# A verified factorization algorithm for integer polynomials with polynomial complexity* 

Jose Divasón Sebastiaan Joosten René Thiemann<br>Akihisa Yamada

September 13, 2023


#### Abstract

Short vectors in lattices and factors of integer polynomials are related. Each factor of an integer polynomial belongs to a certain lattice. When factoring polynomials, the condition that we are looking for an irreducible polynomial means that we must look for a small element in a lattice, which can be done by a basis reduction algorithm. In this development we formalize this connection and thereby one main application of the LLL basis reduction algorithm: an algorithm to factor square-free integer polynomials which runs in polynomial time. The work is based on our previous Berlekamp-Zassenhaus development, where the exponential reconstruction phase has been replaced by the polynomial-time basis reduction algorithm. Thanks to this formalization we found a serious flaw in a textbook.


## Contents

1 Introduction 2
2 Factor bound 5
3 Executable dvdm operation 6
3.1 Uniqueness of division algorithm for polynomials . . . . . . . 7
3.2 Executable division operation modulo $m$ for polynomials . . . 12

4 The LLL factorization algorithm 13
5 Correctness of the LLL factorization algorithm 16
5.1 Basic facts about the auxiliary functions . . . . . . . . . . . . 16
5.2 Facts about Sylvester matrices and norms . . . . . . . . . . . 17

[^0]5.3 Proof of the key lemma 16.20 ..... 21
5.4 Properties of the computed lattice and its connection with Sylvester matrices ..... 22
5.5 Proving that factorization-lattice returns a basis of the lattice ..... 23
5.6 Being in the lattice is being a multiple modulo ..... 24
5.7 Soundness of the LLL factorization algorithm ..... 31
6 Calculating All Possible Sums of Sub-Multisets ..... 46
7 Implementation and soundness of a modified version of Al- gorithm 16.22 ..... 47
7.1 Previous lemmas obtained using local type definitions ..... 47
7.2 The modified version of Algorithm 16.22 ..... 48
7.3 Soundness proof ..... 50
7.3.1 Starting the proof ..... 50
7.3.2 Inner loop ..... 56
7.3.3 Outer loop ..... 72
7.3.4 Final statement ..... 77
8 Mistakes in the textbook Modern Computer Algebra (2nd edition) ..... 78
8.1 A real problem of Algorithm 16.22 ..... 78
8.2 Another potential problem of Algorithm 16.22 ..... 78
8.3 Verified wrong results ..... 80

## 1 Introduction

In order to factor an integer polynomial $f$, we may assume a modular factorization of $f$ into several monic factors $u_{i}: f \equiv \operatorname{lc}(f) \cdot \prod_{i} u_{i}$ modulo $m$ where $m=p^{l}$ is some prime power for user-specified $l$. In Isabelle, we just reuse our verified modular factorization algorithm [1] to obtain the modular factorization of $f$.

We briefly explain how to compute non-trivial integer factors of $f$. The key is the following lemma [2, Lemma 16.20].

Lemma 1 ([2, Lemma 16.20]) Let $f, g, u$ be non-constant integer polynomials. Let $u$ be monic. If $u$ divides $f$ modulo $m, u$ divides $g$ modulo $m$, and $\|f\|^{\text {degree }(g)} \cdot\|g\|^{\text {degree }(f)}<m$, then $h=\operatorname{gcd}(f, g)$ is non-constant.

Let $f$ be a polynomial of degree $n$. Let $u$ be any degree- $d$ factor of $f$ modulo $m$. Now assume that $f$ is reducible, so $f=f_{1} \cdot f_{2}$ where w.l.o.g., we assume that $u$ divides $f_{1}$ modulo $m$ and that $0<\operatorname{degree}\left(f_{1}\right)<n$. Let us further assume that a lattice $L_{u, k}$ encodes the set of all polynomials of
degree below $d+k$ (as vectors of length $d+k$ ) which are divisible by $u$ modulo $m$. Fix $k=n-d$. Then clearly, $f_{1} \in L_{u, k}$.

In order to instantiate Lemma 1, it now suffices to take $g$ as the polynomial corresponding to any short vector in $L_{u, k}: u$ will divide $g$ modulo $m$ by definition of $L_{u, k}$ and moreover $\operatorname{degree}(g)<n$. The short vector requirement will provide an upper bound to satisfy the assumption $\|f\|^{\text {degree }(g)} \cdot\|g\|^{\text {degree }(f)}<m$.

$$
\begin{gather*}
\|g\| \leq 2^{(n-1) / 2} \cdot\left\|f_{1}\right\| \leq 2^{(n-1) / 2} \cdot 2^{n-1}\|f\|=2^{3(n-1) / 2}\|f\|  \tag{1}\\
\|f\|^{\text {degree }(g)} \cdot\|g\|^{\text {degree }(f)} \leq\|f\|^{n-1} \cdot\left(2^{3(n-1) / 2}\|f\|\right)^{n}=\|f\|^{2 n-1} \cdot 2^{3 n(n-1) / 2} \tag{2}
\end{gather*}
$$

Here, the first inequality in (1) is the short vector approximation $\left(f_{1} \in L_{u, k}\right)$. The second inequality in (1) is Mignotte's factor bound ( $f_{1}$ is a factor of $f$ ). Finally, (1) is used as an approximation of $\|g\|$ in (2).

Hence, if $l$ is chosen large enough so that $m=p^{l}>\|f\|^{2 n-1} \cdot 2^{3 n(n-1) / 2}$ then all preconditions of Lemma 1 are satisfied, and $h=g c d(f, g)$ will be a non-constant factor of $f$. Since the degree of $h$ will be strictly less than $n$, $h$ is also a proper factor of $f$, i.e., in particular $h \notin\{1, f\}$.

The textbook [2] also describes the general idea of the factorization algorithm based on the previous lemma in prose, and then presents an algorithm in pseudo-code which slightly extends the idea by directly splitting off $i r$ reducible factors [2, Algorithm 16.22]. We initially implemented and tried to verify this pseudo-code algorithm (see files Factorization_Algorithm_16_22.thy and Modern_Computer_Algebra_Problem.thy). After some work, we had only one remaining goal to prove: the content of the polynomial $g$ corresponding to the short vector is not divisible by the chosen prime $p$. However, we were unable to figure out how to discharge this goal and then also started to search for inputs where the algorithm delivers wrong results. After a while we realized that Algorithm 16.22 indeed has a serious flaw as demonstrated in the upcoming example.

Example 1 Consider the square-free and content-free polynomial $f=(1+$ $x) \cdot\left(1+x+x^{3}\right)$. Then according to Algorithm 16.22 we determine

- the prime $p=2$
- the exponent $l=61$
(our new formalized algorithm uses a tighter bound which results in $l=41$ )
- the leading coefficient $b=1$
- the value $B=96$
- the factorization $\bmod p$ via $h_{1}=1+x, h_{2}=1+x+x^{3}$
- the factorization mod $p^{l}$ via $g_{1}=1+x, g_{2}=1+x+x^{3}$
- $f^{*}=f, T=\{1,2\}, G=\emptyset$.
- we enter the loop and in the first iteration choose
- $u=1+x+x^{3}, d=3, j=4$
- we consider the lattice generated by $(1,1,0,1),\left(p^{l}, 0,0,0\right),\left(0, p^{l}, 0,0\right)$, ( $0,0, p^{l}, 0$ ).
- now we obtain a short vector in the lattice: $g^{*}=(2,2,0,2)$.

Note that $g^{*}$ has not really been computed by Algorithm 16.10, but it satisfies the soundness criterion, i.e., it is a sufficiently short vector in the lattice.

To see this, note that a shortest vector in the lattice is (1, 1, 0,1$)$.

$$
\left\|g^{*}\right\|=2 \cdot \sqrt{3} \leq 2 \cdot \sqrt{2} \cdot \sqrt{3}=2^{(j-1) / 2} \cdot\|(1,1,0,1)\|
$$

So $g^{*}$ has the required precision that was assumed by the short-vector calculation.

- the problem at this point is that $p$ divides the content of $g^{*}$. Consequently, every polynomial divides $g^{*} \bmod p$. Thus in step 9 we compute $S=T, h=1$, enter the then-branch and update $T=\emptyset$, $G=G \cup\left\{1+x+x^{3}\right\}, f^{*}=1, b=1$.
- Then in step 10 we update $G=\left\{1+x+x^{3}, 1\right\}$ and finally return that the factorization of $f$ is $\left(1+x+x^{3}\right) \cdot 1$.

More details about the bug and some other wrong results presented in the book are shown in the file Modern_Computer_Algebra_Problem.thy.

Once we realized the problem, we derived another algorithm based on Lemma 1, which also runs in polynomial-time, and prove its soundness in Isabelle/HOL. The corresponding Isabelle statement is as follows:

## Theorem 1 (LLL Factorization Algorithm)

```
assumes square_free ( \(f::\) int poly)
and degree \(f \neq 0\)
and \(L L L\) factorization \(f=g s\)
shows \(f=\) prod_list \(g s\)
and \(\forall g_{i} \in\) set gs. irreducible \(g_{i}\)
```

Finally, we also have been able to fix Algorithm 16.22 and provide a formal correctness proof of the the slightly modified version. It can be seen as an implementation of the pseudo-code factorization algorithm given by Lenstra, Lenstra, and Lovász [3].

## 2 Factor bound

This theory extends the work about factor bounds which was carried out in the Berlekamp-Zassenhaus development.
theory Factor-Bound-2
imports Berlekamp-Zassenhaus.Factor-Bound LLL-Basis-Reduction.Norms
begin
lemma norm-1-bound-mignotte: norm1 $f \leq 2 \wedge($ degree $f) *$ mahler-measure $f$
proof (cases $f=0$ )
case f0: False
have $c f:$ coeffs $f=\operatorname{map}(\lambda i$. coeff $f i)[0 . .<\operatorname{Suc}($ degree $f)]$ unfolding coeffs-def
using $f 0$ by auto
have real-of-int (sum-list (map abs (coeffs f))) $=\left(\sum i \leq\right.$ degree $f$. real-of-int $\mid$ poly.coeff $\left.f i \mid\right)$ unfolding cf of-int-hom.hom-sum-list unfolding sum-list-sum-nth by (rule sum.cong, force, auto simp: o-def nth-append)
also have $\ldots \leq\left(\sum i \leq\right.$ degree $f$. real (degree $f$ choose $\left.i\right) *$ mahler-measure $f$ ) by (rule sum-mono, rule Mignotte-bound)
also have $\ldots=\operatorname{real}(\operatorname{sum}(\lambda i .($ degree $f$ choose $i))\{$..degree $f\}) *$ mahler-measure $f$
unfolding sum-distrib-right[symmetric] by auto
also have $\ldots=2 \wedge($ degree $f) *$ mahler-measure $f$ unfolding choose-row-sum by auto
finally show ?thesis unfolding norm1-def .
qed (auto simp: mahler-measure-ge-0 norm1-def)
lemma mahler-measure-l2norm: mahler-measure $f \leq \operatorname{sqrt}\left(\right.$ of-int $\left.\|f\|^{2}\right)$
using Landau-inequality-mahler-measure $[o f f$ ] unfolding sq-norm-poly-def
by (auto simp: power2-eq-square)
lemma sq-norm-factor-bound:
fixes $f h$ :: int poly
assumes $d v d: h d v d f$ and $f 0: f \neq 0$
shows $\|h\|^{2} \leq 2^{\wedge}(2 *$ degree $h) *\|f\|^{2}$
proof -
let $? r=$ real-of-int
have h21: ?r $\|h\|^{2} \leq($ ?r $($ norm1 $h))$ ^2 using norm2-le-norm1-int $[$ of $h]$
by (metis of-int-le-iff of-int-power)
also have $\ldots \leq\left(\mathscr{Q}^{\wedge}(\right.$ degree $h) *$ mahler-measure $\left.h\right) \wedge^{\text {® }}$ using power-mono[OF norm-1-bound-mignotte[of h], of 2] by (auto simp: norm1-ge-0)
also have $\ldots=2 \wedge(2 *$ degree $h) *($ mahler-measure $h) \wedge^{\wedge}$ by (simp add: power-even-eq power-mult-distrib)
also have $\ldots \leq 2$ ^ $2 *$ degree $h) *($ mahler-measure $f) \wedge_{2}$ by (rule mult-left-mono[OF power-mono], auto simp: mahler-measure-ge-0 mahler-measure-dvd[OF f0 dvd])

```
    also have ... \leq2`(2 * degree h) * ?r (|f||
    proof (rule mult-left-mono)
    have ?r (|f\mp@subsup{|}{}{2})\geq0 by auto
    from real-sqrt-pow2[OF this]
    show (mahler-measure f)}\mp@subsup{)}{}{2}\leq\mathrm{ ?r ( }|f\mp@subsup{|}{}{2}\mathrm{ )
        using power-mono[OF mahler-measure-l2norm[of f], of 2]
        by (auto simp: mahler-measure-ge-0)
    qed auto
    also have ... = ?r (2`(2*degree h)*|f|⿻土一⿱日一
    by (simp add: ac-simps)
finally show |h||
qed
end
```


## 3 Executable dvdm operation

This theory contains some results about division of integer polynomials which are not part of Polynomial＿Factorization．Dvd＿Int＿Poly．thy．
Essentially，we give an executable implementation of division modulo m ．
theory Missing－Dvd－Int－Poly
imports
Berlekamp－Zassenhaus．Poly－Mod－Finite－Field
Berlekamp－Zassenhaus．Polynomial－Record－Based
Berlekamp－Zassenhaus．Hensel－Lifting
Subresultants．Subresultant
Perron－Frobenius．Cancel－Card－Constraint
begin
lemma degree－div－mod－smult：
fixes $g$ ：：int poly
assumes $g$ ：degree $g<j$
and $r$ ：degree $r<d$
and $u$ ：degree $u=d$
and $g 1: g=q * u+$ smult $m r$
and $q: q \neq 0$ and $m$－not0：$m \neq 0$
shows degree $q<j-d$
proof－
have $u$－not 0 ：$u \neq 0$ using $u r$ by auto
have $d$－uq：$d \leq$ degree（ $u * q$ ）using $u$ degree－mult－right－le［OF q］by auto
have $j: j>$ degree $(q * u+$ smult $m r)$ using $g 1 g$ by auto
have degree（smult $m r$ ）＜d using degree－smult－eq m－not0 $r$ by auto
also have $\ldots \leq$ degree $(u * q)$ using $d$－uq by auto
finally have deg－mr－uq：degree（smult $m r$ ）$<$ degree $(q * u)$
by（simp add：mult．commute）
have j2：degree $(q * u+$ smult $m r)=$ degree $(q * u)$
by（rule degree－add－eq－left［OF deg－mr－uq］）

```
    also have ... = degree q + degree u
    by (rule degree-mult-eq[OF q u-not0])
    finally have degree q}=\mathrm{ degree }g-\mathrm{ degree }u\mathrm{ using g1 by auto
    thus ?thesis
    using j j2 <degree ( }q*u)=\mathrm{ degree q + degree u〉u
    by linarith
qed
```


### 3.1 Uniqueness of division algorithm for polynomials

```
lemma uniqueness-algorithm-division-poly:
    fixes f::'a::{comm-ring,semiring-1-no-zero-divisors} poly
    assumes f1:f=g*q1+r1
            and f2: }f=g*q2+r
            and g:g\not=0
            and r1:r1 = 0 \vee degree r1<degree g
            and r2: r2 = 0 \vee degree r2 < degree g
        shows q1 = q2 ^r1 = r2
proof -
    have 0 = g*q1 +r1-(g*q2 + r2) using f1 f2 by auto
    also have ... = g* (q1 - q2) + r1 - r2
        by (simp add: right-diff-distrib)
    finally have eq: g* (q1 - q2) = r2 - r1 by auto
    have q-eq: q1 = q2
    proof (rule ccontr)
        assume q1-not-q2: q1 }=~q
        hence nz: g* (q1 - q2) \not=0 using g by auto
        hence degree (g* (q1 - q2)) \geq degree g
            by (simp add: degree-mult-right-le)
        moreover have degree (r2 - r1) < degree g
            using eq nz degree-diff-less r1 r2 by auto
            ultimately show False using eq by auto
    qed
    moreover have r1 = r2 using eq q-eq by auto
    ultimately show ?thesis by simp
qed
lemma pdivmod-eq-pdivmod-monic:
    assumes g: monic g
    shows pdivmod fg= pdivmod-monic fg
proof -
    obtain q r where qr: pdivmod f g=( q,r) by simp
    obtain Q R where QR: pdivmod-monic f g = (Q,R) by (meson surj-pair)
    have g0:g\not=0 using g by auto
    have f1:f=g*q+r
    by (metis Pair-inject mult-div-mod-eq qr)
    have r:r=0 V degree r< degree g
    by (metis Pair-inject assms degree-mod-less leading-coeff-0-iff qr zero-neq-one)
    have f2: f}=g*Q+
```

```
    by (simp add:QR assms pdivmod-monic(1))
    have R:R=0\vee degree }R<\mathrm{ degree }
    by (rule pdivmod-monic[OF g QR])
    have q=Q\wedger=R by (rule uniqueness-algorithm-division-poly[OF f1 fo g0 r R])
    thus ?thesis using qr QR by auto
qed
context poly-mod
begin
definition pdivmod2 f g=( if Mpg=0 then (0,f)
    else let ilc = inverse-p m ((lead-coeff (Mp g)));
    h= Polynomial.smult ilc (Mp g); (q,r)= pseudo-divmod (Mp f) (Mph)
    in (Polynomial.smult ilc q,r))
end
context poly-mod-prime-type
begin
lemma dvdm-iff-pdivmod0:
    assumes f:(F :: 'a mod-ring poly) =of-int-poly f
    and g:(G :: 'a mod-ring poly) = of-int-poly g
    shows g dvdm f = (snd (pdivmod F G)=0)
proof -
    have [transfer-rule]:MP-Rel f F unfolding MP-Rel-def
        by (simp add: Mp-f-representative f)
    have [transfer-rule]:MP-Rel g G unfolding MP-Rel-def
        by (simp add: Mp-f-representative g)
    have (snd (pdivmod FG)=0)=(G dvd F)
        unfolding dvd-eq-mod-eq-0 by auto
    from this [untransferred] show ?thesis by simp
qed
lemma of-int-poly-Mp-O[simp]:(of-int-poly (Mp a) = (0:: 'a mod-ring poly)) =
(Mpa=0)
    by (auto, metis Mp-f-representative map-poly-0 poly-mod.Mp-Mp)
lemma uniqueness-algorithm-division-of-int-poly:
    assumes g0: Mp g\not=0
    and f:(F:: 'a mod-ring poly) = of-int-poly f
    and g:(G :: 'a mod-ring poly) = of-int-poly g
    and}F:F=G*Q+
    and R:R=0\vee degree }R<\mathrm{ degree }
    and Mp-f:Mpf=Mpg*q+r
    and r:r = 0 V degree r < degree (Mp g)
shows }Q=of-int-poly q\wedgeR=of-int-poly 
proof (rule uniqueness-algorithm-division-poly[OF F - R])
    have f}\mp@subsup{f}{}{\prime}:Mpf=to-int-poly F unfolding 
    by (simp add: Mp-f-representative)
```

```
have g':Mpg=to-int-poly G unfolding g
    by (simp add: Mp-f-representative)
have f'\prime: of-int-poly (Mpf)=F
    by (metis (no-types, lifting) Dp-Mp-eq Mp-f-representative
        Mp-smult-m-0 add-cancel-left-right f map-poly-zero of-int-hom.map-poly-hom-add
            to-int-mod-ring-hom.hom-zero to-int-mod-ring-hom.injectivity)
have g'\prime: of-int-poly (Mp g)=G
    by (metis (no-types, lifting) Dp-Mp-eq Mp-f-representative
        Mp-smult-m-0 add-cancel-left-right g map-poly-zero of-int-hom.map-poly-hom-add
            to-int-mod-ring-hom.hom-zero to-int-mod-ring-hom.injectivity)
    have F = of-int-poly ( Mp g*q+r) using Mp-f f'l by auto
    also have ... = G* of-int-poly q + of-int-poly r
    by (simp add: g' of-int-poly-hom.hom-add of-int-poly-hom.hom-mult)
finally show F=G* of-int-poly q + of-int-poly r .
show of-int-poly r = 0 \vee degree (of-int-poly r::'a mod-ring poly) < degree G
proof (cases r=0)
    case True
    hence of-int-poly r=0 by auto
    then show ?thesis by auto
next
    case False
    have degree (of-int-poly r::'a mod-ring poly) \leq degree (r)
    by (simp add: degree-map-poly-le)
    also have ... < degree (Mp g) using r False by auto
    also have ... = degree G by (simp add: g')
    finally show ?thesis by auto
qed
show G\not=0 using g0 unfolding g'\[symmetric] by simp
qed
corollary uniqueness-algorithm-division-to-int-poly:
    assumes g0: Mp g\not=0
    and f:(F:: 'a mod-ring poly) = of-int-poly f
    and g:(G :: 'a mod-ring poly) =of-int-poly g
    and F:F=G*Q+R
    and R:R=0\vee degree }R<\mathrm{ degree }
    and Mp-f:Mpf=Mpg*q+r
    and r:r=0\vee degree r<degree (Mp g)
shows Mpq= to-int-poly Q\wedge Mpr= to-int-poly R
using uniqueness-algorithm-division-of-int-poly[OF assms]
by (auto simp add: Mp-f-representative)
lemma uniqueness-algorithm-division-Mp-Rel:
    assumes monic-Mpg: monic (Mp g)
    and f:(F :: 'a mod-ring poly) = of-int-poly f
and g:(G:: 'a mod-ring poly) = of-int-poly g
and qr: pseudo-divmod (Mpf) (Mpg) = (q,r)
```

and $Q R$ : pseudo-divmod $F G=(Q, R)$
shows MP-Rel $q Q \wedge M P$-Rel $r R$
proof (unfold MP-Rel-def, rule uniqueness-algorithm-division-to-int-poly[OF - $f$ g])
show $f-g q-r$ : $M p f=M p g * q+r$
by (rule pdivmod-monic (1)[OF monic-Mpg], simp add: pdivmod-monic-pseudo-divmod qr monic-Mpg)
have monic- $G$ : monic $G$ using monic-Mpg
using $M p$-f-representative $g$ by auto
show $F=G * Q+R$
by (rule pdivmod-monic(1)[OF monic-G], simp add: pdivmod-monic-pseudo-divmod $Q R$ monic- $G$ )
show $M p g \neq 0$ using monic-Mpg by auto
show $R=0 \vee$ degree $R<$ degree $G$
by (rule pdivmod-monic(2)[OF monic-G],
auto simp add: pdivmod-monic-pseudo-divmod monic- $G$ intro: $Q R$ )
show $r=0 \vee$ degree $r<$ degree $(M p g)$
by (rule pdivmod-monic(2)[OF monic-Mpg], auto simp add: pdivmod-monic-pseudo-divmod monic-Mpg intro: qr)
qed
definition MP-Rel-Pair $A B \equiv(\operatorname{let}(a, b)=A ;(c, d)=B$ in MP-Rel a $c \wedge M P$-Rel $b d)$
lemma pdivmod2-rel[transfer-rule]:
(MP-Rel $===>M P-$ Rel $===>M P-$ Rel-Pair $)($ pdivmod2) $($ pdivmod $)$
proof (auto simp add: rel-fun-def MP-Rel-Pair-def)
interpret pm: prime-field $m$
using $m$ unfolding prime-field-def mod-ring-locale-def by auto
have $p$ : prime-field $\operatorname{TYPE}\left({ }^{\prime} a\right) m$
using $m$ unfolding prime-field-def mod-ring-locale-def by auto
fix $f F g G a b$
assume 1 [transfer-rule]: $M P$-Rel f $F$
and $2[$ transfer-rule $]: M P-R e l ~ g G$
and 3: pdivmod2 $f g=(a, b)$
have MP-Rel a $(F \operatorname{div} G) \wedge M P-R e l b(F \bmod G)$
proof (cases Mp $g \neq 0$ )
case True note $M p-g=$ True
have $G: G \neq 0$ using $M p-g 2$ unfolding MP-Rel-def by auto
have $g G[$ transfer-rule $]$ : pm.mod-ring-rel (lead-coeff (Mp g)) (lead-coeff $G$ )
using 2
unfolding pm.mod-ring-rel-def MP-Rel-def
by auto
have $[$ transfer-rule]: (pm.mod-ring-rel $===>$ pm.mod-ring-rel) (inverse-p $m)$ inverse
by (rule prime-field.mod-ring-inverse $[O F \quad p]$ )
hence rel-inverse-p[transfer-rule]:
pm.mod-ring-rel (inverse-p $m((l e a d-c o e f f ~(M p ~ g))))($ inverse (lead-coeff $G))$
using $g G$ unfolding rel-fun-def by auto
let $? h=($ Polynomial.smult (inverse-p $m($ lead-coeff $(M p g))) g)$
define $h$ where $h: h=$ Polynomial.smult (inverse-p m (lead-coeff $\left(\begin{array}{ll}M p & g\end{array}\right)$ )
( $M p g$ )
define $H$ where $H: H=$ Polynomial.smult (inverse (lead-coeff $G$ )) $G$
have $h H^{\prime}: M P$-Rel ? $h ~ H$ unfolding MP-Rel-def unfolding $H$
by (metis (mono-tags, opaque-lifting) 2 MP-Rel-def $M$-to-int-mod-ring Mp-f-representative
rel-inverse-p functional-relation left-total-MP-Rel of-int-hom.map-poly-hom-smult pm.mod-ring-rel-def right-unique-MP-Rel to-int-mod-ring-hom.injectivity to-int-mod-ring-of-int-M)
have Mp (Polynomial.smult (inverse-p m (lead-coeff (Mp g))) g)
$=M p$ (Polynomial.smult (inverse-p $m$ (lead-coeff $(M p g)))(M p g))$ by simp
hence $h H$ : MP-Rel $h H$ using $h H^{\prime} h$ unfolding MP-Rel-def by auto
obtain $q x$ where pseudo-fh: pseudo-divmod $(M p f)(M p h)=(q, x)$ by (meson surj-pair)
hence $l c$ - $G$ : (lead-coeff $G) \neq 0$ using $G$ by auto
have $a: a=$ Polynomial.smult (inverse-p $m(($ lead-coeff $(M p g)))) q$
using 3 pseudo-fh Mp-g
unfolding pdivmod2-def Let-def $h$ by auto
have $b: b=x$ using 3 pseudo-fh $M p-g$
unfolding pdivmod2-def Let-def $h$ by auto
have $M p-$ Rel-FH: MP-Rel $q(F$ div $H) \wedge M P-R e l x(F \bmod H)$
proof (rule uniqueness-algorithm-division-Mp-Rel)
show monic ( $M p h$ )
proof -
have aux: (inverse-p $m$ (lead-coeff $(M p g)))=$ to-int-mod-ring (inverse
(lead-coeff $G$ ))
using rel-inverse-p unfolding pm.mod-ring-rel-def by auto
hence $M$ (inverse-p $m(M$ (poly.coeff $g($ degree $(M p g))))$
$=$ to-int-mod-ring (inverse (lead-coeff $G)$ )
by (simp add: M-to-int-mod-ring Mp-coeff)
thus ?thesis unfolding $h$ unfolding $M p$-coeff by auto
(metis (no-types, lifting) 2 H MP-Rel-def Mp-coeff aux degree-smult-eq $g G$
$h H^{\prime}$
inverse-zero-imp-zero lc-G left-inverse pm.mod-ring-rel-def to-int-mod-ring-hom.degree-map-poly-hom to-int-mod-ring-hom.hom-one to-int-mod-ring-times)
qed
hence monic- $H$ : monic $H$ using $h H H l c-G$ by auto
show $f$ : $F=$ of-int-poly $f$
using 1 unfolding MP-Rel-def
by (simp add: Mp-f-representative poly-eq-iff)
have pdivmod $F H=$ pdivmod-monic $F H$
by (rule pdivmod-eq-pdivmod-monic $[$ OF monic- $H$ ])
also have $\ldots=$ pseudo-divmod $F H$
by (rule pdivmod-monic-pseudo-divmod $[$ OF monic- $H]$ )
finally show pseudo-divmod $F H=(F \operatorname{div} H, F \bmod H)$ by simp
show $H=o f-i n t-p o l y h$
by (meson MP-Rel-def Mp-f-representative hH right-unique-MP-Rel right-unique-def)

```
        show pseudo-divmod (Mpf) (Mph)=(q,x) by (rule pseudo-fh)
    qed
    hence Mp-Rel-F-div-H:MP-Rel q (F div H) and Mp-Rel-F-mod-H:MP-Rel x
(F mod H) by auto
    have F div H= Polynomial.smult (lead-coeff G) (F div G)
        by (simp add: H div-smult-right)
    hence F-div-G:(F div G)= Polynomial.smult (inverse (lead-coeff G)) (F div
H)
        using lc-G by auto
    have MP-Rel a (F div G)
    proof -
        have of-int-poly (Polynomial.smult (inverse-p m ((lead-coeff (Mp g)))) q)
            = smult (inverse (lead-coeff G)) (F div H)
                            by (metis (mono-tags) MP-Rel-def M-to-int-mod-ring Mp-Rel-F-div-H
Mp-f-representative
                of-int-hom.map-poly-hom-smult pm.mod-ring-rel-def rel-inverse-p right-unique-MP-Rel
                    right-unique-def to-int-mod-ring-hom.injectivity to-int-mod-ring-of-int-M)
            thus ?thesis
            using Mp-Rel-F-div-H
            unfolding MP-Rel-def a F-div-G Mp-f-representative by auto
    qed
    moreover have MP-Rel b (F mod G)
            using Mp-Rel-F-mod-H b H inverse-zero-imp-zero lc-G
            by (metis mod-smult-right)
        ultimately show ?thesis by auto
    next
    assume Mp-g-0:\negMpg\not=0
    hence pdivmod2 f g = ( O, f) unfolding pdivmod2-def by auto
    hence }a:a=0\mathrm{ and b:b=f using 3 by auto
    have G0:G=0 using Mp-g-0 2 unfolding MP-Rel-def by auto
    have MP-Rel a (F div G) unfolding MP-Rel-def G0 a by auto
    moreover have MP-Rel b (F mod G) using 1 unfolding MP-Rel-def G0 a b
by auto
    ultimately show ?thesis by simp
    qed
    thus MP-Rel a (F div G) and MP-Rel b (F mod G) by auto
qed
```


