Quantales

Georg Struth

December 7, 2022

Abstract

These mathematical components formalise basic properties of quantales, together with some important models, constructions, and concepts, including quantic nuclei and conuclei.

Contents

1 Introductory Remarks

Quantales are complete lattices equipped with an associative composition that preserves suprema in both arguments. They have been used—under various names and in various guises—in mathematics for almost a century. One important context is the structure of ideals in rings and C^*-algebras, another one the foundations of quantum mechanics, a third one lies in approaches to generalised metric spaces. In computing, quantales occur naturally in program semantics—algebras of predicate transformers, for instance, form quantales, the semantics of linear logic, the foundations of fuzzy systems and program construction; but also languages or binary relations form quantales.

These components formalise the basic concepts and properties of quantales, following by and large Rosenthal's monograph [6]. Because of applications to predicate transformer semantics, families of quantales are considered in which certain Sup-preservation laws are absent (nomenclature diverges from Rosenthal, but is consistent with AFP entries for dioids and Kleene algebras [2]). Beyond basic equational reasoning, some models of quantales are presented, though those that arise from ring theory or C^*-algebras are currently not supported.

Nuclei and conuclei of quantales are also investigated, and some important relationships with quotients and subalgebras of quantales are formalised, following Rosenthal. In particular, I (re)prove his representation theorem that every quantale is isomorphic to a nucleus of a powerset quantale over some semigroup. Beyond that it is shown how left-sided elements give rise to nuclei and conuclei.

Another subject of study are quantale-modules, which have been introduced by Abramsky and Vickers [1] and widely used since, with some original results on semidirect products over these [4] and some new results on the Kleene star in this setting.

Quantales draw heavily on lattice and order theory, Galois connections and the associated monads and comonads. They are also strongly related to complete Heyting algebras, frames and locales [5], for which future AFP entries might be worth creating. Further variants, such as Girard quantales, might also be worth exploring.

2 Quantales

theory *Quantales*
imports
Order-Lattice-Props.Closure-Operators
Kleene-Algebra.Dioid
begin

2.1 Families of Proto-Quantales

Proto-Quanales are complete lattices equipped with an operation of composition or multiplication that need not be associative. The notation in this component differs from Rosenthal's [6], but is consistent with the one we use for semirings and Kleene algebras.

class *proto-near-quantale* = *complete-lattice* + *times* +
assumes *Sup-distr*: $\bigsqcup X \cdot y = (\bigsqcup x \in X.\ x \cdot y)$

lemma *Sup-pres-multr*: *Sup-pres* $(\lambda(z::'a::proto\text{-}near\text{-}quantale).\ z \cdot y)$
⟨*proof*⟩

lemma *sup-pres-multr*: *sup-pres* $(\lambda(z::'a::proto\text{-}near\text{-}quantale).\ z \cdot y)$
⟨*proof*⟩

lemma *bot-pres-multr*: *bot-pres* $(\lambda(z::'a::proto\text{-}near\text{-}quantale).\ z \cdot y)$
⟨*proof*⟩

context *proto-near-quantale*
begin

lemma *mult-botl* [*simp*]: $\bot \cdot x = \bot$
⟨*proof*⟩

lemma *sup-distr*: $(x \sqcup y) \cdot z = (x \cdot z) \sqcup (y \cdot z)$
⟨*proof*⟩

lemma *mult-isor*: $x \le y \Longrightarrow x \cdot z \le y \cdot z$
⟨*proof*⟩

Left and right residuals can be defined in every proto-nearquantale.

definition *bres* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixr** $\rightarrow$ *60*) **where**
$x \rightarrow z = \bigsqcup\{y.\ x \cdot y \le z\}$

definition *fres* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixl** $\leftarrow$ *60*) **where**
$z \leftarrow y = \bigsqcup\{x.\ x \cdot y \le z\}$

The left one is a right adjoint to composition. For the right one, additional assumptions are needed

lemma *bres-galois-imp*: $x \cdot y \le z \Longrightarrow y \le x \rightarrow z$

⟨*proof*⟩

lemma *fres-galois*: $(x \cdot y \leq z) = (x \leq z \leftarrow y)$
⟨*proof*⟩

end

lemma *fres-adj*: $(\lambda(x::'a::proto\text{-}near\text{-}quantale).\ x \cdot y) \dashv (\lambda x.\ x \leftarrow y)$
⟨*proof*⟩

context *proto-near-quantale*
begin

lemma *fres-canc1*: $(y \leftarrow x) \cdot x \leq y$
⟨*proof*⟩

lemma *fres-canc2*: $y \leq (y \cdot x) \leftarrow x$
⟨*proof*⟩

lemma *inf-fres*: $y \cdot x = \bigsqcap\{z.\ y \leq z \leftarrow x\}$
⟨*proof*⟩

lemma *bres-iso*: $x \leq y \Longrightarrow z \rightarrow x \leq z \rightarrow y$
⟨*proof*⟩

lemma *bres-anti*: $x \leq y \Longrightarrow y \rightarrow z \leq x \rightarrow z$
⟨*proof*⟩

lemma *fres-iso*: $x \leq y \Longrightarrow x \leftarrow z \leq y \leftarrow z$
⟨*proof*⟩

lemma *bres-top-top* [*simp*]: $\top \rightarrow \top = \top$
⟨*proof*⟩

lemma *fres-top-top* [*simp*]: $\top \leftarrow \top = \top$
⟨*proof*⟩

lemma *bres-bot-bot* [*simp*]: $\bot \rightarrow \bot = \top$
⟨*proof*⟩

lemma *left-sided-localp*: $\top \cdot y = y \Longrightarrow x \cdot y \leq y$
⟨*proof*⟩

lemma *fres-sol*: $((y \leftarrow x) \cdot x = y) = (\exists z.\ z \cdot x = y)$
⟨*proof*⟩

lemma *sol-fres*: $((y \cdot x) \leftarrow x = y) = (\exists z.\ y = z \leftarrow x)$
⟨*proof*⟩

end

class *proto-pre-quantale* = *proto-near-quantale* +
  assumes *Sup-subdistl*: $(\bigsqcup y \in Y.\ x \cdot y) \leq x \cdot \bigsqcup Y$

begin

lemma *sup-subdistl*: $(x \cdot y) \sqcup (x \cdot z) \leq x \cdot (y \sqcup z)$
  ⟨*proof*⟩

lemma *mult-isol*: $x \leq y \Longrightarrow z \cdot x \leq z \cdot y$
  ⟨*proof*⟩

lemma *fres-anti*: $x \leq y \Longrightarrow z \leftarrow y \leq z \leftarrow x$
  ⟨*proof*⟩

end

class *weak-proto-quantale* = *proto-near-quantale* +
  assumes *weak-Sup-distl*: $Y \neq \{\} \Longrightarrow x \cdot \bigsqcup Y = (\bigsqcup y \in Y.\ x \cdot y)$

begin

subclass *proto-pre-quantale*
⟨*proof*⟩

lemma *sup-distl*: $x \cdot (y \sqcup z) = (x \cdot y) \sqcup (x \cdot z)$
  ⟨*proof*⟩

lemma $y \leq x \rightarrow z \longrightarrow x \cdot y \leq z$
⟨*proof*⟩

end

class *proto-quantale* = *proto-near-quantale* +
  assumes *Sup-distl*: $x \cdot \bigsqcup Y = (\bigsqcup y \in Y.\ x \cdot y)$

lemma *Sup-pres-multl*: *Sup-pres* $(\lambda(z::'a::\mathit{proto\text{-}quantale}).\ x \cdot z)$
  ⟨*proof*⟩

lemma *sup-pres-multl*: *sup-pres* $(\lambda(z::'a::\mathit{proto\text{-}quantale}).\ x \cdot z)$
  ⟨*proof*⟩

lemma *bot-pres-multl*: *bot-pres* $(\lambda(z::'a::\mathit{proto\text{-}quantale}).\ x \cdot z)$
  ⟨*proof*⟩

context *proto-quantale*
begin

subclass *weak-proto-quantale*
⟨*proof*⟩

lemma *mult-botr* [*simp*]: $x \cdot \bot = \bot$
⟨*proof*⟩

Now there is also an adjunction for the other residual.

lemma *bres-galois*: $x \cdot y \leq z \longleftrightarrow y \leq x \rightarrow z$
⟨*proof*⟩

end

lemma *bres-adj*: $(\lambda(y::'a::proto\text{-}quantale).\ x \cdot y) \dashv (\lambda y.\ x \rightarrow y)$
⟨*proof*⟩

context *proto-quantale*
begin

lemma *bres-canc1*: $x \cdot (x \rightarrow y) \leq y$
⟨*proof*⟩

lemma *bres-canc2*: $y \leq x \rightarrow (x \cdot y)$
⟨*proof*⟩

lemma *inf-bres*: $x \cdot y = \bigsqcap\{z.\ y \leq x \rightarrow z\}$
⟨*proof*⟩

lemma *bres-sol*: $(x \cdot (x \rightarrow y) = y) = (\exists z.\ x \cdot z = y)$
⟨*proof*⟩

lemma *sol-bres*: $(x \rightarrow (x \cdot y) = y) = (\exists z.\ y = x \rightarrow z)$
⟨*proof*⟩

end

lemma *bres-fres-clop*: $clop\ (\lambda x::'a::proto\text{-}quantale.\ y \leftarrow (x \rightarrow y))$
⟨*proof*⟩

lemma *fres-bres-clop*: $clop\ (\lambda x::'a::proto\text{-}quantale.\ (y \leftarrow x) \rightarrow y)$
⟨*proof*⟩

2.2 Families of Quantales

class *near-quantale* = *proto-near-quantale* + *semigroup-mult*

sublocale *near-quantale* ⊆ *nsrnq*: *near-dioid* $(\sqcup)$ $(\cdot)$ $(\leq)$ $(<)$
⟨*proof*⟩

context *near-quantale*

begin

lemma *fres-curry*: $(z \leftarrow y) \leftarrow x = z \leftarrow (x \cdot y)$
  ⟨*proof*⟩

end

class *unital-near-quantale* = *near-quantale* + *monoid-mult*

sublocale *unital-near-quantale* $\subseteq$ *nsrnqo*: *near-dioid-one* $(\sqcup)$ $(\cdot)$ *1*$(\leq)$ $(<)$
  ⟨*proof*⟩

context *unital-near-quantale*
begin

definition *iter* :: $'a \Rightarrow 'a$ **where**
  iter $x \equiv \prod i.\ x\ \widehat{}\ i$

lemma *iter-ref* [*simp*]: *iter* $x \leq$ *1*
  ⟨*proof*⟩

lemma *le-top*: $x \leq \top \cdot x$
  ⟨*proof*⟩

lemma *top-times-top* [*simp*]: $\top \cdot \top = \top$
  ⟨*proof*⟩

lemma *bres-one*: *1* $\leq x \rightarrow x$
  ⟨*proof*⟩

lemma *fres-one*: *1* $\leq x \leftarrow x$
  ⟨*proof*⟩

end

class *pre-quantale* = *proto-pre-quantale* + *semigroup-mult*

begin

subclass *near-quantale* ⟨*proof*⟩

lemma *fres-interchange*: $z \cdot (x \leftarrow y) \leq (z \cdot x) \leftarrow y$
  ⟨*proof*⟩

end

sublocale *pre-quantale* $\subseteq$ *psrpq*: *pre-dioid* $(\sqcup)$ $(\cdot)$ $(\leq)$ $(<)$
  ⟨*proof*⟩

class *unital-pre-quantale* = *pre-quantale* + *monoid-mult*

begin

subclass *unital-near-quantale* ⟨*proof*⟩

Abstract rules of Hoare logic without the star can be derived.

