

Nagata Factoriality

Arthur Freitas Ramos* David Barros Hulak
Ruy J. G. B. de Queiroz

May 8, 2026

Abstract

This entry formalizes a prime-generated version of Nagata’s factoriality theorem in Isabelle/HOL. It develops the basic theory of prime-generated multiplicative sets, packages a wrapper interface around the AFP entry `Localization_Ring`, and proves record-based descent theorems showing that factoriality descends from a localization to the base ring under prime-generated and prime-or-unit hypotheses on the multiplicative set. The theorem package also includes closure-based corollaries for arbitrary and finite families of prime generators. The application layer specializes this framework to polynomial rings, both for localization away the polynomial variable X and for localizations generated by constant prime polynomials.

Contents

1	Overview	2
2	Prime-generated multiplicative sets	2
3	Localization helper lemmas	4
4	Record-based Nagata descent lemmas	15
5	Polynomial applications	52
6	Constant-prime localization applications	54
7	Nagata-factoriality scaffolding	56

*Maintainer: arfreita@microsoft.com

1 Overview

This entry formalizes a prime-generated version of Nagata's factoriality theorem for noetherian domains, following the classical commutative-algebra framework developed by Nagata, Samuel, and Matsumura.[2, 3, 1] It packages the descent theorem itself, closure-based corollaries for prime-generated multiplicative sets, and abstract polynomial applications for localization away the polynomial variable X and for localizations generated by constant prime polynomials.

theory *Prime-Generated*

imports *HOL-Computational-Algebra.Factorial-Ring*

begin

2 Prime-generated multiplicative sets

This theory isolates the reusable combinatorial layer behind Nagata's factoriality theorem. The full localization argument is developed in later theories; here we focus on the multiplicative sets generated by prime elements and on the closure lemmas that do not depend on any localization API.

definition *avoids* :: ' a :: *comm-semiring-1 set* \Rightarrow ' $a \Rightarrow$ *bool* **where**
avoids S $p \iff (\forall s \in S. \neg p \text{ dvd } s)$

definition *prime-generated* :: ' a :: *comm-semiring-1 set* \Rightarrow *bool* **where**
prime-generated $S \iff$
 $(\forall s \in S. \exists M. (\forall q. q \in \# M \longrightarrow q \in S \wedge \text{prime-elem } q) \wedge \text{prod-mset } M = s)$

inductive-set *mult-submonoid-closure* :: ' a :: *comm-monoid-mult set* \Rightarrow ' a *set* **for** A **where**

one-closed: $1 \in \text{mult-submonoid-closure } A$

| *generator*: $a \in A \implies a \in \text{mult-submonoid-closure } A$

| *mult-closed*:

$a \in \text{mult-submonoid-closure } A \implies b \in \text{mult-submonoid-closure } A \implies$

$a * b \in \text{mult-submonoid-closure } A$

definition *powers-set* :: ' a :: *monoid-mult* \Rightarrow ' a *set* **where**

powers-set $p = \{x. \exists n. x = p \wedge n\}$

lemma *prime-generatedI*:

assumes $\bigwedge s. s \in S \implies \exists M. (\forall q. q \in \# M \longrightarrow q \in S \wedge \text{prime-elem } q) \wedge \text{prod-mset } M = s$

shows *prime-generated* S

using *assms* **unfolding** *prime-generated-def* **by** *blast*

lemma *prime-generatedE*:

assumes *prime-generated* S $s \in S$

obtains M **where** $(\forall q. q \in \# M \longrightarrow q \in S \wedge \text{prime-elem } q) \text{prod-mset } M = s$

using *assms* **unfolding** *prime-generated-def* **by** *blast*

```

lemma prime-generated-powers-set:
  assumes prime-elem p
  shows prime-generated (powers-set p)
proof (rule prime-generatedI)
  fix s
  assume s ∈ powers-set p
  then obtain n where hs: s = p ^ n
    unfolding powers-set-def by blast
  have factors:
     $\forall q. q \in \# \text{ replicate-mset } n \ p \longrightarrow q \in \text{ powers-set } p \wedge \text{ prime-elem } q$ 
proof
  fix q
  show  $q \in \# \text{ replicate-mset } n \ p \longrightarrow q \in \text{ powers-set } p \wedge \text{ prime-elem } q$ 
  proof (cases n)
    case 0
    then show ?thesis
      by simp
    next
    case (Suc m)
    show ?thesis
    proof
      assume  $q \in \# \text{ replicate-mset } n \ p$ 
      from  $\langle q \in \# \text{ replicate-mset } n \ p \rangle$  have  $n > 0 \wedge q = p$ 
        by (simp only: in-replicate-mset)
      with Suc have q-eq: q = p
        by simp
      have q-in-powers: q ∈ powers-set p
      by (metis (mono-tags, lifting) mem-Collect-eq power-one-right powers-set-def)
    q-eq)
      from q-eq assms have q-prime: prime-elem q
        by simp
      from q-in-powers q-prime show  $q \in \text{ powers-set } p \wedge \text{ prime-elem } q$ 
        by blast
    qed
  qed
  qed
  show  $\exists M. (\forall q. q \in \# M \longrightarrow q \in \text{ powers-set } p \wedge \text{ prime-elem } q) \wedge \text{ prod-mset } M = s$ 
    by (rule exI[of - replicate-mset n p]) (use factors hs in auto)
qed

```

```

lemma prime-generated-mult-submonoid-closure:
  assumes  $\bigwedge q. q \in A \implies \text{ prime-elem } q$ 
  shows prime-generated (mult-submonoid-closure A)
proof (rule prime-generatedI)
  fix s
  assume s ∈ mult-submonoid-closure A
  then show

```

```

     $\exists M. (\forall q. q \in \# M \longrightarrow q \in \text{mult-submonoid-closure } A \wedge \text{prime-elem } q) \wedge$ 
     $\text{prod-mset } M = s$ 
  proof induction
    case one-closed
    show ?case
    by (rule exI[of - {#}]) (auto intro: mult-submonoid-closure.one-closed)
  next
    case (generator a)
    show ?case
    proof (rule exI)
      from generator assms show
       $(\forall q. q \in \# \{ \# a \# \} \longrightarrow q \in \text{mult-submonoid-closure } A \wedge \text{prime-elem } q) \wedge$ 
       $\text{prod-mset } \{ \# a \# \} = a$ 
      by (auto intro: mult-submonoid-closure.generator)
    qed
  next
    case (mult-closed a b)
    show ?case
    by (metis mult-closed.IH Un-iff prod-mset-Un set-mset-union)
  qed
qed

```

```

lemma zero-notin-prime-generated:
  assumes prime-generated S
  shows  $(0 :: 'a :: \text{semidom}) \notin S$ 
  using assms prime-generated-def by force

```

```

end
theory Localization-Interface
  imports
    HOL-Algebra.Ring-Divisibility
    HOL-Algebra.QuotRing
    Localization-Ring.Localization
  begin

```

3 Localization helper lemmas

The AFP entry *Localization-Ring.Localization* develops localizations as quotient rings in the HOL-Algebra hierarchy. For the present development we package a small wrapper layer at the level of equality of representatives, denominator rescaling, units coming from the multiplicative set, and injectivity of the canonical map.

```

context eq-obj-rng-of-frac
begin

```

```

lemma fraction-eq-iff-rel:
  assumes  $(r, s) \in \text{carrier } rel$ 
  and  $(r', s') \in \text{carrier } rel$ 

```

shows $(r \mid_{rel} s) = (r' \mid_{rel} s') \iff (r, s) \cdot_{=rel} (r', s')$
proof
from *assms* **have** $rs: (r, s) \in carrier\ R \times S$ **and** $rs': (r', s') \in carrier\ R \times S$
by (*simp-all add: rel-def*)
assume $(r \mid_{rel} s) = (r' \mid_{rel} s')$
then show $(r, s) \cdot_{=rel} (r', s')$
using *eq-class-to-rel[of r s r' s']* *rs rs'* **by** *blast*
next
assume $hrel: (r, s) \cdot_{=rel} (r', s')$
have $class-of_{rel} (r, s) = class-of_{rel} (r', s')$
using *equiv-obj-rng-of-frac assms hrel* **by** (*rule elem-eq-class*)
then show $(r \mid_{rel} s) = (r' \mid_{rel} s')$
by (*simp add: class-of-to-rel*)
qed

lemma *fraction-zero-rep* [*simp*]:
assumes $s \in S$
shows $(\mathbf{0} \mid_{rel} s) = \mathbf{0}_{rec-rng-of-frac}$
using *assms* **by** (*rule class-of-zero-rng-of-frac*)

lemma *fraction-surj*:
assumes $x \in carrier\ rec-rng-of-frac$
shows $\exists r \in carrier\ R. \exists s \in S. x = (r \mid_{rel} s)$
using *assms*
unfolding *rec-rng-of-frac-def set-eq-class-of-rng-of-frac-def rel-def*
by *auto*

lemma *fraction-rescale*:
assumes $(r, s) \in carrier\ rel$
and $s' \in S$
shows $(r \mid_{rel} s) = (s' \otimes r \mid_{rel} s' \otimes s)$
using *assms* **by** (*rule simp-in-frac*)

lemma *fraction-mult-rep*:
assumes $rs: (r, s) \in carrier\ rel$
and $r's': (r', s') \in carrier\ rel$
shows $(r \mid_{rel} s) \otimes_{rec-rng-of-frac} (r' \mid_{rel} s') =$
 $(r \otimes_R r' \mid_{rel} s \otimes_R s')$

proof –
have *hfund*:
 $(r \mid_{rel} s) \otimes_{rec-monoid-rng-of-frac} (r' \mid_{rel} s') =$
 $(r \otimes_R r' \mid_{rel} s \otimes_R s')$
using *mult-rng-of-frac-fundamental-lemma[OF rs r's']*
by *simp*
have *hmonoid-mult*:
 $(r \mid_{rel} s) \otimes_{rec-monoid-rng-of-frac} (r' \mid_{rel} s') =$
 $mult-rng-of-frac (r \mid_{rel} s) (r' \mid_{rel} s')$
by (*simp add: rec-monoid-rng-of-frac-def*)
have $(r \mid_{rel} s) \otimes_{rec-rng-of-frac} (r' \mid_{rel} s') =$

$mult\text{-rng-of-frac } (r \mid_{rel} s) (r' \mid_{rel} s')$
by (*simp only: rec-rng-of-frac-def ring-record-simps*)
also have $\dots = (r \mid_{rel} s) \otimes_{rec\text{-monoid-rng-of-frac}} (r' \mid_{rel} s')$
using *hmonoid-mult by simp*
also have $\dots = (r \otimes_R r' \mid_{rel} s \otimes_R s')$
by (*rule hfund*)
finally show *?thesis* .
qed

lemma *map-mul-fraction:*

assumes *a-in: a ∈ carrier R*
and *rs: (r, s) ∈ carrier rel*
shows $rng\text{-to-rng-of-frac } a \otimes_{rec\text{-rng-of-frac}} (r \mid_{rel} s) = (a \otimes_R r \mid_{rel} s)$
proof –
have *one-in: 1 ∈ S*
by (*rule one-closed*)
have *a-rel: (a, 1) ∈ carrier rel*
using *a-in one-in by (simp add: rel-def)*
have *s-in: s ∈ S*
using *rs by (simp add: rel-def)*
have *s-carrier: s ∈ carrier R*
using *subset s-in by blast*
have *one-s: 1 ⊗_R s = s*
using *s-carrier by simp*
have $rng\text{-to-rng-of-frac } a \otimes_{rec\text{-rng-of-frac}} (r \mid_{rel} s) = (a \mid_{rel} \mathbf{1}) \otimes_{rec\text{-rng-of-frac}} (r \mid_{rel} s)$
by (*simp add: rng-to-rng-of-frac-def*)
also have $\dots = (a \otimes_R r \mid_{rel} \mathbf{1} \otimes_R s)$
by (*rule fraction-mult-rep[OF a-rel rs]*)
also have $\dots = (a \otimes_R r \mid_{rel} s)$
using *one-s by simp*
finally show *?thesis* .
qed

lemma *fraction-mul-map:*

assumes *rs: (r, s) ∈ carrier rel*
and *a-in: a ∈ carrier R*
shows $(r \mid_{rel} s) \otimes_{rec\text{-rng-of-frac}} rng\text{-to-rng-of-frac } a = (r \otimes_R a \mid_{rel} s)$
proof –
have *one-in: 1 ∈ S*
by (*rule one-closed*)
have *a-rel: (a, 1) ∈ carrier rel*
using *a-in one-in by (simp add: rel-def)*
have *s-in: s ∈ S*
using *rs by (simp add: rel-def)*
have *s-carrier: s ∈ carrier R*
using *subset s-in by blast*
have *s-one: s ⊗_R 1 = s*
using *s-carrier by simp*

have $(r \mid_{rel} s) \otimes_{rec-rng-of-frac} rng-to-rng-of-frac\ a = (r \mid_{rel} s) \otimes_{rec-rng-of-frac} (a \mid_{rel} \mathbf{1})$
by *(simp add: rng-to-rng-of-frac-def)*
also have $\dots = (r \otimes_R a \mid_{rel} s \otimes_R \mathbf{1})$
by *(rule fraction-mult-rep[OF rs a-rel])*
also have $\dots = (r \otimes_R a \mid_{rel} s)$
using *s-one* **by** *simp*
finally show *?thesis* .
qed

lemma *fraction-eq-iff-cross-multiply*:

assumes *rs*: $(r, s) \in carrier\ rel$
and *rs'*: $(r', s') \in carrier\ rel$
and *zero-notin*: $\mathbf{0} \notin S$
and *no-zero-div*: $\forall a \in carrier\ R. \forall b \in carrier\ R. a \otimes b = \mathbf{0} \longrightarrow a = \mathbf{0} \vee b = \mathbf{0}$
shows $(r \mid_{rel} s) = (r' \mid_{rel} s') \longleftrightarrow s' \otimes_R r = s \otimes_R r'$
proof *(intro iffI)*
from *rs rs'* **have** *r-in*: $r \in carrier\ R$ **and** *s-in*: $s \in S$
and *r'-in*: $r' \in carrier\ R$ **and** *s'-in*: $s' \in S$
by *(simp-all add: rel-def)*
have *s-carrier*: $s \in carrier\ R$ **and** *s'-carrier*: $s' \in carrier\ R$
using *s-in s'-in subset rev-subsetD* **by** *auto*
have *lhs-in*: $s' \otimes_R r \in carrier\ R$
using *s'-carrier r-in* **by** *auto*
have *rhs-in*: $s \otimes_R r' \in carrier\ R$
using *s-carrier r'-in* **by** *auto*
assume *eq-frac*: $(r \mid_{rel} s) = (r' \mid_{rel} s')$
have *hrel*: $(r, s) \cdot_{rel} (r', s')$
using *fraction-eq-iff-rel[OF rs rs'] eq-frac* **by** *blast*
then obtain *t* **where** *t-in*: $t \in S$ **and** *t-zero*: $t \otimes ((s' \otimes_R r) \ominus_R (s \otimes_R r')) = \mathbf{0}$
unfolding *rel-def* **by** *auto*
have *t-carrier*: $t \in carrier\ R$
using *t-in subset rev-subsetD* **by** *auto*
have *diff-in*: $(s' \otimes_R r) \ominus_R (s \otimes_R r') \in carrier\ R$
using *lhs-in rhs-in* **by** *auto*
have $t = \mathbf{0} \vee (s' \otimes_R r) \ominus_R (s \otimes_R r') = \mathbf{0}$
using *no-zero-div t-carrier diff-in t-zero* **by** *blast*
moreover have $t \neq \mathbf{0}$
using *t-in zero-notin* **by** *auto*
ultimately have $(s' \otimes_R r) \ominus_R (s \otimes_R r') = \mathbf{0}$
by *auto*
then have $((s' \otimes_R r) \ominus_R (s \otimes_R r')) \oplus_R (s \otimes_R r') = \mathbf{0} \oplus_R (s \otimes_R r')$
by *simp*
then have $(s' \otimes_R r) \oplus_R ((\ominus_R (s \otimes_R r')) \oplus_R (s \otimes_R r')) = s \otimes_R r'$
using *lhs-in rhs-in* **by** *(simp add: a-minus-def a-assoc)*

have *inv-cancel*: $(\ominus_R (s \otimes_R r')) \oplus_R (s \otimes_R r') = \mathbf{0}$
using *rhs-in* **by** (*rule l-neg*)
then have $(s' \otimes_R r) \oplus_R ((\ominus_R (s \otimes_R r')) \oplus_R (s \otimes_R r')) =$
 $(s' \otimes_R r) \oplus_R \mathbf{0}$
by *simp*
then have $(s' \otimes_R r) \oplus_R \mathbf{0} = s \otimes_R r'$
using $\langle (s' \otimes_R r) \oplus_R ((\ominus_R (s \otimes_R r')) \oplus_R (s \otimes_R r')) =$
 $s \otimes_R r' \rangle$
by *simp*
then show $s' \otimes_R r = s \otimes_R r'$
using *lhs-in* **by** (*simp add: r-zero*)
next
assume *cross-mul*: $s' \otimes_R r = s \otimes_R r'$
from *rs rs'* **have** *r-in'*: $r \in \text{carrier } R$ **and** *s'-in'*: $s' \in S$
by (*simp-all add: rel-def*)
have *lhs-in'*: $s' \otimes_R r \in \text{carrier } R$
proof –
have $s' \in \text{carrier } R$
using *s'-in'* *subset rev-subsetD* **by** *auto*
then show *?thesis*
using *r-in'* **by** *auto*
qed
have *diff-zero*: $(s' \otimes_R r) \ominus_R (s \otimes_R r') = \mathbf{0}$
proof –
have $(s' \otimes_R r) \ominus_R (s \otimes_R r') = (s' \otimes_R r) \ominus_R (s' \otimes_R r)$
using *cross-mul* **by** *simp*
also have $\dots = (s' \otimes_R r) \oplus_R (\ominus_R (s' \otimes_R r))$
by (*simp add: a-minus-def*)
also have $\dots = \mathbf{0}$
using *lhs-in'* **by** (*rule r-neg*)
finally show *?thesis* .
qed
have $(r, s) \text{ .}=_\text{rel} (r', s')$
proof –
have $\mathbf{1} \in S$
by (*rule one-closed*)
moreover have $\mathbf{1} \otimes ((s' \otimes_R r) \ominus_R (s \otimes_R r')) = \mathbf{0}$
using *diff-zero* **by** *simp*
ultimately show *?thesis*
unfolding *rel-def* **using** *rs rs'* **by** *auto*
qed
then show $(r \mid_{\text{rel}} s) = (r' \mid_{\text{rel}} s')$
using *fraction-eq-iff-rel[OF rs rs']* **by** *blast*
qed
lemma *fraction-eq-zero-iff*:
assumes *rs*: $(r, s) \in \text{carrier rel}$
and *zero-notin*: $\mathbf{0} \notin S$
and *no-zero-div*: $\forall a \in \text{carrier } R. \forall b \in \text{carrier } R. a \otimes b = \mathbf{0} \longrightarrow a = \mathbf{0} \vee b =$