### 3.2 Executable division operation modulo $m$ for polynomials

lemma dvdm-iff-Mp-pdivmod2:
shows $g d v d m f=(M p($ snd $($ pdivmod2 $f g))=0)$
proof -
let ? $F=($ of-int-poly $f)::^{\prime}$ a mod-ring poly
let ? $G=($ of-int-poly $g):: ' a$ mod-ring poly
have a[transfer-rule]: MP-Rel $f$ ?F
by (simp add: MP-Rel-def Mp-f-representative)
have $b[$ transfer-rule]: $M P$-Rel $g$ ? $G$

```
    by (simp add:MP-Rel-def Mp-f-representative)
    have MP-Rel-Pair (pdivmod2 f g) (pdivmod ?F?G)
    using pdivmod2-rel unfolding rel-fun-def using a b by auto
    hence MP-Rel (snd (pdivmod2 f g)) (snd (pdivmod ?F ?G))
    unfolding MP-Rel-Pair-def by auto
hence (Mp (snd (pdivmod2 fg))=0)=(snd (pdivmod ?F?G)=0)
    unfolding MP-Rel-def by auto
thus ?thesis using dvdm-iff-pdivmod0 by auto
qed
end
```

lemmas (in poly-mod-prime) dvdm-pdivmod $=$ poly-mod-prime-type.dvdm-iff-Mp-pdivmod2
[unfolded poly-mod-type-simps, internalize-sort ' $a$ :: prime-card, OF type-to-set,
unfolded remove-duplicate-premise, cancel-type-definition, OF non-empty]
lemma (in poly-mod) dvdm-code:
$g d v d m f=($ if prime $m$ then $M p($ snd $($ pdivmod2 $f g))=0$
else Code.abort (STR "dvdm error: $m$ is not a prime number") $(\lambda-. g d v d m f)$ )
using poly-mod-prime.dvdm-pdivmod[unfolded poly-mod-prime-def]
by auto
declare poly-mod.pdivmod2-def [code]
declare poly-mod.dvdm-code[code]
end

## 4 The LLL factorization algorithm

This theory contains an implementation of a polynomial time factorization algorithm. It first constructs a modular factorization. Afterwards it recursively invokes the LLL basis reduction algorithm on one lattice to either split a polynomial into two non-trivial factors, or to deduce irreducibility.

theory LLL-Factorization-Impl<br>imports LLL-Basis-Reduction.LLL-Certification<br>Factor-Bound-2<br>Missing-Dvd-Int-Poly<br>Berlekamp-Zassenhaus.Berlekamp-Zassenhaus<br>begin<br>hide-const (open) up-ring.coeff up-ring.monom<br>Unique-Factorization.factors Divisibility.factors<br>Unique-Factorization.factor Divisibility.factor<br>Divisibility.prime

definition factorization-lattice where factorization-lattice ukm map ( $\lambda i$. vec-of-poly-n $(u *$ monom $1 i)($ degree $u+k)$ ) $[k>. .0] @$ map $(\lambda i$. vec-of-poly-n (monom $m i)($ degree $u+k)$ ) [degree $u>. .0]$

```
fun min-degree-poly :: int poly \(\Rightarrow\) int poly \(\Rightarrow\) int poly
    where min-degree-poly \(a b=(\) if degree \(a \leq\) degree \(b\) then \(a\) else \(b)\)
fun choose- \(u\) :: int poly list \(\Rightarrow\) int poly
    where choose-u [] = undefined
    |choose-u \([g i]=g i\)
    |choose-u (gi \# gj \# gs) = min-degree-poly gi (choose-u (gj \# gs))
```

lemma factorization-lattice-code[code]: factorization-lattice u $k m=$ (
let $n=$ degree $u$ in
map
( $\lambda i$. vec-of-poly-n (monom-mult $i u)(n+k))[k>. .0]$
@ map ( $\lambda$ i. vec-of-poly-n (monom mi) $(n+k)$ ) [ $n>. .0]$
) unfolding factorization-lattice-def monom-mult-def
by (auto simp: ac-simps Let-def)

Optimization: directly try to minimize coefficients of polynomial $u$.

```
definition \(L L L\)-short-polynomial where
    LLL-short-polynomial pl \(n u=\) poly-of-vec (short-vector-hybrid 2 (factorization-lattice
    (poly-mod.inv-Mp pl (poly-mod.Mp pl u)) (n - degree u) pl))
locale \(L L L\)-implementation \(=\)
    fixes \(p\) pl :: int
begin
function \(L L L\)-many-reconstruction where
    LLL-many-reconstruction \(f\) us \(=\) (let
        \(d=\) degree \(f ;\)
        \(d 2=d \operatorname{div} 2 ;\)
        f2-opt \(=\) find-map-filter
            ( \(\lambda\) u. gcd \(f\) (LLL-short-polynomial pl (Suc d2) u))
            ( \(\lambda\) f2. let deg \(=\) degree f2 in deg \(>0 \wedge\) deg \(<d\) )
            (filter \((\lambda u\). degree \(u \leq d 2) u s\) )
        in case f2-opt of None \(\Rightarrow[f]\)
        | Some f2 \(\Rightarrow\) let f1 \(=f\) div f2;
            (us1, us2) \(=\) List.partition ( \(\lambda\) gi. poly-mod.dvdm p gi f1) us
            in LLL-many-reconstruction f1 us1 @ LLL-many-reconstruction f2 us2)
    by pat-completeness auto
termination
proof (relation measure ( \(\lambda(f, u s)\). degree \(f\) ), goal-cases \()\)
    case (3 f us d d2 f2-opt f2 f1 pair us1 us2)
    from find-map-filter-Some[OF 3(4)[unfolded 3(3) Let-def]] 3(1,5)
    show ?case by auto
```

```
next
    case (2 f us d d2 f2-opt f2 f1 pair us1 us2)
    from find-map-filter-Some[OF 2(4)[unfolded 2(3) Let-def]] 2(1,5)
    have f:f=f1*f2 and f0: f}=
        and deg: degree f2 > 0 degree f2 < degree f by auto
    have degree f}=\mathrm{ degree f1 + degree f2 using f0 unfolding f
        by (subst degree-mult-eq, auto)
    with deg show ?case by auto
qed auto
function LLL-reconstruction where
    LLL-reconstruction fus = (let
        d = degree f;
        u = choose-u us;
        g=LLL-short-polynomial pl d u;
        f2 = gcd f g;
        deg = degree f2
        in if deg=0\vee deg \geqd then [f]
            else let f1 = f div f2;
            (us1,us2) = List.partition (\lambda gi.poly-mod.dvdm p gi f1) us
            in LLL-reconstruction f1 us1 @ LLL-reconstruction f2 us2)
    by pat-completeness auto
```


## termination

```
proof (relation measure ( }\lambda\mathrm{ ( }f,\mathrm{ us). degree f), goal-cases)
    case (2 f us d u g f2 deg f1 pair us1 us2)
    hence f:f=f1*f2 and f0: f}\not=0\mathrm{ by auto
    have deg: degree f = degree f1 + degree f2 using f0 unfolding f
        by (subst degree-mult-eq, auto)
    from 2 have degree f2 > 0 degree f2 < degree f by auto
    thus ?case using deg by auto
qed auto
end
declare LLL-implementation.LLL-reconstruction.simps[code]
declare LLL-implementation.LLL-many-reconstruction.simps[code]
definition LLL-factorization :: int poly }=>\mathrm{ int poly list where
    LLL-factorization f = (let
        - find suitable prime
        p= suitable-prime-bz f;
        - compute finite field factorization
        (-, fs) = finite-field-factorization-int p f;
        - determine exponent l and B
        n= degree f;
        no =|f|}\mp@subsup{|}{}{2}
        B = sqrt-int-ceiling (2`(5* (n-1)*(n-1))*no^(2* (n-1)));
        l= find-exponent p B;
        - perform hensel lifting to lift factorization to mod pl
```

```
    \(u s=h e n s e l-l i f t i n g ~ p l f s ;\)
    - reconstruct integer factors via LLL algorithm
        \(p l=p\urcorner\)
    in LLL-implementation.LLL-reconstruction p pl fus)
definition LLL-many-factorization :: int poly \(\Rightarrow\) int poly list where
    LLL-many-factorization \(f=(\) let
        - find suitable prime
        \(p=\) suitable-prime-bz f;
        - compute finite field factorization
        \((-, f s)=\) finite-field-factorization-int \(p\) f;
        - determine exponent l and B
        \(n=\) degree \(f\);
        \(n o=\|f\|^{2} ;\)
        \(B=\operatorname{sqrt-int-ceiling}(2 `(5 *(n \operatorname{div} 2) *(n \operatorname{div} 2)) * n o `(2 *(n d i v 2))) ;\)
        \(l=\) find-exponent \(p B\);
        - perform hensel lifting to lift factorization to \(\bmod p^{l}\)
        us \(=\) hensel-lifting \(p l f f\);
        - reconstruct integer factors via LLL algorithm
        \(p l=p\urcorner\)
    in LLL-implementation.LLL-many-reconstruction p pl fus)
```

end

## 5 Correctness of the LLL factorization algorithm

This theory connects short vectors of lattices and factors of polynomials. From this connection, we derive soundness of the lattice based factorization algorithm.

```
theory LLL-Factorization
    imports
        LLL-Factorization-Impl
        Berlekamp-Zassenhaus.Factorize-Int-Poly
begin
```


### 5.1 Basic facts about the auxiliary functions

hide-const (open) module.smult
lemma nth-factorization-lattice:
fixes $u$ and $d$
defines $n \equiv$ degree $u$
assumes $i<n+d$
shows factorization-lattice udm!i=
vec-of-poly-n (if $i<d$ then $u *$ monom $1(d-S u c i)$ else monom $m(n+d-S u c$
i)) $(n+d)$
using assms
by (unfold factorization-lattice-def, auto simp: nth-append smult-monom Let-def not-less)
lemma length-factorization-lattice[simp]:
shows length (factorization-lattice $u$ d $m$ ) $=$ degree $u+d$
by (auto simp: factorization-lattice-def Let-def)
lemma dim-factorization-lattice:
assumes $x<$ degree $u+d$
shows dim-vec (factorization-lattice $u d m!x)=$ degree $u+d$
unfolding factorization-lattice-def using assms nth-append
by (simp add: nth-append Let-def)
lemma dim-factorization-lattice-element:
assumes $x \in$ set (factorization-lattice $u d m$ ) shows dim-vec $x=$ degree $u+d$
using assms by (auto simp: factorization-lattice-def Let-def)
lemma set-factorization-lattice-in-carrier[simp]: set (factorization-lattice u d m) $\subseteq$ carrier-vec (degree $u+d$ )
using dim-factorization-lattice by (auto simp: factorization-lattice-def Let-def)
lemma choose-u-Cons: choose-u $(x \# x s)=$
(if $x s=[]$ then $x$ else min-degree-poly $x($ choose-u xs))
by (cases xs, auto)
lemma choose-u-member: $x s \neq[] \Longrightarrow$ choose-u xs $\in$ set xs
by (induct xs, auto simp: choose-u-Cons)
declare choose-u.simps[simp del]

### 5.2 Facts about Sylvester matrices and norms

lemma (in LLL) lattice-is-span [simp]: lattice-of $x s=$ span-list $x s$ by (unfold lattice-of-def span-list-def lincomb-list-def image-def, auto)
lemma sq-norm-row-sylvester-mat1:
fixes $f g$ :: ' $a$ :: conjugatable-ring poly
assumes $i: i<$ degree $g$
shows $\|($ row $($ sylvester-mat $f g) i)\left\|^{2}=\right\| f \|^{2}$
proof (cases $f=0$ )
case True
thus ?thesis
by (auto simp add: sylvester-mat-def row-def sq-norm-vec-def o-def
interv-sum-list-conv-sum-set-nat i intro!: sum-list-zero)
next
case False note $f=$ False
let ?f $=\lambda j$. if $i \leq j \wedge j-i \leq$ degree $f$ then coeff $f($ degree $f+i-j)$ else 0
let $? h=\lambda j . j+i$
let ?row $=$ vec $($ degree $f+$ degree $g)$ ?f
let $? g=\lambda j$ degree $f-j$
have image-g: ?g ' $\{0 . . .<$ Suc (degree $f)\}=\{0 . .<$ Suc (degree $f)\}$
by (auto simp add: image-def)
(metis (no-types, opaque-lifting) Nat.add-diff-assoc add.commute add-diff-cancel-left'
atLeastLessThan-iff diff-Suc-Suc diff-Suc-less less-Suc-eq-le zero-le)
have bij-h: bij-betw ?h $\{0 . .<$ Suc (degree $f)\}\{i . .<\operatorname{Suc}($ degree $f+i)\}$
unfolding bij-betw-def image-def
by (auto, metis atLeastLessThan-iff le-add-diff-inverse2
less-diff-conv linorder-not-less not-less-eq zero-order(3))
have $\|$ row (sylvester-mat $f$ g) $i\left\|^{2}=\right\|$ ? row $\|^{2}$
by (rule arg-cong[of - sq-norm-vec], insert $i$, auto simp add: row-def sylvester-mat-def sylvester-mat-sub-def)
also have $\ldots=$ sum-list $($ map $($ sq-norm $\circ$ ?f) $[0 . .<$ degree $f+$ degree $g])$
unfolding sq-norm-vec-def by auto
also have $\ldots=\operatorname{sum}($ sq-norm $\circ$ ?f) $\{0 . . .<$ degree $f+$ degree $g\}$
unfolding interv-sum-list-conv-sum-set-nat by auto
also have $\ldots=\operatorname{sum}(s q-n o r m \circ$ ?f) $\{i . .<\operatorname{Suc}($ degree $f+i)\}$
by (rule sum.mono-neutral-right, insert $i$, auto)
also have..$=\operatorname{sum}(($ sq-norm $\circ$ ?f) $) \circ$ ?h) $\{0 . .<$ Suc $($ degree $f)\}$
by (unfold o-def, rule sum.reindex-bij-betw[symmetric, OF bij-h])
also have $\ldots=\operatorname{sum}(\lambda j$. sq-norm $($ coeff $f($ degree $f-j))$ ) $\{0 . . .<$ Suc $($ degree $f)\}$
by (rule sum.cong, auto)
also have $\ldots=\operatorname{sum}((\lambda j$. sq-norm $($ coeff f $j)) \circ$ ?g) $\{0 . .<$ Suc (degree $f)\}$
unfolding $o$-def ..
also have $\ldots=\operatorname{sum}(\lambda j$. sq-norm (coeff fj)) (?g' $\{0 . .<$ Suc (degree $f)\})$
by (rule sum.reindex[symmetric], auto simp add: inj-on-def)
also have $\ldots=\operatorname{sum}($ sq-norm $\circ$ coeff $f)\{0 . .<$ Suc (degree $f$ ) $\}$ unfolding image- $g$ by $\operatorname{simp}$
also have $\ldots=$ sum-list (map sq-norm (coeffs $f$ ))
unfolding coeff $s$-def using $f$
by (simp add: interv-sum-list-conv-sum-set-nat)
finally show ?thesis unfolding sq-norm-poly-def by auto qed
lemma sq-norm-row-sylvester-mat2:
fixes $f g$ :: ' $a$ :: conjugatable-ring poly
assumes $i 1$ : degree $g \leq i$ and $i 2: i<$ degree $f+$ degree $g$
shows $\|$ row (sylvester-mat $f g$ ) $i\left\|^{2}=\right\| g \|^{2}$
proof -
let ?f $=\lambda j$. if $i-$ degree $g \leq j \wedge j \leq i$ then coeff $g(i-j)$ else 0
let ?row $=$ vec $($ degree $f+$ degree $g)$ ?f
let $? h=\lambda j . j+i-$ degree $g$
let $? g=\lambda j$. degree $g-j$
have image-g: ?g ' $\{0 . . .<$ Suc (degree $g)\}=\{0 . . .<$ Suc $($ degree $g)\}$
by (auto simp add: image-def)
(metis atLeastLessThan-iff diff-diff-cancel diff-le-self less-Suc-eq-le zero-le)
have $x: x-(i-$ degree $g) \leq$ degree $g$ if $x: x<$ Suc $i$ for $x$ using $x$ by auto
have bij-h: bij-betw ?h $\{0 . .<$ Suc (degree g) $\}\{i-$ degree $g . .<S u c i\}$
unfolding bij-betw-def inj-on-def using i1 i2 unfolding image-def
by (auto, metis (no-types) Nat.add-diff-assoc atLeastLessThan-iff x less-Suc-eq-le
less-eq-nat.simps(1) ordered-cancel-comm-monoid-diff-class.diff-add)
have $\|$ row (sylvester-mat $f g$ ) $i\left\|^{2}=\right\|$ ?row $\|^{2}$
by (rule arg-cong[of - sq-norm-vec], insert i1 i2, auto simp add: row-def sylvester-mat-def sylvester-mat-sub-def)
also have $\ldots=$ sum-list (map (sq-norm $\circ$ ?f) $[0 . .<$ degree $f+$ degree $g])$
unfolding sq-norm-vec-def by auto
also have $\ldots=\operatorname{sum}($ sq-norm $\circ$ ?f) $\{0 . .<$ degree $f+$ degree $g\}$
unfolding interv-sum-list-conv-sum-set-nat by auto
also have $\ldots=\operatorname{sum}($ sq-norm $\circ$ ?f) $\{i-$ degree $g . .<$ Suc $i\}$
by (rule sum.mono-neutral-right, insert i2, auto)
also have $\ldots=\operatorname{sum}(($ sq-norm $\circ$ ?f) $\circ$ ? $h$ ) $\{0 . .<$ Suc (degree g) $\}$
by (unfold o-def, rule sum.reindex-bij-betw[symmetric, OF bij-h])
also have $\ldots=\operatorname{sum}(\lambda j$. sq-norm (coeff $g($ degree $g-j))$ ) $\{0 . .<$ Suc (degree $g)\}$
by (rule sum.cong, insert i1, auto)
also have $\ldots=\operatorname{sum}((\lambda j$. sq-norm $($ coeff $g j)) \circ ? g)\{0 . .<$ Suc (degree $g)\}$
unfolding o-def ..
also have $\ldots=\operatorname{sum}(\lambda j$. sq-norm $($ coeff $g j))(? g$ ' $\{0 . .<$ Suc (degree $g)\})$
by (rule sum.reindex[symmetric], auto simp add: inj-on-def)
also have $\ldots=\operatorname{sum}($ sq-norm $\circ$ coeff $g)\{0 . .<$ Suc (degree $g)\}$ unfolding image- $g$ by $\operatorname{simp}$
also have $\ldots=$ sum-list (map sq-norm (coeffs g))
unfolding coeffs-def
by (simp add: interv-sum-list-conv-sum-set-nat)
finally show ?thesis unfolding sq-norm-poly-def by auto
qed
lemma Hadamard's-inequality-int:
fixes $A$ :: int mat
assumes $A: A \in$ carrier-mat $n n$
shows $|\operatorname{det} A| \leq \operatorname{sqrt}($ of-int $($ prod-list $($ map sq-norm (rows $A))))$
proof -
let ? $A=$ map-mat real-of-int $A$
have $|\operatorname{det} A|=\mid \operatorname{det}$ ?A $\mid$ unfolding of-int-hom.hom-det by simp
also have $\ldots \leq$ sqrt (prod-list (map sq-norm (rows ?A)))
by (rule Hadamard's-inequality [of ? A n], insert A, auto)
also have $\ldots=\operatorname{sqrt}($ of-int (prod-list (map sq-norm (rows $A))$ )) unfolding of-int-hom.hom-prod-list map-map
by (rule arg-cong[of $-\lambda$ x. sqrt (prod-list $x$ )], rule nth-equalityI, force,
auto simp: sq-norm-of-int[symmetric] row-def intro!: arg-cong[of--sq-norm-vec])
finally show? ?thesis.
qed
lemma resultant-le-prod-sq-norm:
fixes $f g$ ::int poly
defines $n \equiv$ degree $f$ and $k \equiv$ degree $g$

```
    shows |resultant f g| \leq sqrt (of-int (|f||}\mp@subsup{|}{}{`}k*|g|\mp@subsup{|}{}{2`}n)
proof -
    let ?S = sylvester-mat fg
    let ?f = sq-norm \circ row ?S
    have map-rw1: map ?f [0..<degree g] = replicate k |f||
    proof (rule nth-equalityI)
    let ?M = map (sq-norm ○ row (sylvester-mat f g)) [0..<degree g]
    show length ?M = length (replicate k|f\mp@subsup{|}{}{2}}\mathrm{ ) using k-def by auto
    show ?M ! i = replicate k|f\mp@subsup{|}{}{2}!i\mathrm{ if }i:i<length ?M for }
    proof -
        have ik: i<k using i k-def by auto
        hence i-deg-g: i< degree g using k-def by auto
        have replicate k|f\mp@subsup{|}{}{2}!i=|f\mp@subsup{|}{}{2}}\mathrm{ by (rule nth-replicate[OF ik])
        also have ... = (sq-norm ○ row (sylvester-mat fg)) (0+i)
            using sq-norm-row-sylvester-mat1 ik k-def by force
        also have ... = ?M ! i by (rule nth-map-upt[symmetric], simp add: i-deg-g)
        finally show ?M ! i = replicate k|f\mp@subsup{|}{}{2}!i..
    qed
    qed
    have map-rw2: map ?f [degree g..<degree f + degree g] = replicate n |g\mp@subsup{|}{}{2}
    proof (rule nth-equalityI)
    let ?M = map (sq-norm ○ row (sylvester-mat f g)) [degree g..<degree f + degree
g]
    show length ?M = length (replicate n |g| |
    show ?M!i= replicate n |g\mp@subsup{|}{}{2}!i\mathrm{ if }i<\mathrm{ length ?M for }i
    proof -
        have i-n: i<n using n-def that by auto
        hence i-deg-f:i<degree f using n-def by auto
        have replicate n|g\mp@subsup{|}{}{2}!i=|g\mp@subsup{|}{}{2}}\mathrm{ by (rule nth-replicate[OF i-n])
        also have ... = (sq-norm ○ row (sylvester-mat fg)) (degree g + i)
                using i-n n-def
                by (simp add: sq-norm-row-sylvester-mat2)
        also have ... = ?M! i
            by (simp add: i-deg-f)
        finally show ?M ! i = replicate n |g| |
    qed
qed
have p1: prod-list (map ?f [0..<degree g]) =|f||
    unfolding map-rw1 by (rule prod-list-replicate)
have p2: prod-list (map ?f [degree g..<degree f + degree g]) =|g| |}\mp@subsup{|}{}{`}
    unfolding map-rw2 by (rule prod-list-replicate)
have list-rw: [0..<degree f+degree g] = [0..<degree g]@ [degree g..<degree f+
degree g]
    by (metis add.commute upt-add-eq-append zero-le)
have |resultant f g| = |det ?S | unfolding resultant-def ..
also have ... \leqsqrt (of-int (prod-list (map sq-norm (rows ?S))))
    by (rule Hadamard's-inequality-int, auto)
also have map sq-norm (rows ?S) = map ?f [0..<degree f + degree g]
    unfolding Matrix.rows-def by auto
```

also have $\ldots=$ map ?f $([0 . .<$ degree $g] @[$ degree $g . .<$ degree $f+$ degree $g])$
by (simp add: list-rw)
also have prod-list $\ldots=$ prod-list (map ?f $[0 . .<$ degree g])

* prod-list (map ?f $[$ degree $g . .<$ degree $f+$ degree $g]$ ) by auto
finally show ?thesis unfolding p1 p2.
qed


### 5.3 Proof of the key lemma 16.20

```
lemma common-factor-via-short:
    fixes fgu :: int poly
    defines }n\equiv\mathrm{ degree f}\mathrm{ and }k\equiv\mathrm{ degree g
    assumes n0:n>0 and k0:k>0
        and monic: monic u and deg-u: degree }u>
        and uf:poly-mod.dvdm muf and ug: poly-mod.dvdm m ug
        and short: |f\mp@subsup{|}{}{2`}k*|g\mp@subsup{|}{}{2`}n<\mp@subsup{m}{}{2}
        and m:m\geq0
    shows degree (gcd fg)>0
proof -
    interpret poly-mod m .
    have f-not0: f}\not=0\mathrm{ and g-not0: g}\not=
        using n0 k0 k-def n-def by auto
    have deg-f: degree f >0 using n0 n-def by simp
    have deg-g: degree g}>0\mathrm{ using }k0k\mathrm{ -def by simp
    obtain st where deg-s: degree s<degree g and deg-t: degree t<degree f
    and res-eq: [:resultant f g:] =s*f+t*g and s-not0:s\not=0 and t-not0: t\not=
0
    using resultant-as-nonzero-poly[OF deg-f deg-g] by auto
    have res-eq-modulo: [:resultant f g:] =m s*f+t*g using res-eq
    by simp
    have u-dvdm-res: u dvdm [:resultant f g:]
    proof (unfold res-eq, rule dvdm-add)
    show udvdm s*f
            using dvdm-factor[OF uf, of s]
            unfolding mult.commute[of fs] by auto
    show }udvdmt*
            using dvdm-factor[OF ug, of t]
            unfolding mult.commute[of g t] by auto
    qed
    have res-0-mod: resultant f g mod m}=
        by (rule monic-dvdm-constant[OF u-dvdm-res monic deg-u])
    have res0: resultant f g}=
    proof (rule mod-0-abs-less-imp-0)
        show [resultant fg=0] (mod m) using res-0-mod unfolding cong-def by
auto
    have |resultant fg| s sqrt (real-of-int (|f| |
        unfolding k-def n-def
        by (rule resultant-le-prod-sq-norm)
    also have ... < m
```

```
            by (meson m of-int-0-le-iff of-int-power-less-of-int-cancel-iff real-less-lsqrt
short)
    finally show |resultant fg| <m using of-int-less-iff by blast
    qed
    have }\neg\mathrm{ coprime fg
    by (rule resultant-zero-imp-common-factor, auto simp add: deg-f res0)
    thus ?thesis
    using res0 resultant-0-gcd by auto
qed
```


### 5.4 Properties of the computed lattice and its connection with Sylvester matrices

```
lemma factorization-lattice-as-sylvester:
    fixes p :: ' }a\mathrm{ :: semidom poly
    assumes dj:d\leqj and d: degree p=d
    shows mat-of-rows j (factorization-lattice p (j-d) m) = sylvester-mat-sub d
(j-d) p [:m:]
proof (cases p=0)
    case True
    have deg-p:d = 0 using True d by simp
    show ?thesis
        by (auto simp add: factorization-lattice-def True deg-p mat-of-rows-def d)
next
    case p0: False
    note 1 = degree-mult-eq[OF p0, of monom - -, unfolded monom-eq-0-iff, OF
one-neq-zero]
    from dj show ?thesis
        apply (cases m=0)
        apply (auto simp: mat-eq-iff d[symmetric] 1 coeff-mult-monom
    sylvester-mat-sub-index mat-of-rows-index nth-factorization-lattice vec-index-of-poly-n
    degree-monom-eq coeff-const)
        done
qed
```

context inj-comm-semiring-hom begin
lemma map-poly-hom-mult-monom [hom-distribs]:
map-poly hom ( $p$ * monom a $n$ ) = map-poly hom $p * \operatorname{monom}($ hom a) $n$
by (auto intro!: poly-eqI simp:coeff-mult-monom hom-mult)
lemma hom-vec-of-poly-n [hom-distribs]:
map-vec hom (vec-of-poly-n p n) = vec-of-poly-n (map-poly hom $p$ ) $n$
by (auto simp: vec-index-of-poly-n)
lemma hom-factorization-lattice [hom-distribs]:
shows map (map-vec hom) (factorization-lattice $u k m)=$ factorization-lattice
(map-poly hom u) $k$ (hom m)
by (auto intro!:arg-cong[of - - $\lambda p$. vec-of-poly-n p-] simp: list-eq-iff-nth-eq nth-factorization-lattice hom-vec-of-poly-n map-poly-hom-mult-monom)
end