lemma *h-w1*: $x \leq x' \Longrightarrow \; x' \cdot y \leq z \Longrightarrow x \cdot y \leq z$
⟨*proof*⟩

lemma *h-w2*: $x \cdot y \leq z' \Longrightarrow z' \leq z \Longrightarrow x \cdot y \leq z$
⟨*proof*⟩

lemma *h-seq*: $x \cdot v \leq z \Longrightarrow y \cdot w \leq v \Longrightarrow x \cdot y \cdot w \leq z$
⟨*proof*⟩

lemma *h-sup*: $x \cdot w \leq z \Longrightarrow y \cdot w \leq z \Longrightarrow (x \sqcup y) \cdot w \leq z$
⟨*proof*⟩

lemma *h-Sup*: $\forall\, x \in X.\; x \cdot w \leq z \Longrightarrow \bigsqcup X \cdot w \leq z$
⟨*proof*⟩

end

sublocale *unital-pre-quantale* ⊆ *psrpqo*: *pre-dioid-one* $(\sqcup)$ $(\cdot)$ *1* $(\leq)$ $(<)$⟨*proof*⟩

class *weak-quantale* = *weak-proto-quantale* + *semigroup-mult*

begin

subclass *pre-quantale* ⟨*proof*⟩

The following counterexample shows an important consequence of weakness: the absence of right annihilation.

lemma $x \cdot \bot = \bot$
⟨*proof*⟩

end

class *unital-weak-quantale* = *weak-quantale* + *monoid-mult*

lemma (**in** *unital-weak-quantale*) $x \cdot \bot = \bot$
⟨*proof*⟩

subclass (**in** *unital-weak-quantale*) *unital-pre-quantale* ⟨*proof*⟩

sublocale *unital-weak-quantale* ⊆ *wswq*: *dioid-one-zerol* $(\sqcup)$ $(\cdot)$ *1* $\bot$ $(\leq)$ $(<)$
⟨*proof*⟩

class *quantale* = *proto-quantale* + *semigroup-mult*

begin

subclass *weak-quantale* ⟨*proof*⟩

lemma *Inf-subdistl*: $x \cdot \bigsqcap Y \leq (\bigsqcap y \in Y.\ x \cdot y)$
⟨*proof*⟩

lemma *Inf-subdistr*: $\bigsqcap\ X \cdot y \leq (\bigsqcap x \in X.\ x \cdot y)$
⟨*proof*⟩

lemma *fres-bot-bot* [*simp*]: $\bot \leftarrow \bot = \top$
⟨*proof*⟩

lemma *bres-interchange*: $(x \rightarrow y) \cdot z \leq x \rightarrow (y \cdot z)$
⟨*proof*⟩

lemma *bres-curry*: $x \rightarrow (y \rightarrow z) = (y \cdot x) \rightarrow z$
⟨*proof*⟩

lemma *fres-bres*: $x \rightarrow (y \leftarrow z) = (x \rightarrow y) \leftarrow z$
⟨*proof*⟩

end

class *quantale-with-dual* = *quantale* + *complete-lattice-with-dual*

class *unital-quantale* = *quantale* + *monoid-mult*

class *unital-quantale-with-dual* = *unital-quantale* + *quantale-with-dual*

subclass (**in** *unital-quantale*) *unital-weak-quantale* ⟨*proof*⟩

sublocale *unital-quantale* ⊆ *wswq*: *dioid-one-zero* $(\sqcup)$ $(\cdot)$ *1* $\bot$ $(\leq)$ $(<)$
⟨*proof*⟩

class *ab-quantale* = *quantale* + *ab-semigroup-mult*

begin

lemma *bres-fres-eq*: $x \rightarrow y = y \leftarrow x$
⟨*proof*⟩

end

class *ab-unital-quantale* = *ab-quantale* + *unital-quantale*

sublocale *complete-heyting-algebra* $\subseteq$ *chaq*: *ab-unital-quantale* ($\sqcap$) - - - - - - - - $\top$
⟨*proof*⟩

class *distrib-quantale* = *quantale* + *distrib-lattice*

class *bool-quantale* = *quantale* + *complete-boolean-algebra-alt*

class *distrib-unital-quantale* = *unital-quantale* + *distrib-lattice*

class *bool-unital-quantale* = *unital-quantale* + *complete-boolean-algebra-alt*

class *distrib-ab-quantale* = *distrib-quantale* + *ab-quantale*

class *bool-ab-quantale* = *bool-quantale* + *ab-quantale*

class *distrib-ab-unital-quantale* = *distrib-quantale* + *unital-quantale*

class *bool-ab-unital-quantale* = *bool-ab-quantale* + *unital-quantale*

sublocale *complete-boolean-algebra* $\subseteq$ *cba-quantale*: *bool-ab-unital-quantale* *inf* - - - - - - - - - - $\top$
⟨*proof*⟩

context *complete-boolean-algebra*
begin

In this setting, residuation is classical implication.

lemma *cba-bres1*: $x \sqcap y \leq z \longleftrightarrow x \leq$ *cba-quantale.bres* $y\ z$
⟨*proof*⟩

lemma *cba-bres2*: $x \leq -y \sqcup z \longleftrightarrow x \leq$ *cba-quantale.bres* $y\ z$
⟨*proof*⟩

lemma *cba-bres-prop*: *cba-quantale.bres* $x\ y = -x \sqcup y$
⟨*proof*⟩

end

2.3 Quantales Based on Sup-Lattices and Inf-Lattices

These classes are defined for convenience in instantiation and interpretation proofs, or likewise. They are useful, e.g., in the context of predicate transformers, where only one of Sup or Inf may be well behaved.

class *Sup-quantale* = *Sup-lattice* + *semigroup-mult* +
assumes *Supq-distr*: $\bigsqcup X \cdot y = (\bigsqcup x \in X.\ x \cdot y)$
and *Supq-distl*: $x \cdot \bigsqcup Y = (\bigsqcup y \in Y.\ x \cdot y)$

class *unital-Sup-quantale* = *Sup-quantale* + *monoid-mult*

class *Inf-quantale* = *Inf-lattice* + *monoid-mult* +
assumes *Supq-distr*: $\prod X \cdot y = (\prod x \in X.\ x \cdot y)$
and *Supq-distl*: $x \cdot \prod Y = (\prod y \in Y.\ x \cdot y)$

class *unital-Inf-quantale* = *Inf-quantale* + *monoid-mult*

sublocale *Inf-quantale* ⊆ *qdual*: *Sup-quantale* - *Inf* (≥)
⟨*proof*⟩

sublocale *unital-Inf-quantale* ⊆ *uqdual*: *unital-Sup-quantale* - - *Inf* (≥)⟨*proof*⟩

sublocale *Sup-quantale* ⊆ *supq*: *quantale* - *Infs Sup-class.Sup infs* (≤) *le sups bots tops*
⟨*proof*⟩

sublocale *unital-Sup-quantale* ⊆ *usupq*: *unital-quantale* - - *Infs Sup-class.Sup infs* (≤) *le sups bots tops*⟨*proof*⟩

2.4 Products of Quantales

definition *Inf-prod* $X = ((\prod x \in X.\ fst\ x), (\prod x \in X.\ snd\ x))$

definition *inf-prod* $x\ y = (fst\ x \sqcap fst\ y,\ snd\ x \sqcap snd\ y)$

definition *bot-prod* = (*bot*,*bot*)

definition *Sup-prod* $X = ((\bigsqcup x \in X.\ fst\ x), (\bigsqcup x \in X.\ snd\ x))$

definition *sup-prod* $x\ y = (fst\ x \sqcup fst\ y,\ snd\ x \sqcup snd\ y)$

definition *top-prod* = (*top*,*top*)

definition *less-eq-prod* $x\ y \equiv$ *less-eq* $(fst\ x)\ (fst\ y) \wedge$ *less-eq* $(snd\ x)\ (snd\ y)$

definition *less-prod* $x\ y \equiv$ *less-eq* $(fst\ x)\ (fst\ y) \wedge$ *less-eq* $(snd\ x)\ (snd\ y) \wedge x \neq y$

definition *times-prod'* $x\ y = (fst\ x \cdot fst\ y,\ snd\ x \cdot snd\ y)$

definition *one-prod* = (*1*,*1*)

definition *dual-prod* $x = (\partial\ (fst\ x), \partial\ (snd\ x))$

interpretation *prod*: *complete-lattice Inf-prod Sup-prod inf-prod less-eq-prod less-prod sup-prod bot-prod top-prod* :: (*'a*::*complete-lattice* × *'b*::*complete-lattice*)
⟨*proof*⟩

interpretation *prod*: *complete-lattice-with-dual Inf-prod Sup-prod inf-prod less-eq-prod less-prod sup-prod bot-prod top-prod* :: (*'a*::*complete-lattice-with-dual* × *'b*::*complete-lattice-with-dual*)

dual-prod
⟨*proof*⟩

interpretation *prod*: *proto-near-quantale Inf-prod Sup-prod inf-prod less-eq-prod less-prod sup-prod bot-prod top-prod* :: (*'a*::*proto-near-quantale* × *'b*::*proto-near-quantale*) *times-prod'*
⟨*proof*⟩

interpretation *prod*: *proto-quantale Inf-prod Sup-prod inf-prod less-eq-prod less-prod sup-prod bot-prod top-prod* :: (*'a*::*proto-quantale* × *'b*::*proto-quantale*) *times-prod'*
⟨*proof*⟩

interpretation *prod*: *unital-quantale one-prod times-prod' Inf-prod Sup-prod inf-prod less-eq-prod less-prod sup-prod bot-prod top-prod* :: (*'a*::*unital-quantale* × *'b*::*unital-quantale*)

⟨*proof*⟩

2.5 Quantale Morphisms

There are various ways of defining quantale morphisms, depending on the application. Following Rosenthal, I present the most important one.