0
shows $(r \mid_{rel} s) = \mathbf{0}_{rec-rng-of-frac} \longleftrightarrow r = \mathbf{0}$
proof
from rs **have** $s\text{-in}: s \in S$
by (*simp add: rel-def*)
have $s\text{-carrier}: s \in carrier\ R$
using *subset s-in* **by** *blast*
from rs **have** $r\text{-in}: r \in carrier\ R$
by (*simp add: rel-def*)
have $zero\text{-rel}: (\mathbf{0}, \mathbf{1}) \in carrier\ rel$
by (*simp add: rel-def one-closed*)
assume $hzero: (r \mid_{rel} s) = \mathbf{0}_{rec-rng-of-frac}$
have $(r \mid_{rel} s) = (\mathbf{0} \mid_{rel} \mathbf{1})$
using $hzero$ **by** (*simp add: class-of-zero-rng-of-frac[OF one-closed]*)
then have $\mathbf{1} \otimes_R r = s \otimes_R \mathbf{0}$
using *fraction-eq-iff-cross-multiply[OF rs zero-rel zero-notin no-zero-div]*
by *simp*
then have $r = s \otimes_R \mathbf{0}$
using $r\text{-in}$ **by** *simp*
also have $\dots = \mathbf{0}$
using $s\text{-carrier}$ **by** *simp*
finally show $r = \mathbf{0}$.
next
assume $r = \mathbf{0}$
then have $r\text{-zero}: r = \mathbf{0}$.
have $s\text{-in}: s \in S$
using rs **by** (*simp add: rel-def*)
from $r\text{-zero}$ $s\text{-in}$ **show** $(r \mid_{rel} s) = \mathbf{0}_{rec-rng-of-frac}$
by *simp*
qed

lemma *map-eq-zero-iff*:
assumes $a\text{-in}: a \in carrier\ R$
and $zero\text{-notin}: \mathbf{0} \notin S$
and $no\text{-zero-div}: \forall a' \in carrier\ R. \forall b' \in carrier\ R. a' \otimes b' = \mathbf{0} \longrightarrow a' = \mathbf{0} \vee b' = \mathbf{0}$
shows $rng\text{-to-rng-of-frac}\ a = \mathbf{0}_{rec-rng-of-frac} \longleftrightarrow a = \mathbf{0}$
proof –
have $a\text{-rel}: (a, \mathbf{1}) \in carrier\ rel$
using $a\text{-in}$ *one-closed* **by** (*simp add: rel-def*)
show *?thesis*
using *fraction-eq-zero-iff[OF a-rel zero-notin no-zero-div]*
by (*simp add: rng-to-rng-of-frac-def*)
qed

lemma *dvd-map-iff*:
assumes $a\text{-in}: a \in carrier\ R$
and $b\text{-in}: b \in carrier\ R$
and $zero\text{-notin}: \mathbf{0} \notin S$

and no-zero-div: $\forall a' \in \text{carrier } R. \forall b' \in \text{carrier } R. a' \otimes b' = \mathbf{0} \longrightarrow a' = \mathbf{0} \vee b' = \mathbf{0}$
shows $\text{rng-to-rng-of-frac } a \text{ divides}_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } b \longleftrightarrow (\exists s \in S. a \text{ divides}_R (s \otimes_R b))$
proof
assume $hdiv: \text{rng-to-rng-of-frac } a \text{ divides}_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } b$
then obtain x **where** $x\text{-mem}: x \in \text{carrier } \text{rec-rng-of-frac}$
and $hx: \text{rng-to-rng-of-frac } b = \text{rng-to-rng-of-frac } a \otimes_{\text{rec-rng-of-frac}} x$
unfolding factor-def **by** blast
from $\text{fraction-surj}[OF x\text{-mem}]$ **obtain** c **where** $c\text{-in}: c \in \text{carrier } R$
and $hs: \exists s \in S. x = (c \mid_{\text{rel}} s)$
by blast
from hs **obtain** s **where** $s\text{-in}: s \in S$ **and** $x\text{-def}: x = (c \mid_{\text{rel}} s)$
by blast
have $b\text{-rel}: (b, \mathbf{1}) \in \text{carrier } \text{rel}$
using $b\text{-in one-closed}$ **by** $(\text{simp add: rel-def})$
have $cs\text{-rel}: (c, s) \in \text{carrier } \text{rel}$
using $c\text{-in s-in}$ **by** $(\text{simp add: rel-def})$
have $ac\text{-rel}: (a \otimes_R c, s) \in \text{carrier } \text{rel}$
using $a\text{-in c-in s-in}$ **by** $(\text{simp add: rel-def})$
have $(b \mid_{\text{rel}} \mathbf{1}) = (a \otimes_R c \mid_{\text{rel}} s)$
proof –
have $h1: \text{rng-to-rng-of-frac } b = \text{rng-to-rng-of-frac } a \otimes_{\text{rec-rng-of-frac}} (c \mid_{\text{rel}} s)$
using $hx x\text{-def}$ **by** simp
have $h2: \text{rng-to-rng-of-frac } a \otimes_{\text{rec-rng-of-frac}} (c \mid_{\text{rel}} s) = (a \otimes_R c \mid_{\text{rel}} s)$
using $\text{map-mul-fraction}[OF a\text{-in cs-rel}]$ **by** simp
from $h1 h2$ **show** $?thesis$
by $(\text{simp add: rng-to-rng-of-frac-def})$
qed
then have $s \otimes_R b = \mathbf{1} \otimes_R (a \otimes_R c)$
using $\text{fraction-eq-iff-cross-multiply}[OF b\text{-rel ac-rel zero-notin no-zero-div}]$
by simp
then have $s \otimes_R b = a \otimes_R c$
using $a\text{-in c-in}$ **by** simp
then have $a \text{ divides}_R (s \otimes_R b)$
unfolding factor-def **using** $c\text{-in}$ **by** blast
then show $\exists s \in S. a \text{ divides}_R (s \otimes_R b)$
using $s\text{-in}$ **by** blast
next
assume $hdiv: \exists s \in S. a \text{ divides}_R (s \otimes_R b)$
then obtain s **where** $s\text{-in}: s \in S$ **and** $hsab: a \text{ divides}_R (s \otimes_R b)$
by blast
then obtain c **where** $c\text{-in}: c \in \text{carrier } R$ **and** $hc: s \otimes_R b = a \otimes_R c$
unfolding factor-def **by** blast
have $cs\text{-rel}: (c, s) \in \text{carrier } \text{rel}$
using $c\text{-in s-in}$ **by** $(\text{simp add: rel-def})$
have $ac\text{-rel}: (a \otimes_R c, s) \in \text{carrier } \text{rel}$
using $a\text{-in c-in s-in}$ **by** $(\text{simp add: rel-def})$
have $b\text{-rel}: (b, \mathbf{1}) \in \text{carrier } \text{rel}$

```

    using b-in one-closed by (simp add: rel-def)
  have eq-frac:  $(b \mid_{rel} \mathbf{1}) = (a \otimes_R c \mid_{rel} s)$ 
  proof -
    have  $s \otimes_R b = \mathbf{1} \otimes_R (a \otimes_R c)$ 
      using hc a-in c-in by simp
    then show ?thesis
      using fraction-eq-iff-cross-multiply[OF b-rel ac-rel zero-notin no-zero-div]
      by blast
  qed
  have frac-mem:  $(c \mid_{rel} s) \in \text{carrier } \text{rec-rng-of-frac}$ 
    using cs-rel unfolding rec-rng-of-frac-def set-eq-class-of-rng-of-frac-def
    by auto
  have rng-to-rng-of-frac  $b = \text{rng-to-rng-of-frac } a \otimes_{\text{rec-rng-of-frac}} (c \mid_{rel} s)$ 
  proof -
    have h1:  $\text{rng-to-rng-of-frac } b = (b \mid_{rel} \mathbf{1})$ 
      by (simp add: rng-to-rng-of-frac-def)
    have h2:  $(b \mid_{rel} \mathbf{1}) = (a \otimes_R c \mid_{rel} s)$ 
      using eq-frac by simp
    have h3:  $\text{rng-to-rng-of-frac } a \otimes_{\text{rec-rng-of-frac}} (c \mid_{rel} s) = (a \otimes_R c \mid_{rel} s)$ 
      using map-mul-fraction[OF a-in cs-rel] by simp
    from h1 h2 h3 show ?thesis
      by simp
  qed
  then show rng-to-rng-of-frac  $a \text{ divides}_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } b$ 
    unfolding factor-def using frac-mem by blast
  qed

lemma image-submonoid-is-unit:
  assumes  $x \in \text{rng-to-rng-of-frac } S$ 
  shows  $x \in \text{Units } \text{rec-rng-of-frac}$ 
  using assms by (rule Im-rng-to-rng-of-frac-unit)

lemma map-submonoid-elem-is-unit:
  assumes  $s \in S$ 
  shows  $\text{rng-to-rng-of-frac } s \in \text{Units } \text{rec-rng-of-frac}$ 
  using assms image-submonoid-is-unit by blast

lemma map-unit-is-unit:
  assumes u-unit:  $u \in \text{Units } R$ 
  shows  $\text{rng-to-rng-of-frac } u \in \text{Units } \text{rec-rng-of-frac}$ 
  proof -
    have u-in:  $u \in \text{carrier } R$ 
      using u-unit by blast
    have inv-u-in:  $\text{inv } u \in \text{carrier } R$ 
      using u-unit by blast
    have map-u-in:  $\text{rng-to-rng-of-frac } u \in \text{carrier } \text{rec-rng-of-frac}$ 
      using ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom u-in] .
    have map-inv-u-in:  $\text{rng-to-rng-of-frac } (\text{inv } u) \in \text{carrier } \text{rec-rng-of-frac}$ 
      using ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom inv-u-in] .
  qed

```

have *left-inv*:
 $\text{rng-to-rng-of-frac } u \otimes_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } (\text{inv } u) =$
 $\mathbf{1}_{\text{rec-rng-of-frac}}$
proof –
have $\text{rng-to-rng-of-frac } u \otimes_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } (\text{inv } u) =$
 $\text{rng-to-rng-of-frac } (u \otimes_R \text{inv } u)$
using *ring-hom-mult*[*OF rng-to-rng-of-frac-is-ring-hom u-in inv-u-in*]
by *simp*
also have $\dots = \text{rng-to-rng-of-frac } \mathbf{1}$
using *u-unit* **by** *simp*
also have $\dots = \mathbf{1}_{\text{rec-rng-of-frac}}$
using *ring-hom-one*[*OF rng-to-rng-of-frac-is-ring-hom*] .
finally show *?thesis* .
qed
have *right-inv*:
 $\text{rng-to-rng-of-frac } (\text{inv } u) \otimes_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } u =$
 $\mathbf{1}_{\text{rec-rng-of-frac}}$
proof –
have $\text{rng-to-rng-of-frac } (\text{inv } u) \otimes_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } u =$
 $\text{rng-to-rng-of-frac } (\text{inv } u \otimes_R u)$
using *ring-hom-mult*[*OF rng-to-rng-of-frac-is-ring-hom inv-u-in u-in*]
by *simp*
also have $\dots = \text{rng-to-rng-of-frac } \mathbf{1}$
using *u-unit* **by** *simp*
also have $\dots = \mathbf{1}_{\text{rec-rng-of-frac}}$
using *ring-hom-one*[*OF rng-to-rng-of-frac-is-ring-hom*] .
finally show *?thesis* .
qed
show *?thesis*
unfolding *Units-def*
using *map-u-in map-inv-u-in left-inv right-inv* **by** *blast*
qed

lemma *fraction-unit-numerator-is-unit*:
assumes *u-unit*: $u \in \text{Units } R$
and *s-in*: $s \in S$
shows $(u \mid_{\text{rel}} s) \in \text{Units rec-rng-of-frac}$
proof –
have *u-in*: $u \in \text{carrier } R$
using *Units-closed*[*OF u-unit*] .
have *s-carrier*: $s \in \text{carrier } R$
using *subset s-in* **by** *blast*
have *u-rel*: $(u, \mathbf{1}) \in \text{carrier rel}$
using *u-in one-closed* **by** (*simp add: rel-def*)
have *one-rel*: $(\mathbf{1}, s) \in \text{carrier rel}$
using *s-in* **by** (*simp add: rel-def*)
have *frac-eq'*:
 $\text{rng-to-rng-of-frac } u \otimes_{\text{rec-rng-of-frac}} (\mathbf{1} \mid_{\text{rel}} s) =$
 $(u \mid_{\text{rel}} s)$

proof –
have $\text{rng-to-rng-of-frac } u \otimes_{\text{rec-rng-of-frac}} (\mathbf{1} \mid_{\text{rel}} s) =$
 $(u \mid_{\text{rel}} \mathbf{1}) \otimes_{\text{rec-rng-of-frac}} (\mathbf{1} \mid_{\text{rel}} s)$
by (*simp add: rng-to-rng-of-frac-def*)
also have $\dots = (u \otimes_R \mathbf{1} \mid_{\text{rel}} \mathbf{1} \otimes_R s)$
by (*rule fraction-mult-rep[OF u-rel one-rel]*)
also have $\dots = (u \mid_{\text{rel}} s)$
using *u-in s-carrier* **by** *simp*
finally show *?thesis* .
qed
have *frac-eq*:
 $(u \mid_{\text{rel}} s) =$
 $\text{rng-to-rng-of-frac } u \otimes_{\text{rec-rng-of-frac}} (\mathbf{1} \mid_{\text{rel}} s)$
using *frac-eq'* **by** *simp*
have *map-unit*: $\text{rng-to-rng-of-frac } u \in \text{Units rec-rng-of-frac}$
using *u-unit* **by** (*rule map-unit-is-unit*)
have *denom-unit*: $(\mathbf{1} \mid_{\text{rel}} s) \in \text{Units rec-rng-of-frac}$
proof –
have *s-carrier*: $s \in \text{carrier } R$
using *subset s-in* **by** *blast*
have *s-rel*: $(s, \mathbf{1}) \in \text{carrier rel}$
using *s-carrier one-closed* **by** (*simp add: rel-def*)
have *left-inv*:
 $(\mathbf{1} \mid_{\text{rel}} s) \otimes_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } s =$
 $\mathbf{1}_{\text{rec-rng-of-frac}}$
proof –
have $(\mathbf{1} \mid_{\text{rel}} s) \otimes_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } s =$
 $(\mathbf{1} \mid_{\text{rel}} s) \otimes_{\text{rec-rng-of-frac}} (s \mid_{\text{rel}} \mathbf{1})$
by (*simp add: rng-to-rng-of-frac-def*)
also have $\dots = (\mathbf{1} \otimes_R s \mid_{\text{rel}} s \otimes_R \mathbf{1})$
by (*rule fraction-mult-rep[OF one-rel s-rel]*)
also have $\dots = (\mathbf{1} \mid_{\text{rel}} \mathbf{1})$
proof –
have *one-rel'*: $(\mathbf{1}, \mathbf{1}) \in \text{carrier rel}$
using *one-closed* **by** (*simp add: rel-def*)
have $(\mathbf{1} \mid_{\text{rel}} \mathbf{1}) = (s \otimes_R \mathbf{1} \mid_{\text{rel}} s \otimes_R \mathbf{1})$
by (*rule fraction-rescale[OF one-rel' s-in]*)
then show *?thesis*
using *s-carrier* **by** *simp*
qed
also have $\dots = \mathbf{1}_{\text{rec-rng-of-frac}}$
by (*simp only: rec-rng-of-frac-def ring-record-simps*)
finally show *?thesis* .
qed
have *right-inv*:
 $\text{rng-to-rng-of-frac } s \otimes_{\text{rec-rng-of-frac}} (\mathbf{1} \mid_{\text{rel}} s) =$
 $\mathbf{1}_{\text{rec-rng-of-frac}}$
proof –
have $\text{rng-to-rng-of-frac } s \otimes_{\text{rec-rng-of-frac}} (\mathbf{1} \mid_{\text{rel}} s) =$

$(s \mid_{rel} \mathbf{1}) \otimes_{rec-rng-of-frac} (\mathbf{1} \mid_{rel} s)$
by (*simp add: rng-to-rng-of-frac-def*)
also have $\dots = (s \otimes_R \mathbf{1} \mid_{rel} \mathbf{1} \otimes_R s)$
by (*rule fraction-mult-rep[OF s-rel one-rel]*)
also have $\dots = (\mathbf{1} \mid_{rel} \mathbf{1})$
proof –
have *one-rel'*: $(\mathbf{1}, \mathbf{1}) \in carrier\ rel$
using *one-closed* **by** (*simp add: rel-def*)
have $(\mathbf{1} \mid_{rel} \mathbf{1}) = (s \otimes_R \mathbf{1} \mid_{rel} s \otimes_R \mathbf{1})$
by (*rule fraction-rescale[OF one-rel' s-in]*)
then show *?thesis*
using *s-carrier* **by** *simp*
qed
also have $\dots = \mathbf{1}_{rec-rng-of-frac}$
by (*simp only: rec-rng-of-frac-def ring-record-simps*)
finally show *?thesis* .
qed
have *frac-in*: $(\mathbf{1} \mid_{rel} s) \in carrier\ rec-rng-of-frac$
proof –
have *one-s-rel*: $(\mathbf{1}, s) \in carrier\ rel$
using *s-in* **by** (*simp add: rel-def*)
show *?thesis*
using *one-s-rel*
unfolding *rec-rng-of-frac-def set-eq-class-of-rng-of-frac-def*
by *auto*
qed
have *s-carrier*: $s \in carrier\ R$
using *s-in subset rev-subsetD* **by** *blast*
have *map-s-in*: $rng-to-rng-of-frac\ s \in carrier\ rec-rng-of-frac$
using *ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom s-carrier]* .
show *?thesis*
unfolding *Units-def* **using** *frac-in map-s-in left-inv right-inv* **by** *blast*
qed
show *?thesis*
proof –
have *rng-to-rng-of-frac u* $\otimes_{rec-rng-of-frac} (\mathbf{1} \mid_{rel} s) \in Units\ rec-rng-of-frac$
by (*rule monoid.Units-m-closed[OF ring.is-monoid[OF rng-rng-of-frac] map-unit*
denom-unit])
then show *?thesis*
using *frac-eq* **by** *simp*
qed
qed

lemma *map-inj-on*:
assumes $\mathbf{0} \notin S$
and $\forall a \in carrier\ R. \forall b \in carrier\ R. a \otimes b = \mathbf{0} \longrightarrow a = \mathbf{0} \vee b = \mathbf{0}$
shows *inj-on rng-to-rng-of-frac (carrier R)*
proof –
have *a-kernel R rec-rng-of-frac rng-to-rng-of-frac* $= \{\mathbf{0}\}$

```

    using assms by (rule rng-to-rng-of-frac-without-zero-div-is-inj)
  then show ?thesis
    using ring-hom-ring.trivial-ker-imp-inj[
      OF ring-hom-ringI2[OF ring-axioms rng-rng-of-frac rng-to-rng-of-frac-is-ring-hom]
    ]
    by blast
qed

end

end
theory Nagata-Lemmas
  imports Localization-Interface
begin