### 5.5 Proving that factorization-lattice returns a basis of the lattice <br> context $L L L$ <br> begin <br> sublocale idom-vec $n$ TYPE (int).

lemma upper-triangular-factorization-lattice:
fixes $u::{ }^{\prime} a$ :: semidom poly and $d::$ nat
assumes $d: d \leq n$ and $d u: d=$ degree $u$
shows upper-triangular (mat-of-rows $n$ (factorization-lattice $u(n-d) k$ ))
(is upper-triangular ?M)
proof (intro upper-triangularI, unfold mat-of-rows-carrier length-factorization-lattice)
fix $i j$
assume $j i: j<i$ and $i: i<$ degree $u+(n-d)$
with $d d u$ have $j n: j<n$ by auto
show ? $M \$(i, j)=0$
proof (cases $u=0$ )
case True with ji i show ?thesis
by (auto simp: factorization-lattice-def mat-of-rows-def)
next
case False
then show ?thesis
using $d j i i$
apply (simp add: du mat-of-rows-index nth-factorization-lattice)
apply (auto simp: vec-index-of-poly-n[OF jn] degree-mult-eq degree-monom-eq) done
qed
qed
lemma factorization-lattice-diag-nonzero:
fixes $u::{ }^{\prime} a$ :: semidom poly and $d$
assumes $d$ : $d=$ degree $u$
and $d n: d \leq n$
and $u: u \neq 0$
and $m 0: k \neq 0$
and $i: i<n$
shows (factorization-lattice $u(n-d) k)!i \$ i \neq 0$
proof -
have 1: monom $\left(1::^{\prime} a\right)(n-S u c($ degree $u+i)) \neq 0$ using $m 0$ by auto
have 2: $i<$ degree $u+(n-d)$ using $i d$ by auto
let $? p=u *$ monom $1(n-$ Suc $($ degree $u+i))$

```
    have 3:i<n-degree }u\Longrightarrow\mathrm{ degree (?p) = n - Suc i
    using assms by (auto simp: degree-mult-eq[OF - 1] degree-monom-eq)
    show ?thesis
    apply (unfold nth-factorization-lattice[OF 2] vec-index-of-poly-n[OF 2])
    using assms leading-coeff-0-iff[of ?p]
    apply (cases i<n - degree u, auto simp:d 3 degree-monom-eq)
    done
qed
corollary factorization-lattice-diag-nonzero-RAT: fixes d
    assumes d=degree u
        and d\leqn
        and}u\not=
        and }k\not=
        and}i<
    shows RAT (factorization-lattice u (n-d)k)!i$ i\not=0
    using factorization-lattice-diag-nonzero[OF assms] assms
    by (auto simp: nth-factorization-lattice)
sublocale gs: vec-space TYPE(rat) n.
lemma lin-indpt-list-factorization-lattice: fixes d
    assumes d:d= degree }u\mathrm{ and dn:d sn and u:u}=0\mathrm{ and }k:k\not=
    shows gs.lin-indpt-list (RAT (factorization-lattice u (n-d) k)) (is gs.lin-indpt-list
(RAT ?vs))
proof-
    have 1: rows (mat-of-rows n (map (map-vec rat-of-int) ?vs)) = map (map-vec
rat-of-int) ?vs
        using dn d
    by (subst rows-mat-of-rows, auto dest!: subsetD[OF set-factorization-lattice-in-carrier])
    note 2 = factorization-lattice-diag-nonzero-RAT[OF d dn uk]
    show ?thesis
    apply (intro gs.upper-triangular-imp-lin-indpt-list[of mat-of-rows n (RAT ?vs),
unfolded 1])
            using assms 2 by (auto simp: diag-mat-def mat-of-rows-index hom-distribs
intro!:upper-triangular-factorization-lattice)
qed
end
```


### 5.6 Being in the lattice is being a multiple modulo

lemma (in semiring-hom) hom-poly-of-vec: map-poly hom (poly-of-vec v) = poly-of-vec (map-vec hom v)
by (auto simp add: coeff-poly-of-vec poly-eq-iff)
abbreviation of-int-vec $\equiv$ map-vec of-int
context $L L L$

## begin

lemma lincomb-to-dvd-modulo:
fixes $u d$
defines $d \equiv$ degree $u$
assumes $d: d \leq n$
and lincomb: lincomb-list $c($ factorization-lattice $u(n-d) k)=g($ is $? l=? r)$
shows poly-mod.dvdm $k u$ (poly-of-vec $g$ )
proof-
let $? S=$ sylvester-mat-sub $d(n-d) u[: k:]$
define $q$ where $q \equiv$ poly-of-vec (vec-first (vec $n c)(n-d)$ )
define $r$ where $r \equiv$ poly-of-vec (vec-last (vec $n c$ ) $d$ )
have ?l $=? S^{T} *_{v}$ vec $n c$
apply (subst lincomb-list-as-mat-mult)
using $d d$-def apply (force simp:factorization-lattice-def)
apply (fold transpose-mat-of-rows)
using $d d$-def by (simp add: factorization-lattice-as-sylvester)
also have poly-of-vec $\ldots=q * u+$ smult $k r$
apply (subst sylvester-sub-poly) using $d$-def $d q$-def $r$-def by auto
finally have $\ldots=$ poly-of-vec $g$ unfolding lincomb of-int-hom.hom-poly-of-vec by auto
then have poly-of-vec $g=q * u+$ Polynomial.smult $k r$ by auto
then have poly-mod.Mp $k$ (poly-of-vec $g)=$ poly-mod.Mp $k(q * u+$ Polynomial.smult $k r$ ) by auto
also have $\ldots=$ poly-mod.Mp $k(q * u+$ poly-mod.Mp $k$ (Polynomial.smult $k r))$
using poly-mod.plus-Mp(2) by auto
also have $\ldots=$ poly-mod.Mp $k(q * u)$
using poly-mod.plus-Mp(2) unfolding poly-mod.Mp-smult-m-0 by simp
also have $\ldots=$ poly-mod. $M p k(u * q)$ by (simp add: mult.commute)
finally show ?thesis unfolding poly-mod.dvdm-def by auto qed
lemma dvd-modulo-to-lincomb:
fixes $u::$ int poly and $d$
defines $d \equiv$ degree $u$
assumes $d: d<n$
and dvd: poly-mod.dvdm $k u$ (poly-of-vec $g$ )
and $k$-not0: $k \neq 0$
and monic-u: monic $u$
and dim-g: dim-vec $g=n$
and deg-u: degree $u>0$
shows $\exists c$. lincomb-list $c$ (factorization-lattice $u(n-d) k)=g$
proof -
interpret $p$ : poly-mod $k$.
have $u$-not $0: u \neq 0$ using monic- $u$ by auto
hence $n[\operatorname{simp}]: 0<n$ using $d$ by auto
obtain $q^{\prime} r^{\prime}$ where $g$ : poly-of-vec $g=q^{\prime} * u+$ smult $k r^{\prime}$
using $p$.dvdm-imp-div-mod $[O F d v d]$ by auto
obtain $q^{\prime \prime} r^{\prime \prime}$ where $r^{\prime}: r^{\prime}=q^{\prime \prime} * u+r^{\prime \prime}$ and deg- $r^{\prime \prime}:$ degree $r^{\prime \prime}<$ degree $u$ using monic-imp-div-mod-int-poly-degree2[OF monic-u deg-u, of r $\boldsymbol{\prime}]$ by auto
have g1: poly-of-vec $g=\left(q^{\prime}+\right.$ smult $\left.k q^{\prime \prime}\right) * u+$ smult $k r^{\prime \prime}$
unfolding $g r^{\prime}$
by (metis (no-types, lifting) combine-common-factor mult-smult-left smult-add-right)
define $q$ where $q: q=\left(q^{\prime}+\right.$ smult $\left.k q^{\prime \prime}\right)$
define $r$ where $r: r=r^{\prime \prime}$
have degree- $q: q=0 \vee$ degree $\left(q^{\prime}+\right.$ smult $\left.k q^{\prime \prime}\right)<n-d$
proof (cases $q=0$, auto, rule degree-div-mod-smult $[O F--g 1])$
show degree (poly-of-vec g) < $n$ by (rule degree-poly-of-vec-less, auto simp add: dim-g)
show degree $r^{\prime \prime}<d$ using deg- $r^{\prime \prime}$ unfolding $d$-def .
assume $q \neq 0$ thus $q^{\prime}+$ smult $k q^{\prime \prime} \neq 0$ unfolding $q$.
show $k \neq 0$ by fact
show degree $u=d$ using $d$-def by auto
qed
have g2: (vec-of-poly-n $(q * u) n)+($ vec-of-poly- $n($ smult $k r) n)=g$
proof -
have $g=v e c-o f-p o l y-n(p o l y$-of-vec $g) n$
by (rule vec-of-poly-n-poly-of-vec[symmetric], auto simp add: dim-g)
also have $\ldots=$ vec-of-poly- $n\left(\left(q^{\prime}+\right.\right.$ smult $\left.k q^{\prime \prime}\right) * u+$ smult $\left.k r^{\prime \prime}\right) n$
using $g 1$ by auto
also have $\ldots=$ vec-of-poly- $n\left(q * u+\right.$ smult $\left.k r^{\prime \prime}\right) n$ unfolding $q$ by auto
also have $\ldots=$ vec-of-poly- $n(q * u) n+$ vec-of-poly- $n\left(s m u l t ~ k r^{\prime \prime}\right) n$
by (rule vec-of-poly-n-add)
finally show ?thesis unfolding $r$ by simp
qed
let ? $c=\lambda i$. if $i<n-d$ then coeff $q(n-d-1-i)$ else coeff $r(n-$ Suc $i)$
let ?c1 $=\lambda i$. ?c $i \cdot{ }_{v}$ factorization-lattice $u(n-d) k!i$
show ?thesis
proof (rule exI[of - ?c] )
let ?part1 $=$ map $(\lambda i$. vec-of-poly-n $(u *$ monom $1 i) n)[n-d>. .0]$
let ?part2 $=m a p(\lambda i$. vec-of-poly-n $($ monom $k i) n)[d>. .0]$
have $[\operatorname{simp}]$ : dim-vec (M.sumlist (map ?c1 $[0 . .<n-d])$ ) $=n$
by (rule dim-sumlist, auto simp add: dim-factorization-lattice d-def)
have [simp]: dim-vec (M.sumlist (map ?c1 $[n-d . .<n])$ ) $=n$
by (rule dim-sumlist, insert d, auto simp add: dim-factorization-lattice d-def)
have [simp]: factorization-lattice $u(n-d) k!x \in$ carrier-vec $n$ if $x: x<n$ for
using $x$ dim-factorization-lattice-element nth-factorization-lattice $[$ of $x$ un-d]
$d$
by (auto simp: $d$-def)
have $[0 . .<$ length (factorization-lattice $u(n-d) k)]=[0 . .<n]$
using $d$ by (simp add: d-def less-imp-le-nat)
also have...$=[0 . .<n-d] @[n-d . .<n]$
by (rule upt-minus-eq-append, auto)
finally have list-rw: $[0 . .<$ length (factorization-lattice $u(n-d) k)]=[0 . .<n-$ $d]$ @ $[n-d . .<n]$.

```
    have qu1: poly-of-vec (M.sumlist (map ?c1 [0..<n - d])) = q*u
    proof -
    have poly-of-vec (M.sumlist (map ?c1 [0..<n-d])) = poly-of-vec (\bigoplus Vi\in{0..<n-d}.
?c1 i)
    by (subst sumlist-map-as-finsum, auto)
    also have ... = poly-of-vec ( }\mp@subsup{\bigoplus}{V}{}\mathrm{ íset [0..<n-d]. ?c1 i) by auto
    also have ... =sum (\lambdai. poly-of-vec (?c1 i)) (set [0..<n-d])
    by (auto simp:poly-of-vec-finsum)
    also have ... = sum (\lambdai. poly-of-vec (?c1 i)) {0..<n-d} by auto
    also have ... = q*u
    proof -
        have deg: degree (u* monom 1 ( n-Suc (d+i)))<n if i:i<n-d for
i
    proof -
        let ?m=monom (1::int) (n-Suc (d+i))
        have monom-not0: ?m }\not=0\mathrm{ using i by auto
            have deg-m: degree ?m = n - Suc (d+i) by (rule degree-monom-eq,
auto)
            have degree (u*?m) = d+(n-Suc (d+i))
                using degree-mult-eq[OF u-not0 monom-not0] d-def deg-m by auto
            also have ...<n using i by auto
            finally show ?thesis.
    qed
        have lattice-rw: factorization-lattice u(n-d) k!i = vec-of-poly-n (u*
monom 1 (n-Suc (d+i))) n
            if i:i<n - for i apply (subst nth-factorization-lattice) using i by
(auto simp:d-def)
    have q-rw: q = (\sumi=0..<n-d. (smult (coeff q ( n - Suc (d + i)))
(monom 1 (n-Suc (d+i)))))
    proof (auto simp add: poly-eq-iff coeff-sum)
        fix }
        let ?m = n-d-1-j
        let ?f = \lambdax. coeff q(n-Suc (d+x))*(if n - Suc (d+x)=j then 1
else 0)
            have set-rw: {0..<n-d}= insert ?m ({0..<n-d} - {?m}) using d by
auto
    have sum0:(\sumx\in{0..<n-d}-{?m}. ?f }x)=0\mathrm{ by (rule sum.neutral,
auto)
    have (\sumx=0..<n-d. ?f }x)=(\sumx\in\mathrm{ insert ?m ({0..<n-d} - {?m}).
?f }x
        using set-rw by presburger
        also have ... = ?f ?m}+(\sumx\in{0..<n-d}-{?m}. ?f x) by (rule
sum.insert, auto)
    also have ... = ?f ?m unfolding sum0 by auto
    also have ... = coeff q j
    proof (cases j<n-d)
        case True
        then show ?thesis by auto
    next
```

```
        case False
        have j>degree q using degree-q q False d by auto
        then show ?thesis using coeff-eq-0 by auto
    qed
    finally show coeff qj=(\sumi=0..<n-d. coeff q( n - Suc (d+i))
        * (if n - Suc (d+i)=j then 1 else 0))..
    qed
    have sum (\lambdai. poly-of-vec (?c1 i)) {0..<n-d}
    =(\sumi=0..<n-d. poly-of-vec (coeff q ( }n-Suc(d+i))\cdotv factoriza--
tion-lattice u (n-d)k!i))
        by (rule sum.cong, auto)
    also have ... = (\sumi=0..<n-d.(poly-of-vec (coeff q ( n-Suc (d+i))
        v
        by (rule sum.cong, auto simp add: lattice-rw)
    also have ... = (\sumi=0..<n-d. smult (coeff q ( n-Suc (d+i)))(u*
monom 1 ( n-Suc (d+i))))
            by (rule sum.cong, auto simp add: poly-of-vec-scalar-mult[OF deg])
    also have ... = (\sumi=0..<n-d.u*(smult (coeff q (n-Suc (d+i)))
(monom 1 (n-Suc (d+i)))))
            by auto
            also have ... =u*(\sumi=0..<n-d. (smult (coeff q ( n-Suc (d+i)))
(monom 1 (n - Suc (d + i)))))
            by (rule sum-distrib-left[symmetric])
            also have \ldots. =u*q using q-rw by auto
            also have ... = q*u by auto
            finally show ?thesis.
            qed
            finally show ?thesis .
    qed
    have qu:M.sumlist (map ?c1 [0..<n - d]) = vec-of-poly-n (q*u) n
    proof -
    have vec-of-poly-n (q*u) n = vec-of-poly-n (poly-of-vec (M.sumlist (map ?c1
[0..<n-d]))) n
            using qu1 by auto
            also have vec-of-poly-n (poly-of-vec (M.sumlist (map ?c1 [0..<n - d]))) n
            = M.sumlist (map ?c1 [0..<n-d])
            by (rule vec-of-poly-n-poly-of-vec, auto)
            finally show ?thesis ..
    qed
    have rm1: poly-of-vec (M.sumlist (map ?c1 [n-d..<n])) = smult k r
    proof -
    have poly-of-vec (M.sumlist (map ?c1 [ n-d..<n])) = poly-of-vec (\bigoplus V i\in{n-d..<n}.
?c1 i)
            by (subst sumlist-map-as-finsum, auto)
            also have ... = poly-of-vec ( }\mp@subsup{\bigoplus}{V}{}\mathrm{ i}\in\mathrm{ set [ }n-d..<n]. ?c1 i) by aut
            also have ... =sum (\lambdai.poly-of-vec (?c1 i)) {n-d..<n}
            by (auto simp: poly-of-vec-finsum)
    also have ... = smult kr
    proof -
```

have deg: degree (monom $k(n-S u c i))<n$ if $i: n-d \leq i$ and $i 2: i<n$ for
using degree-monom-le i i2
by (simp add: degree-monom-eq $k$-not0)
have lattice-rw: factorization-lattice $u(n-d) k!i=v e c-o f-p o l y-n$ (monom $k(n-$ Suc $i)) n$
if $i: n-d \leq i$ and $i 2: i<n$ for $i$
using i2 $i d d$-def
by (subst nth-factorization-lattice, auto)
have $r-r w: r=\left(\sum i \in\{n-d . .<n\}\right.$. (monom (coeff $\left.r(n-S u c i)\right)(n-S u c$
i)))
proof (auto simp add: poly-eq-iff coeff-sum)
fix $j$
show coeff $r j=\left(\sum i=n-d . .<n\right.$. if $n-S u c i=j$ then coeff $r(n-$
Suc i) else 0)
proof (cases $j<d$ )
case True
have $j$-eq: $n-S u c(n-1-j)=j$ using $d$ True by auto
let ? $i=n-1-j$
let ?f $=\lambda$ i. if $n-S u c i=j$ then coeff $r(n-S u c i)$ else 0
have sum0: sum ?f $(\{n-d . .<n\}-\{? i\})=0$ by (rule sum.neutral, auto)
have $\{n-d . .<n\}=$ insert ? $i(\{n-d . .<n\}-\{? i\})$ using True by auto
hence sum ?f $\{n-d . .<n\}=$ sum ?f (insert ?i $(\{n-d . .<n\}-\{? i\}))$
by auto
also have $\ldots=$ ?f ? $i+\operatorname{sum} ? f(\{n-d . .<n\}-\{? i\})$
by (rule sum.insert, auto)
also have $\ldots=$ coeff $r j$ unfolding sum0 $j$-eq by simp
finally show ?thesis ..
next
case False
hence $\left(\sum i=n-d . .<n\right.$. if $n-S u c i=j$ then coeff $r(n-S u c i)$ else
$0)=0$
by (intro sum.neutral ballI, insert False, simp, linarith)
also have $\ldots=$ coeff $r j$
by (rule coeff-eq- $0[$ symmetric $]$, insert False deg-r ${ }^{\prime \prime} r d$-def, auto)
finally show ?thesis ..
qed
qed
have sum ( $\lambda i$. poly-of-vec (?c1 i)) $\{n-d . .<n\}$
$=\left(\sum i \in\{n-d . .<n\}\right.$. poly-of-vec (coeff $r(n-S u c i) \cdot v$ factorization-lattice $u(n-d) k!i))$
by (rule sum.cong, auto)
also have $\ldots=\left(\sum i \in\{n-d . .<n\}\right.$. (poly-of-vec (coeff r $(n-S u c i)$
$\cdot v$ vec-of-poly-n (monom $k(n-S u c i)) n)))$
by (rule sum.cong, auto simp add: lattice-rw)
also have $\ldots=\left(\sum i \in\{n-d . .<n\}\right.$. smult (coeff $r(n-$ Suc $\left.i)\right)($ monom $k$ ( $n-S u c i)$ ))
by (rule sum.cong, auto simp add: poly-of-vec-scalar-mult[OF deg])
also have $\ldots=\left(\sum i \in\{n-d . .<n\}\right.$. smult $k$ (monom (coeff $\left.r(n-S u c i)\right)$

```
(n-Suc i)))
            by (rule sum.cong, auto simp add: smult-monom smult-sum2)
            also have ... = smult k (\sumi\in{n-d..<n}.(monom (coeff r (n-Suci))
(n-Suc i)))
            by (simp add: smult-sum2)
            also have ... = smult kr using r-rw by auto
            finally show ?thesis .
        qed
        finally show ?thesis .
    qed
    have rm: (M.sumlist (map ?c1 [n-d..<n])) = vec-of-poly-n (smult k r) n
    proof -
            have vec-of-poly-n (smult kr) n
                =vec-of-poly-n (poly-of-vec (M.sumlist (map ?c1 [n-d..<n]))) n
                using rm1 by auto
            also have vec-of-poly-n (poly-of-vec (M.sumlist (map ?c1 [n-d..<n]))) n
                = M.sumlist (map ?c1 [ n-d..<n])
                by (rule vec-of-poly-n-poly-of-vec, auto)
            finally show ?thesis ..
    qed
    have lincomb-list ?c (factorization-lattice u (n-d)k)= M.sumlist (map ?c1
([0..<n-d]@ [n-d..<n]))
            unfolding lincomb-list-def list-rw by auto
    also have ... = M.sumlist (map ?c1 [0..<n - d] @ map ?c1 [n-d..<n]) by
auto
    also have ... = M.sumlist (map ?c1 [0..<n - d]) + M.sumlist (map ?c1
[n-d..<n])
    using d by (auto simp add: d-def nth-factorization-lattice intro!: M.sumlist-append)
    also have ... = vec-of-poly-n (q*u) n + vec-of-poly-n (smult k r) n
            unfolding qu rm by auto
    also have ... = g using g2 by simp
    finally show lincomb-list ?c (factorization-lattice u(n-d)k)=g.
    qed
qed
The factorization lattice precisely characterises the polynomials of a certain degree which divide \(u\) modulo \(M\).
lemma factorization-lattice: fixes \(M\) assumes
deg-u: degree \(u \neq 0\) and \(M: M \neq 0\)
shows degree \(u \leq n \Longrightarrow n \neq 0 \Longrightarrow f \in\) poly-of-vec'lattice-of (factorization-lattice
\(u(n-\) degree \(u) M) \Longrightarrow\)
degree \(f<n \wedge\) poly-mod.dvdm Muf
monic \(u \Longrightarrow\) degree \(u<n \Longrightarrow\)
degree \(f<n \Longrightarrow\) poly-mod.dvdm \(M u f \Longrightarrow f \in\) poly-of-vec 'lattice-of (factorization-lattice
\(u(n-\) degree \(u) M)\)
proof -
from deg-u have deg-u: degree \(u>0\) by auto
let ? \(L=\) factorization-lattice \(u(n-\) degree \(u) M\)
\{
```

```
    assume deg:degree f<n and dvd: poly-mod.dvdm Muf and mon: monic u
        and deg-u-lt: degree }u<
    define fv where fv = vec n (\lambdai.(coeff f (n-Suci)))
    have f:f= poly-of-vec fv unfolding fv-def poly-of-vec-def Let-def using deg
        by (auto intro!: poly-eqI coeff-eq-0 simp: coeff-sum)
    have dim-fv: dim-vec fv = n unfolding fv-def by simp
    from dvd-modulo-to-lincomb[OF deg-u-lt - M mon-deg-u(1), of fv, folded f,
OF dvd dim-fv]
    obtain c where gv: fv = lincomb-list c ?L by auto
    have fv \in lattice-of ?L unfolding gv lattice-is-span by (auto simp: in-span-listI)
    thus f}\in\mathrm{ poly-of-vec` lattice-of ?L unfolding f by auto
}
moreover
{
    assume f}\in\mathrm{ poly-of-vec 'lattice-of ?L and deg-u: degree }u\leqn\mathrm{ and n:n}=
    then obtain fv where f:f= poly-of-vec fv and fv: fv\inlattice-of ?L by auto
    from in-span-listE[OF fv[unfolded lattice-is-span]]
    obtain c}\mathrm{ where fv: fv = lincomb-list c ?L by auto
    from lincomb-to-dvd-modulo[OF-fv[symmetric]] deg-u f
    have dvd: poly-mod.dvdm Muf by auto
    have set ?L \subseteq carrier-vec n unfolding factorization-lattice-def using deg-u
by auto
    hence fv \in carrier-vec n unfolding fv by (metis lincomb-list-carrier)
    hence degree f<n unfolding f using degree-poly-of-vec-less[of fv n] using n
by auto
    with dvd show degree f<n\wedge poly-mod.dvdm Muf by auto
    }
qed
end
```