abbreviation *comp-pres* :: (*'a*::*times* ⇒ *'b*::*times*) ⇒ *bool* **where**
comp-pres f ≡ (∀ *x y*. *f* (*x* · *y*) = *f x* · *f y*)

abbreviation *un-pres* :: (*'a*::*one* ⇒ *'b*::*one*) ⇒ *bool* **where**
un-pres f ≡ (*f 1* = *1*)

definition *comp-closed-set X* = (∀ *x* ∈ *X*. ∀ *y* ∈ *X*. *x* · *y* ∈ *X*)

definition *un-closed-set X* = (*1* ∈ *X*)

definition *quantale-homset* :: (*'a*::*quantale* ⇒ *'b*::*quantale*) *set* **where**
quantale-homset = {*f*. *comp-pres f* ∧ *Sup-pres f*}

lemma *quantale-homset-iff*: *f* ∈ *quantale-homset* = (*comp-pres f* ∧ *Sup-pres f*)
⟨*proof*⟩

definition *unital-quantale-homset* :: (*'a*::*unital-quantale* ⇒ *'b*::*unital-quantale*) *set* **where**
unital-quantale-homset = {*f*. *comp-pres f* ∧ *Sup-pres f* ∧ *un-pres f*}

lemma *unital-quantale-homset-iff*: *f* ∈ *unital-quantale-homset* = (*comp-pres f* ∧ *Sup-pres f* ∧ *un-pres f*)
⟨*proof*⟩

Though Infs can be defined from Sups in any quantale, quantale morphisms do not generally preserve Infs. A different kind of morphism is needed if this is to be guaranteed.

lemma $f \in$ *quantale-homset* $\Longrightarrow$ *Inf-pres f*
⟨*proof*⟩

The images of quantale morphisms are closed under compositions and Sups, hence they form quantales.

lemma *quantale-hom-q-pres*: $f \in$ *quantale-homset* $\Longrightarrow$ *Sup-closed-set* (*range f*) $\wedge$ *comp-closed-set* (*range f*)
⟨*proof*⟩

Yet the image need not be Inf-closed.

lemma $f \in$ *quantale-homset* $\Longrightarrow$ *Inf-closed-set* (*range f*)
⟨*proof*⟩

Of course Sups are preserved by quantale-morphisms, hence they are the same in subsets as in the original set. Infs in the subset, however, exist, since they subset forms a quantale in which Infs can be defined, but these are generally different from the Infs in the superstructure.

This fact is hidden in Isabelle's definition of complete lattices, where Infs are axiomatised. There is no easy way in general to show that images of quantale morphisms form quantales, though the statement for Sup-quantales is straightforward. I show this for quantic nuclei and left-sided elements.

typedef (**overloaded**) ('*a*,'*b*) *quantale-homset* = *quantale-homset*::('*a*::*quantale* $\Rightarrow$ '*b*::*quantale*) *set*
⟨*proof*⟩

setup-lifting *type-definition-quantale-homset*

Interestingly, the following type is not (gobally) inhabited.

typedef (**overloaded**) ('*a*,'*b*) *unital-quantale-homset* = *unital-quantale-homset*::('*a*::*unital-quantale* $\Rightarrow$ '*b*::*unital-quantale*) *set*
⟨*proof*⟩

lemma *quantale-hom-radj*:
fixes *f* :: '*a*::*quantale-with-dual* $\Rightarrow$ '*b*::*quantale-with-dual*
shows $f \in$ *quantale-homset* $\Longrightarrow f \dashv$ *radj f*
⟨*proof*⟩

lemma *quantale-hom-prop1*:
fixes *f* :: '*a*::*quantale-with-dual* $\Rightarrow$ '*b*::*quantale-with-dual*
shows $f \in$ *quantale-homset* $\Longrightarrow$ *radj f* $(f\ x \rightarrow y) = x \rightarrow$ *radj f y*
⟨*proof*⟩

lemma *quantale-hom-prop2*:
fixes *f* :: '*a*::*quantale-with-dual* $\Rightarrow$ '*b*::*quantale-with-dual*
shows $f \in$ *quantale-homset* $\Longrightarrow$ *radj f* $(y \leftarrow f\ x) =$ *radj f y* $\leftarrow x$
⟨*proof*⟩

definition *quantale-closed-maps* :: (*'a*::*quantale* ⇒ *'b*::*quantale*) *set* **where**
quantale-closed-maps = $\{f.\ (\forall\, x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y))\}$

lemma *quantale-closed-maps-iff*: $f \in$ *quantale-closed-maps* = $(\forall\ x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y))$
⟨*proof*⟩

definition *quantale-closed-Sup-maps* :: (*'a*::*quantale* ⇒ *'b*::*quantale*) *set* **where**
quantale-closed-Sup-maps = $\{f.\ (\forall\ x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y)) \wedge$ *Sup-pres* $f\}$

lemma *quantale-closed-Sup-maps-iff*: $f \in$ *quantale-closed-Sup-maps* = $(\forall\ x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y) \wedge$ *Sup-pres* $f)$
⟨*proof*⟩

definition *quantale-closed-unital-maps* :: (*'a*::*unital-quantale* ⇒ *'b*::*unital-quantale*) *set* **where**
quantale-closed-unital-maps = $\{f.\ (\forall\ x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y)) \wedge 1 \leq f\ 1\}$

lemma *quantale-closed-unital-maps-iff*: $f \in$ *quantale-closed-unital-maps* = $(\forall\ x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y) \wedge 1 \leq f\ 1)$
⟨*proof*⟩

definition *quantale-closed-unital-Sup-maps* :: (*'a*::*unital-quantale* ⇒ *'b*::*unital-quantale*) *set* **where**
quantale-closed-unital-Sup-maps = $\{f.\ (\forall\ x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y)) \wedge$ *Sup-pres* $f \wedge 1 \leq f\ 1\}$

lemma *quantale-closed-unital-Sup-maps-iff*: $f \in$ *quantale-closed-unital-Sup-maps* = $(\forall\ x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y) \wedge$ *Sup-pres* $f \wedge 1 \leq f\ 1)$
⟨*proof*⟩

Closed maps are the right adjoints of quantale morphisms.

lemma *quantale-hom-closed-map*:
fixes f :: *'a*::*quantale-with-dual* ⇒ *'b*::*quantale-with-dual*
shows ($f \in$ *quantale-homset*) ⟹ (*radj* $f \in$ *quantale-closed-maps*)
⟨*proof*⟩

lemma *unital-quantale-hom-closed-unital-map*:
fixes f :: *'a*::*unital-quantale-with-dual* ⇒ *'b*::*unital-quantale-with-dual*
shows ($f \in$ *unital-quantale-homset*) ⟹ (*radj* $f \in$ *quantale-closed-unital-maps*)
⟨*proof*⟩

end

3 Star and Fixpoints in Quantales

theory *Quantale-Star*
imports *Quantales*
Kleene-Algebra.Kleene-Algebra

begin

This component formalises properties of the star in quantales. For pre-quantales these are modelled as fixpoints. For weak quantales they are given by iteration.

3.1 Star and Fixpoints in Pre-Quantales

context *unital-near-quantale*
begin

definition $qiter\text{-}fun\ x\ y = (\sqcup)\ x \circ (\cdot)\ y$

definition $qiterl\ x\ y = lfp\ (qiter\text{-}fun\ x\ y)$

definition $qiterg\ x\ y = gfp\ (qiter\text{-}fun\ x\ y)$

abbreviation $qiterl\text{-}id \equiv qiterl\ 1$

abbreviation $qiterq\text{-}id \equiv qiterg\ 1$

definition $qstar\ x = (\bigsqcup i.\ x\ \widehat{}\ i)$

lemma *qiter-fun-exp*: $qiter\text{-}fun\ x\ y\ z = x \sqcup y \cdot z$
⟨*proof*⟩

end

Many properties of fixpoints have been developed for Isabelle's monotone functions. These carry two type parameters, and must therefore be used outside of contexts.

lemma *iso-qiter-fun*: $mono\ (\lambda z::'a::unital\text{-}pre\text{-}quantale.\ qiter\text{-}fun\ x\ y\ z)$
⟨*proof*⟩

I derive the left unfold and induction laws of Kleene algebra. I link with the Kleene algebra components at the end of this section, to bring properties into scope.

lemma *qiterl-unfoldl* [*simp*]: $x \sqcup y \cdot qiterl\ x\ y = qiterl\ (x::'a::unital\text{-}pre\text{-}quantale)\ y$
⟨*proof*⟩

lemma *qiterg-unfoldl* [*simp*]: $x \sqcup y \cdot qiterg\ x\ y = qiterg\ (x::'a::unital\text{-}pre\text{-}quantale)\ y$
⟨*proof*⟩

lemma *qiterl-inductl*: $x \sqcup y \cdot z \leq z \Longrightarrow qiterl\ (x::'a::unital\text{-}near\text{-}quantale)\ y \leq z$
⟨*proof*⟩

lemma *qiterg-coinductl*: $z \leq x \sqcup y \cdot z \Longrightarrow z \leq qiterg\ (x{::}'a{::}unital\text{-}near\text{-}quantale)$ y
  ⟨*proof*⟩

context *unital-near-quantale*
begin

lemma *powers-distr*: $qstar\ x \cdot y = (\bigsqcup i.\ x\ \widehat{}\ i \cdot y)$
  ⟨*proof*⟩

lemma *Sup-iter-unfold*: $x\ \widehat{}\ 0 \sqcup (\bigsqcup n.\ x\ \widehat{}\ (Suc\ n)) = (\bigsqcup n.\ x\ \widehat{}\ n)$
  ⟨*proof*⟩

lemma *Sup-iter-unfold-var*: $1 \sqcup (\bigsqcup n.\ x \cdot x\ \widehat{}\ n) = (\bigsqcup n.\ x\ \widehat{}\ n)$
  ⟨*proof*⟩

lemma *power-inductr*: $z \sqcup y \cdot x \leq y \Longrightarrow z \cdot x\ \widehat{}\ i \leq y$
  ⟨*proof*⟩

end

context *unital-pre-quantale*
begin

lemma *powers-subdistl*: $(\bigsqcup i.\ x \cdot y\ \widehat{}\ i) \leq x \cdot qstar\ y$
  ⟨*proof*⟩

lemma *qstar-subcomm*: $qstar\ x \cdot x \leq x \cdot qstar\ x$
  ⟨*proof*⟩

lemma *power-inductl*: $z \sqcup x \cdot y \leq y \Longrightarrow x\ \widehat{}\ i \cdot z \leq y$
  ⟨*proof*⟩

end

3.2 Star and Iteration in Weak Quantales

context *unital-weak-quantale*
begin

In unital weak quantales one can derive the star axioms of Kleene algebra for iteration. This generalises the language and relation models from our Kleene algebra components.

lemma *powers-distl*: $x \cdot qstar\ y = (\bigsqcup i.\ x \cdot y\ \widehat{}\ i)$
  ⟨*proof*⟩

lemma *qstar-unfoldl*: $1 \sqcup x \cdot qstar\ x \leq qstar\ x$
  ⟨*proof*⟩

lemma *qstar-comm*: $x \cdot qstar\ x = qstar\ x \cdot x$
⟨*proof*⟩

lemma *qstar-unfoldr* [*simp*]: $1 \sqcup qstar\ x \cdot x = qstar\ x$
⟨*proof*⟩

lemma *qstar-inductl*: $z \sqcup x \cdot y \leq y \Longrightarrow qstar\ x \cdot z \leq y$
⟨*proof*⟩

lemma *qstar-inductr*: $z \sqcup y \cdot x \leq y \Longrightarrow z \cdot qstar\ x \leq y$
⟨*proof*⟩

Hence in this setting one also obtains the right Kleene algebra axioms.