```

4 Record-based Nagata descent lemmas

```

definition ring-avoids ::
  ('a, 'b) ring-scheme  $\Rightarrow$  'a set  $\Rightarrow$  'a  $\Rightarrow$  bool
where
  ring-avoids R S p  $\longleftrightarrow$  ( $\forall s \in S. \neg p \text{ divides}_R s$ )

```

```

definition ring-prime-generated ::
  ('a, 'b) ring-scheme  $\Rightarrow$  'a set  $\Rightarrow$  bool
where
  ring-prime-generated R S  $\longleftrightarrow$ 
    ( $\forall s \in S. \exists fs.$ 
      set fs  $\subseteq S \wedge$ 
      ( $\forall q \in \text{set } fs. \text{ring-prime}_R q$ )  $\wedge$ 
      foldr ( $\otimes_R$ ) fs  $\mathbf{1}_R = s$ )

```

```

lemma ring-prime-generatedI:
  assumes  $\bigwedge s. s \in S \implies \exists fs.$ 
    set fs  $\subseteq S \wedge$ 
    ( $\forall q \in \text{set } fs. \text{ring-prime}_R q$ )  $\wedge$ 
    foldr ( $\otimes_R$ ) fs  $\mathbf{1}_R = s$ 
  shows ring-prime-generated R S
  using assms unfolding ring-prime-generated-def by blast

```

```

lemma ring-prime-generatedE:
  assumes ring-prime-generated R S s  $\in S$ 
  obtains fs where
    set fs  $\subseteq S$ 
     $\forall q \in \text{set } fs. \text{ring-prime}_R q$ 
    foldr ( $\otimes_R$ ) fs  $\mathbf{1}_R = s$ 
  using assms unfolding ring-prime-generated-def by blast

```

```

definition ring-powers-set ::
  ('a, 'b) ring-scheme  $\Rightarrow$  'a  $\Rightarrow$  'a set

```

where

ring-powers-set R $p = \{x. \exists n::nat. x = p [\bigwedge]_R n\}$

inductive-set *ring-mult-submonoid-closure* ::

$(\text{'a}, \text{'b})$ *ring-scheme* \Rightarrow 'a *set* \Rightarrow 'a *set*

for R **and** A

where

one-closed: $\mathbf{1}_R \in \text{ring-mult-submonoid-closure } R A$

| *generator*: $a \in A \Rightarrow a \in \text{ring-mult-submonoid-closure } R A$

| *mult-closed*:

$a \in \text{ring-mult-submonoid-closure } R A \Rightarrow$

$b \in \text{ring-mult-submonoid-closure } R A \Rightarrow$

$a \otimes_R b \in \text{ring-mult-submonoid-closure } R A$

lemma *ring-mult-submonoid-closure-subset*:

assumes *ring-R*: *ring* R

and *A-sub*: $A \subseteq \text{carrier } R$

shows *ring-mult-submonoid-closure* $R A \subseteq \text{carrier } R$

proof –

interpret R : *ring* R

by (*rule ring-R*)

show *?thesis*

proof

fix x

assume *x-in*: $x \in \text{ring-mult-submonoid-closure } R A$

then show $x \in \text{carrier } R$

by *induction (use A-sub in auto)*

qed

qed

lemma *ring-mult-submonoid-closure-submonoid*:

assumes *ring-R*: *ring* R

and *A-sub*: $A \subseteq \text{carrier } R$

shows *submonoid* R (*ring-mult-submonoid-closure* $R A$)

proof –

interpret R : *ring* R

by (*rule ring-R*)

show *?thesis*

proof

show *ring-mult-submonoid-closure* $R A \subseteq \text{carrier } R$

by (*rule ring-mult-submonoid-closure-subset[OF ring-R A-sub]*)

qed (*auto simp: ring-mult-submonoid-closure.one-closed ring-mult-submonoid-closure.mult-closed*)

qed

lemma *foldr-mult-right*:

assumes *ring-R*: *ring* R

and *xs-sub*: $\text{set } xs \subseteq \text{carrier } R$

and *y-in*: $y \in \text{carrier } R$

shows *foldr* (\otimes_R) xs $y =$

```

    foldr ( $\otimes_R$ ) xs  $\mathbf{1}_R \otimes_R y$ 
proof –
  interpret  $R$ : ring  $R$ 
    by (rule ring- $R$ )
  show ?thesis
    using xs-sub y-in
  proof (induction xs)
    case Nil
    then show ?case
      by simp
  next
    case (Cons x xs)
    have x-in:  $x \in \text{carrier } R$ 
      using Cons.prem(1) by simp
    have xs-sub': set xs  $\subseteq \text{carrier } R$ 
      using Cons.prem(1) by simp
    have prod-in: foldr ( $\otimes_R$ ) xs  $\mathbf{1}_R \in \text{carrier } R$ 
      using xs-sub'
    proof (induction xs)
      case Nil
      then show ?case by simp
    next
      case (Cons z zs)
      then show ?case by simp
    qed
  show ?case
    using Cons.IH[OF xs-sub' Cons.prem(2)] x-in prod-in Cons.prem(2)
    by (simp add: R.m-assoc)
  qed
qed

lemma ring-powers-submonoid:
  assumes ring- $R$ : ring  $R$ 
    and p-in:  $p \in \text{carrier } R$ 
  shows submonoid  $R$  (ring-powers-set  $R$  p)
proof –
  interpret  $R$ : ring  $R$ 
    by (rule ring- $R$ )
  show ?thesis
  proof (unfold-locales)
    show ring-powers-set  $R$  p  $\subseteq \text{carrier } R$ 
      using p-in unfolding ring-powers-set-def by auto
    show  $\bigwedge x y. x \in \text{ring-powers-set } R p \implies y \in \text{ring-powers-set } R p \implies x \otimes_R y \in \text{ring-powers-set } R p$ 
  proof –
    fix x y
    assume x-in:  $x \in \text{ring-powers-set } R p$  and y-in:  $y \in \text{ring-powers-set } R p$ 
    then obtain m n :: nat where x-def:  $x = p [\frown]_R m$  and y-def:  $y = p [\frown]_R n$ 
      unfolding ring-powers-set-def by blast

```

```

    show  $x \otimes_R y \in \text{ring-powers-set } R \ p$ 
      using  $R.\text{nat-pow-mult } p\text{-in } \text{ring-powers-set-def } x\text{-def } y\text{-def}$  by fastforce
  qed
  show  $\mathbf{1}_R \in \text{ring-powers-set } R \ p$ 
  proof -
    have  $\mathbf{1}_R = p \ [\frown]_R \ (0::\text{nat})$ 
      using  $p\text{-in}$  by simp
    then show ?thesis
      unfolding  $\text{ring-powers-set-def}$  by blast
  qed
qed
qed
qed

lemma ring-prime-generated-powers-set:
  assumes  $\text{ring-}R$ :  $\text{ring } R$ 
    and  $p\text{-in}$ :  $p \in \text{carrier } R$ 
    and  $hp$ :  $\text{ring-prime}_R \ p$ 
  shows  $\text{ring-prime-generated } R \ (\text{ring-powers-set } R \ p)$ 
proof (rule ring-prime-generatedI)
  interpret  $R$ :  $\text{ring } R$ 
  by (rule  $\text{ring-}R$ )
  fix  $s$ 
  assume  $s\text{-in}$ :  $s \in \text{ring-powers-set } R \ p$ 
  then obtain  $n :: \text{nat}$  where  $s\text{-def}$ :  $s = p \ [\frown]_R \ n$ 
    unfolding  $\text{ring-powers-set-def}$  by blast
  show  $\exists fs.$ 
     $\text{set } fs \subseteq \text{ring-powers-set } R \ p \wedge$ 
     $(\forall q \in \text{set } fs. \text{ring-prime}_R \ q) \wedge$ 
     $\text{foldr } (\otimes_R) \ fs \ \mathbf{1}_R = s$ 
proof (intro exI conjI)
  show  $\text{set } (\text{replicate } n \ p) \subseteq \text{ring-powers-set } R \ p$ 
  proof
    fix  $q$ 
    assume  $q\text{-in}$ :  $q \in \text{set } (\text{replicate } n \ p)$ 
    then have  $q\text{-eq}$ :  $q = p$ 
      by simp
    show  $q \in \text{ring-powers-set } R \ p$ 
      by (metis (mono-tags, lifting)  $R.\text{nat-pow-eone mem-Collect-eq } p\text{-in } q\text{-eq}$ 
         $\text{ring-powers-set-def}$ )
  qed
  show  $(\forall q \in \text{set } (\text{replicate } n \ p). \text{ring-prime}_R \ q)$ 
    using  $hp$  by simp
  show  $\text{foldr } (\otimes_R) \ (\text{replicate } n \ p) \ \mathbf{1}_R = s$ 
  proof -
    have  $\text{foldr } (\otimes_R) \ (\text{replicate } n \ p) \ \mathbf{1}_R = p \ [\frown]_R \ n$ 
      using  $p\text{-in}$ 
    proof (induction  $n$ )
      case 0
      then show ?case by simp
    end
  end
end

```

```

next
  case (Suc n)
  have foldr ( $\otimes_R$ ) (replicate (Suc n) p)  $\mathbf{1}_R =$ 
    p  $\otimes_R$  p [ $\bigwedge_R$ ] n
    using Suc.IH p-in by simp
  also have ... = p [ $\bigwedge_R$ ] (Suc n)
    by (rule sym[OF R.nat-pow-Suc2[OF p-in]])
  finally show ?case .
qed
then show ?thesis
  using s-def by simp
qed
qed
qed

```

lemma ring-prime-generated-mult-submonoid-closure:

```

assumes ring-R: ring R
  and A-sub:  $A \subseteq \text{carrier } R$ 
  and hprime:  $\bigwedge q. q \in A \implies \text{ring-prime}_R q$ 
shows ring-prime-generated R (ring-mult-submonoid-closure R A)
proof (rule ring-prime-generatedI)
  interpret R: ring R
  by (rule ring-R)
  have closure-sub: ring-mult-submonoid-closure R A  $\subseteq$  carrier R
  by (rule ring-mult-submonoid-closure-subset[OF ring-R A-sub])
  fix s
  assume s-in:  $s \in \text{ring-mult-submonoid-closure } R A$ 
  show  $\exists fs.$ 
    set fs  $\subseteq$  ring-mult-submonoid-closure R A  $\wedge$ 
    ( $\forall q \in \text{set } fs. \text{ring-prime}_R q$ )  $\wedge$ 
    foldr ( $\otimes_R$ ) fs  $\mathbf{1}_R = s$ 
  using s-in
proof induction
  case one-closed
  show ?case
  by (intro exI[of - []]) auto
next
  case (generator a)
  show ?case
proof (intro exI[of - [a]] conjI)
  have a-in-closure:  $a \in \text{ring-mult-submonoid-closure } R A$ 
  by (rule ring-mult-submonoid-closure.generator[OF generator.hyps])
  show set [a]  $\subseteq$  ring-mult-submonoid-closure R A
  using a-in-closure by simp
  show  $\forall q \in \text{set } [a]. \text{ring-prime}_R q$ 
  using hprime generator.hyps by simp
  have a-in:  $a \in \text{carrier } R$ 
  using A-sub generator.hyps by blast
  show foldr ( $\otimes_R$ ) [a]  $\mathbf{1}_R = a$ 

```

```

    using a-in by simp
  qed
next
case (mult-closed a b)
from mult-closed.IH(1) obtain fs where
  fs-sub: set fs  $\subseteq$  ring-mult-submonoid-closure R A
  and fs-prime:  $\forall q \in \text{set } fs. \text{ring-prime}_R q$ 
  and fs-prod: foldr ( $\otimes_R$ ) fs  $\mathbf{1}_R = a$ 
by blast
from mult-closed.IH(2) obtain gs where
  gs-sub: set gs  $\subseteq$  ring-mult-submonoid-closure R A
  and gs-prime:  $\forall q \in \text{set } gs. \text{ring-prime}_R q$ 
  and gs-prod: foldr ( $\otimes_R$ ) gs  $\mathbf{1}_R = b$ 
by blast
have fs-carr: set fs  $\subseteq$  carrier R
using fs-sub closure-sub by blast
have b-in: b  $\in$  carrier R
using mult-closed.hyps(2) closure-sub by blast
have prod-append:
  foldr ( $\otimes_R$ ) (fs @ gs)  $\mathbf{1}_R = a \otimes_R b$ 
proof -
  have foldr ( $\otimes_R$ ) (fs @ gs)  $\mathbf{1}_R =$ 
    foldr ( $\otimes_R$ ) fs (foldr ( $\otimes_R$ ) gs  $\mathbf{1}_R$ )
  by simp
  also have ... = foldr ( $\otimes_R$ ) fs b
  by (simp add: gs-prod)
  also have ... = foldr ( $\otimes_R$ ) fs  $\mathbf{1}_R \otimes_R b$ 
  by (rule foldr-mult-right[OF ring-R fs-carr b-in])
  also have ... = a  $\otimes_R b$ 
  by (simp add: fs-prod)
  finally show ?thesis .
qed
show ?case
by (intro exI[of - fs @ gs])
  (use fs-sub fs-prime gs-sub gs-prime prod-append in auto)
qed
qed

locale nagata-localization = eq-obj-rng-of-frac R S + domain R for R (structure)
and S
begin

lemma no-zero-divisors:
   $\forall a \in \text{carrier } R. \forall b \in \text{carrier } R. a \otimes_R b = \mathbf{0} \longrightarrow a = \mathbf{0} \vee b = \mathbf{0}$ 
  using integral by blast

lemma multlist-closed:
  assumes xs-sub: set xs  $\subseteq$  carrier R
  shows foldr ( $\otimes_R$ ) xs  $\mathbf{1}_R \in$  carrier R

```

```

using xs-sub
by (induction xs) auto

lemma multlist-mem-submonoid:
  assumes fs-sub: set fs  $\subseteq S$ 
  shows foldr ( $\otimes_R$ ) fs  $\mathbf{1}_R \in S$ 
  using fs-sub
proof (induction fs)
  case Nil
  show ?case
    by simp
next
  case (Cons q qs)
  have q-in: q  $\in S$ 
    using Cons.prems by simp
  have qs-in: set qs  $\subseteq S$ 
    using Cons.prems by simp
  show ?case
    using q-in Cons.IH[OF qs-in] by simp
qed

lemma multlist-nonzero-of-prime-factors:
  assumes fs-sub: set fs  $\subseteq S$ 
    and hf:  $\forall q \in \text{set } fs. \text{ring-prime}_R q$ 
  shows foldr ( $\otimes_R$ ) fs  $\mathbf{1}_R \neq \mathbf{0}$ 
  using fs-sub hf
proof (induction fs)
  case Nil
  show ?case
    by simp
next
  case (Cons q qs)
  have q-in: q  $\in S$ 
    using Cons.prems(1) by simp
  have q-carr: q  $\in \text{carrier } R$ 
    using q-in subset rev-subsetD by blast
  have q-prime: ring-primeR q
    using Cons.prems(2) by simp
  have q-nz: q  $\neq \mathbf{0}$ 
    using ring-primeE(1)[OF q-carr q-prime] .
  have qs-sub: set qs  $\subseteq S$ 
    using Cons.prems(1) by simp
  have qs-prime:  $\forall r \in \text{set } qs. \text{ring-prime}_R r$ 
    using Cons.prems(2) by simp
  have qs-nz: foldr ( $\otimes_R$ ) qs  $\mathbf{1}_R \neq \mathbf{0}$ 
    using Cons.IH[OF qs-sub qs-prime] .
  have qs-carr: set qs  $\subseteq \text{carrier } R$ 
    using qs-sub subset by blast
  have prod-carr: foldr ( $\otimes_R$ ) qs  $\mathbf{1}_R \in \text{carrier } R$ 

```

by (rule multlist-closed[OF qs-carr])
 show ?case
 by (simp add: integral-iff prod-carr q-carr q-nz qs-nz)
 qed

lemma zero-notin-submonoid-of-prime-generated:

assumes hS : ring-prime-generated R S

shows $0 \notin S$

proof

assume zero-in: $0 \in S$

obtain fs where

fs -sub: set $fs \subseteq S$

and hf : $\forall q \in set\ fs. ring\ prime_R\ q$

and $hprod$: $foldr\ (\otimes_R)\ fs\ \mathbf{1}_R = \mathbf{0}$

using ring-prime-generatedE[OF hS zero-in] by blast

have $foldr\ (\otimes_R)\ fs\ \mathbf{1}_R \neq \mathbf{0}$

using multlist-nonzero-of-prime-factors[OF fs -sub hf].