### 5.7 Soundness of the LLL factorization algorithm

lemma LLL-short-polynomial: assumes deg-u-0: degree $u \neq 0$ and deg-le: degree $u \leq n$
and $p l 1: p l>1$
and monic: monic $u$
shows degree (LLL-short-polynomial pl $n u)<n$
and LLL-short-polynomial pl n $u \neq 0$
and poly-mod.dvdm pl u(LLL-short-polynomial pl nu)
and degree $u<n \Longrightarrow f \neq 0 \Longrightarrow$
poly-mod.dvdm pl $u f \Longrightarrow$ degree $f<n \Longrightarrow \|$ LLL-short-polynomial pl $n u \|^{2} \leq$
$2^{\wedge}(n-1) *\|f\|^{2}$
proof -
interpret poly-mod-2 pl
by (unfold-locales, insert pl1, auto)
from $p l 1$ have $p l 0: p l \neq 0$ by auto
let $? d=$ degree $u$
let $? u=M p u$
let $? i u=i n v-M p ? u$
from $M p-i n v-M p-i d[o f ? u]$ have ? $i u=m ? u$.
also have $\ldots=m u$ by simp
finally have $i u-u$ : ? $i u=m u$ by simp
have degu [simp]: degree ? $u=$ degree $u$ using monic by simp
have mon: monic ?u using monic by (rule monic-Mp)
have degree ? iu $=$ degree ? u unfolding inv-Mp-def
by (rule degree-map-poly, unfold mon, insert mon pl1, auto simp: inv-M-def)
with degu have deg-iu: degree ? iu $=$ degree $u$ by simp
have mon-iu: monic ? iu unfolding deg-iu unfolding inv-Mp-def Mp-def inv-M-def M-def
by (insert pl1, auto simp: coeff-map-poly monic)
let $? L=$ factorization-lattice ? iu $(n-$ ? $d) p l$
let ? $s v=$ short-vector-hybrid 2 ? $L$
from deg-u-0 deg-le have $n: n \neq 0$ by auto
from deg-u-0 have $u 0: u \neq 0$ by auto
have id: LLL-short-polynomial pl $n u=$ poly-of-vec ?sv
unfolding LLL-short-polynomial-def by blast
have $i d^{\prime}: \|$ ?sv $\left\|^{2}=\right\| L L L$-short-polynomial pl $n u \|^{2}$ unfolding id by simp
interpret vec-module TYPE (int) $n$.
interpret $L: L L L n$ ? ? $L 2$.
from deg-le deg-iu have deg-iu-le: degree ? $i u \leq n$ by simp
have len: length ? $L=n$
unfolding factorization-lattice-def using deg-le deg-iu by auto
from deg-u-0 deg-iu have deg-iu0: degree ?iu $\neq 0$ by auto
hence $i u 0: ~ ? i u \neq 0$ by auto
from L.lin-indpt-list-factorization-lattice[OF refl deg-iu-le iu0 pl0]
have $*: 4 / 3 \leq(2$ :: rat) L.gs.lin-indpt-list (L.RAT ?L) by (auto simp: deg-iu)
interpret $L$ : LLL-with-assms $n n$ ? $L 2$
by (unfold-locales, insert *, auto simp: deg-iu deg-le)
note short $=L . s h o r t-v e c t o r-h y b r i d\left[O F\right.$ refl $n$, unfolded id ${ }^{\prime}$ L.L-def]
from short(2) have mem: LLL-short-polynomial pl $n u \in$ poly-of-vec' lattice-of ? $L$
unfolding id by auto
note fact $=$ L.factorization-lattice(1)[OF deg-iu0 pl0 deg-iu-le n, unfolded deg-iu, OF mem]
show degree (LLL-short-polynomial pl $n u)<n$ using fact by auto
from fact have ?iu dvdm (LLL-short-polynomial pl $n u$ ) by auto
then obtain $h$ where LLL-short-polynomial pl $n u=m$ ? iu $* h$ unfolding
dvdm-def by auto
also have $? i u * h=m M p$ ? $i u * h$ unfolding $m u l t-M p$ by simp
also have $M p$ ? iu*h=mu*h unfolding iu-u unfolding mult-Mp by simp
finally show $u$ dvdm (LLL-short-polynomial pl $n u$ ) unfolding dvdm-def by auto
from short have sv1: ?sv $\in$ carrier-vec $n$ by auto
from short have ? $s v \neq 0_{v} j$ for $j$ by auto
thus LLL-short-polynomial pl $n u \neq 0$ unfolding id by simp
assume degu: degree $u<n$ and $d v d: u d v d m f$
and degf: degree $f<n$ and $f 0: f \neq 0$
from dvd obtain $h$ where $f=m u * h$ unfolding dvdm-def by auto
also have $u * h=m M p u * h$ unfolding mult-Mp by simp
also have $M p u * h=m M p$ ? $i u * h$ unfolding $i u-u$ by simp
also have $M p$ ? iu * $h=m$ ? $i u * h$ unfolding mult- $M p$ by simp
finally have $d v d$ : ?iu $d v d m f$ unfolding $d v d m$-def by auto
from degu deg-iu have deg-iun: degree ? iu $<n$ by auto
from L.factorization-lattice(2)[OF deg-iu0 pl0 mon-iu deg-iun degf dvd]
have $f \in$ poly-of-vec' lattice-of ? L using deg-iu by auto
then obtain $f v$ where $f: f=$ poly-of-vec $f v$ and $f v: f v \in l a t t i c e-o f ? L$ by auto
have norm: $\|f v\|^{2}=\|f\|^{2}$ unfolding $f$ by simp
have $f v 0: f v \neq O_{v} n$ using $f 0$ unfolding $f$ by auto
with $f v$ have $f v L: f v \in$ lattice-of ? $L-\left\{0_{v} n\right\}$ by auto
from short(3)[OF this, unfolded norm]
have rat-of-int $\| L L L$-short-polynomial pl $\left.n u\left\|^{2} \leq \operatorname{rat-of-int(2}{ }^{\wedge}(n-1) *\right\| f \|^{2}\right)$
by simp
thus $\| L L L$-short-polynomial pl $n u\left\|^{2} \leq 2^{\wedge}(n-1) *\right\| f \|^{2}$ by linarith
qed
context $L L L$-implementation
begin
lemma LLL-reconstruction: assumes $L L L$-reconstruction $f u s=f s$
and degree $f \neq 0$
and poly-mod.unique-factorization-m pl f(lead-coeff f, mset us)
and $f d v d F$
and $\bigwedge u i . u i \in$ set $u s \Longrightarrow$ poly-mod. Mp pl $u i=u i$
and $F 0: F \neq 0$
and cop: coprime (lead-coeff F) $p$
and sf: poly-mod.square-free-m pF
and $p l 1: p l>1$
and $p l p: p l=p^{\wedge} l$
and $p$ : prime $p$
and large: ${ }^{2}(5 *($ degree $F-1) *($ degree $F-1)) *\|F\|^{2 \uparrow}(2 *($ degree $F-$
1)) $<p l^{2}$
shows $f=$ prod-list fs $\wedge\left(\forall f i \in\right.$ set fs. irreducible $\left._{d} f i\right)$
proof -
interpret $p$ : poly-mod-prime $p$ by (standard, rule $p$ )
interpret pl: poly-mod-2 pl by (standard, rule pl1)
from pl1 plp have $l 0: l \neq 0$ by (cases $l$, auto)
show ?thesis using assms (1-5)
proof (induct $f$ us arbitrary: fs rule: LLL-reconstruction.induct)
case ( $1 \mathrm{f} u \mathrm{f}$ s)
define $u$ where $u=$ choose- $u$ us
define $g$ where $g=L L L$-short-polynomial pl (degree f) $u$
define $k$ where $k=g c d f g$
note res $=1$ (3)
note $\operatorname{degf}=1$ (4)
note $u f=1(5)$
note $f F=1(6)$
note norm $=1$ (7)
note to-fact $=$ pl.unique-factorization-m-imp-factorization
note fact $=$ to-fact $[O F u f]$
have mon-gs: ui $\in$ set us $\Longrightarrow$ monic ui for ui using norm fact
unfolding pl.factorization-m-def by auto
from p.coprime-lead-coeff-factor[OF p.prime] fF cop
have cop: coprime (lead-coeff f) $p$ unfolding dvd-def by blast
have plf0: pl. $M p f \neq 0$
using fact pl.factorization-m-lead-coeff pl.unique-factorization-m-zero uf by
fastforce
have degree $f=$ pl.degree-m $f$
by (rule sym, rule poly-mod.degree-m-eq[OF - pl.m1],
insert cop p, simp add: l0 p.coprime-exp-mod plp)
also have $\ldots=$ sum-mset (image-mset pl.degree-m (mset us))
unfolding pl.factorization-m-degree[OF fact plf0]..
also have ... = sum-list (map pl.degree-m us)
unfolding sum-mset-sum-list[symmetric] by auto
also have $\ldots=$ sum-list (map degree us)
by (rule arg-cong[OF map-cong, OF refl], rule pl.monic-degree-m, insert mon-gs, auto)
finally have degf-gs: degree $f=$ sum-list (map degree us) by auto
hence gs: us $\neq[]$ using degf by (cases us, auto)
from choose-u-member $[O F g s]$ have $u$-gs: $u \in$ set us unfolding $u$-def by auto
from fact $u$-gs have irred: pl.irreducible ${ }_{d}-m u$ unfolding pl.factorization-m-def by auto
hence deg-u: degree $u \neq 0$ unfolding pl.irreducible $_{d}$-m-def norm $[O F u-g s]$ by auto
have deg-uf: degree $u \leq$ degree $f$ unfolding degf-gs using split-list[OF u-gs] by auto
from mon-gs[OF $u-g s]$ have mon-u: monic $u$ and $u 0: u \neq 0$ by auto
have $f 0: f \neq 0$ using degf by auto
from norm have norm': image-mset pl.Mp (mset us) = mset us by (induct us, auto)
have $p l 0: p l \neq 0$ using $p l 1$ by auto
note short-main $=L L L$-short-polynomial $[O F$ deg-u deg-uf pl1 mon-u]
from short-main(1-2)[folded $g$-def]
have degree $k<$ degree $f$ unfolding $k$-def
by (smt Suc-leI Suc-less-eq degree-gcd1 gcd.commute le-imp-less-Suc le-trans)
hence deg-fk: (degree $k=0 \vee$ degree $f \leq$ degree $k)=($ degree $k=0)$ by auto note res $=$ res[unfolded LLL-reconstruction.simps[off us] Let-def, folded u-def,
folded $g$-def, folded $k$-def, unfolded deg-fk]
show ? case
proof (cases degree $k=0$ )
case True
with res have $f s: f s=[f]$ by auto
from sf $f F$ have $s f$ : p.square-free-m $f$
using p.square-free-m-factor (1)[of f] unfolding dvd-def by auto
have irr: irreducible $_{d} f$

```
    proof (rule ccontr)
    assume }\neg\mp@subsup{\mathrm{ irreducible }}{d}{}
    from reducible ed E[OF this] degf obtain f1 f2 where
        f:f=f1*f2 and
        deg12: degree f1 }\not=0\mathrm{ degree f2 }\not=0\mathrm{ degree f1 < degree f degree f2 < degree
f
    by (simp, metis)
    from pl.unique-factorization-m-factor[OF p uf[unfolded f], folded f, OF cop
sf l0 plp]
    obtain us1 us2 where
        uf12: pl.unique-factorization-m f1 (lead-coeff f1, us1)
            pl.unique-factorization-m f2 (lead-coeff f2, us2)
        and gs: mset us = us1 + us2
        and norm12: image-mset pl.Mp us2 = us2 image-mset pl.Mp us1 =us1
        unfolding pl.Mf-def norm' split by (auto simp: pl.Mf-def)
    note norm-u = norm[OF u-gs]
    from u-gs have u-gs':u\in# mset us by auto
    with pl.factorization-m-mem-dvdm[OF fact, of u]
    have u-f: pl.dvdm uf by auto
    from u-gs'[unfolded gs] have u\in#us1 \vee u\in# us2 by auto
    with pl.factorization-m-mem-dvdm[OF to-fact[OF uf12(1)], of u]
        pl.factorization-m-mem-dvdm[OF to-fact[OF uf12(2)], of u]
    have pl.dvdm u f1 \vee pl.dvdm u f2 unfolding norm12 norm-u by auto
    from this have }\exists\mathrm{ f1 f2. f =f1*f2 ^
        degree f1 }\not=0\wedge\mathrm{ degree f2 }\not=0\wedge\mathrm{ degree f1 < degree f }\wedge\mathrm{ degree f2 < degree
f^
        pl.dvdm u f1
    proof
        assume pl.dvdm u f1 thus ?thesis using f deg12 by auto
    next
        from f have f:f=f2 * f1 by auto
        assume pl.dvdm u f2 thus ?thesis using f deg12 by auto
    qed
    then obtain f1 f2 where prod: f=f1 * f2
        and deg: degree f1 }\not=0\mathrm{ degree f2 }\not=0\mathrm{ degree f1 < degree f degree f2 <
degree f
        and uf1:pl.dvdm u f1 by auto
    from pl.unique-factorization-m-factor[OF p uf[unfolded prod], folded prod,
OF cop sf l0 plp]
    obtain us1 where fact-f1: pl.unique-factorization-m f1 (lead-coeff f1, us1)
by auto
    have plf1: pl.Mp f1 f=0
        using to-fact[OF fact-f1] pl.factorization-m-lead-coeff
            pl.unique-factorization-m-zero fact-f1 by fastforce
    have degree }u\leq\mathrm{ degree f1
        by (rule pl.dvdm-degree[OF mon-u uf1 plf1])
    with deg have deg-uf: degree u < degree f by auto
    have pl0:pl\not=0 using pl.m1 plp by linarith
    let ?n = degree f
```

let $? n 1=$ degree $f 1$
let $? d=$ degree $u$
from prod $f F$ have $f 1 F$ ：$f 1$ dvd $F$ unfolding $d v d$－def by auto
from deg－uf have $d e g-u f^{\prime}: ? d \leq ? n$ by auto
from deg have f1－0：$f 1 \neq 0$ by auto
have $u g$ ：pl．dvdm $u g$ using short－main（3）unfolding $g$－def．
have $g 0: g \neq 0$ using short－main（2）unfolding $g$－def．
have deg－gf：degree $g<$ degree $f$ using short－main（1）unfolding $g$－def ．
let $? N=$ degree $F$
from $f F$ prod have $f 1 F$ ：$f 1$ dvd $F$ unfolding dvd－def by auto
have $\|g\|^{2} \leq 2^{\wedge}(? n-1) *\|f 1\|^{2}$ unfolding $g$－def
by（rule short－main（4）［OF deg－uf－uf1］，insert deg，auto）
also have $\ldots \leq 2^{\wedge}(? n-1) *\left(2^{\wedge}\left(2 *\right.\right.$ degree f1）$\left.*\|F\|^{2}\right)$
by（rule mult－left－mono［OF sq－norm－factor－bound［OF f1F F0］］，simp）
also have $\ldots=2^{\wedge}((? n-1)+2 *$ degree $f 1) *\|F\|^{2}$
unfolding power－add by simp
also have $\ldots \leq 2^{\wedge}((? n-1)+2 *(? n-1)) *\|F\|^{2}$
by（rule mult－right－mono，insert $\operatorname{deg}(3)$ ，auto）
also have $\ldots=2^{\wedge}(3 *(? n-1)) *\|F\|^{2}$ by $\operatorname{simp}$
finally have ineq－g：$\|g\|^{2} \leq 2^{\wedge}(3 *(? n-1)) *\|F\|^{2}$ ．
from power－mono［OF this，of ？n1］
have ineq1：$\|g\|^{2}$＾？$n 1 \leq\left(2^{\wedge}(3 *(? n-1)) *\|F\|^{2}\right)^{\wedge} ? n 1$ by auto
from $F 0$ have normF：$\|F\|^{2} \geq 1$ using sq－norm－poly－pos $[o f ~ F]$ by presburger
from $g 0$ have normg：$\|g\|^{2} \geq 1$ using sq－norm－poly－pos $[$ of $g]$ by presburger
from $f 0$ have normf：$\|f\|^{2} \geq 1$ using sq－norm－poly－pos $[o f f]$ by presburger
from f1－0 have normf1：$\|f 1\|^{2} \geq 1$ using sq－norm－poly－pos［of f1］by presburger
from power－mono［OF sq－norm－factor－bound［OF f1F F0］，of degree g］
have ineq2：$\|f 1\|^{2}$ へ degree $g \leq\left(\right.$ 2 $\left.^{\wedge}(2 * ? n 1) *\|F\|^{2}\right)$ へ degree $g$ by auto
also have $\ldots \leq\left(2^{\wedge}(2 * ? n 1) *\|F\|^{2}\right) \wedge(? n-1)$
by（rule pow－mono－exp，insert deg－gf normF，auto）
finally have ineq2：$\|f 1\|^{2}$ へ degree $g \leq\left(2^{\wedge}(2 * ? n 1) *\|F\|^{2}\right) \wedge(? n-1)$ ．
have $n N$ ：？$n \leq ? N$ using $f F F 0$ by（metis dvd－imp－degree－le）
from deg $n N$ have $n 1 N$ ：？$n 1 \leq ? N-1$ by auto
have $\|f 1\|^{2}$＾degree $g *\|g\|^{2}$＾？$n 1 \leq$
$\left(2^{\wedge}(2 * ? n 1) *\|F\|^{2}\right) \wedge(? n-1) *\left(2^{\wedge}(3 *(? n-1)) *\|F\|^{2}\right)^{\wedge} ? n 1$
by（rule mult－mono［OF ineq2 ineq1］，force + ）
also have $\ldots \leq\left(2^{\wedge}(2 *(? N-1)) *\|F\|^{2}\right) \wedge(? N-1) *$
$\left(2^{\wedge}(3 *(? N-1)) *\|F\|^{2}\right)^{\wedge}(? N-1)$
by（rule mult－mono［OF power－both－mono［OF－mult－mono］
power－both－mono］，insert normF $n 1 N n N$ ，auto intro：power－both－mono mult－mono）
also have $\ldots=2^{\wedge}(2 *(? N-1) *(? N-1)+3 *(? N-1) *(? N-$
1））

$$
*\left(\|F\|^{2}\right) \wedge((? N-1)+(? N-1))
$$

unfolding power－mult－distrib power－add power－mult by simp
also have $2 *(? N-1) *(? N-1)+3 *(? N-1) *(? N-1)=5 *$
$(? N-1) *(? N-1)$ by $\operatorname{simp}$
also have ？$N-1+(? N-1)=2 *(? N-1)$ by $\operatorname{simp}$
also have $\mathcal{Z}^{\wedge}(5 *(? N-1) *(? N-1)) *\|F\|^{2}(2 *(? N-1))<p l$ ^2 by (rule large)
finally have large: $\|f 1\|^{2} \uparrow$ degree $g *\|g\|^{2} \uparrow$ degree f1<pl${ }^{2}$.
have deg-ug: degree $u \leq$ degree $g$
proof (rule pl.dvdm-degree[OF mon-u ug], standard)
assume pl.Mp $g=0$
from arg-cong $[$ OF this, of $\lambda p$. coeff $p$ (degree $g)$ ]
have $p l . M($ coeff $g($ degree $g))=0$ by (auto simp: pl.Mp-def coeff-map-poly)
from this[unfolded pl.M-def] obtain $c$ where $l g$ : lead-coeff $g=p l * c$ by

## auto

with $g 0$ have $c 0: c \neq 0$ by auto
hence $p l \wedge 2 \leq(l e a d$-coeff $g)$ ^2 unfolding $l g$ abs-le-square-iff[symmetric]
by (rule aux-abs-int)
also have $\ldots \leq\|g\|^{2} \wedge 1$ using coeff-le-sq-norm $[$ of $g]$ by auto
also have $\ldots \leq\|g\|^{2}$ - degree $f 1$
by (rule pow-mono-exp, insert deg normg, auto)
also have $\ldots=1 * \ldots$ by $\operatorname{simp}$
also have $\ldots \leq\|f 1\|^{2} \uparrow$ degree $g *\|g\|^{2}$ degree $f 1$
by (rule mult-right-mono, insert normf1, auto)
also have $\ldots<p l^{2}$ by (rule large)
finally show False by auto
qed
from deg deg-u deg-ug have degree $f 1>0$ degree $g>0$ by auto
from common-factor-via-short[OF this mon-u-uf1 ug large] deg-u pl.m1
have $0<$ degree $(g c d f 1 g)$ by auto
moreover from True[unfolded $k$-def] have degree ( $g c d f g$ ) $=0$.
moreover have $d v d: g c d f 1 g d v d g c d f g$ using $f 0$ unfolding prod by simp
ultimately show False using divides-degree[OF dvd] using f0 by simp
qed
show ?thesis unfolding $f s$ using irr by auto
next
case False
define $f 1$ where $f 1=f$ div $k$
have $f: f=f 1 * k$ unfolding $f 1$-def $k$-def by auto
with arg-cong[OF this, of degree] f0 have deg-f1k: degree $f=$ degree $f 1+$ degree $k$
by (auto simp: degree-mult-eq)
from $f f F$ have $d v d: f 1$ dvd $F k$ dvd $F$ unfolding dvd-def by auto
obtain gs1 gs2 where part: List.partition ( $\lambda$ gi. p.dvdm gif1) us $=(g s 1, g s 2$ )
by force
note $I H=1(1-2)[$ OF refl $u$-def $g$-def $k$-def refl, unfolded deg-fk, OF False f1-def part[symmetric] refl]
obtain $f s 1$ where $f s 1$ : LLL-reconstruction $f 1$ gs1 $=f s 1$ by auto
obtain $f_{s 2} 2$ where $f_{s} 2$ : LLL-reconstruction $k$ gs2 $=f_{s 2} 2$ by auto
from False res[folded f1-def, unfolded part split fs1 fs2]
have $f s$ : $f s=f s 1$ @ fs2 by auto
from short-main(1)
have deg-gf: degree $g<$ degree $f$ unfolding $g$-def by auto
from short-main(2)
have $g 0: g \neq 0$ unfolding $g$-def by auto
have deg-kg: degree $k \leq$ degree $g$ unfolding $k$-def gcd.commute[of f $g$ ]
by (rule degree-gcd1[OF gO])
from deg-gf deg-kg have deg-kf: degree $k<$ degree $f$ by auto
with deg-f1k have deg-f1: degree $f 1 \neq 0$ by auto
have sf-f: p.square-free-m $f$ using sf fF p.square-free-m-factor unfolding $d v d-d e f$ by blast
from $p$.unique-factorization-m-factor-partition[OF l0 uf[unfolded plp]f cop sf-f part]
have uf: pl.unique-factorization-m f1 (lead-coeff f1, mset gs1)
pl.unique-factorization-m $k$ (lead-coeff $k$, mset gs2) by (auto simp: plp)
have set us $=$ set gs1 $\cup$ set gs2 using part by auto
with norm have norm-12: gi $\operatorname{set} g s 1 \vee g i \in \operatorname{set} g s 2 \Longrightarrow p l . M p g i=g i$ for gi by auto
note $I H 1=I H(1)[O F f s 1$ deg-f1 uf(1) dvd(1) norm-12]
note $I H 2=I H(2)[O F$ fs2 False uf(2) dvd(2) norm-12]
show ?thesis unfolding $f s f$ using IH1 IH2 by auto
qed
qed
qed
lemma LLL-many-reconstruction: assumes LLL-many-reconstruction $f u s=f s$
and degree $f \neq 0$
and poly-mod.unique-factorization-m plf(lead-coeff $f$, mset us)
and $f d v d F$
and $\bigwedge u i . u i \in$ set $u s \Longrightarrow$ poly-mod. $M p$ pl $u i=u i$
and $F 0: F \neq 0$
and cop: coprime (lead-coeff F) p
and sf: poly-mod.square-free-m p F
and $p l 1: p l>1$
and $p l p: p l=p^{\wedge} l$
and $p$ : prime $p$
and large: $\mathfrak{Z}^{\wedge}(5 *($ degree $F$ div 2 $) *($ degree $F$ div 2$)) *\|F\|^{2}$ 〔 $\mathcal{Z} *($ degree $F$ div 2)) $<p l^{2}$
shows $f=$ prod-list $f s \wedge\left(\forall f i \in\right.$ set fs. irreducible $\left.{ }_{d} f\right)$
proof -
interpret $p$ : poly-mod-prime $p$ by (standard, rule $p$ )
interpret pl: poly-mod-2 pl by (standard, rule pl1)
from pl1 plp have $l 0: l \neq 0$ by (cases $l$, auto)
show ?thesis using assms (1-5)
proof (induct $f$ us arbitrary: $f s$ rule: LLL-many-reconstruction.induct)
case ( $1 f u s f s$ )
note res $=1$ (3)
note $\operatorname{degf}=1$ (4)
note $u f=1$ (5)
note $f F=1(6)$
note norm $=1$ (7)
note to-fact $=$ pl.unique-factorization-m-imp-factorization
note fact $=$ to-fact $[O F u f]$
have mon-gs: $u i \in$ set $u s \Longrightarrow$ monic ui for $u i$ using norm fact unfolding pl.factorization-m-def by auto
from $p$.coprime-lead-coeff-factor[OF p.prime] fF cop
have cop: coprime (lead-coeff f) $p$ unfolding dvd-def by blast
have plf0: pl. $M p f \neq 0$
using fact pl.factorization-m-lead-coeff pl.unique-factorization-m-zero uf by
fastforce
have degree $f=$ pl.degree-m $f$
by (rule sym, rule poly-mod.degree-m-eq[OF - pl.m1],
insert cop p, simp add: l0 p.coprime-exp-mod plp)
also have $\ldots=$ sum-mset (image-mset pl.degree-m (mset us))
unfolding pl.factorization-m-degree[OF fact plf0] ..
also have ... = sum-list (map pl.degree-m us)
unfolding sum-mset-sum-list[symmetric] by auto
also have $\ldots=$ sum-list (map degree us)
by (rule arg-cong[OF map-cong, OF refl], rule pl.monic-degree-m, insert mon-gs, auto)
finally have degf-gs: degree $f=$ sum-list (map degree us) by auto
hence gs: us $\neq[]$ using degf by (cases us, auto)
from 1 (4) have $f 0: f \neq 0$ and $d f 0$ : degree $f \neq 0$ by auto
from norm have norm': image-mset pl.Mp (mset us) $=$ mset us by (induct us, auto)
have $p l 0: p l \neq 0$ using $p l 1$ by auto
let $? \mathrm{D} 2=$ degree $F$ div 2
let $? d 2=$ degree $f$ div 2
define $g g$ where $g g=L L L$-short-polynomial pl (Suc ?d2)
let ?us $=$ filter $(\lambda u$. degree $u \leq$ ?d2) us
note res $=$ res[unfolded LLL-many-reconstruction.simps[offus], unfolded Let-def, folded gg-def]
let ?fD-opt $=$ find-map-filter $(\lambda u . g c d f(g g u))$
( $\lambda$ f2. $0<$ degree f2 $\wedge$ degree f2 $<$ degree $f$ ) ? us
show ?case
proof (cases ?f2-opt)
case (Some f2)
from find-map-filter-Some[OF this]
obtain $g$ where deg-f2: degree f2 $\neq 0$ degree $f 2<$ degree $f$
and $d v d$ : $f 2 d v d f$ and $g c d: f 2=g c d f g$ by auto
note res $=$ res[unfolded Some option.simps]
define $f 1$ where $f 1=f$ div $f 2$
have $f: f=f 1 * f 2$ unfolding $f 1$-def using dvd by auto
with arg-cong[OF this, of degree] f0 have deg-sum: degree $f=$ degree $f 1+$ degree f2
by (auto simp: degree-mult-eq)
with deg-f2 have deg-f1: degree $f 1 \neq 0$ degree $f 1<$ degree $f$ by auto
from $f f F$ have $d v d: f 1$ dvd $F f 2$ dvd $F$ unfolding dvd-def by auto
obtain gs1 gs2 where part: List.partition ( $\lambda$ gi. p.dvdm gif1) us $=(g s 1, g s 2)$
by force
note $I H=1(1-2)[$ OF refl refl refl, unfolded Let-def, folded gg-def, OF Some f1-def part[symmetric] refl]
obtain $f_{s 1} 1$ where $f_{s} 1$ : LLL-many-reconstruction f1 gs1 $=f s 1$ by blast
obtain $f s 2$ where $f_{s} 2$ : LLL-many-reconstruction f2 gs2 $=f s 2$ by blast
from res[folded f1-def, unfolded part split fs1 fs2]
have $f_{s}$ : $f_{s}=f_{s} 1$ @ $f_{s 2}$ by auto
have sf-f: p.square-free-m $f$ using sf fF p.square-free-m-factor unfolding $d v d-d e f$ by blast
from p.unique-factorization-m-factor-partition[OF l0 uf[unfolded plp] f cop sf-f part]
have uf: pl.unique-factorization-m f1 (lead-coeff f1, mset gs1)
pl.unique-factorization-m f2 (lead-coeff f2, mset gs2) by (auto simp: plp)
have set us $=$ set gs $1 \cup$ set gs2 using part by auto
with norm have norm-12: gi $\operatorname{set} g s 1 \vee g i \in \operatorname{set} g s 2 \Longrightarrow p l . M p g i=g i$ for $g i$ by auto
note $I H 1=I H(1)[O F f s 1$ deg-f1(1) uf(1) dvd(1) norm-12]
note $I H 2=I H(2)[O F f s 2$ deg-f2(1) uf(2) $\operatorname{dvd}(2)$ norm-12 $]$
show ?thesis unfolding fs $f$ using IH1 IH2 by auto