end

sublocale *unital-weak-quantale* ⊆ *uwqlka*: *left-kleene-algebra-zerol* $(\sqcup)$ $(\cdot)$ 1 $\bot$ $(\leq)$ $(<)$ *qstar*
⟨*proof*⟩

sublocale *unital-quantale* ⊆ *uqka*: *kleene-algebra* $(\sqcup)$ $(\cdot)$ 1 $\bot$ $(\leq)$ $(<)$ *qstar*
⟨*proof*⟩

The star is indeed the least fixpoint.

lemma *qstar-qiterl*: *qstar* (*x*::'*a*::*unital-weak-quantale*) = *qiterl-id x*
⟨*proof*⟩

context *unital-weak-quantale*
begin

3.3 Deriving the Star Axioms of Action Algebras

Finally the star axioms of action algebras are derived.

lemma *act-star1*: $1 \sqcup x \sqcup (qstar\ x) \cdot (qstar\ x) \leq (qstar\ x)$
⟨*proof*⟩

lemma (**in** *unital-quantale*) *act-star3*: $qstar\ (x \rightarrow x) \leq x \rightarrow x$
⟨*proof*⟩

lemma *act-star3-var*: $qstar\ (x \leftarrow x) \leq x \leftarrow x$
⟨*proof*⟩

end

An integration of action algebras requires first resolving some notational issues within the components where these algebras are located.

end

4 Quantale Modules and Semidirect Products

theory *Quantale-Modules*
  imports *Quantale-Star*

begin

4.1 Quantale Modules

Quantale modules are extensions of semigroup actions in that a quantale acts on a complete lattice. These have been investigated by Abramsky and Vickers [1] and others, predominantly in the context of pointfree topology.

locale *unital-quantale-module* =
  fixes *act* :: $'a$::*unital-quantale* ⇒ $'b$::*complete-lattice-with-dual* ⇒ $'b$ (α)
  assumes *act1*: $\alpha\ (x \cdot y)\ p = \alpha\ x\ (\alpha\ y\ p)$
    and *act2* [*simp*]: $\alpha\ 1\ p = p$
    and *act3*: $\alpha\ (\bigsqcup X)\ p = (\bigsqcup x \in X.\ \alpha\ x\ p)$
    and *act4*: $\alpha\ x\ (\bigsqcup P) = (\bigsqcup p \in P.\ \alpha\ x\ p)$

context *unital-quantale-module*
begin

Actions are morphisms. The curried notation is particularly convenient for this.

lemma *act-morph1*: $\alpha\ (x \cdot y) = (\alpha\ x) \circ (\alpha\ y)$
  ⟨*proof*⟩

lemma *act-morph2* [*simp*]: $\alpha\ 1 = id$
  ⟨*proof*⟩

lemma *emp-act* [*simp*]: $\alpha\ (\bigsqcup \{\})\ p = \bot$
  ⟨*proof*⟩

lemma *emp-act-var* [*simp*]: $\alpha\ \bot\ p = \bot$
  ⟨*proof*⟩

lemma *act-emp* [*simp*]: $\alpha\ x\ (\bigsqcup \{\}) = \bot$
  ⟨*proof*⟩

lemma *act-emp-var* [*simp*]: $\alpha\ x\ \bot = \bot$
  ⟨*proof*⟩

lemma *act-sup-distl*: $\alpha\ x\ (p \sqcup q) = (\alpha\ x\ p) \sqcup (\alpha\ x\ q)$
  ⟨*proof*⟩

lemma *act-sup-distr*: $\alpha\ (x \sqcup y)\ p = (\alpha\ x\ p) \sqcup (\alpha\ y\ p)$
  ⟨*proof*⟩

lemma *act-sup-distr-var*: $\alpha\ (x \sqcup y) = (\alpha\ x) \sqcup (\alpha\ y)$
⟨*proof*⟩

4.2 Semidirect Products and Weak Quantales

Next, the semidirect product of a unital quantale and a complete lattice is defined.

definition *sd-prod* $x\ y = (fst\ x \cdot fst\ y,\ snd\ x \sqcup \alpha\ (fst\ x)\ (snd\ y))$

lemma *sd-distr*: *sd-prod* (*Sup-prod* X) y = *Sup-prod* $\{$*sd-prod* $x\ y\ |x.\ x \in X\}$
⟨*proof*⟩

lemma *sd-distl-aux*: $Y \neq \{\} \Longrightarrow p \sqcup (\bigsqcup y \in Y.\ \alpha\ x\ (snd\ y)) = (\bigsqcup y \in Y.\ p \sqcup \alpha\ x\ (snd\ y))$
⟨*proof*⟩

lemma *sd-distl*: $Y \neq \{\} \Longrightarrow$ *sd-prod* x (*Sup-prod* Y) = *Sup-prod* $\{$*sd-prod* $x\ y\ |y.\ y \in Y\}$
⟨*proof*⟩

definition *sd-unit* = $(1,\bot)$

lemma *sd-unitl* [*simp*]: *sd-prod sd-unit* $x = x$
⟨*proof*⟩

lemma *sd-unitr* [*simp*]: *sd-prod* x *sd-unit* $= x$
⟨*proof*⟩

The following counterexamples rule out that semidirect products of quantales and complete lattices form quantales. The reason is that the right annihilation law fails.

lemma *sd-prod* x (*Sup-prod* Y) = *Sup-prod* $\{$*sd-prod* $x\ y\ |y.\ y \in Y\}$
⟨*proof*⟩

lemma *sd-prod* x *bot-prod* = *bot-prod*
⟨*proof*⟩

However one can show that semidirect products of (unital) quantales with complete lattices form weak (unital) quantales. This provides an example of how weak quantales arise quite naturally.

sublocale *dp-quantale*: *weak-quantale sd-prod Inf-prod Sup-prod inf-prod less-eq-prod less-prod sup-prod bot-prod top-prod*
⟨*proof*⟩

sublocale *dpu-quantale*: *unital-weak-quantale sd-unit sd-prod Inf-prod Sup-prod inf-prod less-eq-prod less-prod sup-prod bot-prod top-prod*
⟨*proof*⟩

4.3 The Star in Semidirect Products

I define the star operation for a semidirect product of a quantale and a complete lattice, and prove a characteristic property.

abbreviation *sd-power* $\equiv$ *dpu-quantale.power*

abbreviation *sd-star* $\equiv$ *dpu-quantale.qstar*

lemma *sd-power-zero* [*simp*]: *sd-power x 0* $= (1,\bot)$
⟨*proof*⟩

lemma *sd-power-prop-aux*: $\alpha\ (x\ \widehat{}\ 0)\ y \sqcup (\bigsqcup\{\alpha\ (x\ \widehat{}\ k)\ y\ |k.\ \ 0 < k \wedge k \leq Suc\ i\}) = \bigsqcup\{\alpha\ (x\ \widehat{}\ k)\ y\ |k.\ k \leq Suc\ i\}$
⟨*proof*⟩

lemma *sd-power-prop1* [*simp*]: *sd-power* $(x,y)\ (Suc\ i) = (x\ \widehat{}\ (Suc\ i), \bigsqcup\{\alpha\ (x\ \widehat{}\ k)\ y|k.\ k \leq i\})$
⟨*proof*⟩

lemma *sd-power-prop2* [*simp*]: *sd-power* $(x,y)\ i = (x\ \widehat{}\ i, \bigsqcup\{\alpha\ (x\ \widehat{}\ k)\ y|k.\ k < i\})$
⟨*proof*⟩

lemma *sdstar-prop*: *sd-star* $(x,y) = (qstar\ x, \alpha\ (qstar\ x)\ y)$
⟨*proof*⟩

end

end

5 Models of Quantales

theory *Quantale-Models*
imports *Quantales*
Dioid-Models-New

begin

Most of the models in this section expand those of dioids (and hence Kleene algebras). They really belong here, because quantales form a stronger setting than dioids or Kleene algebras. They are, however, subsumed by the general lifting results of partial semigroups and monoids from another AFP entry [3].

5.1 Quantales of Booleans

instantiation *bool* :: *bool-ab-unital-quantale*
begin

definition *one-bool* = *True*

definition *times-bool* = $(\wedge)$

instance
  ⟨*proof*⟩

end

lemma *bool-sep-eq-conj* [*simp*]: $((x :: bool) * y) = (x \wedge y)$
  ⟨*proof*⟩

lemma *bool-impl-eq* [*simp*]: $(x :: bool) \rightarrow y = -x \sqcup y$
  ⟨*proof*⟩

5.2 Powerset Quantales of Semigroups and Monoids

instance *set* :: (*semigroup-mult*) *quantale*
  ⟨*proof*⟩

With the definition of products on powersets (from the component of models of dioids) one can prove the following mixed distributivity law.

lemma *comp-dist-mix*: $\bigsqcup (X{::}'a{::}quantale\ set) \cdot \bigsqcup Y = \bigsqcup (X \cdot Y)$
⟨*proof*⟩

Powerset quantales of monoids can nevertheless be formalised as instances.

instance *set* :: (*monoid-mult*) *unital-quantale*⟨*proof*⟩

There is much more to this example. In fact, every quantale can be represented, up to isomorphism, as a certain quotient of a powerset quantale over some semigroup [6]. This representation theorem is formalised in the section on nuclei.

5.3 Language, Relation, Trace and Path Quantales

The language quantale is implcit in the powerset quantale over a semigroup or monoid. The free list monoid has already been linked with the class of monoid as an instance in Isabelle's dioid components [2]. I provide an alternative interpretation. In all of the following locale statements, an interpretation for Sup-quantales fails, due to the occurance of some low level less operations in the underlying model...

interpretation *lan-quantale*: *unital-quantale* $1{::}'a$ *lan* $(\cdot)$ *Inf Sup inf* $(\subseteq)$ $(\subset)$ *sup* *0 UNIV*
  ⟨*proof*⟩

The relation quantale follows.

interpretation *rel-quantale*: *unital-quantale Id relcomp Inf Sup inf* $(\subseteq)$ $(\subset)$ *sup* $\{\}$ *UNIV*
$\langle proof \rangle$

Traces alternate between state and action symbols, the first and last symbol of a trace being state symbols. They can be associated with behaviours of automata or executions of programs.

interpretation *trace-quantale*: *unital-quantale t-one t-prod Inf Sup inf* $(\subseteq)$ $(\subset)$ *sup t-zero UNIV*
$\langle proof \rangle$

The final model corresponds to paths as sequences of states of an automata, transition system or graph.

interpretation *path-quantale*: *unital-quantale p-one p-prod Inter Union* $(\cap)$ $(\subseteq)$ $(\subset)$ $(\cup)$ $\{\}$ *UNIV*
$\langle proof \rangle$

Rosenthal's book contains a wealth of other examples. Many of them come from ring theory (e.g. the additive subgroups of a ring form a quantale and so do the left, right and two-sided ideals). Others are based on the interval $[0, \infty]$. The first line of models was the original motivation for the study of quantales, the second one is important to Lawvere's categorical generalisation of metric spaces. These examples are left for future consideration.

end

6 Quantic Nuclei and Conuclei

theory *Quantic-Nuclei-Conuclei*
imports *Quantale-Models*

begin

Quantic nuclei and conuclei are an important part of the structure theory of quantales. I formalise the basics, following Rosenthal's book [6]. In the structure theorems, I collect all parts of the proof, but do not present the theorems in compact form due to difficulties to speak about subalgebras and homomorphic images without explicit carrier sets. Nuclei also arise in the context of complete Heyting algebras, frames and locales [5]. Their formalisation seems an interesting future task.