with $hprod$ show False

by contradiction

qed

lemma zero-notin-submonoid-of-prime-or-unit:

assumes hS : $\bigwedge s. s \in S \implies ring\ prime_R\ s \vee s \in Units\ R$

shows $0 \notin S$

proof

assume zero-in: $0 \in S$

from hS [OF zero-in] show False

proof

assume ring-prime R 0

then show False

by (simp add: ring-prime-def)

next

assume zero-unit: $0 \in Units\ R$

have inv-zero-in: $inv\ 0 \in carrier\ R$

using zero-unit by simp

have $(0 \otimes_R inv\ 0) = 0$

using inv-zero-in by simp

moreover have $(0 \otimes_R inv\ 0) = 1$

using zero-unit by simp

ultimately show False

using one-not-zero by simp

qed

qed

lemma ring-prime-imp-ring-irreducible:

assumes p -in: $p \in carrier\ R$

and hp : ring-prime R p

shows ring-irreducible R p

proof –

```

from ring-primeE[OF p-in hp] have p-nz:  $p \neq \mathbf{0}$ 
  and p-prime-mult: prime (mult-of R) p
  by auto
have irreducible (mult-of R) p
  using p-prime-mult by (rule mult-of.prime-irreducible)
have p-nz-in:  $p \in \text{carrier } R - \{\mathbf{0}\}$ 
  using p-in p-nz by blast
from p-nz-in and ⟨irreducible (mult-of R) p⟩ show ?thesis
by (rule ring-irreducibleI')
qed

lemma prime-of-irreducible-of-dvd-mem:
  assumes hS:  $\bigwedge s. s \in S \implies \text{ring-prime}_R s \vee s \in \text{Units } R$ 
  and p-in:  $p \in \text{carrier } R$ 
  and hp: ring-irreducibleR p
  and s-in:  $s \in S$ 
  and p-dvd-s: p dividesR s
  shows ring-primeR p
proof –
  have p-not-unit:  $p \notin \text{Units } R$ 
  using ring-irreducibleE(4)[OF p-in hp] .
from hS[OF s-in] show ?thesis
proof
  assume hs-prime: ring-primeR s
  have s-in-carrier:  $s \in \text{carrier } R$ 
  using s-in subset rev-subsetD by blast
  have s-irreducible: ring-irreducibleR s
  using s-in-carrier hs-prime by (rule ring-prime-imp-ring-irreducible)
  have p-assoc-s:  $p \sim_R s$ 
  by (meson divides-irreducible-condition p-dvd-s p-in p-not-unit ring-irreducible-def
    s-irreducible)
  have s-prime: prime R s
  using ring-primeE(3)[OF s-in-carrier hs-prime] .
  have s-nz:  $s \neq \mathbf{0}$ 
  using ring-primeE(1)[OF s-in-carrier hs-prime] .
  have p-nz:  $p \neq \mathbf{0}$ 
  using ring-irreducibleE(1)[OF p-in hp] .
  have s-assoc-p-mult:  $s \sim_{\text{mult-of } R} p$ 
  using assoc-iff-assoc-mult[OF s-in-carrier p-in] associated-sym[OF p-assoc-s]
by blast
  have s-prime-mult: prime (mult-of R) s
  using prime-eq-prime-mult[OF s-in-carrier] s-prime by blast
  have prime R p
  proof –
  have prime (mult-of R) p
  proof (rule mult-of.prime-cong[OF s-prime-mult s-assoc-p-mult])
    show  $s \in \text{carrier (mult-of R)}$ 
    using s-in-carrier s-nz by simp
    show  $p \in \text{carrier (mult-of R)}$ 

```

```

      using p-in p-nz by simp
    qed
  then show ?thesis
    using prime-eq-prime-mult[OF p-in] by blast
  qed
  then have p-prime: prime R p .
  from p-nz p-prime show ?thesis
    by (rule ring-primeI)
next
  assume hs-unit: s ∈ Units R
  have p ∈ Units R
    using divides-unit[OF p-dvd-s p-in hs-unit] .
  with p-not-unit show ?thesis
    by contradiction
  qed
qed

lemma prime-of-irreducible-of-dvd-prime-factors:
  assumes fs-sub: set fs ⊆ S
    and hf: ∀ q ∈ set fs. ring-prime_R q
    and p-in: p ∈ carrier R
    and hp: ring-irreducible_R p
    and hdiv: p divides_R foldr (⊗_R) fs 1_R
  shows ring-prime_R p
proof -
  have hmain:
    ∧ gs. set gs ⊆ S ⇒
      (∀ q ∈ set gs. ring-prime_R q) ⇒
      p divides_R foldr (⊗_R) gs 1_R ⇒
      ring-prime_R p
  proof -
    fix gs
    show set gs ⊆ S ⇒
      (∀ q ∈ set gs. ring-prime_R q) ⇒
      p divides_R foldr (⊗_R) gs 1_R ⇒
      ring-prime_R p
    proof (induction gs)
      case Nil
      have p-dvd-one: p divides_R 1_R
        using Nil.prem(3) by simp
      have p ∈ Units R
        by (rule divides-unit[OF p-dvd-one p-in Units-one-closed])
      with ring-irreducibleE(4)[OF p-in hp] show ?case
        by contradiction
    next
      case (Cons q gs)
      have q-in: q ∈ S
        using Cons.prem(1) by simp
      have q-carr: q ∈ carrier R

```

```

    using q-in subset rev-subsetD by blast
  have q-ring-prime: ring-primeR q
    using Cons.prem(2) by simp
  have q-prime: prime R q
    using ring-primeE(3)[OF q-carr q-ring-prime] .
  have q-nz: q ≠ 0
    using ring-primeE(1)[OF q-carr q-ring-prime] .
  have q-ring-irred: ring-irreducibleR q
    using ring-prime-imp-ring-irreducible[OF q-carr q-ring-prime] .
  have q-not-unit: q ∉ Units R
    using ring-irreducibleE(4)[OF q-carr q-ring-irred] .
  have qs-sub: set qs ⊆ S
    using Cons.prem(1) by simp
  have qs-prime: ∀ r ∈ set qs. ring-primeR r
    using Cons.prem(2) by simp
  have qs-carr: set qs ⊆ carrier R
    using qs-sub subset by blast
  have rest-carr: foldr (⊗R) qs 1R ∈ carrier R
    by (rule multlist-closed[OF qs-carr])
  obtain d where
    d-in: d ∈ carrier R
    and hd: foldr (⊗R) (q # qs) 1R = p ⊗R d
    using Cons.prem(3) unfolding factor-def by blast
  have q-dvd-pd: q dividesR (p ⊗R d)
  proof -
    have q ⊗R foldr (⊗R) qs 1R = p ⊗R d
      using hd by simp
    then show ?thesis
      using rest-carr by force
  qed
  have q-dvd-p-or-d: q dividesR p ∨ q dividesR d
    using primeE[OF q-prime] p-in d-in q-dvd-pd by blast
  from q-dvd-p-or-d show ?case
  proof
    assume q-dvd-p: q dividesR p
    have q-assoc-p: q ∼R p
      by (meson divides-irreducible-condition hp q-carr q-dvd-p q-not-unit
ring-irreducible-def)
    have q-assoc-p-mult: q ∼mult-of R p
      using assoc-iff-assoc-mult[OF q-carr p-in] q-assoc-p by blast
    have q-prime-mult: prime (mult-of R) q
      using prime-eq-prime-mult[OF q-carr] q-prime by blast
    have prime R p
  proof -
    have prime (mult-of R) p
  proof (rule mult-of.prime-cong[OF q-prime-mult q-assoc-p-mult])
    show q ∈ carrier (mult-of R)
      using q-carr q-nz by simp
    show p ∈ carrier (mult-of R)

```

```

      using p-in ring-irreducibleE(1)[OF p-in hp] by simp
    qed
  then show ?thesis
    using prime-eq-prime-mult[OF p-in] by blast
  qed
  then have p-prime: prime R p .
  from ring-irreducibleE(1)[OF p-in hp] p-prime show ?thesis
    by (rule ring-primeI)
next
  assume q-dvd-d: q dividesR d
  obtain e where e-in: e ∈ carrier R and he: d = q ⊗R e
    using q-dvd-d unfolding factor-def by blast
  have pe-carr: p ⊗R e ∈ carrier R
    using p-in e-in by auto
  have rest-eq: foldr (⊗R) qs 1R = p ⊗R e
  proof -
    have q ⊗R foldr (⊗R) qs 1R = q ⊗R (p ⊗R e)
      using hd he q-carr rest-carr p-in e-in by (simp add: m-assoc m-comm
m-lcomm)
    then show ?thesis
      using q-nz q-carr rest-carr pe-carr by (simp add: m-lcancel)
  qed
  have p-dvd-rest: p dividesR foldr (⊗R) qs 1R
    unfolding factor-def using e-in rest-eq by blast
  show ?thesis
    by (rule Cons.IH[OF qs-sub qs-prime p-dvd-rest])
  qed
  qed
  qed
  show ?thesis
    by (rule hmain[OF fs-sub hf hdiv])
  qed

lemma prime-of-irreducible-of-dvd-mem-prime-generated:
  assumes hS: ring-prime-generated R S
    and p-in: p ∈ carrier R
    and hp: ring-irreducibleR p
    and s-in: s ∈ S
    and p-dvd-s: p dividesR s
  shows ring-primeR p
  proof -
  obtain fs where
    fs-sub: set fs ⊆ S
    and hf: ∀ q ∈ set fs. ring-primeR q
    and hprod: foldr (⊗R) fs 1R = s
    using ring-prime-generatedE[OF hS s-in] by blast
  have p dividesR foldr (⊗R) fs 1R
    using p-dvd-s by (simp add: hprod)
  then show ?thesis

```

by (rule prime-of-irreducible-of-dvd-prime-factors[OF fs-sub hf p-in hp])
qed

lemma *dvd-of-localization-dvd*:

assumes $hS: \bigwedge s. s \in S \implies \text{ring-prime}_R s \vee s \in \text{Units } R$
and $p\text{-in}: p \in \text{carrier } R$
and $a\text{-in}: a \in \text{carrier } R$
and $hp: \text{ring-irreducible}_R p$
and $havoid: \text{ring-avoids } R S p$
and $hdiv: \text{rng-to-rng-of-frac } p \text{ divides}_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } a$
shows $p \text{ divides}_R a$

proof –

have $zero\text{-notin}: \mathbf{0} \notin S$
using hS by (rule zero-notin-submonoid-of-prime-or-unit)
have $hdiv'$:
 $\exists s \in S. p \text{ divides}_R (s \otimes_R a)$
using $dvd\text{-map-iff}[OF p\text{-in } a\text{-in } zero\text{-notin } no\text{-zero-divisors}] hdiv$
by blast
then obtain s **where** $s\text{-in}: s \in S$ **and** $p\text{-dvd-sa}: p \text{ divides}_R (s \otimes_R a)$
by blast
have $s\text{-in-carrier}: s \in \text{carrier } R$
using $s\text{-in subset rev-subsetD}$ by blast
then obtain c **where** $c\text{-in}: c \in \text{carrier } R$ **and** $hc: s \otimes_R a = p \otimes_R c$
using $p\text{-dvd-sa}$ **unfolding** $factor\text{-def}$ by blast
from $hS[OF s\text{-in}]$ **show** $?thesis$

proof

assume $hs\text{-prime}: \text{ring-prime}_R s$
have $s\text{-nz}: s \neq \mathbf{0}$ **and** $s\text{-prime}: \text{prime } R s$
using $ring\text{-primeE}[OF s\text{-in-carrier } hs\text{-prime}]$ by auto
have $s\text{-not-unit}: s \notin \text{Units } R$
using $primeE[OF s\text{-prime}]$ by blast
have $s\text{-dvd-pc}: s \text{ divides}_R (p \otimes_R c)$
using $a\text{-in } hc$ by force
have $s\text{-dvd-p-or-c}: s \text{ divides}_R p \vee s \text{ divides}_R c$
using $primeE[OF s\text{-prime}] p\text{-in } c\text{-in } s\text{-dvd-pc}$ by blast
from $s\text{-dvd-p-or-c}$ **show** $?thesis$

proof

assume $s\text{-dvd-p}: s \text{ divides}_R p$
have $s\text{-assoc-p}: s \sim_R p$
by (meson $divides\text{-irreducible-condition } hp \text{ ring-irreducible-def } s\text{-dvd-p}$

$s\text{-in-carrier}$

$s\text{-not-unit}$)

have $p \text{ divides}_R s$
using $associatedD[OF associated\text{-sym}[OF s\text{-assoc-p}]]$.
moreover have $\neg p \text{ divides}_R s$
using $havoid s\text{-in}$ **unfolding** $ring\text{-avoids-def}$ by blast
ultimately **show** $?thesis$
by contradiction

next

```

    assume s-dvd-c: s dividesR c
    then obtain d where d-in: d ∈ carrier R and hd: c = s ⊗R d
      unfolding factor-def by blast
    have pd-in: p ⊗R d ∈ carrier R
      using p-in d-in by auto
    have s ⊗R a = s ⊗R (p ⊗R d)
      using hc hd s-in-carrier a-in p-in d-in by (simp add: m-assoc m-comm
m-lcomm)
    then have a = p ⊗R d
      using s-nz s-in-carrier a-in pd-in by (simp add: m-lcancel)
    then show ?thesis
      unfolding factor-def using d-in by blast
  qed
next
  assume hs-unit: s ∈ Units R
  have sa-assoc-a: (s ⊗R a) ~R a
    using hs-unit a-in s-in-carrier by (intro associatedI2') (simp add: m-comm)
  have p dividesR (s ⊗R a)
    using p-dvd-sa .
  then show ?thesis
    using divides-cong-r[OF - sa-assoc-a p-in] by blast
  qed
qed

lemma prime-of-localization-prime:
  assumes hS: ⋀s. s ∈ S ⇒ ring-primeR s ∨ s ∈ Units R
    and p-in: p ∈ carrier R
    and hp: ring-irreducibleR p
    and havoid: ring-avoids R S p
    and hploc: ring-primerec-rng-of-frac (rng-to-rng-of-frac p)
  shows ring-primeR p
proof (rule ring-primeI)
  show p ≠ 0
    using ring-irreducibleE(1)[OF p-in hp] .
next
  show prime R p
proof (rule primeI)
  show p ∉ Units R
    using ring-irreducibleE(4)[OF p-in hp] .
next
  fix a b
  assume a-in: a ∈ carrier R
    and b-in: b ∈ carrier R
    and p-dvd-ab: p dividesR (a ⊗R b)
  obtain c where c-in: c ∈ carrier R and hc: a ⊗R b = p ⊗R c
    using p-dvd-ab unfolding factor-def by blast
  have a-frac-in: rng-to-rng-of-frac a ∈ carrier rec-rng-of-frac
    by (rule ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom a-in])
  have b-frac-in: rng-to-rng-of-frac b ∈ carrier rec-rng-of-frac

```

by (rule ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom b-in])
 have c-frac-in: rng-to-rng-of-frac c ∈ carrier rec-rng-of-frac
 by (rule ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom c-in])
 have hloc-div:
 rng-to-rng-of-frac p divides_{rec-rng-of-frac}
 (rng-to-rng-of-frac a ⊗_{rec-rng-of-frac} rng-to-rng-of-frac b)
proof –
 have map-ab:
 rng-to-rng-of-frac (a ⊗_R b) =
 rng-to-rng-of-frac a ⊗_{rec-rng-of-frac} rng-to-rng-of-frac b
 using ring-hom-mult[OF rng-to-rng-of-frac-is-ring-hom a-in b-in] .
 have map-pc:
 rng-to-rng-of-frac (p ⊗_R c) =
 rng-to-rng-of-frac p ⊗_{rec-rng-of-frac} rng-to-rng-of-frac c
 using ring-hom-mult[OF rng-to-rng-of-frac-is-ring-hom p-in c-in] .
 have rng-to-rng-of-frac p divides_{rec-rng-of-frac} rng-to-rng-of-frac (a ⊗_R b)
 unfolding factor-def using c-frac-in hc map-pc by auto
 then show ?thesis
 using map-ab by simp
qed
 have hloc-prime: prime rec-rng-of-frac (rng-to-rng-of-frac p)
 using hploc unfolding ring-prime-def by simp
 have hloc-cases:
 rng-to-rng-of-frac p divides_{rec-rng-of-frac} rng-to-rng-of-frac a ∨
 rng-to-rng-of-frac p divides_{rec-rng-of-frac} rng-to-rng-of-frac b
 by (metis a-frac-in b-frac-in hloc-div hloc-prime primeE)
 then show p divides_R a ∨ p divides_R b
 using dvd-of-localization-dvd[OF hS p-in a-in hp havoid]
 dvd-of-localization-dvd[OF hS p-in b-in hp havoid]
 by blast
qed
qed

lemma dvd-of-mul-eq-prime-factors:
assumes fs-sub: set fs ⊆ S
and hf: ∀ q ∈ set fs. ring-prime_R q
and p-in: p ∈ carrier R
and a-in: a ∈ carrier R
and hp: ring-irreducible_R p
and hnot: ∀ q ∈ set fs. ¬ p divides_R q
and c-in: c ∈ carrier R
and hEq: foldr (⊗_R) fs 1_R ⊗_R a = p ⊗_R c
shows p divides_R a
proof –
have hmain:
 ∧ gs d. set gs ⊆ S ⇒
 (∀ q ∈ set gs. ring-prime_R q) ⇒
 (∀ q ∈ set gs. ¬ p divides_R q) ⇒
 d ∈ carrier R ⇒

```

    foldr ( $\otimes_R$ ) gs  $\mathbf{1}_R \otimes_R a = p \otimes_R d \implies$ 
    p dividesR a
proof –
  fix gs d
  show set gs  $\subseteq S \implies$ 
    ( $\forall q \in \text{set } gs. \text{ring-prime}_R q \implies$ 
    ( $\forall q \in \text{set } gs. \neg p \text{ divides}_R q \implies$ 
     $d \in \text{carrier } R \implies$ 
    foldr ( $\otimes_R$ ) gs  $\mathbf{1}_R \otimes_R a = p \otimes_R d \implies$ 
    p dividesR a
  proof (induction gs arbitrary: d)
  case Nil
  from Nil.prems have d-in:  $d \in \text{carrier } R$  and eq:  $a = p \otimes_R d$ 
  using a-in by simp-all
  show ?case
  using d-in eq unfolding factor-def by blast
next
  case (Cons q qs)
  have q-in:  $q \in S$ 
  using Cons.prems(1) by simp
  have q-carr:  $q \in \text{carrier } R$ 
  using q-in subset rev-subsetD by blast
  have q-ring-prime: ring-primeR q
  using Cons.prems(2) by simp
  have q-prime: prime R q
  using ring-primeE(3)[OF q-carr q-ring-prime] .
  have q-nz:  $q \neq \mathbf{0}$ 
  using ring-primeE(1)[OF q-carr q-ring-prime] .
  have q-ring-irred: ring-irreducibleR q
  using ring-prime-imp-ring-irreducible[OF q-carr q-ring-prime] .
  have q-not-unit:  $q \notin \text{Units } R$ 
  using ring-irreducibleE(4)[OF q-carr q-ring-irred] .
  have qs-sub: set qs  $\subseteq S$ 
  using Cons.prems(1) by simp
  have qs-prime:  $\forall r \in \text{set } qs. \text{ring-prime}_R r$ 
  using Cons.prems(2) by simp
  have qs-not:  $\forall r \in \text{set } qs. \neg p \text{ divides}_R r$ 
  using Cons.prems(3) by simp
  have qs-carr: set qs  $\subseteq \text{carrier } R$ 
  using qs-sub subset by blast
  have rest-carr: foldr ( $\otimes_R$ ) qs  $\mathbf{1}_R \in \text{carrier } R$ 
  by (rule multlist-closed[OF qs-carr])
  have resta-carr: foldr ( $\otimes_R$ ) qs  $\mathbf{1}_R \otimes_R a \in \text{carrier } R$ 
  using rest-carr a-in by auto
  have q-dvd-pd:  $q \text{ divides}_R (p \otimes_R d)$ 
proof –
  have hqd:  $q \otimes_R (\text{foldr } (\otimes_R) qs \mathbf{1}_R \otimes_R a) = p \otimes_R d$ 
  using Cons.prems(5) a-in m-comm m-lcomm q-carr rest-carr by force
  then show ?thesis