## next

case None
from res[unfolded None option.simps] have $f s-f: f s=[f]$ by simp
from sf $f F$ have $s f$ : p.square-free-m $f$
using p.square-free-m-factor ( 1 )[of f] unfolding dvd-def by auto
have irreducible $_{d} f$
proof (rule ccontr)
assume $\neg$ irreducible $_{d} f$
from reducible $_{d} E[O F$ this] degf obtain $f 1$ f2 where
$f: f=f 1 * f 2$ and
deg12: degree $f 1 \neq 0$ degree f2 $\neq 0$ degree $f 1<$ degree $f$ degree f2 $<$ degree
$f$
by (simp, metis)
from f0 have degree $f=$ degree $f 1+$ degree f2 unfolding $f$
by (auto simp: degree-mult-eq)
hence degree $f 1 \leq$ degree $f$ div $2 \vee$ degree f2 $\leq$ degree $f$ div 2 by auto
then obtain $f 1$ f2 where
$f: f=f 1 * f 2$ and
deg12: degree $f 1 \neq 0$ degree f2 $\neq 0$ degree $f 1 \leq$ degree $f$ div 2 degree $f 2<$ degree $f$
proof (standard, goal-cases)
case 1
from 1 (1)[off1 f2] 1(2) $f$ deg12 show ?thesis by auto
next
case 2
from 2(1)[of f2 f1] 2(2) $f$ deg12 show ?thesis by auto
qed
from f0 $f$ have $f 10: f 1 \neq 0$ by auto
from sf $f$ have sf1: p.square-free-m f1
using $p$.square-free-m-factor (1)[of f1] by auto
from p.coprime-lead-coeff-factor[OF p.prime cop[unfolded f]]
ave cop1: coprime (lead-coeff f1) p by auto
have deg-m1: pl.degree-m $f 1=$ degree f1
by (rule poly-mod.degree-m-eq[OF - pl.m1], insert cop1 $p$, simp add: l0 p.coprime-exp-mod plp)
from pl.unique-factorization-m-factor[OF puf[unfolded f], folded f, OF cop sf 10 plp ]
obtain us1 us2 where
uf12: pl.unique-factorization-m f1 (lead-coeff f1, us1)
pl.unique-factorization-m f2 (lead-coeff f2, us2)
and $g s$ : mset us $=u s 1+u s 2$
and norm12: image-mset pl.Mp us2 $=u s 2$ image-mset pl. $M p$ us1 $=u s 1$
unfolding pl.Mf-def norm' split by (auto simp: pl.Mf-def)
from $g s$ have $x \in \#$ us $1 \Longrightarrow x \in \#$ mset us for $x$ by auto
hence sub1: $x \in \#$ us $1 \Longrightarrow x \in$ set us for $x$ by auto
from to-fact[OF uf12(1)]
have fact1: pl.factorization-m f1 (lead-coeff f1, us1).
have plf10: pl.Mp f1 $\neq 0$
using fact1 pl.factorization-m-lead-coeff pl.unique-factorization-m-zero uf12(1) by fastforce
have degree f1 = pl.degree-m f1 using deg-m1 by simp
also have $\ldots=$ sum-mset (image-mset pl.degree-m us1)
unfolding pl.factorization-m-degree[OF fact1 plf10] ..
also have $\ldots=$ sum-mset (image-mset degree us1)
by (rule arg-cong[of -- sum-mset], rule image-mset-cong, rule pl.monic-degree-m, rule mon-gs, rule sub1)
finally have degf1-sum: degree f1 $=$ sum-mset (image-mset degree us1) by auto
with deg12 have us1 $\neq\{\#\}$ by auto
then obtain $u$ us11 where us1: us1 $=\{\# u \#\}+u s 11$
by (cases us1, auto)
hence $u 1: u \in \# u s 1$ by auto
hence $u: u \in$ set us by (rule sub1)
let ? $g=g g u$
from pl.factorization-m-mem-dvdm[OF fact1, of $u] u 1$ have $u$-f1: pl.dvdm uf1 by auto
note norm-u $=$ norm $[$ OF $u$ ]
from fact $u$ have irred: pl.irreducible ${ }_{d}-m u$ unfolding pl.factorization-m-def by auto
hence deg-u: degree $u \neq 0$ unfolding pl.irreducible $_{d}$-m-def norm $[O F u]$ by auto
have degree $u \leq$ degree f1 unfolding degf1-sum unfolding us1 by simp
also have $\ldots \leq$ degree $f$ div 2 by fact
finally have deg-uf: degree $u \leq$ degree $f$ div 2 .
hence deg-uf': degree $u \leq S u c$ (degree $f$ div 2) degree $u<S u c$ (degree $f$ div 2) by auto
from mon-gs $[O F u]$ have mon-u: monic $u$.
note short $=L L L$-short-polynomial[OF deg-u deg-uf ${ }^{\prime}(1)$ pl1 mon-u, folded $g g-d e f]$
note short $=\operatorname{short}(1-3) \operatorname{short}(4)\left[O F \operatorname{deg}-u f^{\prime}(2)\right]$
from $\operatorname{short}(1,2) \operatorname{deg} 12(1,3)$ f10 have degree $(g c d f ? g) \leq$ degree $f$ div 2
by（metis Suc－leI Suc－le－mono degree－gcd1 gcd．commute le－trans）
also have $\ldots$ ． degree $f$ using degf by simp
finally have degree $(g c d f$ ？$g)<$ degree $f$ by simp
with find－map－filter－None［OF None，simplified，rule－format，of u］deg－uf u
have deg－gcd：degree $(g c d f(? g))=0$ by（auto simp：gcd．commute）
have gcd f1（？g）dvd gcd $f(? g)$ using $f 0$ unfolding $f$ by simp
from divides－degree［OF this，unfolded deg－gcd］f0
have deg－gcd1：degree（gcd f1（？g））$=0$ by auto
from $F 0$ have normF：$\|F\|^{2} \geq 1$ using sq－norm－poly－pos $[o f F]$ by presburger
have $g 0$ ：？$g \neq 0$ using short（2）．
from $g 0$ have normg：$\|? g\|^{2} \geq 1$ using sq－norm－poly－pos［of ？$\left.g\right]$ by presburger
from f10 have normf1：$\|f 1\|^{2} \geq 1$ using sq－norm－poly－pos［of f1］by presburger
from $f F f$ have $f 1 F$ ：f1 dvd $F$ unfolding $d v d-d e f$ by auto
have pl－ge0：pl $\geq 0$ using pl．poly－mod－2－axioms poly－mod－2－def by auto
from $f F$ have degree $f \leq$ degree $F$ using $F 0 f 0$ by（metis dvd－imp－degree－le）
hence d2D2：？d2 $\leq$ ？D2 by simp
with deg12（3）have df1－D2：degree f1 $\leq$ ？D2 by linarith
from short（1）d2D2 have dg－D2：degree（ $g g u$ ）$\leq$ ？D2 by linarith
have $\|f 1\|^{2}$ へ degree $(g g u) *\|g g u\|^{2}$ へ degree f1 $\leq\|f 1\|^{2}$ へ ？D2＊$\|g g u\|^{2}$ へ？$D 2$
by（rule mult－mono［OF pow－mono－exp pow－mono－exp］， insert normf1 normg，auto intro：df1－D2 dg－D2）
also have $\ldots=\left(\|f 1\|^{2} *\|g g u\|^{2}\right)^{\wedge} ? D 2$
by（simp add：power－mult－distrib）
also have $\ldots \leq\left(\|f 1\|^{2} *\left(\text { 2 }^{\wedge} \text { ？D2 } *\|f 1\|^{2}\right)\right)^{\wedge}$ ？D2
by（rule power－mono［OF mult－left－mono［OF order．trans $[$ OF $\operatorname{short}(4)[O F$ f10 u－f1］］］］，
insert deg12 d2D2，auto intro！：mult－mono）
also have $\ldots=\|f 1\|^{2}$ へ $(? D 2+? D 2) * 2 \uparrow(? D 2 * ? D 2)$
unfolding power－add power－mult－distrib power－mult by simp
also have $\ldots \leq\left(\mathcal{Z}^{\wedge}(2 * ? D 2) *\|F\|^{2}\right)^{\wedge}(? D 2+? D 2) * \mathcal{Z}^{\wedge}(? D 2 * ? D 2)$
by（rule mult－right－mono $[$ OF order．trans $[$ OF power－mono $[$ OF sq－norm－factor－bound $[O F$ f1F F0］］］］，
auto intro！：power－mono mult－right－mono df1－D2）
also have $\ldots=2^{\wedge}(2 * ? D 2 *(? D 2+? D 2)+? D 2 * ? D 2) *\|F\|^{2}$＾
（？D2＋？D2）
unfolding power－mult－distrib power－mult power－add by simp
also have $2 * ? D 2 *(? D 2+? D 2)+? D 2 * ? D 2=5 * ? D 2 * ? D 2$ by simp
also have ？D2＋？D2 $=2 *$ ？D2 by simp
finally have large：
$\|f 1\|^{2}$ へ degree $(g g u) *\|g g u\|^{2}$ へ degree $f 1<p l \wedge 2$ using large by simp
have degree $u \leq$ degree（？g）
proof（rule pl．dvdm－degree［OF mon－u short（3）］，standard）
assume pl．Mp（？g）$=0$
from arg－cong［OF this，of $\lambda$ p．coeff $p$（degree ？g）］
have pl．M（coeff ？g（degree ？g））$=0$ by（auto simp：pl．Mp－def co－ eff－map－poly）
from this［unfolded pl．M－def］obtain $c$ where $l g$ ：lead－coeff ？$g=p l * c$
by auto
with $g 0$ have $c 0: c \neq 0$ by auto
hence $p l$＾2 $\leq$（lead－coeff ？g）${ }^{\text {＾2 }} 2$ unfolding $l g$ abs－le－square－iff［symmetric］ by（rule aux－abs－int）
also have $\ldots \leq\|? g\|^{2}$ using coeff－le－sq－norm $[o f ? g]$ by auto
also have $\ldots=\|? g\|^{2} \wedge 1$ by simp
also have $\ldots \leq\|? g\|^{2}$ へ degree $f 1$
by（rule pow－mono－exp，insert deg12 normg，auto）
also have $\ldots=1 * \ldots$ by $\operatorname{simp}$
also have $\ldots \leq\|f 1\|^{2} \uparrow$ degree $? g *\|? g\|^{2}$＾degree f1
by（rule mult－right－mono，insert normf1，auto）
also have $\ldots<p l^{2}$ by（rule large）
finally show False by auto
qed
with deg－u have deg－g： $0<$ degree（ $g g u$ ）by auto
have pl－ge0：$p l \geq 0$ using pl．poly－mod－2－axioms poly－mod－2－def by auto
from $f F$ have degree $f \leq$ degree $F$ using $F 0 f 0$ by（metis dvd－imp－degree－le）
hence d2D2：？d2 $\leq$ ？D2 by simp
with deg12（3）have df1－D2：degree f1 $\leq$ ？D2 by linarith
from short（1）d2D2 have dg－D2：degree（ $g g u$ ）$\leq$ ？D2 by linarith
have $0<$ degree f1 $0<$ degree $u$ using deg12 deg－$u$ by auto
from common－factor－via－short［of f1 gg $u$ ，OF this（1）deg－g mon－u this（2）
u－f1 short（3）－pl－ge0］deg－gcd1
have $p l \wedge 2 \leq\|f 1\|^{2}$ へ degree $(g g u) *\|g g u\|^{2}$ へ degree f1 by linarith
also have $\ldots<p l \leadsto 2$ by（rule large）
finally show False by simp
qed
thus ？thesis using $f s$－$f$ by simp
qed
qed
qed
end
lemma LLL－factorization：
assumes res：LLL－factorization $f=g s$
and sff：square－free $f$
and deg：degree $f \neq 0$
shows $f=$ prod－list gs $\wedge\left(\forall g \in\right.$ set gs． irreducible $\left._{d} g\right)$
proof－
let ？lc＝lead－coeff $f$
define $p$ where $p \equiv$ suitable－prime－bz $f$
obtain $c$ gs where fff：finite－field－factorization－int $p f=(c, g s)$ by force
let ？degs＝map degree gs
note res $=$ res［unfolded LLL－factorization－def Let－def，folded p－def， unfolded fff split，folded］

```
    from suitable-prime-bz[OF sff refl]
    have prime: prime p and cop: coprime ?lc p and sf: poly-mod.square-free-m p f
        unfolding p-def by auto
    note res
    from prime interpret p: poly-mod-prime p by unfold-locales
    define K where K= 2`(5* (degree f-1)* (degree f-1))*|f| 2^(2*
(degree f-1))
    define }N\mathrm{ where N=sqrt-int-ceiling K
    have K0: K\geq0 unfolding K-def by fastforce
    have NO: N\geq0 unfolding N-def sqrt-int-ceiling using KO
    by (smt of-int-nonneg real-sqrt-ge-0-iff zero-le-ceiling)
    define }n\mathrm{ where }n=\mathrm{ find-exponent p N
    note res = res[folded n-def[unfolded N-def K-def]]
    note n = find-exponent[OF p.m1, of N, folded n-def]
    note bh = p.berlekamp-and-hensel-separated(1)[OF cop sf refl fff n(2)]
    from deg have f0: f\not=0 by auto
    from n p.m1 have pn1: p` n> 1 by auto
    note res = res[folded bh(1)]
    note * = p.berlekamp-hensel-unique[OF cop sf bh n(2)]
    note ** = p.berlekamp-hensel-main[OF n(2) bh cop sf fff]
    from res * **
    have uf: poly-mod.unique-factorization-m ( }\mp@subsup{p}{}{`}n)f\mathrm{ (lead-coeff f, mset (berlekamp-hensel
p nf))
    and norm: \bigwedgeui. ui \in set (berlekamp-hensel p nf) \Longrightarrow poly-mod.Mp ( p^n) ui
= ui
    unfolding berlekamp-hensel-def fff split by auto
    have K:K< ( }\mp@subsup{p}{}{`}n\mp@subsup{)}{}{2}\mathrm{ using n sqrt-int-ceiling-bound[OF KO]
    by (smt NO N-def n(1) power2-le-imp-le)
    show ?thesis
        by (rule LLL-implementation.LLL-reconstruction[OF res deg uf dvd-refl norm
f0 cop sf pn1
                refl prime K[unfolded K-def]])
qed
lemma LLL-many-factorization:
    assumes res:LLL-many-factorization f}=g
    and sff: square-free f
    and deg: degree f}\not=
    shows f}=\mathrm{ prod-list gs }\wedge(\forallg\in\mathrm{ set gs. irreducible d g)
proof -
    let ?lc = lead-coeff f
    define p}\mathrm{ where }p\equiv\mathrm{ suitable-prime-bz f
    obtain c gs where fff: finite-field-factorization-int p f=(c,gs) by force
    let ?degs = map degree gs
    note res = res[unfolded LLL-many-factorization-def Let-def, folded p-def,
        unfolded fff split, folded]
    from suitable-prime-bz[OF sff refl]
    have prime: prime p and cop: coprime ?lc p and sf: poly-mod.square-free-m p f
        unfolding p-def by auto
```

note res
from prime interpret $p$ : poly-mod-prime $p$ by unfold-locales
define $K$ where $K=\mathcal{Z}^{\wedge}\left(5 *\left(\right.\right.$ degree $f$ div 2) * (degree $f$ div 2) ) * $\|f\|^{2 \wedge}(2 *$ (degree f div 2))
define $N$ where $N=$ sqrt-int-ceiling $K$
have $K 0: K \geq 0$ unfolding $K$-def by fastforce
have $N 0: N \geq 0$ unfolding $N$-def sqrt-int-ceiling using KO
by (smt of-int-nonneg real-sqrt-ge-0-iff zero-le-ceiling)
define $n$ where $n=$ find-exponent $p N$
note res $=$ res[folded $n$-def[unfolded $N$-def $K$-def]]
note $n=$ find-exponent[OF p.m1, of $N$, folded $n$-def]
note $b h=p$.berlekamp-and-hensel-separated(1)[OF cop sf refl fff $n(2)]$
from deg have $f 0: f \neq 0$ by auto
from $n$ p.m1 have $p n 1: p^{\wedge} n>1$ by auto
note res $=$ res[folded bh(1)]
note $*=p$.berlekamp-hensel-unique[OF cop sf bh n(2)]
note $* *=$ p.berlekamp-hensel-main[OF n(2) bh cop sf fff]
from res $* * *$
have uf: poly-mod.unique-factorization-m $\left(p^{\wedge} n\right) f$ (lead-coeff $f$, mset (berlekamp-hensel $p n f$ )
and norm: $\bigwedge u i$. ui $\in$ set (berlekamp-hensel $p n f) \Longrightarrow \operatorname{poly-mod.Mp(p\wedge n)ui}$ $=u i$
unfolding berlekamp-hensel-def fff split by auto
have $K$ : $K<\left(p^{\wedge} n\right)^{2}$ using $n$ sqrt-int-ceiling-bound $[O F K 0$ ]
by (smt NO $N$-def $n(1)$ power2-le-imp-le)
show ?thesis
by (rule LLL-implementation.LLL-many-reconstruction[OF res deg uf dvd-refl norm f0 cop sf pn1 refl prime $K[$ unfolded $K$-def]])
qed
lift-definition one-lattice-LLL-factorization :: int-poly-factorization-algorithm is LLL-factorization using $L L L$-factorization by auto
lift-definition many-lattice-LLL-factorization :: int-poly-factorization-algorithm is LLL-many-factorization using LLL-many-factorization by auto
lemma LLL-factorization-primitive: assumes LLL-factorization $f=f s$
square-free f
$0<$ degree $f$
primitive $f$
shows $f=$ prod-list $f s \wedge(\forall f i \in$ set fs. irreducible $f i \wedge 0<$ degree $f i \wedge$ primitive $f i)$
using assms(1)
by (intro int-poly-factorization-algorithm-irreducible[of one-lattice-LLL-factorization,

$$
\text { OF }-\operatorname{assms}(2-)], \text { transfer, auto })
$$

thm factorize-int-poly[of one-lattice-LLL-factorization]
thm factorize-int-poly[of many-lattice-LLL-factorization]
end

## 6 Calculating All Possible Sums of Sub-Multisets

```
theory Sub-Sums
    imports
        Main
        HOL-Library.Multiset
begin
fun sub-mset-sums :: 'a :: comm-monoid-add list = ' a set where
    sub-mset-sums [] ={0}
| sub-mset-sums (x# xs)=(let S = sub-mset-sums xs in S\cup((+)x)'S)
lemma subset-add-mset: ys \subseteq# add-mset x zs \longleftrightarrow (ys\subseteq# zs \vee (\exists xs. xs \subseteq# zs
^ys = add-mset x xs))
    (is ?l = ?r)
proof
    have sub: ys \subseteq#zs \Longrightarrowys\subseteq# add-mset x zs
        by (metis add-mset-remove-trivial diff-subset-eq-self subset-mset.dual-order.trans)
    assume ?r
    thus ?l using sub by auto
next
    assume l: ?l
    show ?r
    proof (cases x G# ys)
        case True
        define xs where xs = (ys - {# x #})
        from True have ys:ys=add-mset x xs unfolding xs-def by auto
        from l[unfolded ys] have xs \subseteq# zs by auto
        thus ?r unfolding ys by auto
    next
        case False
        with l have ys \subseteq# zs by (simp add: subset-mset.le-iff-sup)
        thus ?thesis by auto
    qed
qed
lemma sub-mset-sums[simp]: sub-mset-sums xs = sum-mset '{ ys. ys \subseteq# mset xs
}
proof (induct xs)
    case (Cons x xs)
    have id: {ys.ys\subseteq# mset (x # xs)}={ys.ys\subseteq# mset xs}\cup{add-mset x ys
ys. ys \subseteq# mset xs}
    unfolding mset.simps subset-add-mset by auto
    show ?case unfolding sub-mset-sums.simps Let-def Cons id image-Un
        by force
qed auto
```

end

## 7 Implementation and soundness of a modified version of Algorithm 16.22

Algorithm 16.22 is quite similar to the LLL factorization algorithm that was verified in the previous section. Its main difference is that it has an inner loop where each inner loop iteration has one invocation of the LLL basis reduction algorithm. Algorithm 16.22 of the textbook is therefore closer to the factorization algorithm as it is described by Lenstra, Lenstra, and Lovász [3], which also uses an inner loop.
The advantage of the inner loop is that it can find factors earlier, and then small lattices suffice where without the inner loop one invokes the basis reduction algorithm on a large lattice. The disadvantage of the inner loop is that if the input is irreducible, then one cannot find any factor early, so that all but the last iteration have been useless: only the last iteration will prove irreducibility.

We will describe the modifications w.r.t. the original Algorithm 16.22 of the textbook later in this theory.

```
theory Factorization-Algorithm-16-22
    imports
        LLL-Factorization
        Sub-Sums
begin
```


### 7.1 Previous lemmas obtained using local type definitions

context poly-mod-prime-type
begin
lemma irreducible-m-dvdm-prod-list-connect:
assumes irr: irreducible-m a
and dvd: a dvdm (prod-list xs)
shows $\exists b \in$ set xs. $a d v d m b$
proof -
let ? $A=($ of-int-poly a)::'a mod-ring poly
let ? $X$ = (map of-int-poly xs)::'a mod-ring poly list
let ?XS1 = (of-int-poly (prod-list xs))::'a mod-ring poly
have [transfer-rule]: MP-Rel a ?A
by (simp add: MP-Rel-def Mp-f-representative)
have [transfer-rule]: MP-Rel (prod-list xs) ?XS1
by (simp add: MP-Rel-def Mp-f-representative)
have [transfer-rule]: list-all2 MP-Rel xs ? XS
by (simp add: MP-Rel-def Mp-f-representative list-all2-conv-all-nth)
have A: ?A dvd ?XS1 using dvd by transfer

```
    have \exists b \in set ?XS. ?A dvd b
    by (rule irreducible-dvd-prod-list, insert irr, transfer, auto simp add: A)
    from this[untransferred] show ?thesis .
qed
end
lemma (in poly-mod-prime) irreducible-m-dvdm-prod-list:
    assumes irr: irreducible-m a
    and dvd: a dvdm (prod-list xs)
    shows \exists b\in set xs.a dvdm b
    by (rule poly-mod-prime-type.irreducible-m-dvdm-prod-list-connect[unfolded poly-mod-type-simps,
    internalize-sort 'a :: prime-card, OF type-to-set, unfolded remove-duplicate-premise,
        cancel-type-definition, OF non-empty irr dvd])
```


### 7.2 The modified version of Algorithm 16.22

```
definition B2-LLL :: int poly }=>\mathrm{ int where
```

    B2-LLL \(f=2\) へ \((2 *\) degree \(f) *\|f\|^{2}\)
    hide-const (open) factors
hide-const (open) factors
hide-const (open) factor
hide-const (open) factor

```
context
    fixes p:: int and l :: nat
begin
context
    fixes gs :: int poly list
        and f :: int poly
        and u :: int poly
        and Degs :: nat set
begin
```

This is the critical inner loop.
In the textbook there is a bug, namely that the filter is applied to $g^{\prime}$ and not to the primitive part of $g^{\prime}$. (Problems occur if the content of $g^{\prime}$ is divisible by $p$.) We have fixed this problem in the obvious way.
However, there also is a second problem, namely it is only guaranteed that $g^{\prime}$ is divisible by $u$ modulo $p^{l}$. However, for soundness we need to know that then also the primitive part of $g^{\prime}$ is divisible by $u$ modulo $p^{l}$. This is not necessary true, e.g., if $g^{\prime}=p^{l}$, then the primitive part is 1 which is not divisible by $u$ modulo $p^{l}$. It is open, whether such a large $g^{\prime}$ can actually occur. Therefore, the current fix is to manually test whether the leading
coefficient of $g^{\prime}$ is strictly smaller than $p^{l}$.
With these two modifications, Algorithm 16.22 will become sound as proven below.

```
definition LLL-reconstruction-inner \(j \equiv\)
    let \(j^{\prime}=j-1\) in
    - optimization: check whether degree \(\mathrm{j}^{\prime}\) is possible
    if \(j^{\prime} \notin\) Degs then None else
    - short vector computation
    let
    \(l l=\left(\right.\) let \(n=\) sqrt-int-ceiling \(\left(\|f\|^{2} \wedge\left(2 * j^{\prime}\right) * 2 \wedge\left(5 * j^{\prime} * j^{\prime}\right)\right)\);
    \(l l^{\prime}=\) find-exponent \(p\) in if \(l l^{\prime}<l\) then \(l l^{\prime}\) else \(\left.l\right)\);
    - optimization: dynamically adjust the modulus
    \(p l=p ` l l ;\)
    \(g^{\prime}=\) LLL-short-polynomial pl \(j u\)
- fix: forbid multiples of \(p^{l}\) as short vector, unclear whether this is really required
    in if abs (lead-coeff \(\left.g^{\prime}\right) \geq p l\) then None else
    let ppg \(=\) primitive-part \(g^{\prime}\)
    in
    - slight deviation from textbook: we check divisibility instead of norm-inequality
    case div-int-poly \(f\) ppg of Some \(f^{\prime} \Rightarrow\)
    - fix: consider modular factors of ppg and not of g'
    Some (filter ( \(\lambda\) gi. \(\neg\) poly-mod.dvdm p gi ppg) gs, lead-coeff \(f^{\prime}, f^{\prime}, p p g\) )
    | None \(\Rightarrow\) None
```

function $L L L$-reconstruction-inner-loop where
LLL-reconstruction-inner-loop $j=$
(if $j>$ degree $f$ then ( []$, 1,1, f$ )
else case LLL-reconstruction-inner $j$
of Some tuple $\Rightarrow$ tuple
| None $\Rightarrow$ LLL-reconstruction-inner-loop $(j+1))$
by auto
termination by (relation measure $(\lambda j$. Suc (degree $f)-j$ ), auto)
end
partial-function (tailrec) LLL-reconstruction" where [code]:
LLL-reconstruction" gs b factors $=$
(if gs $=[]$ then factors
else
let $u=$ choose-u gs;
$d=$ degree $u ;$
$g s^{\prime}=$ remove1 $u g s ;$
degs $=$ map degree $g^{\prime}$;
Degs $=((+) d)$ 'sub-mset-sums degs;
$\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.$, factor $)=$ LLL-reconstruction-inner-loop gs $f$ u Degs $(d+1)$
in LLL-reconstruction" gs' $b^{\prime} f^{\prime}$ (factor\#factors)

```
    )
```

definition reconstruction-of-algorithm-16-22 gs $f \equiv$
let $G=[] ;$
$b=$ lead-coeff $f$
in LLL-reconstruction" gs bf $G$
end
definition factorization-algorithm-16-22 :: int poly $\Rightarrow$ int poly list where
factorization-algorithm-16-22 $f=$ (let

- find suitable prime
$p=$ suitable-prime-bz f;
- compute finite field factorization
$\left(-, f_{s}\right)=$ finite-field-factorization-int $p f$;
- determine 1 and B
$n=$ degree $f$;
- bound improved according to textbook, which uses no $=(n+1) *(\max -$ normf $)^{2}$
$n o=\|f\|^{2} ;$
- possible improvement: $B=\operatorname{sqrt}\left(2^{5 * n *(n-1)} * n o^{2 * n-1}\right.$, cf. LLL-factorization
$B=$ sqrt-int-ceiling $\left(2^{\wedge}(5 * n * n) * n o\right.$ へ $\left.(2 * n)\right)$;
$l=$ find-exponent $p B$;
- perform hensel lifting to lift factorization to $\bmod p^{l}$
$v s=$ hensel-lifting $p l f f s$
- reconstruct integer factors
in reconstruction-of-algorithm-16-22 $p l$ vs $f$ )


### 7.3 Soundness proof

### 7.3.1 Starting the proof

Key lemma to show that forbidding values of $p^{l}$ or larger suffices to find correct factors.
lemma (in poly-mod-prime) Mp-smult-p-removal: poly-mod.Mp $(p * p \wedge k)(s m u l t$ $p f)=0 \Longrightarrow$ poly-mod.Mp $\left(p^{\wedge} k\right) f=0$
by (smt add.left-neutral m1 poly-mod.Dp-Mp-eq poly-mod.Mp-smult-m-0 sdiv-poly-smult smult-smult)
lemma (in poly-mod-prime) eq-m-smult-p-removal: poly-mod.eq-m $(p * p \wedge k)$ (smult $p f$ ) (smult $p g$ )
$\Longrightarrow$ poly-mod.eq-m ( $\left.p^{\wedge} k\right) f g$ using Mp-smult- $p$-removal $[o f k f-g]$
by (metis add-diff-cancel-left' diff-add-cancel diff-self poly-mod.Mp-0 poly-mod.minus-Mp(2)
smult-diff-right)
lemma content-le-lead-coeff: abs (content $(f::$ int poly) $) \leq a b s(l e a d-c o e f f ~ f)$
proof (cases $f=0$ )
case False
from content-dvd-coeff [of f degree $f]$ have abs (content $f$ ) dvd abs (lead-coeff f)
by auto
moreover have abs (lead-coeff f) $\neq 0$ using False by auto
ultimately show ?thesis by (smt dvd-imp-le-int)
qed auto
lemma poly-mod-dvd-drop-smult: assumes $u$ : monic $u$ and $p$ : prime $p$ and $c: c$ $\neq 0|c|<p^{\wedge} l$
and dvd: poly-mod.dvdm $\left(p^{\wedge} l\right) u($ smult $c f)$
shows poly-mod.dvdm puf
using $c$ dvd
proof (induct l arbitrary: c rule: less-induct)
case (less lc)
interpret poly-mod-prime $p$ by (unfold-locales, insert $p$, auto)
note $c=\operatorname{less}(2-3)$
note $d v d=\operatorname{less}(4)$
note $I H=\operatorname{less}(1)$
show ?case
proof (cases $l=0$ )
case True
thus ?thesis using $c d v d$ by auto
next
case 10: False
interpret pl: poly-mod-2 $p^{\wedge} l$ by (unfold-locales, insert m1 l0, auto)
show ?thesis
proof (cases pdvd c)
case False
let ? $i=$ inverse-mod $c\left(p^{\wedge} l\right)$
have $g c d$ c $p=1$ using $p$ False
by (metis Primes.prime-int-iff gcd-ge-0-int semiring-gcd-class.gcd-dvd1 semir-ing-gcd-class.gcd-dvd2)
hence coprime c $p$ by (metis dvd-refl gcd-dvd-1)
from pl.inverse-mod-coprime-exp[OF refl $p$ l0 this]
have id: pl.M $(? i * c)=1$.
have pl.Mp $($ smult ?i $($ smult c f $))=$ pl.Mp $($ smult $(p l . M(? i * c)) f)$ by simp also have $\ldots=p l . M p f$ unfolding id by simp
finally have pl.dvdm u $f$ using $p l . d v d m-s m u l t[O F ~ d v d$, of ? $i]$ unfolding $p l . d v d m$-def by simp
thus $u d v d m f$ using $l 0$ pl-dvdm-imp- $p-d v d m$ by blast
next
case True
then obtain $d$ where $c p d: c=p * d$ unfolding $d v d-d e f$ by auto
from $c p d c$ have $d 0: d \neq 0$ by auto
note to- $p=M p-M p$-pow-is-Mp[OF 10 ml$]$
from dvd obtain $v$ where eq: pl.eq- $m(u * v)($ smult $p(s m u l t d f))$
unfolding $p l . d v d m$-def cpd by auto
from arg-cong $[O F$ this, of $M p$, unfolded to-p]
have $M p(u * v)=0$ unfolding $M p$-smult-m-0 .
with $u$ have $M p v=0$
by (metis Mp-0 add-eq-0-iff-both-eq-0 degree-0
degree-m-mult-eq monic-degree-0 monic-degree-m mult-cancel-right2)
from Mp-O-smult-sdiv-poly[OF this]
obtain $w$ where $v: v=$ smult $p w$ by metis
with eq have eq: pl.eq-m (smult $p(u * w)$ ) (smult $p$ (smult $d f$ )) by simp
from $l 0$ obtain $l l$ where $l=S u c l l$ by (cases $l$, auto)
hence $p l: p^{\wedge} l=p * p^{\wedge} l l$ and $l l: l l<l$ by auto
from $c$ (2) have $d$-small: $|d|<p^{\wedge} l l$ unfolding pl cpd abs-mult
using mult-less-cancel-left-pos[of $p$ d $\left.p^{\wedge} l l\right] m 1$ by auto
from eq-m-smult-p-removal[OF eq[unfolded pl]]
have poly-mod.eq-m ( $\left.p^{\wedge} l l\right)(u * w)(s m u l t d f)$.
hence dvd: poly-mod.dvdm ( $\left.p^{\wedge} l l\right) u(s m u l t d f)$ unfolding poly-mod.dvdm-def
by metis
show ?thesis by (rule $I H[O F$ ll d0 d-small dvd])
qed
qed
qed
context
fixes $p::$ int and $F::$ int poly and $N::$ nat
and $l::$ nat
defines $[$ simp $]: N \equiv$ degree $F$
assumes $p$ : prime $p$
and $N O: N>0$
and bound-l: $\boldsymbol{2}^{\wedge} N^{2} * B 2-L L L F^{\wedge}(2 * N) \leq\left(p^{\wedge} l\right)^{2}$
begin
private lemma $F 0: F \neq 0$ using $N 0$
by fastforce
private lemma $p 1: p>1$ using $p$ prime-gt-1-int by auto
interpretation $p$ : poly-mod-prime $p$ using $p$ by unfold-locales
interpretation $p l:$ poly- $\bmod p \wedge$.
lemma B2-2: $2 \leq$ B2-LLL F
proof -
from $F 0$ have $\|F\|^{2} \neq 0$ by simp
hence $F 1:\|F\|^{2} \geq 1$ using sq-norm-poly-pos[of $\left.F\right]$ F0 by linarith
have $(2::$ int $)=2 \wedge 1 * 1$ by simp
also have $\ldots \leq B 2-L L L F$ unfolding B2-LLL-def
by (intro mult-mono power-increasing F1, insert N0, auto)
finally show $2 \leq B 2-L L L F$.
qed
lemma l-gt-0: $l>0$
proof (cases l)

```
    case 0
    have 1*2 < 2^ N N*B2-LLL F^(2 *N)
    proof (rule mult-mono)
    have 2 * 1\leq(2 :: int) * (2 ` (2*N - 1)) by (rule mult-left-mono, auto)
    also have ... = 2 ^ (2 *N) using NO by (cases N, auto)
    also have ... \leqB2-LLL F^ (2 *N)
        by (rule power-mono[OF B2-2], force)
    finally show 2 \leq B2-LLL F^
qed auto
also have ... \leq 1 using bound-l[unfolded 0] by auto
finally show ?thesis by auto
qed auto
lemma l0:l\not=0 using l-gt-0 by auto
lemma pl-not0: p^ l f 0 using p1 l0 by auto
interpretation pl: poly-mod-2 p^l
    by (standard, insert p1 l0, auto)
private lemmas pl-dvdm-imp-p-dvdm = p.pl-dvdm-imp-p-dvdm[OF l0]
lemma p-Mp-pl-Mp[simp]: p.Mp (pl.Mp k)=p.Mp k
    using Mp-Mp-pow-is-Mp[OF l0 p.m1].
context
    fixes u :: int poly
        and d and f and n
        and gs :: int poly list
        and Degs :: nat set
    defines [simp]:d \equiv degree u
    assumes d0:d>0
        and u:monic u
        and irred-u: p.irreducible-m u
        and u-f:p.dvdm uf
        and f-dvd-F: f dvd F
        and [simp]: n== degree f
        and f-gs: pl.unique-factorization-m f (lead-coeff f, mset gs)
        and cop: coprime (lead-coeff f) p
        and sf: p.square-free-m f
        and sf-F: square-free f
        and u-gs:u\in set gs
        and norm-gs: map pl.Mp gs = gs
        and Degs: \ factor. factor dvd f \Longrightarrow p.dvdm u factor }\Longrightarrow\mathrm{ degree factor }
Degs
begin
interpretation pl: poly-mod-2 p`l using l0 p1 by (unfold-locales, auto)
private lemma f0:f\not=0 using sf-F unfolding square-free-def by fastforce
```

```
private lemma MpfO: pl.Mp f}=
```