6.1 Nuclei

definition *nucleus* :: $('a::quantale \Rightarrow 'a::quantale) \Rightarrow bool$ **where**
$nucleus\ f = (clop\ f \wedge (\forall x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y)))$

lemma *nuc-lax*: $nucleus\ f \Longrightarrow f\ x \cdot f\ y \leq f\ (x \cdot y)$

⟨*proof*⟩

definition *unucleus* :: ($'a$::*unital-quantale* ⇒ $'a$::*unital-quantale*) ⇒ *bool* **where**
unucleus f = (*nucleus* $f \wedge 1 \leq f\ 1$)

lemma *nucleus* $f \Longrightarrow f\ \bot = \bot$
⟨*proof*⟩

lemma *conucleus* $f \Longrightarrow f\ \top = \top$
⟨*proof*⟩

lemma *nuc-prop1*: *nucleus* $f \Longrightarrow f\ (x \cdot y) = f\ (x \cdot f\ y)$
⟨*proof*⟩

lemma *nuc-prop2*: *nucleus* $f \Longrightarrow f\ (x \cdot y) = f\ (f\ x \cdot y)$
⟨*proof*⟩

lemma *nuc-comp-prop*: *nucleus* $f \Longrightarrow f\ (f\ x \cdot f\ y) = f\ (x \cdot y)$
⟨*proof*⟩

lemma *nucleus-alt-def1*: *nucleus* $f \Longrightarrow f\ x \rightarrow f\ y = x \rightarrow f\ y$
⟨*proof*⟩

lemma *nucleus-alt-def2*: *nucleus* $f \Longrightarrow f\ y \leftarrow f\ x = f\ y \leftarrow x$
⟨*proof*⟩

lemma *nucleus-alt-def3*:
fixes f :: $'a$::*unital-quantale* ⇒ $'a$
shows $\forall\, x\ y.\ f\ x \rightarrow f\ y = x \rightarrow f\ y \Longrightarrow \forall\, x\ y.\ f\ y \leftarrow f\ x = f\ y \leftarrow x \Longrightarrow$ *nucleus* f
⟨*proof*⟩

lemma *nucleus-alt-def*:
fixes f :: $'a$::*unital-quantale* ⇒ $'a$
shows *nucleus* $f = (\forall\ x\ y.\ f\ x \rightarrow f\ y = x \rightarrow f\ y \wedge f\ y \leftarrow f\ x = f\ y \leftarrow x)$
⟨*proof*⟩

lemma *nucleus-alt-def-cor1*: *nucleus* $f \Longrightarrow f\ (x \rightarrow y) \leq x \rightarrow f\ y$
⟨*proof*⟩

lemma *nucleus-alt-def-cor2*: *nucleus* $f \Longrightarrow f\ (y \leftarrow x) \leq f\ y \leftarrow x$
⟨*proof*⟩

lemma *nucleus-ab-unital*:
fixes f :: $'a$::*ab-unital-quantale* ⇒ $'a$
shows *nucleus* $f = (\forall\, x\ y.\ f\ x \rightarrow f\ y = x \rightarrow f\ y)$
⟨*proof*⟩

lemma *nuc-comp-assoc*: *nucleus* $f \Longrightarrow f\ (x \cdot f\ (y \cdot z)) = f\ (f\ (x \cdot y) \cdot z)$
⟨*proof*⟩

lemma *nuc-Sup-closed*: $nucleus\ f \Longrightarrow f \circ Sup \circ (\text{`})\ f = (f \circ Sup)$
⟨*proof*⟩

lemma *nuc-Sup-closed-var*: $nucleus\ f \Longrightarrow f\ (\bigsqcup x \in X.\ f\ x) = f\ (\bigsqcup X)$
⟨*proof*⟩

lemma *nuc-Inf-closed*: $nucleus\ f \Longrightarrow Sup\text{-}closed\text{-}set\ (Fix\ f)$
⟨*proof*⟩

lemma *nuc-Inf-closed*: $nucleus\ f \Longrightarrow Inf\text{-}closed\text{-}set\ (Fix\ f)$
⟨*proof*⟩

lemma *nuc-comp-distl*: $nucleus\ f \Longrightarrow f\ (x \cdot f\ (\bigsqcup Y)) = f\ (\bigsqcup y \in Y.\ f\ (x \cdot y))$
⟨*proof*⟩

lemma *nuc-comp-distr*: $nucleus\ f \Longrightarrow f\ (f\ (\bigsqcup X) \cdot y) = f\ (\bigsqcup x \in X.\ f\ (x \cdot y))$
⟨*proof*⟩

lemma $nucleus\ f \Longrightarrow f\ (x \cdot y) = f\ x \cdot f\ y$
⟨*proof*⟩

lemma *nuc-bres-closed*: $nucleus\ f \Longrightarrow f\ (f\ x \rightarrow f\ y) = f\ x \rightarrow f\ y$
⟨*proof*⟩

lemma $nucleus\ f \Longrightarrow f\ (x \rightarrow y) = f\ x \rightarrow f\ y$
⟨*proof*⟩

lemma *nuc-fres-closed*: $nucleus\ f \Longrightarrow f\ (f\ x \leftarrow f\ y) = f\ x \leftarrow f\ y$
⟨*proof*⟩

lemma *nuc-fres-closed*: $nucleus\ f \Longrightarrow f\ (x \leftarrow y) = f\ x \leftarrow f\ y$
⟨*proof*⟩

lemma *nuc-inf-closed*: $nucleus\ f \Longrightarrow inf\text{-}closed\text{-}set\ (Fix\ f)$
⟨*proof*⟩

lemma *nuc-inf-closed-var*: $nucleus\ f \Longrightarrow f\ (f\ x \sqcap f\ y) = f\ x \sqcap f\ y$
⟨*proof*⟩

Taken together these facts show that, for $f : Q \to Q$, $f[Q]$ forms a quantale with composition $f\,(- \cdot -)$ and sup $f\,(\bigsqcup -)$, and that $f : Q \to f[Q]$ is a quantale morphism. This is the first part of Theorem 3.1.1 in Rosenthal's book.

class *quantale-with-nuc* = *quantale* + *cl-op* +
assumes *cl-op-nuc*: $cl\text{-}op\ x \cdot cl\text{-}op\ y \leq cl\text{-}op\ (x \cdot y)$

begin

subclass *clattice-with-clop*⟨*proof*⟩

end

class *unital-quantale-with-nuc* = *quantale-with-nuc* + *unital-quantale* +
  assumes *one-nuc*: $1 \leq$ *cl-op 1*

lemma *nucleus-cl-op*: *nucleus* (*cl-op*::′*a*::*quantale-with-nuc* ⇒ ′*a*)
  ⟨*proof*⟩

lemma *unucleus-cl-op*: *unucleus* (*cl-op*::′*a*::*unital-quantale-with-nuc* ⇒ ′*a*)
  ⟨*proof*⟩

instantiation *cl-op-im* :: (*quantale-with-nuc*) *quantale*
begin

lift-definition *times-cl-op-im* :: ′*a*::*quantale-with-nuc cl-op-im* ⇒ ′*a cl-op-im* ⇒ ′*a cl-op-im* **is** $\lambda x\ y.$ *cl-op* $(x \cdot y)$
  ⟨*proof*⟩

instance
  ⟨*proof*⟩

end

instantiation *cl-op-im* :: (*unital-quantale-with-nuc*) *unital-quantale*
begin

lift-definition *one-cl-op-im* :: ′*a*::*unital-quantale-with-nuc cl-op-im* **is** *cl-op 1*
  ⟨*proof*⟩

instance
  ⟨*proof*⟩

end

The usefulness of these theorems remains unclear; it seems difficult to make them collaborate with concrete nuclei.

lemma *nuc-hom*: *Abs-cl-op-im* ∘ *cl-op* ∈ *quantale-homset*
  ⟨*proof*⟩

This finishes the first statement of Theorem 3.1.1. The second part follows. It states that for every surjective quantale homomorphism there is a nucleus such that the range of the nucleus is isomorphic to the range of the surjection.

lemma *quant-morph-nuc*:
  fixes *f* :: ′*a*::*quantale-with-dual* ⇒ ′*b*::*quantale-with-dual*
  assumes *f* ∈ *quantale-homset*
  shows *nucleus* ((*radj f*) ∘ *f*)
⟨*proof*⟩

lemma *surj-quantale-hom-bij-on*:
fixes f :: $'a$::*quantale-with-dual* $\Rightarrow$ $'b$::*quantale-with-dual*
assumes *surj* f
and $f \in$ *quantale-homset*
shows *bij-betw* f (*range* (*radj* $f \circ f$)) *UNIV*
⟨*proof*⟩

This establishes the bijection, extending a similar fact about closure operators and complete lattices (surj-Sup-pres-bij). It remains to show that f is a quantale morphism, that is, it preserves Sups and compositions of closed elements with operations defined as in the previous instantiation statement. Sup-preservation holds already for closure operators on complete lattices (surj-Sup-pres-iso). Hence it remains to prove preservation of compositions.

lemma *surj-comp-pres-iso*:
fixes f :: $'a$::*quantale-with-dual* $\Rightarrow$ $'b$::*quantale-with-dual*
assumes $f \in$ *quantale-homset*
shows f ((*radj* $f \circ f$) $(x \cdot y)) = f\ x \cdot f\ y$
⟨*proof*⟩

This establishes the quantale isomorphism and finishes the proof of Theorem 3.1.1.