```

```

    by (metis dividesI resta-carr)
qed
have q-dvd-p-or-d: q dividesR p ∨ q dividesR d
  using primeE[OF q-prime] p-in Cons.prem(4) q-dvd-pd by blast
from q-dvd-p-or-d show ?case
proof
  assume q-dvd-p: q dividesR p
  have q-assoc-p: q ~R p
  by (metis divides-irreducible-condition hp q-carr q-dvd-p q-not-unit ring-irreducible-def)
  have p dividesR q
    using associatedD[OF associated-sym[OF q-assoc-p]] .
  with Cons.prem(3) show ?thesis
  by simp
next
  assume q-dvd-d: q dividesR d
  obtain e where e-in: e ∈ carrier R and he: d = q ⊗R e
    using q-dvd-d unfolding factor-def by blast
  have pe-carr: p ⊗R e ∈ carrier R
    using p-in e-in by auto
  have rest-eq: foldr (⊗R) qs 1R ⊗R a = p ⊗R e
  proof -
    have q ⊗R (foldr (⊗R) qs 1R ⊗R a) =
      q ⊗R (p ⊗R e)
    using Cons.prem(5) he q-carr rest-carr a-in p-in e-in
    by (simp add: m-assoc m-comm m-lcomm)
  then show ?thesis
    using q-nz q-carr resta-carr pe-carr by (simp add: m-lcancel)
qed
show ?thesis
  by (rule Cons.IH[OF qs-sub qs-prime qs-not e-in rest-eq])
qed
qed
qed
show ?thesis
  by (rule hmain[OF fs-sub hf hnot c-in hEq])
qed

```

```

lemma dvd-of-localization-dvd-prime-generated:
  assumes hS: ring-prime-generated R S
  and p-in: p ∈ carrier R
  and a-in: a ∈ carrier R
  and hp: ring-irreducibleR p
  and havoid: ring-avoids R S p
  and hdiv: rng-to-rng-of-frac p dividesrec-rng-of-frac rng-to-rng-of-frac a
  shows p dividesR a
proof -
  have zero-notin: 0 ∉ S
    using zero-notin-submonoid-of-prime-generated[OF hS] .
  have hdiv':

```

$\exists s \in S. p \text{ divides}_R (s \otimes_R a)$
using *dvd-map-iff*[*OF p-in a-in zero-notin no-zero-divisors*] *hdiv*
by *blast*
then obtain *s* **where** *s-in*: $s \in S$ **and** *p-dvd-sa*: $p \text{ divides}_R (s \otimes_R a)$
by *blast*
obtain *c* **where** *c-in*: $c \in \text{carrier } R$ **and** *hc*: $s \otimes_R a = p \otimes_R c$
using *p-dvd-sa unfolding factor-def* **by** *blast*
obtain *fs* **where**
fs-sub: $\text{set } fs \subseteq S$
and *hf*: $\forall q \in \text{set } fs. \text{ring-prime}_R q$
and *hprod*: $\text{foldr } (\otimes_R) fs \mathbf{1}_R = s$
using *ring-prime-generatedE*[*OF hS s-in*] **by** *blast*
have *hnot*: $\forall q \in \text{set } fs. \neg p \text{ divides}_R q$
using *havoid fs-sub unfolding ring-avoids-def* **by** *blast*
have *hEq*: $\text{foldr } (\otimes_R) fs \mathbf{1}_R \otimes_R a = p \otimes_R c$
using *hc* **by** (*simp add: hprod*)
show *?thesis*
using *dvd-of-mul-eq-prime-factors*[*OF fs-sub hf p-in a-in hp hnot c-in hEq*].
qed

lemma *map-irreducible-not-unit-of-zero-notin*:

assumes *zero-notin*: $\mathbf{0} \notin S$
and *loc-dom*: *domain rec-rng-of-frac*
and *p-in*: $p \in \text{carrier } R$
and *hp*: *ring-irreducible*_{*R*} *p*
and *havoid*: *ring-avoids* *R S p*
shows *rng-to-rng-of-frac p* $\notin \text{Units } \text{rec-rng-of-frac}$
proof
interpret *L*: *domain rec-rng-of-frac*
by (*rule loc-dom*)
assume *map-p-unit*: *rng-to-rng-of-frac p* $\in \text{Units } \text{rec-rng-of-frac}$
have *map-p-dvd-one*:
 $\text{rng-to-rng-of-frac } p \text{ divides}_{\text{rec-rng-of-frac}} \mathbf{1}_{\text{rec-rng-of-frac}}$
using *L.unit-divides*[*OF map-p-unit L.one-closed*].
have *map-p-dvd-map-one*:
 $\text{rng-to-rng-of-frac } p \text{ divides}_{\text{rec-rng-of-frac}} \text{rng-to-rng-of-frac } \mathbf{1}$
using *map-p-dvd-one ring-hom-one*[*OF rng-to-rng-of-frac-is-ring-hom*]
by *simp*
have *one-in*: $\mathbf{1} \in \text{carrier } R$
by *simp*
have *hdiv*:
 $\exists s \in S. p \text{ divides}_R (s \otimes_R \mathbf{1})$
using *dvd-map-iff*[*OF p-in one-in zero-notin no-zero-divisors*] *map-p-dvd-map-one*
by *blast*
then obtain *s* **where** *s-in*: $s \in S$ **and** *p-dvd-s*: $p \text{ divides}_R (s \otimes_R \mathbf{1})$
by *blast*
have *s-carr*: $s \in \text{carrier } R$
using *s-in subset rev-subsetD* **by** *blast*
have *p divides*_{*R*} *s*

using $p\text{-dvd-}s$ $s\text{-carr}$ **by** simp
moreover have $\neg p \text{ divides}_R s$
using $\text{havoid } s\text{-in}$ **unfolding** ring-avoids-def **by** blast
ultimately show False
by contradiction
qed

lemma $\text{map-irreducible-not-unit}$:

assumes hS : $\bigwedge s. s \in S \implies \text{ring-prime}_R s \vee s \in \text{Units } R$
and loc-dom : $\text{domain rec-rng-of-frac}$
and p-in : $p \in \text{carrier } R$
and hp : $\text{ring-irreducible}_R p$
and havoid : $\text{ring-avoids } R S p$
shows $\text{rng-to-rng-of-frac } p \notin \text{Units rec-rng-of-frac}$
proof –
have zero-notin : $\mathbf{0} \notin S$
using hS **by** $(\text{rule zero-notin-submonoid-of-prime-or-unit})$
show ?thesis
using $\text{map-irreducible-not-unit-of-zero-notin}$ $[OF \text{ zero-notin loc-dom p-in hp havoid}]$.
qed

lemma $\text{localization-irreducible-of-irreducible}$:

assumes hS : $\bigwedge s. s \in S \implies \text{ring-prime}_R s \vee s \in \text{Units } R$
and loc-dom : $\text{domain rec-rng-of-frac}$
and p-in : $p \in \text{carrier } R$
and hp : $\text{ring-irreducible}_R p$
and havoid : $\text{ring-avoids } R S p$
shows $\text{ring-irreducible}_{\text{rec-rng-of-frac}} (\text{rng-to-rng-of-frac } p)$
proof –
interpret L : $\text{domain rec-rng-of-frac}$
by (rule loc-dom)
have zero-notin : $\mathbf{0} \notin S$
using hS **by** $(\text{rule zero-notin-submonoid-of-prime-or-unit})$
have map-p-in : $\text{rng-to-rng-of-frac } p \in \text{carrier rec-rng-of-frac}$
by $(\text{rule ring-hom-closed}[OF \text{ rng-to-rng-of-frac-is-ring-hom p-in}])$
have map-p-nz : $\text{rng-to-rng-of-frac } p \neq \mathbf{0}_{\text{rec-rng-of-frac}}$
using map-eq-zero-iff $[OF \text{ p-in zero-notin no-zero-divisors}]$
 $\text{ring-irreducibleE}(1)$ $[OF \text{ p-in hp}]$
by blast
have map-p-not-unit : $\text{rng-to-rng-of-frac } p \notin \text{Units rec-rng-of-frac}$
using $\text{map-irreducible-not-unit}$ $[OF \text{ hS loc-dom p-in hp havoid}]$.
show ?thesis
proof $(\text{rule } L.\text{ring-irreducibleI})$
show $\text{rng-to-rng-of-frac } p \in \text{carrier rec-rng-of-frac} - \{\mathbf{0}_{\text{rec-rng-of-frac}}\}$
using map-p-in map-p-nz **by** blast
next
show $\text{rng-to-rng-of-frac } p \notin \text{Units rec-rng-of-frac}$
using map-p-not-unit .

```

next
  fix x y
  assume x-in:  $x \in \text{carrier rec-rng-of-frac}$ 
  and y-in:  $y \in \text{carrier rec-rng-of-frac}$ 
  and xy:  $\text{rng-to-rng-of-frac } p = x \otimes_{\text{rec-rng-of-frac}} y$ 
  from fraction-surj[OF x-in] obtain a where a-in:  $a \in \text{carrier } R$ 
  and hs:  $\exists s \in S. x = (a \mid_{\text{rel}} s)$ 
  by blast
  from hs obtain s where s-in:  $s \in S$  and x-def:  $x = (a \mid_{\text{rel}} s)$ 
  by blast
  from fraction-surj[OF y-in] obtain b where b-in:  $b \in \text{carrier } R$ 
  and ht:  $\exists t \in S. y = (b \mid_{\text{rel}} t)$ 
  by blast
  from ht obtain t where t-in:  $t \in S$  and y-def:  $y = (b \mid_{\text{rel}} t)$ 
  by blast
  have as-rel:  $(a, s) \in \text{carrier rel}$ 
  using a-in s-in by (simp add: rel-def)
  have bt-rel:  $(b, t) \in \text{carrier rel}$ 
  using b-in t-in by (simp add: rel-def)
  have st-in:  $s \otimes_R t \in S$ 
  using s-in t-in by simp
  have st-carrier:  $s \otimes_R t \in \text{carrier } R$ 
  using st-in subset rev-subsetD by blast
  have p-rel:  $(p, \mathbf{1}) \in \text{carrier rel}$ 
  using p-in one-closed by (simp add: rel-def)
  have ab-in:  $a \otimes_R b \in \text{carrier } R$ 
  using a-in b-in by auto
  have ab-rel:  $(a \otimes_R b, s \otimes_R t) \in \text{carrier rel}$ 
  using a-in b-in st-in by (simp add: rel-def)
  have frac-prod:
     $x \otimes_{\text{rec-rng-of-frac}} y = (a \otimes_R b \mid_{\text{rel}} s \otimes_R t)$ 
  using fraction-mult-rep[OF as-rel bt-rel] x-def y-def by simp
  have eq-frac:
     $\text{rng-to-rng-of-frac } p = (a \otimes_R b \mid_{\text{rel}} s \otimes_R t)$ 
  using xy frac-prod by (simp add: rng-to-rng-of-frac-def)
  have p-eq-raw:  $(s \otimes_R t) \otimes_R p = \mathbf{1}_R \otimes_R (a \otimes_R b)$ 
  proof -
    have eq-cross:
       $\text{rng-to-rng-of-frac } p = (a \otimes_R b \mid_{\text{rel}} s \otimes_R t) \longleftrightarrow$ 
       $(s \otimes_R t) \otimes_R p = \mathbf{1}_R \otimes_R (a \otimes_R b)$ 
    using fraction-eq-iff-cross-multiply[OF p-rel ab-rel zero-notin no-zero-divisors]
    unfolding rng-to-rng-of-frac-def by simp
    from eq-cross eq-frac show ?thesis
    by blast
  qed
  have p-eq:  $(s \otimes_R t) \otimes_R p = a \otimes_R b$ 
  using p-eq-raw ab-in by simp
  from hS[OF st-in] show  $x \in \text{Units rec-rng-of-frac} \vee y \in \text{Units rec-rng-of-frac}$ 
  proof

```

```

assume hst-prime: ring-primeR (s ⊗R t)
have st-prime: prime R (s ⊗R t)
  and st-nz: s ⊗R t ≠ 0
  using ring-primeE[OF st-carrier hst-prime] by auto
have st-dvd-ab: (s ⊗R t) dividesR (a ⊗R b)
proof –
  have ab-eq: a ⊗R b = (s ⊗R t) ⊗R p
    using p-eq by simp
  show ?thesis
    by (rule dividesI[OF p-in ab-eq])
qed
have st-dvd-a-or-b:
  (s ⊗R t) dividesR a ∨ (s ⊗R t) dividesR b
  using primeE[OF st-prime] a-in b-in st-dvd-ab by blast
from st-dvd-a-or-b show ?thesis
proof
  assume st-dvd-a: (s ⊗R t) dividesR a
  then obtain d where d-in: d ∈ carrier R and hd: a = (s ⊗R t) ⊗R d
    unfolding factor-def by blast
  have db-in: d ⊗R b ∈ carrier R
    using d-in b-in by auto
  have p = d ⊗R b
  proof –
    have (s ⊗R t) ⊗R p = a ⊗R b
      by (rule p-eq)
    also have ... = ((s ⊗R t) ⊗R d) ⊗R b
      using hd by simp
    also have ... = (s ⊗R t) ⊗R (d ⊗R b)
      using st-carrier d-in b-in by (simp add: m-assoc)
    finally have (s ⊗R t) ⊗R p =
      (s ⊗R t) ⊗R (d ⊗R b) .
    then show ?thesis
      using st-nz st-carrier p-in db-in by (simp add: m-lcancel)
  qed
then have d-or-b-unit: d ∈ Units R ∨ b ∈ Units R
  using ring-irreducibleE(5)[OF p-in hp d-in b-in] by blast
then show ?thesis
proof
  assume d-unit: d ∈ Units R
  have t-in-carrier: t ∈ carrier R
    using t-in subset rev-subsetD by blast
  have td-in: t ⊗R d ∈ carrier R
    using t-in-carrier d-in by auto
  have td-rel: (t ⊗R d, 1) ∈ carrier rel
    using td-in one-closed by (simp add: rel-def)
  have x-eq-map-td: x = rng-to-rng-of-frac (t ⊗R d)
  proof –
    have s-in-carrier: s ∈ carrier R
      using s-in subset rev-subsetD by blast

```

```

have x = (a |rel s)
  using x-def by simp
also have ... = ((s ⊗R t) ⊗R d |rel s)
  using hd by (simp add: m-assoc)
also have ... = (s ⊗R (t ⊗R d) |rel s)
  using s-in-carrier t-in-carrier d-in by (simp add: m-assoc)
also have ... = (s ⊗R (t ⊗R d) |rel s ⊗R 1)
  using s-in-carrier by simp
also have ... = (t ⊗R d |rel 1)
  using fraction-rescale[OF td-rel s-in] by simp
also have ... = rng-to-rng-of-frac (t ⊗R d)
  by (simp add: rng-to-rng-of-frac-def)
finally show ?thesis .
qed
have t-unit-loc: rng-to-rng-of-frac t ∈ Units rec-rng-of-frac
  using map-submonoid-elem-is-unit[OF t-in] .
have d-unit-loc: rng-to-rng-of-frac d ∈ Units rec-rng-of-frac
  using map-unit-is-unit[OF d-unit] .
have rng-to-rng-of-frac (t ⊗R d) =
  rng-to-rng-of-frac t ⊗rec-rng-of-frac rng-to-rng-of-frac d
  using ring-hom-mult[OF rng-to-rng-of-frac-is-ring-hom t-in-carrier d-in]

then show ?thesis
  using x-eq-map-td t-unit-loc d-unit-loc by simp
next
assume b-unit: b ∈ Units R
have (b |rel t) ∈ Units rec-rng-of-frac
  by (rule fraction-unit-numerator-is-unit[OF b-unit t-in])
then show ?thesis
  using y-def by blast
qed
next
assume st-dvd-b: (s ⊗R t) dividesR b
then obtain d where d-in: d ∈ carrier R and hd: b = (s ⊗R t) ⊗R d
  unfolding factor-def by blast
have ad-in: a ⊗R d ∈ carrier R
  using a-in d-in by auto
have p = a ⊗R d
proof -
  have (s ⊗R t) ⊗R p = a ⊗R b
    by (rule p-eq)
  also have ... = a ⊗R ((s ⊗R t) ⊗R d)
    using hd by simp
  also have ... = (s ⊗R t) ⊗R (a ⊗R d)
    using st-carrier a-in d-in by (simp add: m-assoc m-comm m-lcomm)
  finally have (s ⊗R t) ⊗R p =
    (s ⊗R t) ⊗R (a ⊗R d) .
then show ?thesis
  using st-nz st-carrier p-in ad-in by (simp add: m-lcancel)

```

```

qed
then have a-or-d-unit: a ∈ Units R ∨ d ∈ Units R
  using ring-irreducibleE(5)[OF p-in hp a-in d-in] by blast
then show ?thesis
proof
  assume a-unit: a ∈ Units R
  have (a |rel s) ∈ Units rec-rng-of-frac
    by (rule fraction-unit-numerator-is-unit[OF a-unit s-in])
  then show ?thesis
    using x-def by blast
next
  assume d-unit: d ∈ Units R
  have s-in-carrier: s ∈ carrier R
    using s-in subset rev-subsetD by blast
  have sd-in: s ⊗R d ∈ carrier R
    using s-in-carrier d-in by auto
  have sd-rel: (s ⊗R d, 1) ∈ carrier rel
    using sd-in one-closed by (simp add: rel-def)
  have y-eq-map-sd: y = rng-to-rng-of-frac (s ⊗R d)
  proof -
    have t-in-carrier: t ∈ carrier R
      using t-in subset rev-subsetD by blast
    have y = (b |rel t)
      using y-def by simp
    also have ... = ((s ⊗R t) ⊗R d |rel t)
      using hd by (simp add: m-assoc)
    also have ... = (t ⊗R (s ⊗R d) |rel t)
      using t-in-carrier s-in-carrier d-in by (simp add: m-assoc m-comm
m-lcomm)
    also have ... = (t ⊗R (s ⊗R d) |rel t ⊗R 1)
      using t-in-carrier by simp
    also have ... = (s ⊗R d |rel 1)
      using fraction-rescale[OF sd-rel t-in] by simp
    also have ... = rng-to-rng-of-frac (s ⊗R d)
      by (simp add: rng-to-rng-of-frac-def)
    finally show ?thesis .
  qed
  have s-unit-loc: rng-to-rng-of-frac s ∈ Units rec-rng-of-frac
    using map-submonoid-elem-is-unit[OF s-in] .
  have d-unit-loc: rng-to-rng-of-frac d ∈ Units rec-rng-of-frac
    using map-unit-is-unit[OF d-unit] .
  have rng-to-rng-of-frac (s ⊗R d) =
    rng-to-rng-of-frac s ⊗rec-rng-of-frac rng-to-rng-of-frac d
    using ring-hom-mult[OF rng-to-rng-of-frac-is-ring-hom s-in-carrier d-in]
  .
  then show ?thesis
    using y-eq-map-sd s-unit-loc d-unit-loc by simp
qed
qed