    by (metis \(p\).square-free-m-def \(p-M p-p l-M p s f\) )
    private lemma $p M p f 0: p . M p f \neq 0$
using $p$.square-free-m-def sf by auto
private lemma $d n: d \leq n$ using $p . d v d m$-imp-degree-le[OF u-f u pMpf0 p1] by
auto
private lemma n0: $n>0$ using $d 0 d n$ by auto
private lemma $B 2-0[$ intro!]: B2-LLL $F>0$ using $B 2-2$ by auto
private lemma deg-u: degree $u>0$ using $d 0 d$-def by auto
private lemma $n$-le- $N$ : $n \leq N$ by (simp add: dvd-imp-degree-le[OF f-dvd-F F0])

```
lemma \(d v d m\)-power: assumes \(g d v d f\)
    shows \(p . d v d m u g \longleftrightarrow p l . d v d m u g\)
proof
    assume pl.dvdm ug
    thus \(p . d v d m u g\) by (rule pl-dvdm-imp-p-dvdm)
next
    assume dvd: p.dvdm ug
    from norm-gs have norm-gsp: \(\bigwedge f . f \in\) set \(g s \Longrightarrow p l . M p f=f\) by (induct gs,
auto)
    with \(f\)-gs[unfolded pl.unique-factorization-m-alt-def pl.factorization-m-def split]
    have gs-irred-mon: \(\bigwedge f . f \in \#\) mset gs \(\Longrightarrow\) pl.irreducible \({ }_{d}-m f \wedge\) monic \(f\) by
auto
    from norm-gs have norm-gs: image-mset pl.Mp (mset gs) \(=\) mset gs by (induct
gs, auto)
    from assms obtain \(h\) where \(f: f=g * h\) unfolding dvd-def by auto
    from pl.unique-factorization-m-factor[OF p.prime f-gs[unfolded f] - l0 refl,
folded \(f\),
            OF cop sf, unfolded pl.Mf-def split] norm-gs
    obtain hs \(f s\) where uf: pl.unique-factorization-m \(h\) (lead-coeff \(h, h s\) )
            pl.unique-factorization-m \(g\) (lead-coeff \(g, f_{s}\) )
            and \(i d\) : mset \(g s=f s+h s\)
            and norm: image-mset pl.Mp fs \(=\) fs image-mset pl.Mp \(h s=h s\) by auto
    from p.square-free-m-prod-imp-coprime-m[OF sf[unfolded f]]
    have cop- \(h\)-f: p.coprime-m \(g h\) by auto
    show pl.dvdm ug
    proof (cases \(u \in \# f s\) )
            case True
            hence \(p l . M p u \in \#\) image-mset pl.Mp fs by auto
    from pl.factorization-m-mem-dvdm[OF pl.unique-factorization-m-imp-factorization[OF
uf(2)] this]
            show ?thesis.
    next
```

```
    case False
    from u-gs have u\in# mset gs by auto
    from this[unfolded id] False have u\in# hs by auto
    hence pl.Mp u\in# image-mset pl.Mp hs by auto
    from pl.factorization-m-mem-dvdm[OF pl.unique-factorization-m-imp-factorization[OF
uf(1)] this]
    have pl.dvdm u h by auto
    from pl-dvdm-imp-p-dvdm[OF this]
    have p.dvdm u h by auto
    from cop-h-f[unfolded p.coprime-m-def, rule-format, OF dvd this]
    have p.dvdm u 1.
    from p.dvdm-imp-degree-le[OF this u-p.m1] have degree u=0 by auto
    with deg-u show ?thesis by auto
    qed
qed
```

private lemma $u f$ : pl.dvdm uf using dvdm-power[OF dvd-refl] u-f by simp
lemma exists-reconstruction: $\exists h 0$. irreducible ${ }_{d} h 0 \wedge$ p.dvdm u h0 $\wedge h 0 \operatorname{dvd} f$
proof -
have deg-f: degree $f>0$ using $\langle n \equiv$ degree $f\rangle n 0$ by blast
from berlekamp-zassenhaus-factorization-irreducible ${ }_{d}[O F$ refl sf-F deg-f]
obtain $f s$ where $f$-fs: $f=$ prod-list $f s$
and $c:\left(\forall f i \in\right.$ set $f s$. irreducible $e_{d} f \wedge 0<$ degree $\left.f_{i}\right)$ by blast
have $p l . d v d m u$ (prod-list fs) using $u f f-f s$ by simp
hence $p . d v d m$ u (prod-list fs) by (rule pl-dvdm-imp-p-dvdm)
from this obtain $h 0$ where $h 0: h 0 \in$ set $f s$ and $d v d m-u$-h0: p.dvdm u h0
using $p$.irreducible-m-dvdm-prod-list [OF irred-u] by auto
moreover have $h 0$ dvd $f$ by (unfold $f$ - $f s$, rule prod-list-dvd[OF h0])
moreover have irreducible $e_{d} 0$ using $c h 0$ by auto
ultimately show ?thesis by blast
qed
lemma factor-dvd-f-0: assumes factor dvd $f$
shows pl.Mp factor $\neq 0$
proof -
from assms obtain $h$ where $f: f=$ factor $* h$ unfolding dvd-def ..
from arg-cong[OF this, of pl.Mp] have $0 \neq p l . M p$ (pl.Mp factor $* h$ )
using Mpf0 by auto
thus ?thesis by fastforce
qed
lemma degree-factor-ge-degree-u:
assumes $u$-dvdm-factor: p.dvdm u factor
and factor-dvd: factor dvd $f$ shows degree $u \leq$ degree factor
proof -
from factor-dvd-f-0[OF factor-dvd] have factor0: pl.Mp factor $\neq 0$.
from $u$-dvdm-factor[unfolded dvdm-power[OF factor-dvd] pl.dvdm-def] obtain $v$
where

```
    *: pl.Mp factor = pl.Mp (u* pl.Mp v) by auto
    with factor0 have v0: pl.Mp v\not=0 by fastforce
    hence }0\not=\mathrm{ lead-coeff (pl.Mp v) by auto
    also have lead-coeff (pl.Mp v)=pl.M (lead-coeff (pl.Mp v))
    by (auto simp: pl.Mp-def coeff-map-poly)
    finally have **: lead-coeff (pl.Mp v) \not= p^ l*r for r by (auto simp: pl.M-def)
    from * have degree factor }\geq\mathrm{ pl.degree-m ( }u*\mathrm{ *l.Mp v) using pl.degree-m-le[of
factor] by auto
    also have pl.degree-m (u* pl.Mp v)= degree ( }u*\mathrm{ * pl.Mp v)
        by (rule pl.degree-m-eq, unfold lead-coeff-mult, insert u pl.m1 **, auto)
    also have ... = degree u + degree (pl.Mp v)
    by (rule degree-mult-eq, insert v0 u, auto)
    finally show ?thesis by auto
qed
```


### 7.3.2 Inner loop

context
fixes $j^{\prime}::$ nat
assumes $d j^{\prime}: d \leq j^{\prime}$
and $j^{\prime} n: j^{\prime}<n$
and deg: $\bigwedge$ factor. p.dvdm u factor $\Longrightarrow$ factor dvd $f \Longrightarrow$ degree factor $\geq j^{\prime}$
begin
private abbreviation (input) $j \equiv S u c j^{\prime}$
private lemma $j n: j \leq n$ using $j^{\prime} n$ by auto
private lemma factor-irreducible ${ }_{d} I$ : assumes $h f: h d v d f$
and puh: p.dvdm u $h$
and degh: degree $h>0$
and degh-j: degree $h \leq j^{\prime}$
shows irreducible $_{d} h$
proof -
from dvdm-power[OF hf] puh have pluh: pl.dvdm u h by simp
note uf-partition $=$ p.unique-factorization-m-factor-partition[OF l0]
obtain $g s 1$ gs2 where part: List.partition ( $\lambda$ gi. p.dvdm gi $h$ ) $g s=(g s 1, g s 2)$ by
force
from part u-gs puh
have $u$-gs1: $u \in$ set gs1 unfolding $p$ by auto
have gs1: gs1 = filter ( $\lambda$ gi. p.dvdm gi $h$ ) gs using part by auto
obtain $k$ where $f: f=h * k$ using hf unfolding dvd-def by auto
from uf-partition[OF f-gs f cop sf part]
have uf-h: pl.unique-factorization-m $h$ (lead-coeff $h$, mset gs1) by auto
show ?thesis
proof (intro irreducible $_{d} I$ degh)
fix $q r$
assume deg- $q$ : degree $q>0$ degree $q<$ degree $h$
and deg-r: degree $r>0$ degree $r<$ degree $h$
and $h: h=q * r$
then have $r$ dvd $h$ by auto
with $h$ dvd-trans $[O F-h f]$ have 1: $q$ dvd $f r d v d f$ by auto
from $\operatorname{cop}[$ unfolded $f$ ] have cop: coprime (lead-coeff $h$ ) $p$
using p.prime pl.coprime-lead-coeff-factor(1) by blast
from $s f[$ unfolded $f]$ have $s f$ : p.square-free-m $h$ using $p$.square-free-m-factor by metis
have norm-gs1: image-mset pl.Mp (mset gs1) = mset gs1 using norm-gs unfolding $g s 1$
by (induct gs, auto)
from pl.unique-factorization-m-factor[OF p uf-h[unfolded h], folded h, OF cop sf 10 refl]
obtain fs gs where uf-q: pl.unique-factorization-m $q$ (lead-coeff $q, f s$ )
and $u f$-r: pl.unique-factorization-m $r$ (lead-coeff $r, g s$ )
and $i d$ : mset gs1 $=f s+g s$
unfolding pl.Mf-def split using norm-gs1 by auto
from degh degh-j deg-q deg-r have $q j^{\prime}$ : degree $q<j^{\prime}$ and $r j^{\prime}:$ degree $r<j^{\prime}$ by auto
have intro: $u \in \# r \Longrightarrow$ pl.Mp $u \in \#$ image-mset $p l . M p r$ for $r$ by auto
note $d v d I=$ pl.factorization-m-mem-dvdm[OF pl.unique-factorization-m-imp-factorization intro]
from $u$-gs1 id have $u \in \# f s \vee u \in \# g s$ unfolding in-multiset-in-set[symmetric] by auto
with $d v d I[O F u f-q] d v d I[O F u f-r]$ have $p l . d v d m u q \vee p l . d v d m u r$ by auto
hence $p . d v d m$ u $q \vee p$.dvdm u $r$ using $p l-d v d m-i m p-p-d v d m$ by blast
with $1 q j^{\prime} r j^{\prime}$ show False
by (elim disjE, auto dest!: deg)
qed
qed
private definition $l l=\left(\right.$ let $n=$ sqrt-int-ceiling $\left(\|f\|^{2}\right.$ へ $\left(2 * j^{\prime}\right) * 2^{\wedge}\left(5 * j^{\prime} *\right.$ $\left.j^{\prime}\right)$ );
$l l^{\prime}=$ find-exponent $p n$ in if $l l^{\prime}<l$ then $l l^{\prime}$ else $l$ )
lemma $l l: l l \leq l$ unfolding $l l-d e f$ Let-def by auto
lemma $l l 0: l l \neq 0$ using $l 0$ find-exponent[OF p.m1]
unfolding ll-def Let-def by auto
lemma pll1: $p^{\wedge} l l>1$ using $l l 0 p . m 1$ by auto
interpretation pll: poly-mod-2 p ${ }^{\wedge} l l$
using llo p.m1 by (unfold-locales, auto)
lemma $p l l 0: p \wedge l l \neq 0$ using $p$ by auto
lemma $d v d m-l-l l$ : assumes $p l . d v d m$ a $b$
shows pll.dvdm a b

```
proof -
    have id: p`l= p^ll* p^(l-ll) using ll unfolding power-add[symmetric] by
auto
    from assms[unfolded pl.dvdm-def] obtain c where eq: pl.eq-m b (a*c) by blast
        from pll.Mp-shrink-modulus[OF eq[unfolded id]] p have pll.eq-m b (a*c) by
auto
    thus ?thesis unfolding pll.dvdm-def ..
qed
private definition g \equivLLL-short-polynomial ( (^^ll) ju
lemma deg-g-j: degree g<j
    and g0:g\not=0
    and ug:pll.dvdm ug
```



```
|h|}\mp@subsup{|}{}{2
proof (atomize(full), goal-cases)
    case 1
    from deg-u have degu0: degree }u\not=0\mathrm{ by auto
    have ju: j\geq degree u using d-def dj' le-Suc-eq by blast
    have ju': j > degree u using d-def dj' by auto
    note short = LLL-short-polynomial[OF degu0 ju pll1 u, folded g-def]
    from short(1-3) short(4)[OF ju` show ?case by auto
qed
lemma LLL-reconstruction-inner-simps: LLL-reconstruction-inner p lgs f u Degs
j
    =(if j' }\not=\mathrm{ Degs then None else if p` ll \ |lead-coeff g| then None
    else case div-int-poly f (primitive-part g) of None }=>\mathrm{ None
            | Some f' }=>\mathrm{ Some ([giזgs . ᄀ p.dvdm gi (primitive-part g)], lead-coeff f',
f},\mathrm{ , primitive-part g))
proof -
    have Suc: Suc j' - 1 = j' by simp
    show ?thesis unfolding LLL-reconstruction-inner-def Suc Let-def ll-def[unfolded
Let-def, symmetric]
        g-def[unfolded Let-def, symmetric] by simp
qed
lemma LLL-reconstruction-inner-complete:
    assumes ret:LLL-reconstruction-inner plgs fu Degs j=None
    shows \factor. p.dvdm u factor }\Longrightarrow\mathrm{ factor dvd f # degree factor }\geq
proof (rule ccontr)
    fix factor
    assume pu-factor: p.dvdm u factor
        and factor-f: factor dvd f
        and deg-factor2: ᄀj\leq degree factor
    with deg[OF this(1,2)] have deg-factor-j [simp]: degree factor = j' and deg-factor-lt-j:
degree factor < j by auto
    from Degs[OF factor-f pu-factor] have Degs: ( j' \not=Degs ) = False by auto
```

from dvdm－power［OF factor－f］pu－factor have u－factor：pl．dvdm u factor by auto
from dvdm－l－ll［OF u－factor］have pll－u－factor：pll．dvdm u factor by auto
have deg－factor：degree factor $>0$
using $d 0$ deg－factor－j $d j^{\prime}$ by linarith
from $f 0$ deg－factor divides－degree $[O F$ factor－$f]$ have deg－$f$ ：degree $f>0$ by auto
from deg－factor have $j^{\prime} 0: j^{\prime}>0$ by simp
from factor－f f0 have factor 0 ：factor $\neq 0$ by auto
from factor－f obtain f2 where $f: f=$ factor $*$ f2 unfolding $d v d-d e f$ by auto
from deg－u have deg－u0：degree $u \neq 0$ by auto
from $p u$－factor $u$ have $u$－$j^{\prime}$ ：degree $u \leq j^{\prime}$ unfolding deg－factor－$j$［symmetric］ using $d$－def deg－factor－$j d j^{\prime}$ by blast
hence $u$－$j$ ：degree $u \leq j$ degree $u<j$ by auto
note $L L L=L L L$－short－polynomial $[O F$ deg－u0 $u$－j（1）pll1 $u$ ，folded $g$－def］
note ret $=$ ret［unfolded LLL－reconstruction－inner－simps Degs if－False］
note $L L L=L L L(1-3) L L L(4)[O F u$－j（2）factor0 pll－u－factor deg－factor－lt－j］
hence deg－g：degree $g \leq j^{\prime}$ by simp
from $L L L$（2）have normg：$\|g\|^{2} \geq 1$ using sq－norm－poly－pos $[$ of $g]$ by presburger from $f 0$ have normf：$\|f\|^{2} \geq 1$ using sq－norm－poly－pos $[$ of $f]$ by presburger
from factor0 have normf1：$\|$ factor $\|^{2} \geq 1$ using sq－norm－poly－pos［of factor］by presburger
from $F 0$ have normF：$\|F\|^{2} \geq 1$ using sq－norm－poly－pos［of $\left.F\right]$ by presburger
from factor－f $\langle f d v d F\rangle$ have factor－$F$ ：factor $d v d F$ by（rule dvd－trans）
have $\|$ factor $\|^{2}$ へ degree $g *\|g\|^{2}$＾degree factor $\leq \|$ factor $\left\|^{2} \uparrow j^{\prime} *\right\| g \|^{2} \wedge j^{\prime}$
by（rule mult－mono［OF power－increasing］，insert normg normf1 deg－g，auto）
also have $\ldots=\left(\| \text { factor }\left\|^{2} *\right\| g \|^{2}\right)^{\wedge} j^{\prime}$ by（simp add：power－mult－distrib）
also have $\ldots \leq\left(\| \text { factor } \|^{2} *\left(2^{\wedge} j^{\prime} * \| \text { factor } \|^{2}\right)\right)^{\wedge} j^{\prime}$
by（rule power－mono［OF mult－left－mono］，insert LLL（4），auto）
also have $\ldots=\|$ factor $\|^{2} \wedge\left(2 * j^{\prime}\right) * \mathcal{Z}^{\wedge}\left(j^{\prime} * j^{\prime}\right)$
unfolding power－mult－distrib power－mult power－add mult－2 by simp
finally have approx－part－1：$\|$ factor $\|^{2}$ へ degree $g *\|g\|^{2}$ へ degree factor $\leq \|$ fac－ tor $\|^{2} \wedge\left(2 * j^{\prime}\right) * \mathcal{2}^{\wedge}\left(j^{\prime} * j^{\prime}\right)$ 。
\｛
fix $f::$ int poly
assume $*$ ：factor dvd ff$\neq 0$
note approx－part－1
also have $\|$ factor $\|^{2}$ へ $\left(2 * j^{\prime}\right) * 2^{\wedge}\left(j^{\prime} * j^{\prime}\right) \leq\left(2 \wedge\left(2 * j^{\prime}\right) *\|f\|^{2}\right) \wedge\left(2 * j^{\prime}\right) *$ $2^{\wedge}\left(j^{\prime} * j^{\prime}\right)$
by（rule mult－right－mono［OF power－mono］，insert sq－norm－factor－bound $[O F$
＊］，auto）
also have $\ldots=\|f\|^{2}\left(2 * j^{\prime}\right) * \mathcal{Z}^{\wedge}\left(2 * j^{\prime} * 2 * j^{\prime}+j^{\prime} * j^{\prime}\right)$
unfolding power－mult－distrib power－add by（simp add：power－mult［symmetric］）
also have $2 * j^{\prime} * 2 * j^{\prime}+j^{\prime} * j^{\prime}=5 * j^{\prime} * j^{\prime}$ by simp
finally have $\|$ factor $\|^{2}$ へ degree $g *\|g\|^{2}$ へ degree factor $\leq\|f\|^{2}$ へ $\left(2 * j^{\prime}\right) * 2$
～$\left(5 * j^{\prime} * j^{\prime}\right)$ ．
$\}$ note approx $=$ this
note approx－1 $=$ approx［OF factor－f f0］
note approx－2－part $=$ approx $[O F$ factor－F F0］
have large：$\|$ factor $\|^{2}$ へ degree $g *\|g\|^{2}$ へ degree factor $<\left(p^{\wedge} l l\right)^{2}$

```
proof (cases \(l l=l\) )
    case False
    let ? \(n=\|f\|^{2}\) へ \(\left(2 * j^{\prime}\right) * 2^{\wedge}\left(5 * j^{\prime} * j^{\prime}\right)\)
    have \(n: ? n \geq 0\) by auto
    let ?s \(=\) sqrt-int-ceiling ? \(n\)
    from False have \(l l=\) find-exponent \(p\) ?s unfolding \(l l\)-def Let-def by auto
    hence spll: ?s < p ll using find-exponent(1)[OF p.m1] by auto
    have sqrt ? \(n \geq 0\) by auto
    hence sqrt: sqrt? \(n>-1\) by linarith
    have \(n s: ? n \leq\) ? \(s^{\wedge} 2\) using sqrt-int-ceiling-bound \([O F n]\).
    also have \(\ldots<\left(p^{\wedge} l l\right)^{\wedge}\) の
        by (rule power-strict-mono[OF spll], insert sqrt, auto)
    finally show ?thesis using approx-1 by auto
next
    case True
    hence \(l l: p \wedge l l=p \wedge l\) by \(\operatorname{simp}\)
    show ?thesis unfolding \(l l\)
    proof (rule less-le-trans[OF le-less-trans[OF approx-2-part] bound-l])
        have \(\|F\|^{2}\) へ \(\left(2 * j^{\prime}\right) * 2^{\wedge}\left(5 * j^{\prime} * j^{\prime}\right)\)
            \(=2^{\wedge}\left(2 * j^{\prime} * j^{\prime}+3 * j^{\prime} * j^{\prime}\right) *\|F\|^{2}\left(j^{\prime}+j^{\prime}\right)\)
            unfolding mult-2 by simp
        also have \(\ldots<2^{\wedge}\left(N^{2}+4 * N * N\right) *\|F\|^{2} へ(2 * N)\)
        proof (rule mult-less-le-imp-less[OF power-strict-increasing pow-mono-exp])
            show \(1 \leq\|F\|^{2}\) by (rule normF)
            have \(j N^{\prime}: j^{\prime}<N\) and \(j N: j^{\prime} \leq N\) using \(j n\) divides-degree \([O F\langle f d v d F\rangle] F 0\)
by auto
            have \(j^{\prime}+j^{\prime} \leq j^{\prime}+j^{\prime}\) using deg-g \(j^{\prime} n\) by auto
            also have \(\ldots=2 * j^{\prime}\) by auto
            also have \(\ldots \leq 2 * N\) using \(j N\) by auto
            finally show \(j^{\prime}+j^{\prime} \leq 2 * N\).
            show \(0<\|F\|^{2}\) へ \(\left(j^{\prime}+j^{\prime}\right)\)
                by (rule zero-less-power, insert normF, auto)
            have \(2 * j^{\prime} * j^{\prime}+3 * j^{\prime} * j^{\prime} \leq 2 * j^{\prime} * j^{\prime}+3 * j^{\prime} * j^{\prime}\) by auto
            also have \(\ldots=5 *\left(j^{\prime} * j^{\prime}\right)\) by auto
            also have \(\ldots<5 *(N * N)\)
                by (rule mult-strict-left-mono[OF mult-strict-mono], insert \(j N^{\prime}\), auto)
            also have \(\ldots=N^{2}+4 * N * N\) by (simp add: power2-eq-square)
            finally show \(2 * j^{\prime} * j^{\prime}+3 * j^{\prime} * j^{\prime}<N^{2}+4 * N * N\).
    qed auto
    also have \(\ldots=2^{\wedge} N^{2} *\left(\mathcal{2}^{\wedge}(2 * N) *\|F\|^{2}\right)^{\wedge}(2 * N)\)
    unfolding power-mult-distrib power-add by (simp add: power-mult[symmetric])
    finally show \(\|F\|^{2}\) へ \(\left(2 * j^{\prime}\right) * 2 へ\left(5 * j^{\prime} * j^{\prime}\right)<2^{\wedge} N^{2} * B 2-L L L F\) (2
* \(N\) )
            unfolding B2-LLL-def by simp
        qed
qed
have \((\mid \text { lead-coeff } g \mid)^{\wedge} 2<\left(p^{\wedge} l l\right)^{\wedge} 2\)
proof (rule le-less-trans[OF - large])
    have \(1 *\left(\mid \text { lead-coeff }\left.g\right|^{2}\right)^{\wedge} 1 \leq \|\) factor \(\|^{2}\) へ degree \(g *\|g\|^{2}\) へ degree factor
```