Next I prove Theorem 3.1.2 in Rosenthal's book. nuc-Inf-closed shows that *Fix* f is Inf-closed. Hence the two following lemmas show one direction.

lemma *nuc-bres-pres*: *nucleus* $f \Longrightarrow y \in$ *Fix* $f \Longrightarrow x \rightarrow y \in$ *Fix* f
⟨*proof*⟩

lemma *nuc-fres-pres*: *nucleus* $f \Longrightarrow y \in$ *Fix* $f \Longrightarrow y \leftarrow x \in$ *Fix* f
⟨*proof*⟩

lemma *lax-aux*:
fixes X :: $'a$::*quantale set*
assumes $\forall x. \forall y \in X.\ x \rightarrow y \in X$
and $\forall x.\ \forall y \in X.\ y \leftarrow x \in X$
shows $\bigsqcap\{z \in X.\ x \leq z\} \cdot \bigsqcap\{z \in X.\ y \leq z\} \leq \bigsqcap\{z \in X.\ x \cdot y \leq z\}$
⟨*proof*⟩

lemma *Inf-closed-res-nuc*:
fixes X :: $'a$::*quantale set*
assumes *Inf-closed-set* X
and $\forall x.\ \forall y \in X.\ x \rightarrow y \in X$
and $\forall x.\ \forall y \in X.\ y \leftarrow x \in X$
shows *nucleus* $(\lambda y.\ \bigsqcap\{x \in X.\ y \leq x\})$
⟨*proof*⟩

lemma *Inf-closed-res-Fix*:
fixes X :: $'a$::*quantale set*

assumes *Inf-closed-set* X
and $\forall x.\ \forall y \in X.\ x \to y \in X$
and $\forall x.\ \forall y \in X.\ y \leftarrow x \in X$
shows $X = \mathit{Fix}\ (\lambda y.\ \bigsqcap\{x \in X.\ y \leq x\})$
⟨*proof*⟩

This finishes the proof of Theorem 3.1.2

6.2 A Representation Theorem

The final proofs for nuclei lead to Rosenthal's representation theorem for quantales (Theorem 3.1.2).

lemma *down-set-lax-morph*: $(\downarrow \circ\ \mathit{Sup})\ (X{::}'a{::}\mathit{quantale\ set}) \cdot (\downarrow \circ\ \mathit{Sup})\ Y \subseteq (\downarrow \circ\ \mathit{Sup})\ (X \cdot Y)$
⟨*proof*⟩

lemma *downset-Sup-nuc*: *nucleus* $(\downarrow \circ\ (\mathit{Sup}{::}'a{::}\mathit{quantale\ set} \Rightarrow 'a))$
⟨*proof*⟩

lemma *downset-surj*: *surj-on* $\downarrow$ $(\mathit{range}\ (\downarrow \circ\ \mathit{Sup}))$
⟨*proof*⟩

In addition, $\downarrow$ is injective by Lemma downset-inj. Hence it is a bijection between the quantale and the powerset quantale. It remains to show that $\downarrow$ is a quantale morphism.

lemma *downset-Sup-pres-var*: $\downarrow (\bigsqcup X) = (\downarrow \circ\ \mathit{Sup})\ (\Downarrow (X{::}'a{::}\mathit{quantale\ set}))$
⟨*proof*⟩

lemma *downset-Sup-pres*: $\downarrow (\bigsqcup X) = (\downarrow \circ\ \mathit{Sup})\ (\bigcup\ (\downarrow\ `\ (X{::}'a{::}\mathit{quantale\ set})))$
⟨*proof*⟩

lemma *downset-comp-pres*: $\downarrow ((x{::}'a{::}\mathit{quantale}) \cdot y) = (\downarrow \circ\ \mathit{Sup})\ (\downarrow x \cdot \downarrow y)$
⟨*proof*⟩

This finishes the proof of Theorem 3.1.2.

6.3 Conuclei

definition *conucleus* :: $('a{::}\mathit{quantale} \Rightarrow 'a{::}\mathit{quantale}) \Rightarrow \mathit{bool}$ **where**
conucleus $f = (\mathit{coclop}\ f \wedge (\forall x\ y.\ f\ x \cdot f\ y \leq f\ (x \cdot y)))$

lemma *conuc-lax*: *conucleus* $f \Longrightarrow f\ x \cdot f\ y \leq f\ (x \cdot y)$
⟨*proof*⟩

definition *uconucleus* :: $('a{::}\mathit{unital\text{-}quantale} \Rightarrow 'a{::}\mathit{unital\text{-}quantale}) \Rightarrow \mathit{bool}$ **where**

uconucleus $f = (\mathit{conucleus}\ f \wedge f\ 1 \leq 1)$

Next I prove Theorem 3.1.3.

lemma *conuc-Sup-closed*: $conucleus\ f \Longrightarrow f \circ Sup \circ (\text{`})\ f = Sup \circ (\text{`})\ f$
⟨*proof*⟩

lemma *conuc-comp-closed*: $conucleus\ f \Longrightarrow f\ (f\ x \cdot f\ y) = f\ x \cdot f\ y$
⟨*proof*⟩

The sets of fixpoints of conuclei are closed under Sups and composition; hence they form subquantales.

lemma *conuc-Sup-qclosed*: $conucleus\ f \Longrightarrow Sup\text{-}closed\text{-}set\ (Fix\ f) \wedge comp\text{-}closed\text{-}set\ (Fix\ f)$
⟨*proof*⟩

This shows that the subset $Fix f$ of a quantale, for conucleus f, is closed under Sups and composition. It is therefore a subquantale. $f : f[Q] \to Q$ is an embedding. As before, this could be shown by formalising a type class of quantales with a conucleus operation, converting the range of the conucleus into a type and providing a sublocale proof. First, this would require showing that the coclosed elements of a complete lattice form a complete sublattice. Relative to the proofs for closure operators and nuclei there is nothing to be learned. I provide this proof in the section on left-sided elements, where the conucleus can be expressed within the language of quantales.

The second part of Theorem 3.1.3 states that every subquantale of a given quantale is equal to $Fix f$ for some conucleus f.

lemma *lax-aux2*:
fixes $X :: 'a::quantale\ set$
assumes *Sup-closed-set X*
and *comp-closed-set X*
shows $\bigsqcup\{z \in X.\ z \leq x\} \cdot \bigsqcup\{z \in X.\ z \leq y\} \leq \bigsqcup\{z \in X.\ z \leq x \cdot y\}$
⟨*proof*⟩

lemma *subquantale-conucleus*:
fixes $X :: 'a::quantale\ set$
assumes *Sup-closed-set X*
and *comp-closed-set X*
shows $conucleus\ (\lambda x.\ \bigsqcup\{y \in X.\ y \leq x\})$
⟨*proof*⟩

lemma *subquantale-Fix*:
fixes $X :: 'a::quantale\ set$
assumes *Sup-closed-set X*
and *comp-closed-set X*
shows $X = Fix\ (\lambda x.\ \bigsqcup\{y \in X.\ y \leq x\})$
⟨*proof*⟩

This finishes the proof of Theorem 3.1.3

end

7 Left-Sided Elements in Quantales

theory *Quantale-Left-Sided*
 imports *Quantic-Nuclei-Conuclei*
begin

context *quantale*
begin

7.1 Basic Definitions

Left-sided elements are well investigated in quantale theory. They could be defined in weaker settings, for instance, ord with a top element.

definition $lsd\ x = (\top \cdot x \leq x)$

definition $rsd\ x = (x \cdot \top \leq x)$

definition $twosd\ x = (lsd\ x \wedge rsd\ x)$

definition $slsd\ x = (\top \cdot x = x)$

definition $srsd\ x = (x \cdot \top = x)$

definition $stwsd\ x = (slsd\ x \wedge srsd\ x)$

lemma *lsided-bres*: $(lsd\ x) = (x \leq \top \rightarrow x)$
 ⟨*proof*⟩

lemma *lsided-fres*: $(lsd\ x) = (\top \leq x \leftarrow x)$
 ⟨*proof*⟩

lemma *lsided-fres-eq*: $(lsd\ x) = (x \leftarrow x = \top)$
 ⟨*proof*⟩

lemma *lsided-slsided*: $lsd\ x \Longrightarrow x \cdot x = x \Longrightarrow slsd\ x$
 ⟨*proof*⟩

lemma *lsided-prop*: $lsd\ x \Longrightarrow y \cdot x \leq x$
 ⟨*proof*⟩

lemma *rsided-fres*: $(rsd\ x) = (x \leq x \leftarrow \top)$
 ⟨*proof*⟩

lemma *rsided-bres*: $(rsd\ x) = (\top \leq x \rightarrow x)$
 ⟨*proof*⟩

lemma *rsided-bres-eq*: $(rsd\ x) = (x \rightarrow x = \top)$
 ⟨*proof*⟩

lemma *rsided-srsided*: $rsd\ x \Longrightarrow x \cdot x = x \Longrightarrow srsd\ x$
$\langle proof \rangle$

lemma *rsided-prop*: $rsd\ x \Longrightarrow x \cdot y \leq x$
$\langle proof \rangle$

lemma *lsided-top*: $lsd\ \top$
$\langle proof \rangle$

lemma *lsided-bot*: $lsd\ \bot$
$\langle proof \rangle$

lemma *rsided-top*: $rsd\ \top$
$\langle proof \rangle$

lemma *rsided-bot*: $rsd\ \bot$
$\langle proof \rangle$

end

Right-sided elements are henceforth not considered. Their properties arise by opposition duality, which is not formalised.

The following functions have left-sided elements as fixpoints.

definition *lsl*:: $'a::quantale \Rightarrow 'a\ (\nu)$ **where**
$\nu\ x = \top \cdot x$

definition *lsu* :: $'a::quantale \Rightarrow 'a\ (\nu^\natural)$ **where**
$\nu^\natural\ x = \top \rightarrow x$

These functions are adjoints.

lemma *ls-galois*: $\nu \dashv \nu^\natural$
$\langle proof \rangle$

Due to this, the following properties hold.

lemma *lsl-iso*: $mono\ \nu$
$\langle proof \rangle$

lemma *lsl-iso-var*: $x \leq y \Longrightarrow \nu\ x \leq \nu\ y$
$\langle proof \rangle$

lemma *lsu-iso*: $mono\ \nu^\natural$
$\langle proof \rangle$

lemma *lsu-iso-var*: $x \leq y \Longrightarrow \nu^\natural\ x \leq \nu^\natural\ y$
$\langle proof \rangle$

lemma *lsl-bot* [*simp*]: $\nu\ \bot = \bot$
$\langle proof \rangle$

lemma *lsu-top* [*simp*]: $\nu^\natural\ \top = \top$
⟨*proof*⟩

lemma *lsu-Inf-pres*: *Inf-pres* $\nu^\natural$
⟨*proof*⟩

lemma *lsl-Sup-pres*: *Sup-pres* $(\nu::'a::quantale \Rightarrow 'a)$
⟨*proof*⟩

lemma *lsu-Inf-closed*: *Inf-closed-set* (*range* $\nu^\natural$)
⟨*proof*⟩

lemma *lsl-Sup-closed*: *Sup-closed-set* (*range* $(\nu::'a::quantale \Rightarrow 'a)$)
⟨*proof*⟩

lemma *lsl-lsu-canc1*: $\nu \circ \nu^\natural \leq id$
⟨*proof*⟩

lemma *lsl-lsu-canc2*: $id \leq \nu^\natural \circ \nu$
⟨*proof*⟩

lemma *clop-lsu-lsl*: *clop* $(\nu^\natural \circ \nu)$
⟨*proof*⟩

lemma *coclop-lsl-lsu*: *coclop* $(\nu \circ \nu^\natural)$
⟨*proof*⟩

lemma *dang1*: $\nu\ (\nu\ x \sqcap y) \leq \nu\ x$
⟨*proof*⟩

lemma *lsl-trans*: $\nu \circ \nu \leq \nu$
⟨*proof*⟩

lemma *lsl-lsu-prop*: $\nu \circ \nu^\natural \leq \nu^\natural$
⟨*proof*⟩

lemma *lsu-lsl-prop*: $\nu \leq \nu^\natural \circ \nu$
⟨*proof*⟩

context *unital-quantale*
begin

Left-sidedness and strict left-sidedness now coincide.

