

# VectorSpace

Holden Lee\*

February 23, 2021

## Abstract

I present a formalisation of basic linear algebra based completely on locales, building off HOL-Algebra. It includes the following:

1. basic definitions: linear combinations, span, linear independence
2. linear transformations
3. interpretation of function spaces as vector spaces
4. direct sum of vector spaces, sum of subspaces
5. the replacement theorem
6. existence of bases in finite-dimensional vector spaces, definition of dimension
7. rank-nullity theorem.

Note that some concepts are actually defined and proved for modules as they also apply there.

In the process, I also prove some basic facts about rings, modules, and fields, as well as finite sums in monoids/modules.

Note that infinite-dimensional vector spaces are supported, but dimension is only supported for finite-dimensional vector spaces.

The proofs are standard; the proofs of the replacement theorem and rank-nullity theorem roughly follow the presentation in [?]. The rank-nullity theorem generalises the existing development in [?] (originally using type classes, now using a mix of type classes and locales).

## Contents

<b>1 Basic facts about rings and modules</b>	<b>2</b>
1.1 Basic facts . . . . .	2
1.2 Units group . . . . .	3
<b>2 Basic lemmas about functions</b>	<b>4</b>
<b>3 Sums in monoids</b>	<b>4</b>

---

\*This work was funded by the Post-Masters Consultancy and the Computer Laboratory at the University of Cambridge.

<b>4</b>	<b>Linear Combinations</b>	<b>6</b>
4.1	Lemmas for simplification . . . . .	6
4.2	Linear combinations . . . . .	6
4.3	Linear dependence and independence. . . . .	8
4.4	Submodules . . . . .	10
<b>5</b>	<b>The direct sum of modules.</b>	<b>16</b>
<b>6</b>	<b>Basic theory of vector spaces, using locales</b>	<b>18</b>
6.1	Basic definitions and facts carried over from modules . . . . .	18
6.1.1	Facts specific to vector spaces . . . . .	21
6.2	Basic facts about span and linear independence . . . . .	22
6.3	The Replacement Theorem . . . . .	22
6.4	Defining dimension and bases. . . . .	23
6.5	The rank-nullity (dimension) theorem . . . . .	26

# 1 Basic facts about rings and modules

```
theory RingModuleFacts
imports Main
  HOL-Algebra.Module
  HOL-Algebra.Coset
```

begin

## 1.1 Basic facts

In a field, every nonzero element has an inverse.

```
lemma (in field) inverse-exists [simp, intro]:
  assumes h1: a ∈ carrier R and h2: a ≠ 0_R
  shows inv_R a ∈ carrier R
⟨proof⟩
```

Multiplication by 0 in  $R$  gives 0. (Note that this fact encompasses smult-l-null as this is for module while that is for algebra, so smult-l-null is redundant.)

```
lemma (in module) lmult-0 [simp]:
  assumes 1: m ∈ carrier M
  shows 0_R ⊙_M m = 0_M
⟨proof⟩
```

Multiplication by 0 in  $M$  gives 0.

```
lemma (in module) rmult-0 [simp]:
  assumes 0: r ∈ carrier R
  shows r ⊙_M 0_M = 0_M
⟨proof⟩
```

Multiplication by  $-1$  is the same as negation. May be useful as a simp rule.

**lemma** (in module) *smult-minus-1*:  
**fixes**  $v$   
**assumes**  $0:v \in \text{carrier } M$   
**shows**  $(\ominus_R \mathbf{1}_R) \odot_M v = (\ominus_M v)$

*<proof>*

The version with equality reversed.

**lemmas** (in module) *smult-minus-1-back = smult-minus-1[THEN sym]*

$-1$  is not  $0$

**lemma** (in field) *neg-1-not-0 [simp]*:  $\ominus_R \mathbf{1}_R \neq \mathbf{0}_R$   
*<proof>*

Note *smult-assoc1* is the wrong way around for simplification. This is the reverse of *smult-assoc1*.

**lemma** (in module) *smult-assoc-simp*:  
 $\llbracket a \in \text{carrier } R; b \in \text{carrier } R; x \in \text{carrier } M \rrbracket ==>$   
 $a \odot_M (b \odot_M x) = (a \otimes b) \odot_M x$   
*<proof>*

**lemmas** (in abelian-group) *show-r-zero = add.l-cancel-one*

**lemmas** (in abelian-group) *show-l-zero = add.r-cancel-one*

A nontrivial ring has  $0 \neq 1$ .

**lemma** (in ring) *nontrivial-ring [simp]*:  
**assumes**  $\text{carrier } R \neq \{\mathbf{0}_R\}$   
**shows**  $\mathbf{0}_R \neq \mathbf{1}_R$   
*<proof>*

Use as simp rule. To show  $a - b = 0$ , it suffices to show  $a = b$ .

**lemma** (in abelian-group) *minus-other-side [simp]*:  
 $\llbracket a \in \text{carrier } G; b \in \text{carrier } G \rrbracket \implies (a \ominus_G b = \mathbf{0}_G) = (a = b)$   
*<proof>*

## 1.2 Units group

Define the units group  $R^\times$  and show it is actually a group.