```

```

next
  assume hst-unit:  $s \otimes_R t \in \text{Units } R$ 
  have ab-assoc-p:  $(a \otimes_R b) \sim_R p$ 
  proof -
    have  $a \otimes_R b = (s \otimes_R t) \otimes_R p$ 
      using p-eq by simp
    also have  $\dots = p \otimes_R (s \otimes_R t)$ 
      using p-in Units-closed[OF hst-unit] by (simp add: m-assoc m-comm
m-lcomm)
    finally have  $a \otimes_R b = p \otimes_R (s \otimes_R t)$  .
    then show ?thesis
      using hst-unit p-in by (rule associatedI2')
  qed
  have p-mult-irreducible: irreducible (mult-of R) p
    using ring-irreducibleE(3)[OF p-in hp] .
  have p-assoc-ab-mult:  $p \sim_{\text{mult-of } R} (a \otimes_R b)$ 
    using assoc-iff-assoc-mult[OF p-in ab-in] associated-sym[OF ab-assoc-p] by
blast
  have st-carr:  $s \otimes_R t \in \text{carrier } R$ 
    using Units-closed[OF hst-unit] .
  have st-nz:  $s \otimes_R t \neq \mathbf{0}$ 
    using st-in zero-notin by auto
  have p-nz:  $p \neq \mathbf{0}$ 
    using ring-irreducibleE(1)[OF p-in hp] .
  have ab-nz:  $a \otimes_R b \neq \mathbf{0}$ 
    using ab-assoc-p divides-cong-r p-nz zero-divides by blast
  have a-nz:  $a \neq \mathbf{0}$ 
    using ab-nz b-in by fastforce
  have b-nz:  $b \neq \mathbf{0}$ 
    using a-in ab-nz by force
  have a-in-mult:  $a \in \text{carrier } (\text{mult-of } R)$ 
    using a-in a-nz by simp
  have b-in-mult:  $b \in \text{carrier } (\text{mult-of } R)$ 
    using b-in b-nz by simp
  have ab-mult-irreducible: irreducible (mult-of R)  $(a \otimes_R b)$ 
  proof (rule mult-of.irreducible-cong[OF p-mult-irreducible p-assoc-ab-mult])
    show  $p \in \text{carrier } (\text{mult-of } R)$ 
      using p-in p-nz by simp
    show  $(a \otimes_R b) \in \text{carrier } (\text{mult-of } R)$ 
      using ab-in ab-nz by simp
  qed
  have xy-unit:  $x \in \text{Units rec-rng-of-frac} \vee y \in \text{Units rec-rng-of-frac}$ 
  proof (rule mult-of.irreducible-prodE[OF ab-mult-irreducible a-in-mult b-in-mult])
    assume a-irreducible: irreducible (mult-of R) a and b-unit-mult:  $b \in \text{Units}$ 
(mult-of R)
    have b-unit:  $b \in \text{Units } R$ 
      using b-unit-mult by simp
    have y-unit:  $y \in \text{Units rec-rng-of-frac}$ 
      unfolding y-def using fraction-unit-numerator-is-unit[OF b-unit t-in] by

```

```

blast
  show  $x \in \text{Units } \text{rec-rng-of-frac} \vee y \in \text{Units } \text{rec-rng-of-frac}$ 
  using y-unit by auto
next
  assume a-unit-mult:  $a \in \text{Units } (\text{mult-of } R)$  and b-irreducible: irreducible
  (mult-of R) b
  have a-unit:  $a \in \text{Units } R$ 
  using a-unit-mult by simp
  have x-unit:  $x \in \text{Units } \text{rec-rng-of-frac}$ 
  unfolding x-def using fraction-unit-numerator-is-unit[OF a-unit s-in] by
blast
  show  $x \in \text{Units } \text{rec-rng-of-frac} \vee y \in \text{Units } \text{rec-rng-of-frac}$ 
  by (simp add: x-unit)
qed
then show ?thesis .
qed
qed
qed

```

lemma *nagata-key-lemma*:

```

assumes hS:  $\bigwedge s. s \in S \implies \text{ring-prime}_R s \vee s \in \text{Units } R$ 
  and loc-fd: factorial-domain rec-rng-of-frac
  and p-in:  $p \in \text{carrier } R$ 
  and hp: ring-irreducibleR p
shows ring-primeR p
proof (cases  $\exists s \in S. p \text{ divides}_R s$ )
  case True
  then obtain s where s-in:  $s \in S$  and p-dvd-s:  $p \text{ divides}_R s$ 
  by blast
  show ?thesis
  using prime-of-irreducible-of-dvd-mem[OF hS p-in hp s-in p-dvd-s] .
next
  case False
  interpret Lfd: factorial-domain rec-rng-of-frac
  by (rule loc-fd)
  interpret L: domain rec-rng-of-frac
  by (rule Lfd.domain-axioms)
  have havoid: ring-avoids R S p
  using False unfolding ring-avoids-def by blast
  have loc-irreducible: ring-irreduciblerec-rng-of-frac (rng-to-rng-of-frac p)
  using localization-irreducible-of-irreducible[OF hS Lfd.domain-axioms p-in hp
havoid] .
  have loc-irred-mult: irreducible (mult-of rec-rng-of-frac) (rng-to-rng-of-frac p)
  using Lfd.ring-irreducibleE(3)[OF ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom
p-in] loc-irreducible] .
  have loc-in-nz:  $\text{rng-to-rng-of-frac } p \in \text{carrier } \text{rec-rng-of-frac} - \{\mathbf{0}_{\text{rec-rng-of-frac}}\}$ 
  using ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom p-in]
  Lfd.ring-irreducibleE(1)[OF ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom
p-in] loc-irreducible]

```

by *blast*
have *loc-in-mult*: $\text{rng-to-rng-of-frac } p \in \text{carrier } (\text{mult-of-rec-rng-of-frac})$
using *loc-in-nz* **by** *simp*
have *loc-prime-mult*: $\text{prime } (\text{mult-of-rec-rng-of-frac}) (\text{rng-to-rng-of-frac } p)$
by (*rule factorial-monoid.irreducible-prime*[*OF Lfd.factorial-monoid-axioms*
loc-irred-mult loc-in-mult])
have *loc-ring-prime*: $\text{ring-prime}_{\text{rec-rng-of-frac}} (\text{rng-to-rng-of-frac } p)$
using *Lfd.ring-primeI'*[*OF loc-in-nz loc-prime-mult*] .
show *?thesis*
using *prime-of-localization-prime*[*OF hS p-in hp havoid loc-ring-prime*] .
qed

lemma *split-prime-factors-of-mul-eq*:

assumes *fs-sub*: $\text{set } fs \subseteq S$
and *hf*: $\forall q \in \text{set } fs. \text{ring-prime}_R q$
and *p-in*: $p \in \text{carrier } R$
and *a-in*: $a \in \text{carrier } R$
and *b-in*: $b \in \text{carrier } R$
and *hEq*: $p \otimes_R \text{foldr } (\otimes_R) fs \mathbf{1}_R = a \otimes_R b$
shows $\exists fs1 fs2 a' b'.$
 $fs <\sim\sim> fs1 @ fs2 \wedge$
 $\text{set } fs1 \subseteq S \wedge (\forall q \in \text{set } fs1. \text{ring-prime}_R q) \wedge$
 $\text{set } fs2 \subseteq S \wedge (\forall q \in \text{set } fs2. \text{ring-prime}_R q) \wedge$
 $a' \in \text{carrier } R \wedge b' \in \text{carrier } R \wedge$
 $a = \text{foldr } (\otimes_R) fs1 \mathbf{1}_R \otimes_R a' \wedge$
 $b = \text{foldr } (\otimes_R) fs2 \mathbf{1}_R \otimes_R b' \wedge$
 $p = a' \otimes_R b'$
using *fs-sub hf p-in a-in b-in hEq*
proof (*induction fs arbitrary: a b*)
case *Nil*
have *hpab-eq*: $p \otimes_R \mathbf{1}_R = a \otimes_R b$
using *Nil.premis(6)* **by** *simp*
have *hpab*: $p = a \otimes_R b$
using *Nil.premis(3)* *hpab-eq* **by** *simp*
show *?case*
proof (*intro exI conjI*)
show $\square <\sim\sim> \square @ \square$
by *simp*
show $\text{set } \square \subseteq S$
by *simp*
show $\forall q \in \text{set } \square. \text{ring-prime}_R q$
by *simp*
show $\text{set } \square \subseteq S$
by *simp*
show $\forall q \in \text{set } \square. \text{ring-prime}_R q$
by *simp*
show $a \in \text{carrier } R$
by (*rule Nil.premis(4)*)

```

show  $b \in \text{carrier } R$ 
  by (rule Nil.prem5)
show  $a = \text{foldr } (\otimes_R) [] \mathbf{1}_R \otimes_R a$ 
  using Nil.prem4 by simp
show  $b = \text{foldr } (\otimes_R) [] \mathbf{1}_R \otimes_R b$ 
  using Nil.prem5 by simp
show  $p = a \otimes_R b$ 
  by (rule hpab)
qed
next
case (Cons q qs)
have q-in:  $q \in S$ 
  using Cons.prem1 by simp
have q-carr:  $q \in \text{carrier } R$ 
  using q-in subset rev-subsetD by blast
have q-ring-prime: ring-primeR q
  using Cons.prem2 by simp
have q-prime: prime R q
  using ring-primeE(3)[OF q-carr q-ring-prime] .
have q-nz:  $q \neq \mathbf{0}$ 
  using ring-primeE(1)[OF q-carr q-ring-prime] .
have qs-sub: set qs  $\subseteq S$ 
  using Cons.prem1 by simp
have qs-prime:  $\forall r \in \text{set } qs. \text{ring-prime}_R r$ 
  using Cons.prem2 by simp
have qs-carr: set qs  $\subseteq \text{carrier } R$ 
  using qs-sub subset by blast
have rest-carr: foldr ( $\otimes_R$ ) qs  $\mathbf{1}_R \in \text{carrier } R$ 
  by (rule multlist-closed[OF qs-carr])
have prest-carr:  $p \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R \in \text{carrier } R$ 
  using Cons.prem3 rest-carr by auto
have q-dvd-ab:  $q \text{ divides}_R (a \otimes_R b)$ 
proof -
  have hEq-cons:  $p \otimes_R (q \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R) = a \otimes_R b$ 
    using Cons.prem6 by simp
  have  $q \otimes_R (p \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R) =$ 
     $p \otimes_R (q \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R)$ 
    using Cons.prem3 q-carr rest-carr by (simp add: m-assoc m-comm m-lcomm)
  also have  $\dots = a \otimes_R b$ 
    using hEq-cons .
  finally have  $q \otimes_R (p \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R) = a \otimes_R b$  .
  then have ab-eq:  $a \otimes_R b =$ 
     $q \otimes_R (p \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R)$ 
    by simp
  show ?thesis
    by (rule dividesI[OF prest-carr ab-eq])
qed
have q-dvd-a-or-b:  $q \text{ divides}_R a \vee q \text{ divides}_R b$ 
  using primeE[OF q-prime] Cons.prem4 Cons.prem5 q-dvd-ab by blast

```

from $q\text{-dvd-}a\text{-or-}b$ **show** $?case$
proof
assume $q\text{-dvd-}a$: q divides $_R$ a
obtain $a1$ **where** $a1\text{-in}$: $a1 \in \text{carrier } R$ **and** $ha1$: $a = q \otimes_R a1$
using $q\text{-dvd-}a$ **unfolding** factor-def **by** $blast$
have $a1b\text{-carr}$: $a1 \otimes_R b \in \text{carrier } R$
using $a1\text{-in}$ $\text{Cons.prem}(5)$ **by** $auto$
have hEq' : $p \otimes_R \text{foldr } (\otimes_R) \text{qs } \mathbf{1}_R = a1 \otimes_R b$
proof –
have $q \otimes_R (p \otimes_R \text{foldr } (\otimes_R) \text{qs } \mathbf{1}_R) =$
 $p \otimes_R (q \otimes_R \text{foldr } (\otimes_R) \text{qs } \mathbf{1}_R)$
using $\text{Cons.prem}(3)$ $q\text{-carr}$ rest-carr **by** $(\text{simp add: } m\text{-assoc } m\text{-comm } m\text{-lcomm})$
also have $\dots = a \otimes_R b$
using $\text{Cons.prem}(6)$ **by** simp
also have $\dots = (q \otimes_R a1) \otimes_R b$
using $ha1$ **by** simp
also have $\dots = q \otimes_R (a1 \otimes_R b)$
using $q\text{-carr } a1\text{-in}$ $\text{Cons.prem}(5)$ **by** $(\text{simp add: } m\text{-assoc})$
finally have $q \otimes_R (p \otimes_R \text{foldr } (\otimes_R) \text{qs } \mathbf{1}_R) =$
 $q \otimes_R (a1 \otimes_R b)$.
then show $?thesis$
using $q\text{-nz}$ $q\text{-carr}$ prest-carr $a1b\text{-carr}$ **by** $(\text{simp add: } m\text{-lcancel})$
qed
obtain $fs1$ $fs2$ a' b' **where**
 part : $qs <\sim\sim> fs1 @ fs2$
and $fs1\text{-sub}$: $\text{set } fs1 \subseteq S$
and $fs1\text{-prime}$: $\forall r \in \text{set } fs1. \text{ring-prime}_R r$
and $fs2\text{-sub}$: $\text{set } fs2 \subseteq S$
and $fs2\text{-prime}$: $\forall r \in \text{set } fs2. \text{ring-prime}_R r$
and $a'\text{-in}$: $a' \in \text{carrier } R$
and $b'\text{-in}$: $b' \in \text{carrier } R$
and ha : $a1 = \text{foldr } (\otimes_R) fs1 \mathbf{1}_R \otimes_R a'$
and hb : $b = \text{foldr } (\otimes_R) fs2 \mathbf{1}_R \otimes_R b'$
and hp' : $p = a' \otimes_R b'$
using $\text{Cons.IH}[OF \text{qs-sub } \text{qs-prime } \text{Cons.prem}(3) \text{a1-in } \text{Cons.prem}(5) \text{hEq}']$
by $blast$
show $?thesis$
proof (intro exI conjI)
show $q \# qs <\sim\sim> (q \# fs1) @ fs2$
using part **by** simp
show $\text{set } (q \# fs1) \subseteq S$
using $q\text{-in } fs1\text{-sub}$ **by** simp
show $\forall r \in \text{set } (q \# fs1). \text{ring-prime}_R r$
using $q\text{-ring-prime } fs1\text{-prime}$ **by** simp
show $\text{set } fs2 \subseteq S$
by $(\text{rule } fs2\text{-sub})$
show $\forall r \in \text{set } fs2. \text{ring-prime}_R r$
by $(\text{rule } fs2\text{-prime})$

```

show  $a' \in \text{carrier } R$ 
  by (rule  $a'$ -in)
show  $b' \in \text{carrier } R$ 
  by (rule  $b'$ -in)
show  $a = \text{foldr } (\otimes_R) (q \# fs1) \mathbf{1}_R \otimes_R a'$ 
proof -
  have  $fs1\text{-carr}: \text{set } fs1 \subseteq \text{carrier } R$ 
    using  $fs1\text{-sub subset}$  by blast
  have  $fs1\text{-prod-carr}: \text{foldr } (\otimes_R) fs1 \mathbf{1}_R \in \text{carrier } R$ 
    by (rule  $\text{multlist-closed}[OF fs1\text{-carr}]$ )
  have  $a = q \otimes_R a'$ 
    by (rule  $ha1$ )
  also have  $\dots = q \otimes_R (\text{foldr } (\otimes_R) fs1 \mathbf{1}_R \otimes_R a')$ 
    using  $ha$  by simp
  also have  $\dots = (q \otimes_R \text{foldr } (\otimes_R) fs1 \mathbf{1}_R) \otimes_R a'$ 
    using  $q\text{-carr } fs1\text{-prod-carr } a'\text{-in}$  by (simp add:  $m\text{-assoc}$ )
  also have  $\dots = \text{foldr } (\otimes_R) (q \# fs1) \mathbf{1}_R \otimes_R a'$ 
    by simp
  finally show ?thesis .
qed
show  $b = \text{foldr } (\otimes_R) fs2 \mathbf{1}_R \otimes_R b'$ 
  by (rule  $hb$ )
show  $p = a' \otimes_R b'$ 
  by (rule  $hp'$ )
qed
next
assume  $q\text{-dvd-}b: q \text{ divides}_R b$ 
obtain  $b1$  where  $b1\text{-in}: b1 \in \text{carrier } R$  and  $hb1: b = q \otimes_R b1$ 
  using  $q\text{-dvd-}b$  unfolding  $\text{factor-def}$  by blast
have  $ab1\text{-carr}: a \otimes_R b1 \in \text{carrier } R$ 
  using  $\text{Cons.prem}(4)$   $b1\text{-in}$  by auto
have  $hEq': p \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R = a \otimes_R b1$ 
proof -
  have  $q \otimes_R (p \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R) =$ 
     $p \otimes_R (q \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R)$ 
    using  $\text{Cons.prem}(3)$   $q\text{-carr } \text{rest-carr}$  by (simp add:  $m\text{-assoc } m\text{-comm } m\text{-lcomm}$ )
  also have  $\dots = a \otimes_R b$ 
    using  $\text{Cons.prem}(6)$  by simp
  also have  $\dots = a \otimes_R (q \otimes_R b1)$ 
    using  $hb1$  by simp
  also have  $\dots = q \otimes_R (a \otimes_R b1)$ 
    using  $q\text{-carr } \text{Cons.prem}(4)$   $b1\text{-in}$  by (simp add:  $m\text{-assoc } m\text{-comm } m\text{-lcomm}$ )
  finally have  $q \otimes_R (p \otimes_R \text{foldr } (\otimes_R) qs \mathbf{1}_R) =$ 
     $q \otimes_R (a \otimes_R b1)$  .
  then show ?thesis
    using  $q\text{-nz } q\text{-carr } \text{prest-carr } ab1\text{-carr}$  by (simp add:  $m\text{-lcancel}$ )
qed
obtain  $fs1 fs2 a' b'$  where