lemma *lsided-eq*: *lsd* = *slsd*
⟨*proof*⟩

lemma *lsided-eq-var1*: $(x \leq \top \to x) = (x = \top \to x)$
⟨*proof*⟩

lemma *lsided-eq-var2*: $lsd\ x = (x = \top \to x)$
⟨*proof*⟩

end

lemma *lsided-def3*: $(\nu\ (x::'a::unital\text{-}quantale) = x) = (\nu^\natural\ x = x)$
⟨*proof*⟩

lemma *Fix-lsl-lsu*: $Fix\ (\nu::'a::unital\text{-}quantale \Rightarrow 'a) = Fix\ \nu^\natural$
⟨*proof*⟩

lemma *Fix-lsl-left-slsided*: $Fix\ \nu = \{(x::'a::unital\text{-}quantale).\ lsd\ x\}$
⟨*proof*⟩

lemma *Fix-lsl-iff* [*simp*]: $(x \in Fix\ \nu) = (\nu\ x = x)$
⟨*proof*⟩

lemma *Fix-lsu-iff* [*simp*]: $(x \in Fix\ \nu^\natural) = (\nu^\natural\ x = x)$
⟨*proof*⟩

lemma *lsl-lsu-prop-eq* [*simp*]: $(\nu::'a::unital\text{-}quantale \Rightarrow 'a) \circ \nu^\natural = \nu^\natural$
⟨*proof*⟩

lemma *lsu-lsl-prop-eq* [*simp*]: $\nu^\natural \circ \nu = (\nu::'a::unital\text{-}quantale \Rightarrow 'a)$
⟨*proof*⟩

7.2 Connection with Closure and Coclosure Operators, Nuclei and Conuclei

lsl is therefore a closure operator, lsu a cocolosure operator.

lemma *lsl-clop*: $clop\ (\nu::'a::unital\text{-}quantale \Rightarrow 'a)$
⟨*proof*⟩

lemma *lsu-coclop*: $coclop\ (\nu^\natural::'a::unital\text{-}quantale \Rightarrow 'a)$
⟨*proof*⟩

lemma *lsl-range-fix*: $range\ (\nu::'a::unital\text{-}quantale \Rightarrow 'a) = Fix\ \nu$
⟨*proof*⟩

lemma *lsu-range-fix*: $range\ (\nu^\natural::'a::unital\text{-}quantale \Rightarrow 'a) = Fix\ \nu^\natural$
⟨*proof*⟩

lemma *range-lsl-iff* [*simp*]: $((x::'a::unital\text{-}quantale) \in range\ \nu) = (\nu\ x = x)$
⟨*proof*⟩

lemma *range-lsu-iff* [*simp*]: $((x::'a::unital\text{-}quantale) \in range\ \nu^\natural) = (\nu^\natural\ x = x)$
⟨*proof*⟩

lsl and lsu are therefore both Sup and Inf closed.

lemma *lsu-Sup-closed*: *Sup-closed-set* (*Fix* ($\nu^\natural$::'*a*::*unital-quantale* $\Rightarrow$ '*a*))
⟨*proof*⟩

lemma *lsl-Inf-closed*: *Inf-closed-set* (*Fix* (ν::'*a*::*unital-quantale* $\Rightarrow$ '*a*))
⟨*proof*⟩

lsl is even a quantic conucleus.

lemma *lsu-lax*: $\nu^\natural$ (*x*::'*a*::*unital-quantale*) $\cdot$ $\nu^\natural$ *y* $\leq$ $\nu^\natural$ ($x \cdot y$)
⟨*proof*⟩

lemma *lsu-one*: $\nu^\natural$ *1* $\leq$ (*1*::'*a*::*unital-quantale*)
⟨*proof*⟩

lemma *lsl-one*: *1* $\leq$ ν (*1*::'*a*::*unital-quantale*)
⟨*proof*⟩

lemma *lsu-conuc*: *conucleus* ($\nu^\natural$::'*a*::*unital-quantale* $\Rightarrow$ '*a*)
⟨*proof*⟩

It is therefore closed under composition.

lemma *lsu-comp-closed-var* [*simp*]: $\nu^\natural$ ($\nu^\natural$ $x \cdot \nu^\natural$ *y*) = $\nu^\natural$ $x \cdot \nu^\natural$ (*y*::'*a*::*unital-quantale*)
⟨*proof*⟩

lemma *lsu-comp-closed*: *comp-closed-set* (*Fix* ($\nu^\natural$::'*a*::*unital-quantale* $\Rightarrow$ '*a*))
⟨*proof*⟩

lsl is not a quantic nucleus unless composition is commutative.

lemma ν $x \cdot \nu$ $y \leq \nu$ ($x \cdot$ (*y*::'*a*::*unital-quantale*))
⟨*proof*⟩

Yet it is closed under composition (because the set of fixpoints are the same).

lemma *lsl-comp-closed*: *comp-closed-set* (*Fix* (ν::'*a*::*unital-quantale* $\Rightarrow$ '*a*))
⟨*proof*⟩

lemma *lsl-comp-closed-var* [*simp*]: ν (ν $x \cdot \nu$ (*y*::'*a*::*unital-quantale*)) = ν $x \cdot \nu$ y
⟨*proof*⟩

The following simple properties go beyond nuclei and conuclei.

lemma *lsl-lsu-ran*: *range* $\nu^\natural$ = *range* (ν::'*a*::*unital-quantale* $\Rightarrow$ '*a*)
⟨*proof*⟩

lemma *lsl-top* [*simp*]: ν $\top$ = ($\top$::'*a*::*unital-quantale*)
⟨*proof*⟩

lemma *lsu-bot* [*simp*]: $\nu^\natural$ $\bot$ = ($\bot$::'*a*::*unital-quantale*)
⟨*proof*⟩

lemma *lsu-inj-on*: *inj-on* $\nu^\natural$ (*Fix* ($\nu^\natural$::*'a*::*unital-quantale* $\Rightarrow$ *'a*))
⟨*proof*⟩

lemma *lsl-inj-on*: *inj-on* ν (*Fix* (ν::*'a*::*unital-quantale* $\Rightarrow$ *'a*))
⟨*proof*⟩

Additional preservation properties of lsl and lsu are useful in proofs.

lemma *lsl-Inf-closed-var* [*simp*]: ν ($\bigsqcap x \in X.\ \nu\ x$) = ($\bigsqcap x \in X.\ \nu$ (x::*'a*::*unital-quantale*))
⟨*proof*⟩

lemma *lsl-Sup-closed-var* [*simp*]: ν ($\bigsqcup x \in X.\ \nu\ x$) = ($\bigsqcup x \in X.\ \nu$ (x::*'a*::*unital-quantale*))
⟨*proof*⟩

lemma *lsu-Inf-closed-var* [*simp*]: $\nu^\natural$ ($\bigsqcap x \in X.\ \nu^\natural\ x$) = ($\bigsqcap x \in X.\ \nu^\natural$ (x::*'a*::*unital-quantale*))
⟨*proof*⟩

lemma *lsu-Sup-closed-var* [*simp*]: $\nu^\natural$ ($\bigsqcup x \in X.\ \nu^\natural\ x$) = ($\bigsqcup x \in X.\ \nu^\natural$ (x::*'a*::*unital-quantale*))
⟨*proof*⟩

lemma *lsl-inf-closed-var* [*simp*]: ν ($\nu\ x \sqcap \nu\ y$) = ν (x::*'a*::*unital-quantale*) $\sqcap\ \nu\ y$
⟨*proof*⟩

lemma *lsl-sup-closed-var* [*simp*]: ν ($\nu\ x \sqcup \nu\ y$) = ν (x::*'a*::*unital-quantale*) $\sqcup\ \nu\ y$
⟨*proof*⟩

lemma *lsu-inf-closed-var* [*simp*]: $\nu^\natural$ ($\nu^\natural\ x \sqcap \nu^\natural\ y$) = $\nu^\natural$ (x::*'a*::*unital-quantale*) $\sqcap$ $\nu^\natural\ y$
⟨*proof*⟩

lemma *lsu-sup-closed-var* [*simp*]: $\nu^\natural$ ($\nu^\natural\ x \sqcup \nu^\natural\ y$) = $\nu^\natural$ (x::*'a*::*unital-quantale*) $\sqcup$ $\nu^\natural\ y$
⟨*proof*⟩

lemma *lsu-fres-closed* [*simp*]: $\nu^\natural$ ($\nu^\natural\ x \leftarrow \nu^\natural\ y$) = $\nu^\natural\ x \leftarrow \nu^\natural$ (y::*'a*::*unital-quantale*)
⟨*proof*⟩

lemma *lsl-fres-closed* [*simp*]: ν ($\nu\ x \leftarrow \nu\ y$) = $\nu\ x \leftarrow \nu$ (y::*'a*::*unital-quantale*)
⟨*proof*⟩

lemma *lsl-almost-lax*: $x \cdot \nu\ y \leq \nu$ (y::*'a*::*unital-quantale*)
⟨*proof*⟩

Finally, here are two counterexamples.

lemma $\nu^\natural$ ($\nu^\natural\ x \rightarrow \nu^\natural\ y$) = $\nu^\natural\ x \rightarrow \nu^\natural$ (y::*'a*::*unital-quantale*)
⟨*proof*⟩

lemma ν ($\nu\ x \rightarrow \nu\ y$) = $\nu\ x \rightarrow \nu$ (y::*'a*::*unital-quantale*)
⟨*proof*⟩

context *ab-quantale*
begin

lemma *lsided-times-top*: *lsd* $(\top \cdot x)$
⟨*proof*⟩

lemma *lsided-le2*: *lsd* $x \Longrightarrow x \cdot y \leq x \sqcap (y \cdot \top)$
⟨*proof*⟩

lemma *lsided-le3*: *lsd* $x \Longrightarrow (x \sqcap y) \cdot z \leq x$
⟨*proof*⟩

lemma *lsided-le4*: *lsd* $x \Longrightarrow (x \sqcap y) \cdot z \leq x \sqcap (y \cdot z)$
⟨*proof*⟩

lemma *lsided-times-le-inf*: *lsd* $x \Longrightarrow$ *lsd* $y \Longrightarrow x \cdot y \leq x \sqcap y$
⟨*proof*⟩

end

Now lsl is a quantic nucleus.

lemma *lsl-lax*: $\nu\ x \cdot \nu\ y \leq \nu\ (x \cdot (y{::}'aa{::}ab\text{-}unital\text{-}quantale))$
⟨*proof*⟩

lemma *lsl-nuc*: *nucleus* $(\nu{::}'a{::}ab\text{-}unital\text{-}quantale \Rightarrow 'a)$
⟨*proof*⟩

The following properties go beyond nuclei.

