eq-def mult-associative tarski-mult-omega-omega*)
**also have** $... \leq (w \ ; \ x^\omega + w \ ; \ x^* \ ; \ y)^* \ ; \ (w \ ; \ x^\omega + w \ ; \ x^* \ ; \ y)^\omega + (w \ ; \ x^\omega + w \ ; \ x^* \ ; \ y)^\omega + (w \ ; \ x^\omega + w \ ; \ x^* \ ; \ y)^* \ ; \ w \ ; \ z$
**by** (*metis add-isotone add-left-upper-bound add-right-upper-bound mult-isotone mult-left-isotone omega-isotone star.circ-isotone*)
**also have** $... = (w \ ; \ x^\omega + w \ ; \ x^* \ ; \ y)^\omega + (w \ ; \ x^\omega + w \ ; \ x^* \ ; \ y)^* \ ; \ w \ ; \ z$
**by** (*metis add-idempotent star-mult-omega*)
**finally have** $w \ ; \ ((x^\omega + x^* \ ; \ y \ ; \ w)^\omega + (x^\omega + x^* \ ; \ y \ ; \ w)^* \ ; \ z) \leq w \ ; \ ((x^\omega + x^* \ ; \ y) \ ; \ w)^\omega + w \ ; \ ((x^\omega + x^* \ ; \ y) \ ; \ w)^* \ ; \ z$
**by** (*smt mult-associative mult-left-dist-add omega-slide star.circ-slide*)
**hence** *2*: $w \ ; \ ((x * (y \ ; \ w)) * z) \leq w \ ; \ (((x * y) \ ; \ w) * z)$
**by** (*smt mult-associative mult-left-dist-add while-def while-slide*)
**have** $w \ ; \ (((x * y) \ ; \ w) * z) \leq w \ ; \ ((x * (y \ ; \ w)) * z)$

**by** (*metis mult-right-isotone while-left-isotone while-sub-associative*)
**thus** *?thesis* **using** *2*
**by** (*metis antisym*)
**qed**

**lemma** *while-denest-3*: $(x * w) * x^\omega = (x * w)^\omega$
**proof** −
**have** *1*: $(x * w) * x^\omega = (x * w)^\omega + (x * w)^* \ ; \ x^{\omega\omega}$
**by** (*metis tarski-omega-idempotent while-def*)
**also have** ... $\leq (x * w)^\omega + (x * w)^* \ ; \ (x^\omega + x^* \ ; \ w)^\omega$
**by** (*metis add-left-upper-bound add-right-isotone mult-right-isotone omega-isotone*)
**also have** ... $= (x * w)^\omega$
**by** (*metis add-idempotent star-mult-omega while-def*)
**finally show** *?thesis* **using** *1*
**by** (*metis add-left-upper-bound antisym-conv*)
**qed**

**lemma** *while-denest-4a*: $(x * w) * (x * (y \ ; \ z)) = (x * w) * ((x * y) \ ; \ z)$
**proof** −
**have** $(x * w) * (x * (y \ ; \ z)) = (x * w)^\omega + ((x * w) * (x^* \ ; \ y \ ; \ z))$
**by** (*smt mult-associative while-denest-3 while-def while-left-dist-add*)
**also have** ... $\leq (x * w)^\omega + ((x * w) * ((x * y) \ ; \ z))$
**by** (*metis add-right-isotone mult-left-isotone star-below-while while-right-isotone*)
**finally have** *1*: $(x * w) * (x * (y \ ; \ z)) \leq (x * w) * ((x * y) \ ; \ z)$
**by** (*smt add-left-upper-bound less-eq-def while-def*)
**have** $(x * w) * ((x * y) \ ; \ z) \leq (x * w) * (x * (y \ ; \ z))$
**by** (*metis while-right-isotone while-sub-associative*)
**thus** *?thesis* **using** *1*
**by** (*metis antisym*)
**qed**

**subclass** *binary-itering-1-T*
**apply** *unfold-locales*
**by** (*smt mult-associative while-denest-2a while-denest-4a while-increasing while-slide*)

**lemma** *while-mult-top*: $(x \ ; \ T) * z = z + x \ ; \ T$
**proof** −
**have** *1*: $z + x \ ; \ T \leq (x \ ; \ T) * z$
**by** (*metis add-least-upper-bound while-denest-1 while-increasing while-one-top*)
**have** $(x \ ; \ T) * z = z + x \ ; \ T \ ; \ ((x \ ; \ T) * z)$
**by** (*metis while-left-unfold*)
**also have** ... $\leq z + x \ ; \ T$
**by** (*metis add-right-isotone mult-associative mult-right-isotone top-greatest*)
**finally show** *?thesis* **using** *1*
**by** (*metis antisym*)
**qed**

**lemma** *tarski-top-omega-below-2*: $x\ ;\ T \leq (x\ ;\ T) * 0$
**by** (*metis add-right-divisibility while-mult-top*)

**lemma** *tarski-top-omega-2*: $x\ ;\ T = (x\ ;\ T) * 0$
**by** (*metis add-left-zero while-mult-top*)

**lemma** *tarski-below-top-omega-2*: $x \leq (x\ ;\ T) * 0$
**by** (*metis tarski-top-omega-2 top-right-mult-increasing*)

**end**

**class** *nonstrict-itering-tarski-zero* = *nonstrict-itering-tarski* + *nonstrict-itering-zero*

**begin**

**lemma** $1 = (x\ ;\ 0) * 1$
**by** (*metis mult-right-zero while-zero*)

**end**

**context** *binary-itering-1-T*

**begin**

**end**

**end**