```

```

part: qs <~~> fs1 @ fs2
and fs1-sub: set fs1 ⊆ S
and fs1-prime: ∀ r ∈ set fs1. ring-prime_R r
and fs2-sub: set fs2 ⊆ S
and fs2-prime: ∀ r ∈ set fs2. ring-prime_R r
and a'-in: a' ∈ carrier R
and b'-in: b' ∈ carrier R
and ha: a = foldr (⊗_R) fs1 1_R ⊗_R a'
and hb: b1 = foldr (⊗_R) fs2 1_R ⊗_R b'
and hp': p = a' ⊗_R b'
using Cons.IH[OF qs-sub qs-prime Cons.prem(3) Cons.prem(4) b1-in hEq]
by blast
show ?thesis
proof (intro exI conjI)
show q # qs <~~> fs1 @ (q # fs2)
  using part by simp
show set fs1 ⊆ S
  by (rule fs1-sub)
show ∀ r ∈ set fs1. ring-prime_R r
  by (rule fs1-prime)
show set (q # fs2) ⊆ S
  using q-in fs2-sub by simp
show ∀ r ∈ set (q # fs2). ring-prime_R r
  using q-ring-prime fs2-prime by simp
show a' ∈ carrier R
  by (rule a'-in)
show b' ∈ carrier R
  by (rule b'-in)
show a = foldr (⊗_R) fs1 1_R ⊗_R a'
  by (rule ha)
show b = foldr (⊗_R) (q # fs2) 1_R ⊗_R b'
proof -
  have fs2-carr: set fs2 ⊆ carrier R
    using fs2-sub subset by blast
  have fs2-prod-carr: foldr (⊗_R) fs2 1_R ∈ carrier R
    by (rule multlist-closed[OF fs2-carr])
  have b = q ⊗_R b1
    by (rule hb1)
  also have ... = q ⊗_R (foldr (⊗_R) fs2 1_R ⊗_R b')
    using hb by simp
  also have ... = (q ⊗_R foldr (⊗_R) fs2 1_R) ⊗_R b'
    using q-carr fs2-prod-carr b'-in by (simp add: m-assoc)
  also have ... = foldr (⊗_R) (q # fs2) 1_R ⊗_R b'
    by simp
  finally show ?thesis .
qed
show p = a' ⊗_R b'
  by (rule hp')
qed

```

qed
qed

lemma *localization-irreducible-of-irreducible-prime-generated:*

assumes *hS: ring-prime-generated R S*
and *loc-dom: domain rec-rng-of-frac*
and *p-in: p ∈ carrier R*
and *hp: ring-irreducible_R p*
and *havoid: ring-avoids R S p*
shows *ring-irreducible_{rec-rng-of-frac} (rng-to-rng-of-frac p)*

proof –

interpret *L: domain rec-rng-of-frac*

by (*rule loc-dom*)

have *zero-notin: 0 ∉ S*

using *zero-notin-submonoid-of-prime-generated[OF hS]* .

have *map-p-in: rng-to-rng-of-frac p ∈ carrier rec-rng-of-frac*

by (*rule ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom p-in]*)

have *map-p-nz: rng-to-rng-of-frac p ≠ 0_{rec-rng-of-frac}*

using *map-eq-zero-iff[OF p-in zero-notin no-zero-divisors]*

ring-irreducibleE(1)[OF p-in hp]

by *blast*

have *map-p-not-unit: rng-to-rng-of-frac p ∉ Units rec-rng-of-frac*

using *map-irreducible-not-unit-of-zero-notin[OF zero-notin loc-dom p-in hp*

havoid] **by** *blast*

show *?thesis*

proof (*rule L.ring-irreducibleI*)

show *rng-to-rng-of-frac p ∈ carrier rec-rng-of-frac – {0_{rec-rng-of-frac}}*

using *map-p-in map-p-nz* **by** *blast*

show *rng-to-rng-of-frac p ∉ Units rec-rng-of-frac*

by (*rule map-p-not-unit*)

fix *x y*

assume *x-in: x ∈ carrier rec-rng-of-frac*

and *y-in: y ∈ carrier rec-rng-of-frac*

and *xy: rng-to-rng-of-frac p = x ⊗_{rec-rng-of-frac} y*

from *fraction-surj[OF x-in]* **obtain** *a* **where**

a-in: a ∈ carrier R

and *hs: ∃ s ∈ S. x = (a |_{rel} s)*

by *blast*

from *hs* **obtain** *s* **where**

s-in: s ∈ S

and *x-def: x = (a |_{rel} s)*

by *blast*

from *fraction-surj[OF y-in]* **obtain** *b* **where**

b-in: b ∈ carrier R

and *ht: ∃ t ∈ S. y = (b |_{rel} t)*

by *blast*

from *ht* **obtain** *t* **where**

t-in: t ∈ S

and *y-def: y = (b |_{rel} t)*

```

    by blast
  have s-carr: s ∈ carrier R
    using s-in subset rev-subsetD by blast
  have t-carr: t ∈ carrier R
    using t-in subset rev-subsetD by blast
  have st-in: s ⊗R t ∈ S
    using s-in t-in by (simp add: m-closed)
  obtain fs where
    fs-sub: set fs ⊆ S
    and hf: ∀ q ∈ set fs. ring-primeR q
    and hprod: foldr (⊗R) fs 1R = s ⊗R t
    using ring-prime-generatedE[OF hS st-in] by blast
  have p-rel: (p, 1) ∈ carrier rel
    using p-in one-closed by (simp add: rel-def)
  have as-rel: (a, s) ∈ carrier rel
    using a-in s-in by (simp add: rel-def)
  have bt-rel: (b, t) ∈ carrier rel
    using b-in t-in by (simp add: rel-def)
  have ab-rel: (a ⊗R b, s ⊗R t) ∈ carrier rel
    using a-in b-in s-in t-in by (simp add: rel-def)
  have frac-prod: x ⊗rec-rng-of-frac y = (a ⊗R b |rel s ⊗R t)
    using fraction-mult-rep[OF as-rel bt-rel] x-def y-def by simp
  have eq-frac: rng-to-rng-of-frac p = (a ⊗R b |rel s ⊗R t)
    using xy frac-prod by (simp add: rng-to-rng-of-frac-def)
  have hEq-cross-raw: (s ⊗R t) ⊗R p =
    1R ⊗R (a ⊗R b)
    using fraction-eq-iff-cross-multiply[OF p-rel ab-rel zero-notin no-zero-divisors]
  eq-frac
    unfolding rng-to-rng-of-frac-def by simp
  have hEq-cross: (s ⊗R t) ⊗R p = a ⊗R b
    using hEq-cross-raw a-in b-in by simp
  have hEq: p ⊗R foldr (⊗R) fs 1R = a ⊗R b
    using hEq-cross hprod m-comm p-in s-carr t-carr by auto
  obtain fs1 fs2 a' b' where
    part: fs <~> fs1 @ fs2
    and fs1-sub: set fs1 ⊆ S
    and fs1-prime: ∀ q ∈ set fs1. ring-primeR q
    and fs2-sub: set fs2 ⊆ S
    and fs2-prime: ∀ q ∈ set fs2. ring-primeR q
    and a'-in: a' ∈ carrier R
    and b'-in: b' ∈ carrier R
    and ha: a = foldr (⊗R) fs1 1R ⊗R a'
    and hb: b = foldr (⊗R) fs2 1R ⊗R b'
    and hpab: p = a' ⊗R b'
    using split-prime-factors-of-mul-eq[OF fs-sub hf p-in a-in b-in hEq] by blast
  have fs1-carr: set fs1 ⊆ carrier R
    using fs1-sub subset by blast
  have fs2-carr: set fs2 ⊆ carrier R
    using fs2-sub subset by blast

```

```

have prod1-in: foldr ( $\otimes_R$ ) fs1  $\mathbf{1}_R \in \text{carrier } R$ 
  by (rule multlist-closed[OF fs1-carr])
have prod2-in: foldr ( $\otimes_R$ ) fs2  $\mathbf{1}_R \in \text{carrier } R$ 
  by (rule multlist-closed[OF fs2-carr])
have prod1-mem: foldr ( $\otimes_R$ ) fs1  $\mathbf{1}_R \in S$ 
  using multlist-mem-submonoid[OF fs1-sub] .
have prod2-mem: foldr ( $\otimes_R$ ) fs2  $\mathbf{1}_R \in S$ 
  using multlist-mem-submonoid[OF fs2-sub] .
have a's-rel:  $(a', s) \in \text{carrier rel}$ 
  using a'-in s-in by (simp add: rel-def)
have b't-rel:  $(b', t) \in \text{carrier rel}$ 
  using b'-in t-in by (simp add: rel-def)
have x-eq:
   $x = \text{rng-to-rng-of-frac} (\text{foldr } (\otimes_R) \text{ fs1 } \mathbf{1}_R) \otimes_{\text{rec-rng-of-frac}} (a' \mid_{\text{rel}} s)$ 
  by (simp add: a's-rel ha map-mul-fraction prod1-in x-def)
have y-eq:
   $y = \text{rng-to-rng-of-frac} (\text{foldr } (\otimes_R) \text{ fs2 } \mathbf{1}_R) \otimes_{\text{rec-rng-of-frac}} (b' \mid_{\text{rel}} t)$ 
  using b't-rel hb map-mul-fraction prod2-in y-def by presburger
have a'-unit-or-b'-unit:  $a' \in \text{Units } R \vee b' \in \text{Units } R$ 
  using ring-irreducibleE(5)[OF p-in hp a'-in b'-in] hpab by blast
then show  $x \in \text{Units rec-rng-of-frac} \vee y \in \text{Units rec-rng-of-frac}$ 
proof
  assume a'-unit:  $a' \in \text{Units } R$ 
  have prod1-unit:  $\text{rng-to-rng-of-frac} (\text{foldr } (\otimes_R) \text{ fs1 } \mathbf{1}_R) \in \text{Units rec-rng-of-frac}$ 
    using map-submonoid-elem-is-unit[OF prod1-mem] .
  have frac1-unit:  $(a' \mid_{\text{rel}} s) \in \text{Units rec-rng-of-frac}$ 
    by (rule fraction-unit-numerator-is-unit[OF a'-unit s-in])
  show ?thesis
    using x-eq prod1-unit frac1-unit by simp
next
  assume b'-unit:  $b' \in \text{Units } R$ 
  have prod2-unit:  $\text{rng-to-rng-of-frac} (\text{foldr } (\otimes_R) \text{ fs2 } \mathbf{1}_R) \in \text{Units rec-rng-of-frac}$ 
    using map-submonoid-elem-is-unit[OF prod2-mem] .
  have frac2-unit:  $(b' \mid_{\text{rel}} t) \in \text{Units rec-rng-of-frac}$ 
    by (rule fraction-unit-numerator-is-unit[OF b'-unit t-in])
  show ?thesis
    using y-eq prod2-unit frac2-unit by simp
qed
qed
qed

```

```

lemma prime-of-localization-prime-prime-generated:
  assumes hS: ring-prime-generated  $R S$ 
  and p-in:  $p \in \text{carrier } R$ 
  and hp: ring-irreducible $_R p$ 
  and havoid: ring-avoids  $R S p$ 
  and hploc: ring-prime $_{\text{rec-rng-of-frac}} (\text{rng-to-rng-of-frac } p)$ 
  shows ring-prime $_R p$ 
proof -

```

```

have p-nz:  $p \neq 0$ 
  using ring-irreducibleE(1)[OF p-in hp] .
have hloc-prime: prime rec-rng-of-frac (rng-to-rng-of-frac p)
  using hploc unfolding ring-prime-def by simp
have p-prime: prime R p
proof (rule primeI)
  show  $p \notin \text{Units } R$ 
    using ring-irreducibleE(4)[OF p-in hp] .
  fix a b
  assume a-in:  $a \in \text{carrier } R$ 
    and b-in:  $b \in \text{carrier } R$ 
    and hdiv:  $p \text{ divides}_R (a \otimes_R b)$ 
  then obtain d where d-in:  $d \in \text{carrier } R$  and hfactor:  $a \otimes_R b = p \otimes_R d$ 
    unfolding factor-def by blast
  then have map-p-dvd-ab:
    rng-to-rng-of-frac p dividesrec-rng-of-frac
      (rng-to-rng-of-frac a  $\otimes_{\text{rec-rng-of-frac}}$  rng-to-rng-of-frac b)
  by (smt (verit, del-insts) a-in b-in dividesI p-in ring-hom-closed ring-hom-mult

      rng-to-rng-of-frac-is-ring-hom)
  have map-a-in: rng-to-rng-of-frac a  $\in \text{carrier rec-rng-of-frac}$ 
    using ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom a-in] .
  have map-b-in: rng-to-rng-of-frac b  $\in \text{carrier rec-rng-of-frac}$ 
    using ring-hom-closed[OF rng-to-rng-of-frac-is-ring-hom b-in] .
  have map-p-dvd-a-or-b:
    rng-to-rng-of-frac p dividesrec-rng-of-frac rng-to-rng-of-frac a  $\vee$ 
    rng-to-rng-of-frac p dividesrec-rng-of-frac rng-to-rng-of-frac b
  by (meson hloc-prime map-a-in map-b-in map-p-dvd-ab primeE)
  then show  $p \text{ divides}_R a \vee p \text{ divides}_R b$ 
    using dvd-of-localization-dvd-prime-generated[OF hS p-in a-in hp havoid]
      dvd-of-localization-dvd-prime-generated[OF hS p-in b-in hp havoid]
    by blast
  qed
from p-nz p-prime show ?thesis
  by (rule ring-primeI)
qed

lemma nagata-key-lemma-prime-generated:
  assumes hS: ring-prime-generated R S
    and loc-fd: factorial-domain rec-rng-of-frac
    and p-in:  $p \in \text{carrier } R$ 
    and hp: ring-irreducibleR p
  shows ring-primeR p
proof (cases  $\exists s \in S. p \text{ divides}_R s$ )
  case True
  then obtain s where s-in:  $s \in S$  and p-dvd-s:  $p \text{ divides}_R s$ 
    by blast
  show ?thesis
    using prime-of-irreducible-of-dvd-mem-prime-generated[OF hS p-in hp s-in

```

p -*dvd-s*] .
next
 case *False*
interpret *Lfd*: factorial-domain rec-rng-of-frac
 by (rule *loc-fd*)
interpret *L*: domain rec-rng-of-frac
 by (rule *Lfd.domain-axioms*)
have *havoid*: ring-avoids *R S p*
 using *False* **unfolding** *ring-avoids-def* by *blast*
have *loc-irreducible*: ring-irreducible_{rec-rng-of-frac} (rng-to-rng-of-frac *p*)
 using *localization-irreducible-of-irreducible-prime-generated*[*OF hS Lfd.domain-axioms*
p-in hp havoid] .
have *loc-irred-mult*: irreducible (mult-of rec-rng-of-frac) (rng-to-rng-of-frac *p*)
 using *Lfd.ring-irreducibleE*(*?*)[*OF ring-hom-closed*[*OF rng-to-rng-of-frac-is-ring-hom*
p-in] *loc-irreducible*] .
have *loc-in-nz*: rng-to-rng-of-frac *p* ∈ carrier rec-rng-of-frac − {**0**_{rec-rng-of-frac}}
 using *ring-hom-closed*[*OF rng-to-rng-of-frac-is-ring-hom p-in*]
Lfd.ring-irreducibleE(*1*)[*OF ring-hom-closed*[*OF rng-to-rng-of-frac-is-ring-hom*
p-in] *loc-irreducible*]
 by *blast*
have *loc-in-mult*: rng-to-rng-of-frac *p* ∈ carrier (mult-of rec-rng-of-frac)
 using *loc-in-nz* by *simp*
have *loc-prime-mult*: prime (mult-of rec-rng-of-frac) (rng-to-rng-of-frac *p*)
 by (rule *factorial-monoid.irreducible-prime*[*OF Lfd.factorial-monoid-axioms*
loc-irred-mult loc-in-mult])
have *loc-ring-prime*: ring-prime_{rec-rng-of-frac} (rng-to-rng-of-frac *p*)
 using *Lfd.ring-primeI*'[*OF loc-in-nz loc-prime-mult*] .
show *?thesis*
 using *prime-of-localization-prime-prime-generated*[*OF hS p-in hp havoid loc-ring-prime*]
 .
qed

lemma *nagata-theorem*:

assumes *noeth*: noetherian-domain *R*
 and *hS*: ring-prime-generated *R S*
 and *loc-fd*: factorial-domain rec-rng-of-frac
shows factorial-domain *R*

proof −

interpret *N*: noetherian-domain *R*
 by (rule *noeth*)
interpret *PC*: primeness-condition-monoid mult-of *R*
proof

fix *a*
assume *a-in*: *a* ∈ carrier (mult-of *R*)
 and *a-irred*: irreducible (mult-of *R*) *a*
have *a-in-R*: *a* ∈ carrier *R* − {**0**}
 using *a-in* by *simp*
have *a-ring-irred*: ring-irreducible_{*R*} *a*
 using *ring-irreducibleI*'[*OF a-in-R a-irred*] .

```

have a-prime: ring-primeR a
  using nagata-key-lemma-prime-generated[OF hS loc-fd DiffD1[OF a-in-R]
a-ring-irred] .
have a-prime-mult: prime (mult-of R) a
  using ring-primeE(2)[OF DiffD1[OF a-in-R] a-prime] .
show prime (mult-of R) a
  by (rule a-prime-mult)
qed
have wfactors-exist-mult:
   $\bigwedge a. \llbracket a \in \text{carrier (mult-of } R) ; a \notin \text{Units (mult-of } R) \rrbracket \implies$ 
   $\exists \text{fs. set fs} \subseteq \text{carrier (mult-of } R) \wedge \text{wfactors (mult-of } R) \text{ fs } a$ 
proof –
  fix a
  assume a-in:  $a \in \text{carrier (mult-of } R)$ 
  and a-not-unit:  $a \notin \text{Units (mult-of } R)$ 
  have a-in-R:  $a \in \text{carrier } R - \{0\}$ 
  using a-in by simp
  have a-not-unit-R:  $a \notin \text{Units } R$ 
  using a-not-unit by simp
  show  $\exists \text{fs. set fs} \subseteq \text{carrier (mult-of } R) \wedge \text{wfactors (mult-of } R) \text{ fs } a$ 
  using N.factorization-property[OF a-in-R a-not-unit-R] .
qed
have factorial-mult: factorial-monoid (mult-of R)
  by (rule mult-of.factorial-monoidI[OF wfactors-exist-mult PC.wfactors-unique])
show ?thesis
  unfolding factorial-domain-def using domain-axioms factorial-mult by simp
qed

lemma nagata-theorem-of-prime-or-unit:
  assumes noeth: noetherian-domain R
  and hS:  $\bigwedge s. s \in S \implies \text{ring-prime}_R s \vee s \in \text{Units } R$ 
  and loc-fd: factorial-domain rec-rng-of-frac
  shows factorial-domain R
proof –
  interpret N: noetherian-domain R
  by (rule noeth)
  interpret PC: primeness-condition-monoid mult-of R
proof
  fix a
  assume a-in:  $a \in \text{carrier (mult-of } R)$ 
  and a-irred: irreducible (mult-of R) a
  have a-in-R:  $a \in \text{carrier } R - \{0\}$ 
  using a-in by simp
  have a-ring-irred: ring-irreducibleR a
  using ring-irreducibleI'[OF a-in-R a-irred] .
  have a-ring-prime: ring-primeR a
  using nagata-key-lemma[OF hS loc-fd DiffD1[OF a-in-R] a-ring-irred] .
  show prime (mult-of R) a
  using ring-primeE(2)[OF DiffD1[OF a-in-R] a-ring-prime] .