lemma *lsl-comp-pres*: $\nu\ (x \cdot y) = \nu\ x \cdot \nu\ (y{::}'a{::}ab\text{-}unital\text{-}quantale)$
⟨*proof*⟩

lemma *lsl-bres-closed* [*simp*]: $\nu\ (\nu\ x \to \nu\ y) = \nu\ x \to \nu\ (y{::}'a{::}ab\text{-}unital\text{-}quantale)$
⟨*proof*⟩

lemma *lsu-bres-pres* [*simp*]: $\nu^\natural\ (\nu^\natural\ x \to \nu^\natural\ y) = \nu^\natural\ x \to \nu^\natural\ (y{::}'a{::}ab\text{-}unital\text{-}quantale)$
⟨*proof*⟩

lemma *lsl-prelocalic-var*: $\nu\ x \cdot y \leq \nu\ x \sqcap \nu\ (y{::}'a{::}ab\text{-}unital\text{-}quantale)$
⟨*proof*⟩

lemma *dang3*: $(\nu\ x \sqcap y) \cdot z \leq \nu\ x \sqcap (y \cdot (z{::}'a{::}ab\text{-}unital\text{-}quantale))$
⟨*proof*⟩

lemma *lsl-prelocalic*: $\nu\ x \cdot \nu\ y \leq \nu\ x \sqcap \nu\ (y{::}'a{::}ab\text{-}unital\text{-}quantale)$
⟨*proof*⟩

lemma *lsl-local*: $x \cdot \nu\ y \leq \nu\ (x \cdot (y{::}'a{::}ab\text{-}unital\text{-}quantale))$
⟨*proof*⟩

lemma *lsl-local-eq*: $x \cdot \nu\ y = \nu\ (x \cdot (y::'a::\mathit{ab\text{-}unital\text{-}quantale}))$
⟨*proof*⟩

Relative pseudocomplementation can be defined on the subquantale induced by lsu.

definition $\mathit{rpc}\ x\ y = \nu^\natural\ (-x \sqcup (y::'a::\mathit{bool\text{-}unital\text{-}quantale}))$

lemma *lsl-rpc* [*simp*]: $\nu\ (\mathit{rpc}\ x\ y) = \mathit{rpc}\ x\ y$
⟨*proof*⟩

lemma *lsu-rpc* [*simp*]: $\nu^\natural\ (\mathit{rpc}\ x\ y) = \mathit{rpc}\ x\ y$
⟨*proof*⟩

lemma *lsl-rpc-galois*: $(x \sqcap \nu\ z \leq y) = (z \leq \mathit{rpc}\ x\ (y::'a::\mathit{bool\text{-}unital\text{-}quantale}))$
⟨*proof*⟩

lemma *lsl-rpc-galois-var*: $x \sqcap \nu\ z \leq y \longleftrightarrow \nu\ z \leq \mathit{rpc}\ x\ y$
⟨*proof*⟩

lemma *rpc-expl-aux*: $\bigsqcup\{z.\ x \sqcap \nu\ z \leq y\} = \bigsqcup\{z.\ x \sqcap z \leq y \wedge \nu\ z = (z::'a::\mathit{bool\text{-}unital\text{-}quantale})\}$
⟨*proof*⟩

lemma *rpc-expl*: $\mathit{rpc}\ x\ y = \bigsqcup\{z.\ x \sqcap z \leq y \wedge \nu\ z = (z::'a::\mathit{bool\text{-}unital\text{-}quantale})\}$
⟨*proof*⟩

7.3 Non-Preservation and Lack of Closure

context *bool-ab-unital-quantale*
begin

A few counterexamples deal in particular with lack of closure with respect to boolean complementation.

lemma $\nu^\natural\ (x \cdot y) = \nu^\natural\ x \cdot \nu^\natural\ y$
⟨*proof*⟩

lemma $\nu\ 1 = 1$
⟨*proof*⟩

lemma $\nu^\natural\ x = \nu\ x$
⟨*proof*⟩

lemma $\nu^\natural\ (\bigsqcup\ P) = \bigsqcup\{\nu^\natural\ p\ |p.\ p \in P\}$
⟨*proof*⟩

lemma $\mathit{rpc}\ (\nu^\natural\ x)\ (\nu^\natural\ y) = -(\nu^\natural\ x) \sqcup (\nu^\natural\ y)$
⟨*proof*⟩

lemma $\mathit{rpc}\ (\nu\ x)\ (\nu\ y) = -(\nu\ x) \sqcup (\nu\ y)$
⟨*proof*⟩

lemma $\nu\ (-(\nu^\natural\ x) \sqcup (\nu^\natural\ y)) = -(\nu^\natural\ x) \sqcup (\nu^\natural\ y)$
⟨*proof*⟩

lemma $\nu\ (-(\nu\ x) \sqcup (\nu\ y)) = -(\nu\ x) \sqcup (\nu\ y)$
⟨*proof*⟩

lemma $\nu\ x \cdot \nu\ y = \nu\ x \sqcap \nu\ y$
⟨*proof*⟩

lemma $\nu\ (-\ (\nu\ x)) = -\ (\nu\ x)$
⟨*proof*⟩

lemma $\nu^\natural\ (-\ (\nu^\natural\ x)) = -\ (\nu^\natural\ x)$
⟨*proof*⟩

lemma $\nu\ (-\ (\nu\ x) \sqcup (\nu\ y)) = -\ (\nu\ x) \sqcup (\nu\ y)$
⟨*proof*⟩

lemma $\nu^\natural\ (-\ (\nu^\natural\ x) \sqcup (\nu^\natural\ y)) = -\ (\nu^\natural\ x) \sqcup (\nu^\natural\ y)$
⟨*proof*⟩

end

7.4 Properties of Quotient Algebras and Subalgebras

Finally I replay Rosenthal's quotient and subalgebra proofs for nuclei and conuclei in the concrete setting of left-sided elements. Ideally it should be sufficient to show that lsl is a (quantale with) nucleus and then pick up the quotient algebra proof from the nucleus section. But there is no way of achieving this, apart from creating a type class for quantales with the lsl operation. This seems bizarre, since lsl is just a definition. On the other hand, an approach in which interpretations are used instead of instantiations might do the job.

The first proof shows that lsu, as a conucleus, yields a subalgebra.

typedef (**overloaded**) *'a lsu-set* = *Fix* ($\nu^\natural$::*'a*::*unital-quantale* ⇒ *'a*)
⟨*proof*⟩

setup-lifting *type-definition-lsu-set*

instantiation *lsu-set* :: (*unital-quantale*) *quantale*
begin

lift-definition *times-lsu-set* :: *'a lsu-set* ⇒ *'a lsu-set* ⇒ *'a lsu-set* **is** *times*
⟨*proof*⟩

lift-definition *Inf-lsu-set* :: *'a lsu-set set* ⇒ *'a lsu-set* **is** *Inf*

⟨*proof*⟩

lift-definition *Sup-lsu-set* :: *'a lsu-set set* ⇒ *'a lsu-set* **is** *Sup*
⟨*proof*⟩

lift-definition *bot-lsu-set* :: *'a lsu-set* **is** $\bot$
⟨*proof*⟩

lift-definition *sup-lsu-set* :: *'a lsu-set* ⇒ *'a lsu-set* ⇒ *'a lsu-set* **is** *sup*
⟨*proof*⟩

lift-definition *top-lsu-set* :: *'a lsu-set* **is** $\top$
⟨*proof*⟩

lift-definition *inf-lsu-set* :: *'a lsu-set* ⇒ *'a lsu-set* ⇒ *'a lsu-set* **is** *inf*
⟨*proof*⟩

lift-definition *less-eq-lsu-set* :: *'a lsu-set* ⇒ *'a lsu-set* ⇒ *bool* **is** *less-eq* ⟨*proof*⟩

lift-definition *less-lsu-set* :: *'a lsu-set* ⇒ *'a lsu-set* ⇒ *bool* **is** *less* ⟨*proof*⟩

instance
⟨*proof*⟩

end

This proof checks simply closure conditions, as one would expect for establishing a subalgebra.

instance *lsu-set* :: (*bool-ab-unital-quantale*) *distrib-ab-quantale*
⟨*proof*⟩

typedef (**overloaded**) *'a lsl-set* = *range* (ν::*'a*:: *unital-quantale* ⇒ *'a*)
⟨*proof*⟩

setup-lifting *type-definition-lsl-set*

The final statement shows that lsu, as a nucleus, yields a quotient algebra.

instantiation *lsl-set* :: (*ab-unital-quantale*) *ab-unital-quantale*
begin

lift-definition *one-lsl-set* :: *'a*::*ab-unital-quantale lsl-set* **is** ν *1*
⟨*proof*⟩

lift-definition *Inf-lsl-set* :: *'a lsl-set set* ⇒ *'a lsl-set* **is** $\lambda X.\ \nu\ (\bigsqcap X)$
⟨*proof*⟩

lift-definition *Sup-lsl-set* :: *'a lsl-set set* ⇒ *'a lsl-set* **is** $\lambda X.\ \nu\ (\bigsqcup X)$
⟨*proof*⟩

lift-definition *bot-lsl-set* :: *'a lsl-set* **is** $\nu\ \bot$
⟨*proof*⟩

lift-definition *sup-lsl-set* :: *'a lsl-set* $\Rightarrow$ *'a lsl-set* $\Rightarrow$ *'a lsl-set* **is** $\lambda x\ y.\ \nu\ x \sqcup \nu\ y$
⟨*proof*⟩

lift-definition *top-lsl-set* :: *'a lsl-set* **is** $\nu\ \top$
⟨*proof*⟩

lift-definition *inf-lsl-set* :: *'a lsl-set* $\Rightarrow$ *'a lsl-set* $\Rightarrow$ *'a lsl-set* **is** $\lambda x\ y.\ \nu\ x \sqcap \nu\ y$
⟨*proof*⟩

lift-definition *less-eq-lsl-set* :: *'a lsl-set* $\Rightarrow$ *'a lsl-set* $\Rightarrow$ *bool* **is** $\lambda x\ y\ .\ \nu\ x \leq \nu\ y$ ⟨*proof*⟩

lift-definition *less-lsl-set* :: *'a lsl-set* $\Rightarrow$ *'a lsl-set* $\Rightarrow$ *bool* **is** $\lambda x\ y.\ \nu\ x \leq \nu\ y \wedge x \neq y$ ⟨*proof*⟩

lift-definition *times-lsl-set* :: *'a lsl-set* $\Rightarrow$ *'a lsl-set* $\Rightarrow$ *'a lsl-set* **is** $\lambda\ x\ y.\ \nu\ (x \cdot y)$
⟨*proof*⟩

instance
⟨*proof*⟩

end

This proof checks preservation properties instead of closure ones.

end

References

[1] S. Abramsky and S. Vickers. Quantales, observational logic and process semantics. *Mathematical Structures in Computer Science*, 3(2):161–227, 1993.

[2] A. Armstrong, G. Struth, and T. Weber. Kleene algebra. *Archive of Formal Proofs*, 2013.

[3] B. Dongol, V. B. F. Gomes, I. J. Hayes, and G. Struth. Partial semigroups and convolution algebras. *Archive of Formal Proofs*, 2017.

[4] B. Dongol, I. J. Hayes, and G. Struth. Relational convolution, generalised modalities and incidence algebras. *CoRR*, abs/1702.04603, 2017.

[5] P. T. Johnstone. *Stone Spaces*. Cambridge University Press, 1982.

[6] K. I. Rosenthal. *Quantales and their Applications*. Longman Scientific & Technical, 1990.