by（rule mult－mono［OF－order．trans［OF power－mono pow－mono－exp］］，
insert normg normf1 deg－f g0 coeff－le－sq－norm［of g］j＇0，
auto intro：pow－mono－one）
thus $\mid$ lead－coeff $\left.g\right|^{2} \leq \|$ factor $\|^{2}$ へ degree $g *\|g\|^{2}$ へ degree factor by simp
qed
hence（lead－coeff $g$ ）＾2 $<\left(p^{\wedge} l l\right){ }^{\wedge} 2$ by $\operatorname{simp}$
hence $\mid$ lead－coeff $g \mid<p^{\wedge}$ ll using $p . m 1$ abs－le－square－iff［of $p^{\wedge} l l$ lead－coeff $g$ ］by auto
hence $\left(p^{\wedge} l l \leq \mid\right.$ lead－coeff $\left.g \mid\right)=$ False by auto
note ret $=$ ret［unfolded this if－False］
have deg－f：degree $f>0$ using $n 0$ by auto
have deg－ug：degree $u \leq$ degree $g$
proof（rule pll．dvdm－degree［OF u LLL（3）］，standard）
assume pll．Mp $g=0$
from $\arg$－cong $[$ OF this，of $\lambda p$ ．coeff $p$（degree $g$ ）］
have pll．M（coeff $g($ degree $g))=0$ by（auto simp：pll．Mp－def coeff－map－poly）
from this［unfolded pll．M－def］obtain $c$ where $l g$ ：lead－coeff $g=p^{\wedge} l l * c$ by
auto
with $L L L$（2）have $c 0: c \neq 0$ by auto
hence $(p \wedge l l){ }^{\wedge} 2 \leq(l e a d$－coeff $g){ }^{\text {＾2 }} 2$ unfolding $l g$ abs－le－square－iff［symmetric］
by（rule aux－abs－int）
also have $\ldots \leq\|g\|^{2}$ using coeff－le－sq－norm $[o f g]$ by auto
also have $\ldots=\|g\|^{2}$＾ 1 by $\operatorname{simp}$
also have $\ldots \leq\|g\|^{2} \wedge$ degree factor
by（rule pow－mono－exp，insert deg－f normg $j^{\prime} 0$ ，auto）
also have $\ldots=1 * \ldots$ by $\operatorname{simp}$
also have $\ldots \leq \|$ factor $\|^{2}$＾degree $g *\|g\|^{2}$ へ degree factor
by（rule mult－right－mono，insert normf1，auto）
also have $\ldots<\left(p^{\wedge} l l\right)^{2}$ by（rule large）
finally show False by auto
qed
with deg－$u$ have deg－$g$ ：degree $g>0$ by simp
from $j^{\prime} 0$ have deg－factor：degree factor $>0$ by simp
let $? g=$ gcd factor $g$
from common－factor－via－short［OF deg－factor deg－g u deg－u pll－u－factor LLL（3）
large］pll．m1
have $g c d: 0<$ degree ？$g$ by auto
have gcd－factor：？g dvd factor by auto
from dvd－trans $[O F$ this factor－$f]$ have $g c d-f: ? g$ dvd $f$ ．
from deg－g have $g 0: g \neq 0$ by auto
have $g c d$－$g$ ：degree $? g \leq$ degree $g$ using $g 0$ using divides－degree by blast
from $g c d-g L L L(1)$ have $h j^{\prime}$ ：degree ？$g \leq j^{\prime}$ by auto
let ？$p p=$ primitive－part $g$
from ret have div－int－poly $f$ ？pp $=$ None by（auto split：option．splits）
from div－int－poly［of $f$ ？pp，unfolded this］go
have $p p f: \neg$ ？pp dvd $f$ unfolding dvd－def by（auto simp：ac－simps）
have irr－f1：irreducible ${ }_{d}$ factor
by（rule factor－irreducible $e_{d}[$ OF factor－f pu－factor deg－factor $]$ ，simp）
from $g c d-$－factor obtain $h$ where factor：factor $=? g * h$ unfolding $d v d$－def by

```
auto
    from irreducible d}D(2)[OF irr-f1, of ?g h, folded factor] have \neg (degree ?g < j'
^ degree h<j')
    by auto
    moreover have j'= degree ?g + degree h using factor0 arg-cong[OF factor, of
degree]
    by (subst (asm) degree-mult-eq, insert j'0, auto)
    ultimately have degree h=0 using gcd by linarith
    from degree0-coeffs[OF this] factor factor0
    obtain c where h:h=[:c:] and c:c\not=0 by fastforce
    from arg-cong[OF factor, of degree] have id: degree ?g = degree factor
        unfolding }h\mathrm{ using c by auto
    moreover have degree ?g \leq degree g
    by (subst gcd.commute, rule degree-gcd1[OF g0])
    ultimately have degree g\geqdegree factor by auto
    with id deg-factor2 deg-g-j have deg: degree ?g = degree g
    and degree g}=\mathrm{ degree factor by auto
    have ?g dvd g}\mathrm{ by auto
    then obtain q where g:g=? g*q unfolding dvd-def by auto
    from arg-cong[OF this, of degree] deg
    have degree q=0
        by (subst (asm) degree-mult-eq, insert g g0, force, force) simp
    from degree0-coeffs[OF this] g g0
    obtain d}\mathrm{ where p:q=[:d:] and d:d}=0\mathrm{ by fastforce
    from arg-cong[OF factor, of (*)q]
    have q* factor =h*g
    by (subst g, auto simp: ac-simps)
    hence smult d factor =h*g unfolding ph by auto
    hence g dvd smult d factor by simp
    from dvd-smult-int[OF d this]
    have primitive-part g dvd factor .
    from dvd-trans[OF this factor-f] ppf show False by auto
qed
lemma LLL-reconstruction-inner-sound:
    assumes ret:LLL-reconstruction-inner plgs fu Degs j=Some (gs', b',f}\mp@subsup{}{\prime}{\prime},h
    shows f=\mp@subsup{f}{}{\prime}*h (is ?g1)
    and irreducibled h (is ?g2)
    and }\mp@subsup{b}{}{\prime}=lead-coeff f' (is ?g3
    and pl.unique-factorization-m f' (lead-coeff f', mset gs') (is ?g4)
    and p.dvdm uh (is ?g5)
    and degree h=j'(is ?g6)
    and length gs'< length gs (is ?g7)
    and set gs'\subseteq set gs (is ?g8)
    and g\mp@subsup{s}{}{\prime}\not=[] (is ?g9)
proof -
    let ?ppg= primitive-part g
    note ret = ret[unfolded LLL-reconstruction-inner-simps]
    from ret have lc:abs (lead-coeff g)< p^ll by (auto split: if-splits)
```

from ret obtain rest where rest: div-int-poly $f$ (primitive-part $g)=$ Some rest by (auto split: if-splits option.splits)
from ret[unfolded this] div-int-then-rqp[OF this] lc
have out $[$ simp $]: h=$ ?ppg gs' $=$ filter $(\lambda$ gi. $\neg p . d v d m ~ g i ~ ? p p g) ~ g s$
$f^{\prime}=$ rest $b^{\prime}=$ lead-coeff rest
and $f: f=? p p g *$ rest by (auto split: if-splits)
with div-int-then-rqp [OF rest] show ?g1 ?g3 by auto
from 〈? g1〉 f0 have $h 0: h \neq 0$ by auto
let $? c=$ content $g$
from $g 0$ have $c t 0: ? c \neq 0$ by auto
have $|? c| \leq \mid$ lead-coeff $g \mid$ by (rule content-le-lead-coeff)
also have $\ldots<p^{\wedge} l l$ by fact
finally have $c t-p l:|? c|<p \wedge l l$.
from $u g$ have pll.dvdm $u$ (smult ?c ?ppg) by simp
from poly-mod-dvd-drop-smult[OF u p ct0 ct-pl this]
show puh: p.dvdm u h by simp
with $d v d m$-power[of $h] f$
have uh: pl.dvdm $u h$ by (auto simp: dvd-def)
from $f$ have $h f: h d v d f$ by (auto intro: $d v d I$ )
have degh: degree $h>0$
by (metis d-def deg deg-u puh dj' hf le-neq-implies-less not-less0 neq0-conv)
show irr-h: ?g2
by (intro factor-irreducible ${ }_{d} I$ degh hf puh, insert deg-g-j, simp)
show deg-h: ?g6 using deg deg-g-j g-def hf le-less-Suc-eq puh degree-primitive-part
by force
show ? $g 7$ unfolding out
by (rule length-filter-less[of $u$ ], insert pl-dvdm-imp-p-dvdm[OF uh] u-gs, auto)
show ? 88 by auto
from $f$ out have $f h: f=h * f^{\prime}$ and $g s^{\prime}: g s^{\prime}=[g i \leftarrow g s . \neg p . d v d m g i h]$ by auto
note $[$ simp del $]=$ out
let $? f s=$ filter $(\lambda g i . p . d v d m$ gi h) gs
have part: List.partition ( $\lambda$ gi. p.dvdm gi h) gs $=\left(? f s, g s^{\prime}\right)$
unfolding $g s^{\prime}$ by (auto simp: o-def)
from $p$.unique-factorization-m-factor-partition[OF 10 f-gs fh cop sf part]
show uf: pl.unique-factorization-m $f^{\prime}\left(\right.$ lead-coeff $f^{\prime}$, mset $\left.g s^{\prime}\right)$ by auto
show ? g9
proof
assume $g s^{\prime}=[]$
with pl.unique-factorization-m-imp-factorization[OF uf, unfolded pl.factorization-m-def]
have $p l . M p f^{\prime}=p l . M p\left(\right.$ smult (lead-coeff $\left.f^{\prime}\right) 1$ ) by auto
from arg-cong[OF this, of degree] pl.degree-m-le[of smult (lead-coeff f') 1]
have $p l$.degree- $m f^{\prime}=0$ by simp
also have pl.degree-m $f^{\prime}=$ degree $f^{\prime}$
proof (rule poly-mod.degree-m-eq[OF-pl.m1])
have coprime (lead-coeff f') $p$
by (rule p.coprime-lead-coeff-factor[OF p.prime cop[unfolded fh]])
thus lead-coeff $f^{\prime} \bmod p^{\wedge} l \neq 0$ using $l 0$ p.prime by fastforce
qed
finally have $\operatorname{deg} f^{\prime}:$ degree $f^{\prime}=0$ by auto
from degree0-coeffs[OF this] f0 fh obtain $c$ where $f^{\prime}=[: c:]$ and $c: c \neq 0$ and fch: $f=$ smult $c h$
by auto
from 〈irreducible $\left.{ }_{d} h\right\rangle$ have irr-f: irreducible $_{d} f$
using irreducible $_{d}$-smult-int $[O F$ c, of $h]$ unfolding fch by auto
have degree $f=j^{\prime}$ using $h f$ irr-h deg-h
using irr-f $\langle n \equiv$ degree $f\rangle$ degh $j^{\prime} n$
by (metis add.right-neutral degf ${ }^{\prime}$ degree-mult-eq f0 fh mult-not-zero)
thus False using $j^{\prime} n$ by auto
qed
qed
end
interpretation $L L L d$.
lemma LLL-reconstruction-inner-None-upt-j':
assumes $i j: \forall i \in\{d+1 . . j\}$. LLL-reconstruction-inner plgs $f$ u Degs $i=$ None and $d j: d<j$ and $j \leq n$
shows $\bigwedge$ factor. $p . d v d m$ factor $\Longrightarrow$ factor dvd $f \Longrightarrow$ degree factor $\geq j$
using assms
proof (induct $j$ )
case (Suc j)
show ?case
proof (rule LLL-reconstruction-inner-complete)
show $\bigwedge$ factor2. p.dvdm u factor2 $\Longrightarrow$ factor2 $d v d f \Longrightarrow j \leq$ degree factor2
proof (cases d=j)
case False
show $\bigwedge$ factor2. $p . d v d m$ u factor2 $\Longrightarrow$ factor2 dvd $f \Longrightarrow j \leq$ degree factor2
by (rule Suc.hyps, insert Suc.prems False, auto)
next
case True
then show $\bigwedge$ factor2. p.dvdm u factor2 $\Longrightarrow$ factor2 dvd $f \Longrightarrow j \leq$ degree
factor2
using degree-factor-ge-degree-u by auto
qed
qed (insert Suc.prems, auto)
qed auto
corollary LLL-reconstruction-inner-None-upt-j:
assumes $i j: \forall i \in\{d+1 . . j\}$. LLL-reconstruction-inner plgs fu Degs $i=$ None and $d j: d \leq j$ and $j n: j \leq n$
shows $\bigwedge$ factor. p.dvdm u factor $\Longrightarrow$ factor dvd $f \Longrightarrow$ degree factor $\geq j$
proof (cases $d=j$ )
case True
then show $\bigwedge$ factor. p.dvdm u factor $\Longrightarrow$ factor dvd $f \Longrightarrow d=j \Longrightarrow j \leq$ degree factor
using degree-factor-ge-degree-u by auto
next
case False

```
    hence dj2: d<j using dj by auto
    then show \{factor. p.dvdm u factor \Longrightarrow factor dvd f \Longrightarrowd\not=j\Longrightarrowj\leq degree
factor
    using LLL-reconstruction-inner-None-upt-j'[OF ij dj2 jn] by auto
qed
lemma LLL-reconstruction-inner-all-None-imp-irreducible:
    assumes i:\foralli\in{d+1..n}.LLL-reconstruction-inner plgs fu Degs i=None
    shows irreducible e}
proof -
    obtain factor
    where irreducible-factor: irreducible d factor
            and dvdp-u-factor: p.dvdm u factor and factor-dvd-f: factor dvd f
    using exists-reconstruction by blast
    have f0: f\not=0 using n0 by auto
    have deg-factor1: degree u\leq degree factor
    by (rule degree-factor-ge-degree-u[OF dvdp-u-factor factor-dvd-f])
    hence factor-not0: factor }\not=0\mathrm{ using d0 by auto
    hence deg-factor2: degree factor \leq degree f using divides-degree[OF factor-dvd-f]
f0 by auto
    let ?j = degree factor
    show ?thesis
    proof (cases degree factor = degree f)
    case True
    from factor-dvd-f obtain g}\mathrm{ where f-factor: f= factor * g unfolding dvd-def
by auto
    from True[unfolded f-factor] f0[unfolded f-factor] have degree g=0 g\not=0
            by (subst (asm) degree-mult-eq, auto)
    from degree0-coeffs[OF this(1)] this(2) obtain c where g=[:c:] and c:c\not=
0 by auto
    with f-factor have fc: f= smult c factor by auto
    from irreducible-factor irreducible e}\mp@subsup{d}{d}{\mathrm{ -smult-int[OF c, of factor, folded fc]}
    show ?thesis by simp
    next
        case False
    hence Suc-j: Suc ?j \leq degree f using deg-factor2 by auto
    have Suc ?j \leq degree factor
        proof (rule LLL-reconstruction-inner-None-upt-j[OF - - dvdp-u-factor fac-
tor-dvd-f])
            show d\leqSuc ?j using deg-factor1 by auto
                show }\foralli\in{d+1..(Suc ?j)}.LLL-reconstruction-inner p l gs f u Degs i
None
            using Suc-j i by auto
                show Suc ?j }\leqn\mathrm{ using Suc-j by simp
    qed
    then show ?thesis by auto
    qed
qed
```

```
lemma irreducible-imp-LLL-reconstruction-inner-all-None:
    assumes irr-f: irreducible \(f\)
    shows \(\forall i \in\{d+1 . . n\}\). LLL-reconstruction-inner plgs fu Degs \(i=\) None
proof (rule ccontr)
    let \(? L L L\)-inner \(=\lambda i\). LLL-reconstruction-inner plgs fu Degs \(i\)
    let \(? G=\{j . j \in\{d+1 . . n\} \wedge\) ?LLL-inner \(j \neq\) None \(\}\)
    assume \(\neg(\forall i \in\{d+1 . . n\}\). ?LLL-inner \(i=\) None \()\)
    hence \(G\)-not-empty: ? \(G \neq\{ \}\) by auto
    define \(j\) where \(j=\) Min ? \(G\)
    have \(j\)-in- \(G: j \in ? G\) by (unfold \(j\)-def, rule Min-in[OF - G-not-empty], simp)
    hence \(j: j \in\{d+1\)..n \(\}\) and LLL-not-None: ?LLL-inner \(j \neq\) None using \(j\)-in- \(G\)
by auto
    have \(\forall i \in\{d+1 . .<j\}\). ?LLL-inner \(i=\) None
    proof (rule ccontr)
        assume \(\neg(\forall i \in\{d+1 . .<j\}\). ? LLL-inner \(i=\) None \()\)
        from this obtain \(i\) where \(i: i \in\{d+1 . .<j\}\) and LLL-i: ?LLL-inner \(i \neq\)
None by auto
    hence \(i G: i \in\) ? \(G\) using \(j\)-def \(G\)-not-empty by auto
    have \(i<j\) using \(i\) by auto
    moreover have \(j \leq i\) using \(i G j\)-def by auto
    ultimately show False by linarith
    qed
    hence all-None: \(\forall i \in\{d+1 . . j-1\}\). ?LLL-inner \(i=\) None by auto
    obtain \(g s^{\prime} b^{\prime} f^{\prime}\) factor where LLL-inner-eq: ?LLL-inner \(j=\) Some \(\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.\),
factor)
    using LLL-not-None by force
    have Suc-j1-eq: Suc \((j-1)=j\) using \(j d 0\) by auto
    have \(j n\) : \(j-1<n\) using \(j\) by auto
    have \(d j\) : \(d \leq j-1\) using \(j d 0\) by auto
    have degree: \(\bigwedge\) factor. \(p . d v d m\) u factor \(\Longrightarrow\) factor \(d v d f \Longrightarrow j-1 \leq\) degree factor
    by (rule LLL-reconstruction-inner-None-upt-j[OF all-None dj], insert jn, auto)
    have LLL-inner-Some: ?LLL-inner \((S u c(j-1))=\operatorname{Some}\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.\), factor \()\)
    using LLL-inner-eq Suc-j1-eq by auto
    have deg-factor: degree factor \(=j-1\)
    and \(f f^{\prime}: f=f^{\prime} *\) factor
    and irreducible-factor: irreducible \(_{d}\) factor
        using LLL-reconstruction-inner-sound \([O F\) dj jn degree LLL-inner-Some] by
(metis+)
    have degree \(f^{\prime}=n-(j-1)\) using arg-cong[OF ff', of degree]
    by (subst (asm) degree-mult-eq, insert f0 ff' deg-factor, auto)
    also have \(\ldots<n\) using irreducible-factor \(j n\) unfolding irreducible \(_{d}\)-def deg-factor
by auto
    finally have deg-f': degree \(f^{\prime}<\) degree \(f\) by auto
    from \(f f f^{\prime}\) have factor-dvd-f: factor \(d v d f\) by auto
    have \(\neg\) irreducible \(_{d} f\)
    by (rule reducible \({ }_{d} I\), rule exI[of- \(\left.f^{\prime}\right]\), rule exI[of-factor],
            intro conjI ff', insert deg-factor jn deg-f \({ }^{\prime}\), auto)
```

```
    thus False using irr-f by contradiction
qed
lemma LLL-reconstruction-inner-all-None:
    assumes i: }\foralli\in{d+1..n}.LLL-reconstruction-inner plgs f u Degs i=Non
    and dj: d<j
shows LLL-reconstruction-inner-loop plgs f u Degs j=([],1,1,f)
    using dj
proof (induct j rule:LLL-reconstruction-inner-loop.induct[off plgs u Degs])
    case (1 j)
    let ?innerl = LLL-reconstruction-inner-loop p l gs f u Degs
    let ?inner =LLL-reconstruction-inner p l gs fu Degs
    note hyp = 1.hyps
    note dj = 1.prems(1)
    show ?case
    proof (cases j\leqn)
        case True note jn= True
        have step: ?inner j = None
            by (cases d=j, insert i jn dj, auto)
    have ?innerl j = ?innerl ( j+1)
        using jn step by auto
    also have ... = ([], 1, 1,f)
        by (rule hyp[OF - step], insert jn dj, auto simp add: jn dj)
    finally show ?thesis.
    qed auto
qed
corollary irreducible-imp-LLL-reconstruction-inner-loop-f:
    assumes irr-f: irreducible }\mp@subsup{|}{d}{}f\mathrm{ and dj:d<j
shows LLL-reconstruction-inner-loop pl gs f u Degs j=([],1,1,f)
    using irreducible-imp-LLL-reconstruction-inner-all-None[OF irr-f]
    using LLL-reconstruction-inner-all-None[OF - dj] by auto
lemma exists-index-LLL-reconstruction-inner-Some:
    assumes inner-loop:LLL-reconstruction-inner-loop plgs fu Degs j=(gs',\mp@subsup{b}{}{\prime},\mp@subsup{f}{}{\prime},factor)
        and i:\foralli\in{d+1..<j}.LLL-reconstruction-inner p l gs f u Degs i=None
        and dj:d<j and jn: j\leqn and f:\neg \mp@subsup{irreducible }{d}{}f
    shows }\exists\mp@subsup{j}{}{\prime}.j\leq\mp@subsup{j}{}{\prime}\wedge\mp@subsup{j}{}{\prime}\leqn\wedged<\mp@subsup{j}{}{\prime
        ^(LLL-reconstruction-inner plgs fu Degs j' = Some (gs', b', f', factor))
        \wedge(\foralli\in{d+1..<j'}.LLL-reconstruction-inner plgs f u Degs i=None)
    using inner-loop i dj jn
proof (induct j rule:LLL-reconstruction-inner-loop.induct[of f p l gs u Degs])
    case (1 j)
    let ?innerl = LLL-reconstruction-inner-loop plgs f u Degs
    let ?inner =LLL-reconstruction-inner plgs fu Degs
    note hyp = 1.hyps
    note 1 = 1.prems(1)
    note 2 = 1.prems(2)
    note dj = 1.prems(3)
```

```
note jn = 1.prems(4)
show ?case
proof (cases ?inner j = None)
    case True
    show ?thesis
    proof (cases j=n)
        case True note j-eq-n = True
        show ?thesis
        proof (cases ?inner n = None)
            case True
            have i2: }\foralli\in{d+1..n}. ?inner i=Non
                using 2 j-eq-n True by auto
            have irreducible d}
                by(rule LLL-reconstruction-inner-all-None-imp-irreducible[OF i2])
            thus ?thesis using f by simp
        next
            case False
            have ?inner n = Some (gs', b', f', factor)
                using False 1 j-eq-n by auto
            moreover have }\foralli\in{d+1..<n}. ?inner i=Non
                using 2 j-eq-n by simp
            moreover have d<n using 12 jn j-eq-n
                    using False dn nat-less-le
                    using d-def dj by auto
            ultimately show ?thesis using j-eq-n by fastforce
        qed
    next
        case False
        have }\exists\mp@subsup{j}{}{\prime}\geqj+1.\mp@subsup{j}{}{\prime}\leqn\wedged<\mp@subsup{j}{}{\prime}
                        ?inner j}\mp@subsup{j}{}{\prime}=\mathrm{ Some (gs', b}\mp@subsup{b}{}{\prime},\mp@subsup{f}{}{\prime},\mathrm{ factor })
                (}\foralli\in{d+1..<\mp@subsup{j}{}{\prime}}.\mathrm{ ? ?inner }i=None
    proof (rule hyp)
            show \neg degree f<j using jn by auto
        show ?inner j = None using True by auto
        show ?innerl (j+1) = (g\mp@subsup{s}{}{\prime},\mp@subsup{b}{}{\prime},\mp@subsup{f}{}{\prime},\mathrm{ factor })
            using 1 True jn by auto
        show }\foralli\in{d+1..<j+1}. ?inner i=Non
            by (metis 2 One-nat-def True add.comm-neutral add-Suc-right atLeast-
LessThan-iff
            le-neq-implies-less less-Suc-eq-le)
        show d<j+1 using dj by auto
        show j +1 \leqn using jn False by auto
    qed
    from this obtain j' where a1: j}\geq\mp@code{j + 1 and a2: j' \leqn and a3:d< j
        and a4: ?inner j' = Some (gs', b}\mp@subsup{}{\prime}{\prime},\mp@subsup{f}{}{\prime},\mathrm{ factor)
        and a5: (\foralli\in{d+1..<\mp@subsup{j}{}{\prime}}. ?inner i=None) by auto
    moreover have j}\mp@subsup{j}{}{\prime}\geqj\mathrm{ using a1 by auto
    ultimately show ?thesis by fastforce
qed
```

```
    next
        case False
        have 1: ? inner j = Some (gs', b', f', factor)
            using False 1 jn by auto
        moreover have 2:( }\foralli\in{d+1..<j}. ?inner i=None
        by (rule 2)
    moreover have 3: j\leqn using jn by auto
    moreover have 4:d<j using 2 False dj jn
        using le-neq-implies-less by fastforce
    ultimately show ?thesis by auto
    qed
qed
lemma unique-factorization-m-1: pl.unique-factorization-m 1 (1, {#})
proof (intro pl.unique-factorization-mI)
    fix d gs
    assume pl: pl.factorization-m 1 (d,gs)
    from pl.factorization-m-degree[OF this] have deg0: \g. g\in# gs \Longrightarrowpl.degree-m
g=0 by auto
    {
    assume gs \not={#}
    then obtain ghs where gs: gs ={#g#} + hs by (cases gs,auto)
    with pl have *: pl.irreducible e}\mp@subsup{|}{d}{}-m\mathrm{ (pl.Mp g)
        monic (pl.Mp g) by (auto simp: pl.factorization-m-def)
    with deg0[of g, unfolded gs] have False by (auto simp: pl.irreducible d-m-def)
}
    hence gs ={#} by auto
    with pl show pl.Mf (d,gs)=pl.Mf (1,{#}) by (cases d = 0,
    auto simp: pl.factorization-m-def pl.Mf-def pl.Mp-def)
qed (auto simp: pl.factorization-m-def)
lemma LLL-reconstruction-inner-loop-j-le-n:
    assumes ret:LLL-reconstruction-inner-loop pl gs f u Degs j = (gs', b', f',factor)
    and ij:\foralli\in{d+1..<j}.LLL-reconstruction-inner plgs fu Degs i=None
    and n: n=degree f
    and jn: j\leqn
    and dj:d<j
shows f=\mp@subsup{f}{}{\prime}*\mathrm{ factor (is ?g1)}
    and irreducibled factor (is ?g2)
    and }\mp@subsup{b}{}{\prime}=lead-coeff f' (is ?g3
    and pl.unique-factorization-m f' (b', mset gs') (is ?g4)
    and p.dvdm u factor (is ?g5)
    and gs\not=[]\longrightarrow length gs' < length gs (is ?g6)
    and factor dvd f (is ?g7)
    and f}\mp@subsup{f}{}{\prime}dvdf(is ?g8
    and set gs '\subseteq set gs (is ?g9)
    and g\mp@subsup{s}{}{\prime}=[]\longrightarrow\mp@subsup{f}{}{\prime}=1(\mathrm{ is ?g10)}
using ret ij jn dj
```

```
proof (atomize(full), induct \(j\) )
    case 0
    then show ?case using deg-u by auto
next
    case (Suc j)
    let ?innerl \(=L L L\)-reconstruction-inner-loop plgs f u Degs
    let ? inner \(=L L L\)-reconstruction-inner \(p l\) gs \(f u\) Degs
    have \(i j\) : \(\forall i \in\{d+1 . . j\}\). ?inner \(i=\) None
        using Suc.prems by auto
    have \(d j\) : \(d \leq j\) using Suc.prems by auto
    have \(j n\) : \(j<n\) using Suc.prems by auto
    have deg: Suc \(j \leq\) degree \(f\) using Suc.prems by auto
    have \(c\) : \(\bigwedge\) factor. \(p . d v d m\) u factor \(\Longrightarrow\) factor \(d v d f \Longrightarrow j \leq\) degree factor
    by (rule LLL-reconstruction-inner-None-upt-j[OF ij dj], insert \(n\) jn, auto)
    have 1: ?innerl \((S u c j)=\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.\), factor \()\)
    using Suc.prems by auto
    show ?case
    proof (cases ?inner (Suc \(j\) ) \(=\) None)
        case False
    have \(L L L\)-rw: ? inner \((S u c j)=\) Some \(\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.\), factor \()\)
        using False deg Suc.prems by auto
        show ?thesis using LLL-reconstruction-inner-sound \([O F\) dj jn c \(L L L\)-rw] by
fastforce
    next
        case True note Suc-j-None \(=\) True
    show ?thesis
    proof (cases \(d=j\) )
        case False
        have \(n j: j \leq\) degree \(f\) using Suc.prems False by auto
        moreover have dj2: \(d<j\) using Suc.prems False by auto
        ultimately show ?thesis using Suc.prems Suc.hyps by fastforce
    next
        case True note \(d-e q-j=\) True
        show ?thesis
        proof (cases irreducible \(_{d} f\) )
                case True
                have \(p l-M p-1: p l . M p 1=1\) by auto
                have \(d\)-Suc- \(j: d<S u c j\) using Suc.prems by auto
                have ?innerl \((\) Suc \(j)=([], 1,1, f)\)
                by (rule irreducible-imp-LLL-reconstruction-inner-loop-f[OF True d-Suc-j])
                hence result-eq: \(([], 1,1, f)=\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.\), factor \()\) using Suc.prems by auto
                moreover have thesis1: p.dvdm u factor using \(u\)-f result-eq by auto
                moreover have thesis2: \(f^{\prime}=\) pl.Mp (Polynomial.smult \(b^{\prime}\left(\right.\) prod-list gs \(\left.{ }^{\prime}\right)\) )
                    using result-eq pl-Mp-1 by auto
            ultimately show ?thesis using True by (auto simp: unique-factorization-m-1)
            next
                case False note irreducible-f \(=\) False
                have \(\exists j^{\prime}\). Suc \(j \leq j^{\prime} \wedge j^{\prime} \leq n \wedge d<j^{\prime}\)
                \(\wedge\left(\right.\) ?inner \(j^{\prime}=\) Some \(\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.\), factor \(\left.)\right)\)
```