```

```

qed
have wfactors-exist-mult:
   $\bigwedge a. \llbracket a \in \text{carrier (mult-of } R); a \notin \text{Units (mult-of } R) \rrbracket \implies$ 
     $\exists \text{fs. set fs} \subseteq \text{carrier (mult-of } R) \wedge \text{wfactors (mult-of } R) \text{ fs } a$ 
proof –
  fix a
  assume a-in:  $a \in \text{carrier (mult-of } R)$ 
  and a-not-unit:  $a \notin \text{Units (mult-of } R)$ 
  have a-in-R:  $a \in \text{carrier } R - \{0\}$ 
  using a-in by simp
  have a-not-unit-R:  $a \notin \text{Units } R$ 
  using a-not-unit by simp
  show  $\exists \text{fs. set fs} \subseteq \text{carrier (mult-of } R) \wedge \text{wfactors (mult-of } R) \text{ fs } a$ 
  using N.factorization-property[OF a-in-R a-not-unit-R] .
qed
have factorial-mult: factorial-monoid (mult-of } R)
  by (rule mult-of.factorial-monoidI[OF wfactors-exist-mult PC.wfactors-unique])
show ?thesis
  unfolding factorial-domain-def using domain-axioms factorial-mult by simp
qed

lemma nagata-theorem-of-prime-generators:
  assumes noeth: noetherian-domain } R
  and S-eq:  $S = \text{ring-mult-submonoid-closure } R \ A$ 
  and A-sub:  $A \subseteq \text{carrier } R$ 
  and hprime:  $\bigwedge q. q \in A \implies \text{ring-prime}_R q$ 
  and loc-fd: factorial-domain rec-rng-of-frac
  shows factorial-domain } R
proof –
  have hS: ring-prime-generated } R S
  unfolding S-eq
  by (rule ring-prime-generated-mult-submonoid-closure[OF ring-axioms A-sub
hprime])
  show ?thesis
  by (rule nagata-theorem[OF noeth hS loc-fd])
qed

lemma nagata-theorem-of-finite-prime-generators:
  assumes noeth: noetherian-domain } R
  and finA: finite } A
  and S-eq:  $S = \text{ring-mult-submonoid-closure } R \ A$ 
  and A-sub:  $A \subseteq \text{carrier } R$ 
  and hprime:  $\bigwedge q. q \in A \implies \text{ring-prime}_R q$ 
  and loc-fd: factorial-domain rec-rng-of-frac
  shows factorial-domain } R
  using finA nagata-theorem-of-prime-generators[OF noeth S-eq A-sub hprime loc-fd]
by blast

end

```

```

end
theory Polynomial-Applications
  imports
    Nagata-Lemmas
    HOL-Algebra.Polynomial-Divisibility
begin

```

5 Polynomial applications

This theory packages the first concrete application layer on top of the record-based Nagata descent theorem. The present results isolate the abstract “localize away X” step for HOL-Algebra polynomial rings, together with the standard field-coefficient specialization in which X is prime by the degree-one irreducibility criterion.

```

context domain
begin

```

```

lemma polynomial-prime-X:
  assumes K: subfield K R
  shows ring-primeK[X] X
proof –
  have X-in: X ∈ carrier (K[X])
  using var-closed(1)[OF subfieldE(1)[OF K]] .
  have X-irred: pirreducible K X
  using degree-one-imp-pirreducible[OF K X-in]
  by (simp add: var-def)
  show ?thesis
  using pprime-iff-pirreducible[OF K X-in] X-irred by simp
qed

```

```

lemma polynomial-prime-generated-powers-X:
  assumes K: subring K R
  and hX: ring-primeK[X] X
  shows ring-prime-generated (K[X]) (ring-powers-set (K[X]) X)
  using ring-prime-generated-powers-set[OF univ-poly-is-ring[OF K] var-closed(1)[OF K] hX] .

```

```

end

```

```

locale polynomial-away-X-localization =
  fixes R (structure) and P (structure) and S and K
  assumes poly-axioms: nagata-localization P S
  and base-axioms: domain R
  and P-eq: P = K[X]
  and S-eq: S = ring-powers-set (K[X]) X
begin

```

abbreviation *loc-ring* **where** *loc-ring* \equiv *eq-obj-rng-of-frac.rec-rng-of-frac P S*

Once a localization of $K[X]$ at the powers of X has been fixed, Nagata's theorem reduces factoriality of $K[X]$ to factoriality of that localization, provided X is prime.

lemma *polynomial-factorial-of-localized-X-factorial*:

assumes *K*: *subring K R*

and *noeth*: *noetherian-domain (K[X])*

and *hX*: *ring-prime_{K[X]} X*

and *loc-fd*: *factorial-domain loc-ring*

shows *factorial-domain (K[X])*

proof –

have *noethP*: *noetherian-domain P*

using *noeth* **unfolding** *P-eq* .

have *hXP*: *ring-prime_P X*

using *hX* **unfolding** *P-eq* .

have *hS*: *ring-prime-generated P S*

unfolding *P-eq S-eq*

by (*rule domain.polynomial-prime-generated-powers-X[OF base-axioms K hX]*)

have *fdP*: *factorial-domain P*

by (*rule nagata-localization.nagata-theorem[OF poly-axioms noethP hS loc-fd]*)

show *?thesis*

using *fdP* **unfolding** *P-eq* .

qed

lemma *polynomial-factorial-of-localized-X-factorial-field*:

assumes *K*: *subfield K R*

and *noeth*: *noetherian-domain (K[X])*

and *loc-fd*: *factorial-domain loc-ring*

shows *factorial-domain (K[X])*

proof –

have *hX*: *ring-prime_{K[X]} X*

by (*rule domain.polynomial-prime-X[OF base-axioms K]*)

show *?thesis*

by (*rule polynomial-factorial-of-localized-X-factorial[
OF subfieldE(1)[OF K] noeth hX loc-fd]*)

qed

end

end

theory *Fraction-Field-Applications*

imports

Nagata-Lemmas

Polynomial-Applications

HOL-Algebra.Polynomial-Divisibility

begin

6 Constant-prime localization applications

This theory packages the constant-prime specialization of the polynomial application layer at the same level of abstraction as *Polynomial-Applications*: it specializes Nagata's theorem to multiplicative sets generated by constant prime polynomials and isolates the corresponding descent step for polynomial rings.

context *domain*
begin

lemma *polynomial-prime-generated-constant-closure*:

assumes *Ksub*: *subring* K R
and *A-sub*: $A \subseteq \text{carrier } (R \ (\!| \text{carrier} := K))$
and *hprime*: $\bigwedge q. q \in A \implies \text{ring-prime}_{K[X]} (\text{poly-of-const } q)$

shows

ring-prime-generated $(K[X])$
(ring-mult-submonoid-closure $(K[X])$ *(poly-of-const ' A))*

proof (*rule ring-prime-generated-mult-submonoid-closure*[*OF univ-poly-is-ring*[*OF Ksub*]])

show *poly-of-const ' A* $\subseteq \text{carrier } (K[X])$

proof

fix p

assume *p-in*: $p \in \text{poly-of-const ' A}$

then obtain q **where** *q-in*: $q \in A$ **and** *p-def*: $p = \text{poly-of-const } q$

by *blast*

have *qK*: $q \in \text{carrier } (R \ (\!| \text{carrier} := K))$

using *A-sub* *q-in* **by** *blast*

show $p \in \text{carrier } (K[X])$

unfolding *p-def* **by** (*rule ring-hom-closed*[*OF canonical-embedding-is-hom*[*OF Ksub*] *qK*])

qed

fix p

assume $p \in \text{poly-of-const ' A}$

then obtain q **where** *q-in*: $q \in A$ **and** *p-def*: $p = \text{poly-of-const } q$

by *blast*

show *ring-prime* $_{K[X]}$ p

unfolding *p-def* **by** (*rule hprime*[*OF q-in*])

qed

end

locale *polynomial-constant-prime-localization* =

fixes R (**structure**) **and** P (**structure**) **and** S **and** $K :: 'a \text{ set}$ **and** $A :: 'a \text{ set}$

assumes *poly-axioms*: *nagata-localization* P S

and *base-axioms*: *domain* R

and *P-eq*: $P = K[X]$

and *S-eq*: $S = \text{ring-mult-submonoid-closure } (K[X])$ *(ring.poly-of-const* $(R \ (\!| \text{carrier} := K))$ *' A)*

begin

abbreviation *const-poly* **where**

const-poly \equiv *ring.poly-of-const* (R (\downarrow *carrier* := K))

abbreviation *loc-ring* **where** *loc-ring* \equiv *eq-obj-rng-of-frac.rec-rng-of-frac* P S

Once a localization of $K[X]$ at a constant-prime closure has been fixed, Nagata's theorem immediately reduces factoriality of $K[X]$ to factoriality of that localization.

lemma *polynomial-factorial-of-localized-constant-primes-factorial*:

assumes *Ksub*: *subring* K R

and *A-sub*: $A \subseteq$ *carrier* (R (\downarrow *carrier* := K))

and *noeth*: *noetherian-domain* ($K[X]$)

and *hprime*: $\bigwedge q. q \in A \implies$ *ring-prime* $_{K[X]}$ (*const-poly* q)

and *loc-fd*: *factorial-domain* *loc-ring*

shows *factorial-domain* ($K[X]$)

proof –

interpret *base*: *domain* R

by (*rule base-axioms*)

interpret *K-ring*: *ring* R (\downarrow *carrier* := K)

by (*rule base.subring-is-ring*[*OF* *Ksub*])

have *noethP*: *noetherian-domain* P

using *noeth* **unfolding** *P-eq* .

have *hprimeP*: $\bigwedge q. q \in A \implies$ *ring-prime* $_P$ (*const-poly* q)

using *P-eq* *hprime* **by** *blast*

have *S-closure-eq*: $S =$ *ring-mult-submonoid-closure* P (*const-poly* ‘ A)

using *P-eq* *S-eq* **by** *blast*

have *Aimg-sub*: *const-poly* ‘ $A \subseteq$ *carrier* P

proof

fix p

assume *p-in*: $p \in$ *const-poly* ‘ A

then obtain q **where** *q-in*: $q \in A$ **and** *p-def*: $p =$ *const-poly* q

by *blast*

have *qK*: $q \in$ *carrier* (R (\downarrow *carrier* := K))

using *A-sub* *q-in* **by** *blast*

have *q-poly-in-base*: *base.poly-of-const* $q \in$ *carrier* ($K[X]$)

using *ring-hom-memE*(1)[*OF* *base.canonical-embedding-is-hom*[*OF* *Ksub*] *qK*]

by *simp*

have *p-def-base*: $p =$ *base.poly-of-const* q

using *p-def* *fun-cong*[*OF* *base.poly-of-const-consistent*[*OF* *Ksub*], *of* q] **by** *simp*

show $p \in$ *carrier* P

unfolding *P-eq*

using *p-def-base* *q-poly-in-base* **by** *simp*

qed

have *hprime-img*: $\bigwedge p. p \in$ *const-poly* ‘ $A \implies$ *ring-prime* $_P$ p

using *hprimeP* **by** *blast*

have *fdP*: *factorial-domain* P

by (*rule nagata-localization.nagata-theorem-of-prime-generators*]

```

      OF poly-axioms noethP S-closure-eq Aimg-sub hprime-img loc-fd])
  show ?thesis
    using fdP unfolding P-eq .
qed

end

end

```

```

theory Nagata-Factoriality
  imports
    Prime-Generated
    Nagata-Lemmas
    Polynomial-Applications
    Fraction-Field-Applications
    HOL-Computational-Algebra.Polynomial
begin

```

7 Nagata-factoriality scaffolding

Nagata’s factoriality theorem descends unique factorization from a localization to the base ring under a prime-generated hypothesis on the multiplicative set. The present entry now packages the prime-generated core, a wrapper layer over the AFP localization entry, and a record-based HOL-Algebra proof of both the prime-generated and prime-or-unit descent variants in *Nagata-Lemmas*. That theory now also packages theorem-level entry points for submonoid closures generated by prime families, including a finite-generator wrapper. The additional theories *Polynomial-Applications* and *Fraction-Field-Applications* package abstract polynomial applications for localization away X and for localizations generated by constant prime polynomials, cf. Nagata and Samuel.[2] [3]

```

lemma prime-generated-constant-prime-polynomials:
  fixes A :: 'a :: semidom set
  assumes  $\bigwedge c. c \in A \implies \text{prime-elem } c$ 
  shows prime-generated (mult-submonoid-closure (( $\lambda c. [:c:]$ ) ‘ A))
proof (rule prime-generated-mult-submonoid-closure)
  fix q
  assume  $q \in (\lambda c. [:c:])$  ‘ A
  then obtain c where  $c \in A$  and q-def:  $q = [:c:]$ 
  by blast
  from assms[OF  $\langle c \in A \rangle$ ] show prime-elem q
  unfolding q-def by (rule lift-prime-elem-poly)
qed

```

```

corollary zero-notin-constant-prime-polynomial-closure:
  fixes A :: 'a :: idom set

```

```

assumes  $\bigwedge c. c \in A \implies \text{prime-elem } c$ 
shows  $0 \notin \text{mult-submonoid-closure } ((\lambda c. [:c:]) ' A)$ 
proof –
  have prime-generated (mult-submonoid-closure (( $\lambda c. [:c:]$ ) '  $A$ ))
    using assms by (rule prime-generated-constant-prime-polynomials)
  then show ?thesis
    by (rule zero-notin-prime-generated[where  $S = \text{mult-submonoid-closure } ((\lambda c. [:c:]) ' A)$ ])
qed

```

The last corollary isolates one of the key configurations in the constant-prime polynomial application: the multiplicative set generated by constant prime polynomials is prime-generated and therefore avoids zero.

Separately, the theory *Localization-Interface* exposes an Isabelle/HOL wrapper around the AFP localization construction with lemmas for representative equality, numerator-denominator surjectivity, denominator rescaling, cross-multiplication in the domain case, units coming from both the multiplicative set and base-ring units, and injectivity of the canonical localization map under the usual domain hypotheses. On top of that, *Nagata-Lemmas* proves the descent lemmas needed for Nagata's theorem, together with a record-based multiplicative-closure API

```

[[ring ? $R$ ; ? $A \subseteq \text{carrier } ?R$ ;  $\bigwedge q. q \in ?A \implies \text{ring-prime } ?R q$ ]
 $\implies \text{ring-prime-generated } ?R (\text{ring-mult-submonoid-closure } ?R ?A)$ 

```

for the constant-prime submonoids used in the fraction-field route, culminating in theorem statements

```

[[nagata-localization ? $R$  ? $S$ ; noetherian-domain ? $R$ ;
  ring-prime-generated ? $R$  ? $S$ ;
  factorial-domain (eq-obj-rng-of-frac.rec-rng-of-frac ? $R$  ? $S$ )]
 $\implies \text{factorial-domain } ?R$ 

```

and

```

[[nagata-localization ? $R$  ? $S$ ; noetherian-domain ? $R$ ;
   $\bigwedge s. s \in ?S \implies \text{ring-prime } ?R s \vee s \in \text{Units } ?R$ ;
  factorial-domain (eq-obj-rng-of-frac.rec-rng-of-frac ? $R$  ? $S$ )]
 $\implies \text{factorial-domain } ?R$ 

```

together with the prime-generator closure wrappers

```

[[nagata-localization ? $R$  ? $S$ ; noetherian-domain ? $R$ ;
  ? $S = \text{ring-mult-submonoid-closure } ?R ?A$ ; ? $A \subseteq \text{carrier } ?R$ ;
   $\bigwedge q. q \in ?A \implies \text{ring-prime } ?R q$ ;
  factorial-domain (eq-obj-rng-of-frac.rec-rng-of-frac ? $R$  ? $S$ )]
 $\implies \text{factorial-domain } ?R$ 

```

and

[[nagata-localization ?R ?S; noetherian-domain ?R; finite ?A;
 ?S = ring-mult-submonoid-closure ?R ?A; ?A \subseteq carrier ?R;
 $\bigwedge q. q \in ?A \implies$ ring-prime ?R q;
 factorial-domain (eq-obj-rng-of-frac.rec-rng-of-frac ?R ?S)]
 \implies factorial-domain ?R

. The theory *Polynomial-Applications* then specializes this framework to the polynomial ring case by proving

[[domain ?R; subfield ?K ?R]] \implies pprime ?R ?K X ?R

over fields and the abstract away-X descent theorem

[[polynomial-away-X-localization ?R ?P ?S ?K; subring ?K ?R;
 noetherian-domain (?K [X] ?R); pprime ?R ?K X ?R;
 factorial-domain (eq-obj-rng-of-frac.rec-rng-of-frac ?P ?S)]
 \implies factorial-domain (?K [X] ?R)

, while *Fraction-Field-Applications* packages the companion constant-prime closure theorem

[[polynomial-constant-prime-localization ?R ?P ?S ?K ?A; subring ?K ?R;
 ?A \subseteq carrier (?R(carrier := ?K)); noetherian-domain (?K [X] ?R);
 $\bigwedge q. q \in ?A \implies$ pprime ?R ?K (ring.poly-of-const (?R(carrier := ?K)) q);
 factorial-domain (eq-obj-rng-of-frac.rec-rng-of-frac ?P ?S)]
 \implies factorial-domain (?K [X] ?R)

.

end

References

- [1] H. Matsumura. *Commutative Ring Theory*, volume 8 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1986.
- [2] M. Nagata. *Local Rings*, volume 13 of *Interscience Tracts in Pure and Applied Mathematics*. Interscience Publishers, New York, 1962.
- [3] P. Samuel. *Lectures on Unique Factorization Domains*, volume 30 of *Tata Institute of Fundamental Research Lectures on Mathematics and Physics*. Tata Institute of Fundamental Research, Bombay, 1964.