    \(\wedge\left(\forall i \in\left\{d+1 . .<j^{\prime}\right\}\right.\). ? inner \(i=\) None \()\)
    proof (rule exists-index-LLL-reconstruction-inner-Some[OF - - False])
    show ? innerl (Suc \(j)=\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.\), factor \()\)
        using Suc.prems by auto
    show \(\forall i \in\{d+1 . .<\) Suc \(j\}\).? inner \(i=\) None
        using Suc.prems by auto
    show Suc \(j \leq n\) using \(j n\) by auto
    show \(d<\) Suc \(j\) using Suc.prems by auto
    qed
    from this obtain \(a\) where \(d a\) : \(d<a\) and \(a n: a \leq n\) and \(j a: j \leq a\)
    and a1: ? inner \(a=\) Some ( \(g s^{\prime}, b^{\prime}, f^{\prime}\), factor)
    and a2: \(\forall i \in\{d+1 . .<a\}\). ?inner \(i=\) None by auto
    define \(j^{\prime}\) where \(j^{\prime}[\) simp \(]: j^{\prime} \equiv a-1\)
    have \(d j^{\prime}: d \leq j^{\prime}\) using \(d a\) by auto
    have \(j^{\prime}: j^{\prime} \neq 0\) using \(d j^{\prime} d 0\) by auto
    hence \(j^{\prime} n: j^{\prime}<n\) using an by auto
    have LLL: ?inner \(\left(\right.\) Suc \(\left.j^{\prime}\right)=\operatorname{Some}\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.\), factor \()\)
    using \(a 1 j^{\prime}\) by auto
    have prev-None: \(\forall i \in\left\{d+1 . . j^{\prime}\right\}\). ?inner \(i=\) None
    using a2 \(j^{\prime}\) by auto
    have Suc-rw: Suc \(\left(j^{\prime}-1\right)=j^{\prime}\) using \(j^{\prime}\) by auto
    have \(c\) : \factor. p.dvdm u factor \(\Longrightarrow\) factor \(d v d f \Longrightarrow \operatorname{Suc}\left(j^{\prime}-1\right) \leq\)
    degree factor
by (rule LLL-reconstruction-inner-None-upt-j, insert dj' Suc-rw $j^{\prime} n$
prev-None, auto)
hence $c 2$ : $\bigwedge$ factor. p.dvdm u factor $\Longrightarrow$ factor $\operatorname{dvd} f \Longrightarrow j^{\prime} \leq$ degree factor
using $j^{\prime}$ by force
show ?thesis using LLL-reconstruction-inner-sound $\left[O F d j^{\prime} j^{\prime} n c 2 L L L\right]$ by
fastforce
qed
qed
qed
qed
lemma LLL-reconstruction-inner-loop-j-ge-n:
assumes ret: LLL-reconstruction-inner-loop plgs $f$ u Degs $j=\left(g s^{\prime}, b^{\prime}, f^{\prime}\right.$, factor $)$ and $i j: \forall i \in\{d+1 . . n\}$. LLL-reconstruction-inner plgs $f u$ Degs $i=$ None and $d j: d<j$ and $j n: j>n$
shows $f=f^{\prime} *$ factor (is ?g1)
and irreducible ${ }_{d}$ factor (is ?g2)
and $b^{\prime}=$ lead-coeff $f^{\prime}$ (is ?g3)
and pl.unique-factorization-m $f^{\prime}\left(b^{\prime}\right.$, mset $\left.g s^{\prime}\right)($ is ? $g 4)$
and $p . d v d m$ u factor (is ?g5)
and $g s \neq[] \longrightarrow$ length $g s^{\prime}<$ length gs (is ?g6)
and factor $d v d f$ (is ? $g^{7}$ )
and $f^{\prime} d v d f$ (is ? $g 8$ )
and set $g s^{\prime} \subseteq$ set gs (is ?g9)

```
    and f}\mp@subsup{f}{}{\prime}=1(\mathrm{ is ?g10)
proof -
    have LLL-reconstruction-inner-loop p l gs f u Degs j = ([],1,1,f) using jn by
auto
    hence g\mp@subsup{s}{}{\prime}:g\mp@subsup{s}{}{\prime}=[] \mathrm{ and }\mp@subsup{b}{}{\prime}:\mp@subsup{b}{}{\prime}=1\mathrm{ and }\mp@subsup{f}{}{\prime}:\mp@subsup{f}{}{\prime}=1\mathrm{ and factor: factor }=f\mathrm{ using ret}
by auto
    have irreducible d}
        by (rule LLL-reconstruction-inner-all-None-imp-irreducible[OF ij])
    thus ?g1 ?g2 ?g3 ?g4 ?g5 ?g6 ?g7 ?g8 ?g9 ?g10 using f' factor b' gs' u-f
        by (auto simp: unique-factorization-m-1)
qed
lemma LLL-reconstruction-inner-loop:
    assumes ret: LLL-reconstruction-inner-loop p l gs f u Degs j = (gs', b',f',factor)
        and ij:\foralli\in{d+1..<j}.LLL-reconstruction-inner plgs fu Degs i=None
        and n: n= degree f
        and dj:d<j
    shows f=\mp@subsup{f}{}{\prime}* factor (is ?g1)
        and irreducibled factor (is ?g2)
        and }\mp@subsup{b}{}{\prime}=lead-coeff ff (is ?g3
        and pl.unique-factorization-m f' (b', mset gs') (is ?g4)
        and p.dvdm u factor (is ?g5)
        and gs\not=[]\longrightarrow length gs' < length gs (is ?g6)
        and factor dvd f (is ?g7)
        and f}\mp@subsup{f}{}{\prime}dvdf(\mathrm{ is ? g8)
        and set gs' }\subseteq\mathrm{ set gs (is ?g9)
        and g\mp@subsup{s}{}{\prime}=[]\longrightarrow\mp@subsup{f}{}{\prime}=1(\mathrm{ is ?g10)}
proof (atomize(full),(cases j>n; intro conjI))
    case True
    have ij2: }\foralli\in{d+1..n}.LLL-reconstruction-inner plgs f u Degs i=Non
        using ij True by auto
    show ?g1 ?g2 ?g3 ?g4 ?g5 ?g6 ?g7 ?g8 ?g9 ?g10
        using LLL-reconstruction-inner-loop-j-ge-n[OF ret ij2 dj True] by blast+
next
    case False
    hence jn: j\leqn by simp
    show ?g1 ?g2 ?g3 ?g4 ?g5 ?g6 ?g7 ?g8 ?g9 ?g10
        using LLL-reconstruction-inner-loop-j-le-n[OF ret ij n jn dj] by blast+
qed
end
```


### 7.3.3 Outer loop

```
lemma LLL-reconstruction':
assumes 1: LLL-reconstruction" \(p l\) gs bf \(G=G^{\prime}\)
and irreducible- \(G: \bigwedge\) factor. factor \(\in\) set \(G \Longrightarrow\) irreducible \(_{d}\) factor
and 3: \(F=f *\) prod-list \(G\)
and 4: pl.unique-factorization-m \(f\) (lead-coeff \(f\), mset gs)
and 5: \(g s \neq[]\)
```

```
    and 6: \ gi.gi get gs \Longrightarrow pl.Mp gi=gi
    and 7: \ gi.gi\in set gs \Longrightarrow p.irreducible d}-m g
    and 8: p.square-free-m f
    and 9: coprime (lead-coeff f) p
    and sf-F: square-free F
    shows (\forallg\in set G'. irreducible ed g) ^F= prod-list G'
    using 1 irreducible-G3456789
proof (induction gs arbitrary: bfGG' rule: length-induct)
    case (1 gs)
    note LLL-f' = 1.prems(1)
    note irreducible-G = 1.prems(2)
    note F-f-G = 1.prems (3)
    note f-gs-factor = 1.prems (4)
    note gs-not-empty = 1.prems (5)
    note norm = 1.prems(6)
    note irred-p = 1.prems(7)
    note sf = 1.prems(8)
    note cop = 1.prems(9)
    obtain u where choose-u-result: choose-u gs =u by auto
    from choose-u-member[OF gs-not-empty, unfolded choose-u-result]
    have u-gs: }u\in\mathrm{ set gs by auto
    define d n where [simp]: d= degree u n= degree f
    hence }n\mathrm{ -def: }n=\mathrm{ degree f }n\equiv\mathrm{ degree f by auto
    define gg"' where gs" = remove1 u gs
    define degs where degs = map degree gs"
    define Degs where Degs =(+)d'sub-mset-sums degs
    obtain gs' b' h factor where inner-loop-result:
        LLL-reconstruction-inner-loop plgs fu Degs (d+1) = (gs',},\mp@subsup{b}{}{\prime},h,factor 
        by (metis prod-cases4)
    have a1:
        LLL-reconstruction-inner-loop plgs f u Degs (d+1) = (gs', b', h, factor 
        using inner-loop-result by auto
    have a?:
        \foralli\in{degree u+1..<(d+1)}. LLL-reconstruction-inner pl gs fu Degs i=None
        by auto
    have LLL-reconstruction" plgs bf G=LLL-reconstruction" pl gs' b'h(factor
# G)
    unfolding LLL-reconstruction".simps[of plgs] using gs-not-empty
    unfolding Let-def using choose-u-result inner-loop-result unfolding Degs-def
degs-def gs"'def by auto
    hence LLL-eq: LLL-reconstruction" pl gs'' b'h(factor # G)=G'using LLL-f'
by auto
    from pl.unique-factorization-m-imp-factorization[OF f-gs-factor,
        unfolded pl.factorization-m-def] norm
    have f-gs: pl.eq-m f (smult (lead-coeff f) (prod-mset (mset gs))) and
        mon: g\in set gs \Longrightarrow monic g}\mathrm{ and irred: g set gs C pl.irreducible d-m g for
g by auto
    {
        from split-list[OF u-gs] obtain gs1 gs2 where gs: gs=gs1 @ u # gs2 by
```

auto
from $f$-gs[unfolded $g s]$ have $p l . d v d m u f$ unfolding $p l . d v d m$-def
by (intro exI[of - smult (lead-coeff f) (prod-mset (mset (gs1 @ gs2)))], auto)
\} note $p l$-uf $=$ this
hence $p$-uf: p.dvdm $u f$ by (rule pl-dvdm-imp-p-dvdm)
have monic-u: monic $u$ using mon[OF u-gs].
have irred-u: p.irreducible-m $u$ using irred- $p[$ OF $u-g s]$ by auto
have degree-m-u: p.degree-m $u=$ degree $u$ using monic- $u$ by simp
have degree-u[simp]: $0<$ degree $u$
using irred- $u$ by (fold degree-m-u, auto simp add: p.irreducible-degree)
have deg-u-d: degree $u<d+1$ by auto
from $F-f-G$ have $f$-dvd- $F: f d v d F$ by auto
from square-free-factor $[O F \quad f$-dvd-F sf- $F]$ have $s f$-f: square-free $f$.
from norm have norm-map: map pl.Mp gs $=$ gs by (induct gs, auto)
\{
fix factor
assume factor-f: factor $d v d f$ and $u$-factor: $p . d v d m$ u factor
from factor- $f$ obtain $h$ where $f: f=$ factor $* h$ unfolding $d v d$-def by auto
obtain $g s 1$ gs2 where part: List.partition ( $\lambda$ gi. p.dvdm gi factor) $g s=(g s 1$,
gs2) by force
from $p$.unique-factorization-m-factor-partition[OF l0 f-gs-factor $f$ cop sf part]
have factor: pl.unique-factorization-m factor (lead-coeff factor, mset gs1) by
auto
from $u$-factor part $u$-gs have $u$-gs1: $u \in$ set gs1 by auto
define $g s 1^{\prime}$ where $g s 1^{\prime}=$ remove1 $u$ gs1
from remove1-mset[OF u-gs1, folded gs1'-def]
have $g s 1$ : mset gs1 = add-mset $u\left(m s e t ~ g s 1^{\prime}\right)$ by auto
from remove1-mset[OF u-gs, folded gs ${ }^{\prime \prime}$-def]
have gs: mset gs =add-mset $u$ ( mset gs ${ }^{\prime \prime}$ ) by auto
from part have filter: gs1 $=[g i \leftarrow g s . p . d v d m$ gi factor $]$ by auto
have mset gs1 $\subseteq \#$ mset gs unfolding filter mset-filter by simp
hence sub: mset gs1' $\subseteq \#$ mset gs ${ }^{\prime \prime}$ unfolding gs gs1 by auto
from p.coprime-lead-coeff-factor[OF〈prime p〉cop[unfolded f]]
have cop': coprime (lead-coeff factor) $p$ by auto
have $p$-factor $0: p . M p$ factor $\neq 0$
by (metis f p.Mp-0 p.square-free-m-def poly-mod.square-free-m-factor(1) sf)
have pl-factor0: pl.Mp factor $\neq 0$ using $p$-factor0 $l 0$
by (metis p.Mp-0 p-Mp-pl-Mp)
from pl.factorization-m-degree[OF pl.unique-factorization-m-imp-factorization[OF
factor] pl-factor0]
have pl.degree-m factor $=$ sum-mset $($ image-mset pl.degree-m $($ mset gs1 $))$.
also have image-mset pl.degree-m (mset gs1) = image-mset degree (mset gs1)
by (rule image-mset-cong, rule pl.monic-degree-m[OF mon], insert part, auto)
also have pl.degree-m factor $=$ degree factor
by (rule pl.degree-m-eq[OF p.coprime-exp-mod $\left[O F \operatorname{cop}^{\prime}\right.$ l0] pl.m1])
finally have degree factor $=d+$ sum-mset (image-mset degree (mset gs1'))
unfolding gs 1 by auto
moreover have sum-mset (image-mset degree (mset gs1')) $\in$ sub-mset-sums
degs unfolding degs-def
sub-mset-sums mset-map
by (intro imageI CollectI image-mset-subseteq-mono[OF sub])
ultimately have degree factor $\in$ Degs unfolding Degs-def by auto
\} note Degs $=$ this
have length-less: length gs ${ }^{\prime}<$ length gs
and irreducible-factor: irreducible $_{d}$ factor
and $h-d v d-f: h d v d f$
and $f$-h-factor: $f=h *$ factor
and $h$-eq: pl.unique-factorization-m $h\left(b^{\prime}\right.$, mset $\left.g s^{\prime}\right)$
and $g s^{\prime}-g s$ : set $g s^{\prime} \subseteq$ set $g s$
and $b^{\prime}: b^{\prime}=$ lead-coeff $h$
and $h 1: g s^{\prime}=[] \longrightarrow h=1$
using LLL-reconstruction-inner-loop[OF degree-u monic-u irred-u p-uf f-dvd-F $n-\operatorname{def}(2)$
f-gs-factor cop sf sf-f u-gs norm-map Degs
a1 a2 n-def(1)] deg-u-d gs-not-empty by metis+
have $F$-h-factor- $G$ : $F=h *$ prod-list (factor $\# G$ )
using $F$-f-G f-h-factor by auto
hence $h$-dvd- $F$ : $h$ dvd $F$ using $f$-dvd- $F$ dvd-trans by auto
have irreducible-factor- $G: \bigwedge x . x \in$ set (factor $\# G) \Longrightarrow$ irreducible $_{d} x$
using irreducible-factor irreducible- $G$ by auto
from p.coprime-lead-coeff-factor[OF 〈prime p〉cop[unfolded f-h-factor]]
have cop': coprime (lead-coeff h) $p$ by auto
have $l c^{\prime}$ : lead-coeff (smult (lead-coeff $h$ ) (prod-list gs')) lead-coeff $h$
by (insert gs'-gs, auto intro!: monic-prod-list intro: mon)
have lc: lead-coeff (pl.Mp (smult (lead-coeff h) (prod-list gs'))) =pl.M (lead-coeff h)
proof (subst pl.degree-m-eq-lead-coeff[OF pl.degree-m-eq[OF - pl.m1]]; unfold lc') show lead-coeff $h \bmod p \wedge l \neq 0$ using $p$.coprime-exp-mod $\left[O F \operatorname{cop}{ }^{\prime} l 0\right]$ by auto qed auto
have uh: pl.unique-factorization-m $h$ (lead-coeff $h$, mset gs') using $h$-eq unfolding $b^{\prime}$.
from $p$.square-free-m-factor[OF sf[unfolded $f$-h-factor]] have $s f^{\prime}: p . s q u a r e-f r e e-m$
$h$ by auto
show ?case
proof (cases gs' $\neq[]$ )
case gs'-not-empty: True
show ?thesis
by (rule 1.IH[rule-format, OF length-less LLL-eq irreducible-factor-G F-h-factor-G
uh gs'-not-empty norm irred-p $s f^{\prime}$ cop $\}$, insert gs'-gs, auto)
next
case False
have pl-ge0: $p^{\wedge} l>0$ using $p 1$ by auto
have $G^{\prime}$-eq: $G^{\prime}=$ factor $\# G$ using $L L L$-eq False using LLL-reconstruction'".simps
by auto
have condition1: $\left(\forall a \in\right.$ set $G^{\prime}$. irreducible $\left._{d} a\right)$ using irreducible-factor- $G G^{\prime}$-eq by auto
have $h$-eq2: pl.Mp $h=p l . M p\left[: b^{\prime}:\right]$ using $h$-eq False
unfolding pl.unique-factorization-m-alt-def pl.factorization-m-def by auto
have Mp-const-rw[simp]: pl.Mp $\left[: b^{\prime}:\right]=\left[: b^{\prime} \bmod p \uparrow:\right]$ using pl.Mp-const-poly by blast
have condition2: $F=$ prod-list $G^{\prime}$ using h1 False f-h-factor $G^{\prime}$-eq $F$-h-factor-G by auto
show ?thesis using condition1 condition2 by auto
qed
qed
context
fixes $g s$ :: int poly list
assumes gs-hen: berlekamp-hensel plF=gs
and cop: coprime (lead-coeff F) $p$
and $s f$ : poly-mod.square-free-m p $F$
and $s f$ - $F$ : square-free $F$
begin
lemma gs-not-empty: $g s \neq[]$
proof (rule ccontr, simp)
assume $g s: g s=[]$
obtain $c f s$ where $c$ - $f s$ : finite-field-factorization-int p $F=(c, f s)$ by force
have sort (map degree $f s$ ) $=$ sort ( map degree gs)
by (rule p.berlekamp-hensel-main(2)[OF - gs-hen cop sf c-fs], simp add: l0)
hence $f s$-empty: $f s=[]$ using $g s$ by (cases $f s$, auto)
hence $f s$ : mset $f s=\{\#\}$ by auto
have p.unique-factorization-m $F(c$, mset $f s)$ and $c: c \in\{0 . .<p\}$
using $p$.finite-field-factorization-int $[O F$ sf $c$ - $f s]$ by auto
hence $p$.factorization-m $F(c, m s e t f s)$
using $p$.unique-factorization-m-imp-factorization by auto
hence eq-m-F: p.eq-m $F[: c:]$ unfolding $f s$ p.factorization-m-def by auto
hence $0=p$.degree- $m F$ by (simp add: p.Mp-const-poly)
also have $\ldots=$ degree $F$ by (rule $p$.degree-m-eq[OF - p1], insert cop p1, auto)
finally have degree $F=0$..
thus False using NO by simp
qed
lemma reconstruction-of-algorithm-16-22:
assumes 1: reconstruction-of-algorithm-16-22 plgs $F=G$
shows $\left(\forall g \in\right.$ set $G$. irreducible $\left._{d} g\right) \wedge F=$ prod-list $G$
proof -
note $*=$ p.berlekamp-hensel-unique[OF cop sf gs-hen l0]
obtain $c f s$ where finite-field-factorization-int $p F=(c, f s)$ by force
from p.berlekamp-hensel-main[OF l0 gs-hen cop sf this]
show ?thesis
using 1 unfolding reconstruction-of-algorithm-16-22-def Let-def
by (intro LLL-reconstruction' ${ }^{\prime}[O F \cdots-$ gs-not-empty], insert $*$ sf sf-F cop, auto)
qed
end
end

### 7.3.4 Final statement

lemma factorization-algorithm-16-22:
assumes res: factorization-algorithm-16-22 $f=G$
and sff: square-free $f$
and deg: degree $f>0$
shows $\left(\forall g \in\right.$ set $G$. irreducible $\left.{ }_{d} g\right) \wedge f=$ prod-list $G$
proof -
let $? l c=$ lead-coeff $f$
define $p$ where $p \equiv$ suitable-prime-bz $f$
obtain $c$ gs where fff: finite-field-factorization-int $p f=(c, g s)$ by force
let ?degs = map degree gs
note res $=$ res[unfolded factorization-algorithm-16-22-def Let-def, folded p-def, unfolded fff split, folded]
from suitable-prime-bz[OF sff refl]
have prime: prime $p$ and cop: coprime ?lc $p$ and sf: poly-mod.square-free-m pf
unfolding $p$-def by auto
note res
from prime interpret poly-mod-prime $p$ by unfold-locales
define $K$ where $K=\mathcal{2}^{\wedge}(5 *$ degree $f *$ degree $f) *$ $\|f\|^{2}$ へ $(2 *$ degree $f)$
define $N$ where $N=$ sqrt-int-ceiling $K$
have $K 0: K \geq 0$ unfolding $K$-def by auto
have NO: $N \geq 0$ unfolding $N$-def sqrt-int-ceiling using KO
by (smt of-int-nonneg real-sqrt-ge-0-iff zero-le-ceiling)
define $n$ where $n=$ find-exponent $p N$
note res $=$ res[folded $n$-def[unfolded $N$-def $K$-def]]
note $n=$ find-exponent $[O F m 1$, of $N$, folded $n$-def]
note $b h=$ berlekamp-and-hensel-separated[OF cop sf refl fff $n(2)]$
note res $=$ res[folded bh(1)]
show ?thesis
proof (rule reconstruction-of-algorithm-16-22[OF prime deg - refl cop sf sff res])
from $n(1)$ have $N \leq p^{\wedge} n$ by $\operatorname{simp}$
hence $*: N^{\wedge} 2 \leq\left(p^{\wedge} n\right)^{\wedge} 2$
by (intro power-mono NO, auto)
show $\mathcal{Z}^{\wedge}(\text { degree } f)^{2} * B 2-L L L f \wedge(2 *$ degree $f) \leq\left(p^{\wedge} n\right)^{2}$
proof (rule order.trans $[O F-*]$ )
have $\mathbf{2}^{\wedge}(\text { degree } f)^{2} * B 2-L L L f \wedge(2 *$ degree $f)=K$
unfolding $K$-def B2-LLL-def by (simp add: ac-simps
power-mult-distrib power2-eq-square power-mult[symmetric] power-add[symmetric])
also have $\ldots \leq N^{2}$ unfolding $N$-def by (rule sqrt-int-ceiling-bound [OF K0])
finally show $2^{\wedge}(\text { degree } f)^{2} * B 2-L L L f \wedge(2 *$ degree $f) \leq N^{2}$.
qed
qed
qed
lift-definition increasing-lattices-LLL-factorization :: int-poly-factorization-algorithm
end

## 8 Mistakes in the textbook Modern Computer Algebra (2nd edition)

```
theory Modern-Computer-Algebra-Problem
imports Factorization-Algorithm-16-22
begin
```

```
fun max-degree-poly :: int poly \(\Rightarrow\) int poly \(\Rightarrow\) int poly
    where max-degree-poly \(a b=(\) if degree \(a \geq\) degree \(b\) then \(a\) else \(b\) )
fun choose- \(u\) :: int poly list \(\Rightarrow\) int poly
    where choose-u [] = undefined
    | choose-u \([g i]=g i\)
    \(\mid\) choose-u \((g i \# g j \# g s)=\) max-degree-poly gi (choose-u \((g j \# g s))\)
```


### 8.1 A real problem of Algorithm 16.22

Bogus example for Modern Computer Algebra (2nd edition), Algorithm 16.22, step 9: After having detected the factor [:1, $1,0,1:]$, the remaining polynomial $f^{*}$ will be 1 , and the remaining list of modular factors will be empty.

```
lemma let \(f=[: 1,1:] *[: 1,1,0,1:]\);
    \(p=\) suitable-prime-bz \(f\);
    \(b=\) lead-coeff \(f\);
    \(A=\operatorname{linf}\)-norm-poly \(f ; n=\) degree \(f ; B=\) sqrt-int-ceiling \((n+1) * 2 \uparrow n * A\);
    Bnd \(=\) 2^( \(^{\wedge}\) ^2 div 2) \(* B^{\wedge}(2 * n) ; l=\log\)-ceiling \(p\) Bnd;
    \((-, f s)=\) finite-field-factorization-int pf;
    \(g s=\) hensel-lifting plffs;
    \(u=\) choose-u gs;
    \(d=\) degree \(u\);
    g-star \(=[: 2,2,0,2::\) int :];
    \(\left(g s^{\prime}, h s^{\prime}\right)=\) List.partition ( \(\lambda\) gi. poly-mod.dvdm p gi g-star) gs;
    \(h\)-star \(=\) smult \(b(\) prod-list hs \()\);
    \(f\)-star \(=\) primitive-part h-star
    in \(\left(h s^{\prime}=[] \wedge f\right.\)-star \(\left.=1\right)\) by eval
```


### 8.2 Another potential problem of Algorithm 16.22

Suppose that $g^{*}$ is $p^{l}$. (It is is not yet clear whether lattices exists where this $g^{*}$ is short enough). Then $p p\left(g^{*}\right)=1$ is detected as irreducible factor and the algorithm stops.
definition input-poly $=[: 1,0,0,0,1,1,0,0,1,0,1,0,1::$ int $:]$
For input-poly the factorization will result in a lattice where each initial basis element has a Euclidean norm of at least $p^{l}$ (since the input polynomial $u$ has a norm larger than $p^{l}$.) So, just from the norm of the basis one cannot infer that the lattice contains small vectors.

```
lemma let \(f=\) input-poly;
    \(p=\) suitable-prime-bz f;
    \(b=\) lead-coeff \(f\);
    \(A=\) linf-norm-poly \(f ; n=\) degree \(f ; B=\) sqrt-int-ceiling \((n+1) * 2 \widehat{ } n * A\);
    Bnd \(=\) 2^( \(n\) ^2 div 2) \(* B^{\wedge}(2 * n) ; l=\log\)-ceiling \(p\) Bnd;
    \(\left(-, f_{s}\right)=\) finite-field-factorization-int pf;
    gs \(=\) hensel-lifting plffs;
    \(u=\) choose-u gs;
    \(p l=p^{\wedge} l ;\)
    \(p l 2=p l d i v 2 ;\)
    \(u^{\prime}=\) poly-mod.inv-Mp2 pl pl2 (poly-mod.Mp pl (smult bu))
in sqrt-int-floor (sq-norm \(u^{\prime}\) ) >pl by eval
```

The following calculation will show that the norm of $g^{*}$ is not that much shorter than $p^{l}$ which is an indication that it is not obvious that in general $p^{l}$ cannot be chosen as short polynomial.

```
definition compute-norms \(=(\) let \(f=\) input-poly;
    \(p=\) suitable-prime-bz f;
    \(b=\) lead-coeff \(f\);
    \(A=\) linf-norm-poly \(f ; n=\) degree \(f ; B=\) sqrt-int-ceiling \((n+1) * \mathcal{Z}^{\wedge} n * A\);
    Bnd \(=\) 2^ \(^{\text {( }}\) ^2 div 2 \() ~ * ~ B `(2 * n) ; l=\log\)-ceiling \(p B n d\);
    \(\left(-, f_{s}\right)=\) finite-field-factorization-int \(p f\);
    gs \(=\) hensel-lifting plffs;
    \(u=\) choose-u gs;
    \(p l=p^{\wedge} l ;\)
    \(p l 2=p l d i v 2 ;\)
    \(u^{\prime}=\) poly-mod.inv-Mp2 pl pl2 (poly-mod.Mp pl (smult bu));
    \(d=\) degree \(u\);
    \(p l=p \imath l ;\)
    \(L=\) factorization-lattice \(u^{\prime} 1\) pl;
    \(g\)-star \(=\) short-vector \(2 L\)
    in (
    "p 1 : " @ show pl @ shows-nl [] @
    "norm u: " @ show (sqrt-int-floor (sq-norm-poly u')) @ shows-nl [] @
    "norm g-star: " @ show (sqrt-int-floor (sq-norm-vec g-star)) @ shows-nl [] @
shows-nl []
))
```

export-code compute-norms in Haskell

- $p^{l}: \approx 6.61056 \cdot 10^{122}$, namely 6610559687902485989519153080327710398284046829642812192846
- norm $u: \approx 6.67555 \cdot 10^{122}$, namely 667555058938127908386141559707490406617756492853269306
- norm g-star: $\approx 5.02568 \cdot 10^{110}$, namely 50256787188889378925810759939795033899734873138630


### 8.3 Verified wrong results

An equality in example 16.24 of the textbook which is not valid.

```
lemma let g2 \(=[:-984,1:]\);
        g3 \(=[:-72,1:] ;\)
        \(g 4=[:-6828,1:] ;\)
        rhs \(=[:-1728,-840,-420,6:]\)
    in \(\neg\) poly-mod.eq-m \(\left(5^{\wedge} 6\right)(\) smult \(6(g 2 * g 3 * g 4))(r h s)\) by eval
end
```


## References

[1] J. Divasón, S. J. C. Joosten, R. Thiemann, and A. Yamada. A formalization of the Berlekamp-Zassenhaus factorization algorithm. In $C P P$ 2017, pages 17-29. ACM, 2017.
[2] J. v. z. Gathen and J. Gerhard. Modern Computer Algebra. Cambridge University Press, New York, NY, USA, 2nd edition, 2003.
[3] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen, 261:515-534, 1982.


[^0]:    *Supported by FWF (Austrian Science Fund) project Y757. Jose Divasón is partially funded by the Spanish project MTM2017-88804-P.