**definition** *units-group::('a,'b) ring-scheme  $\Rightarrow$  'a monoid*  
**where** *units-group*  $R = (\text{carrier} = \text{Units } R, \text{mult} = (\lambda x y. x \otimes_R y),$   
 $\text{one} = \mathbf{1}_R)$

The units form a group.

**lemma** (in *ring*) *units-form-group: group (units-group R)*  
⟨*proof*⟩

The units of a *cring* form a commutative group.

**lemma** (in *cring*) *units-form-cgroup: comm-group (units-group R)*  
⟨*proof*⟩

**end**

## 2 Basic lemmas about functions

**theory** *FunctionLemmas*

**imports** *Main*  
*HOL-Library.FuncSet*  
**begin**

These are used in simplification. Note that the difference from *Pi-mem* is that the statement about the function comes first, so Isabelle can more easily figure out what *S* is.

**lemma** *PiE-mem2: f ∈ S →<sub>E</sub> T ⇒ x ∈ S ⇒ f x ∈ T*  
⟨*proof*⟩

**lemma** *Pi-mem2: f ∈ S → T ⇒ x ∈ S ⇒ f x ∈ T*  
⟨*proof*⟩

**end**

## 3 Sums in monoids

**theory** *MonoidSums*

**imports** *Main*  
*HOL-Algebra.Module*  
*RingModuleFacts*  
*FunctionLemmas*  
**begin**

We build on the finite product simplifications in *FiniteProduct.thy* and the analogous ones for finite sums (see "lemmas" in *Ring.thy*).

Use as an intro rule

**lemma** (in *comm-monoid*) *factors-equal:*  
⟦*a=b; c=d*⟧ ⇒ *a ⊗<sub>G</sub> c = b ⊗<sub>G</sub> d*  
⟨*proof*⟩

**lemma** (in *comm-monoid*) *extend-prod*:  
**fixes**  $a A S$   
**assumes**  $fin$ : finite  $S$  **and**  $subset$ :  $A \subseteq S$  **and**  $a$ :  $a \in A \rightarrow carrier\ G$   
**shows**  $(\bigotimes_G x \in S. (if\ x \in A\ then\ a\ x\ else\ \mathbf{1}_G)) = (\bigotimes_G x \in A. a\ x)$   
**is**  $(\bigotimes_G x \in S. ?b\ x) = (\bigotimes_G x \in A. a\ x)$   
 $\langle proof \rangle$

Scalar multiplication distributes over scalar multiplication (on left).

**lemma** (in *module*) *finsum-smult*:  
 $[\![\ c \in carrier\ R; g \in A \rightarrow carrier\ M \ ]\!] ==>$   
 $(c \odot_M finsum\ M\ g\ A) = finsum\ M\ (\%x. c \odot_M\ g\ x)\ A$   
 $\langle proof \rangle$

Scalar multiplication distributes over scalar multiplication (on right).

**lemma** (in *module*) *finsum-smult-r*:  
 $[\![\ v \in carrier\ M; f \in A \rightarrow carrier\ R \ ]\!] ==>$   
 $(finsum\ R\ f\ A \odot_M\ v) = finsum\ M\ (\%x. f\ x \odot_M\ v)\ A$   
 $\langle proof \rangle$

A sequence of lemmas that shows that the product does not depend on the ambient group. Note I had to dig back into the definitions of *foldSet* to show this.

**lemma** *foldSet-not-depend*:  
**fixes**  $A E$   
**assumes**  $h1$ :  $D \subseteq E$   
**shows**  $foldSetD\ D\ f\ e \subseteq foldSetD\ E\ f\ e$   
 $\langle proof \rangle$

**lemma** *foldD-not-depend*:  
**fixes**  $D E B f e A$   
**assumes**  $h1$ :  $LCD\ B\ D\ f$  **and**  $h2$ :  $LCD\ B\ E\ f$  **and**  $h3$ :  $D \subseteq E$  **and**  
 $h4$ :  $e \in D$  **and**  $h5$ :  $A \subseteq B$  **and**  $h6$ : finite  $B$   
**shows**  $foldD\ D\ f\ e\ A = foldD\ E\ f\ e\ A$   
 $\langle proof \rangle$

**lemma** (in *comm-monoid*) *finprod-all1[simp]*:  
**assumes**  $all1$ :  $\bigwedge a. a \in A \implies f\ a = \mathbf{1}_G$   
**shows**  $(\bigotimes_G a \in A. f\ a) = \mathbf{1}_G$

$\langle proof \rangle$

**context** *abelian-monoid*  
**begin**  
**lemmas**  $summands-equal = add.factors-equal$

```

lemmas extend-sum = add.extend-prod
lemmas finsum-all0 = add.finprod-all1
end

end

```

## 4 Linear Combinations

```

theory LinearCombinations
imports Main
         HOL-Algebra.Module
         HOL-Algebra.Coset
         RingModuleFacts
         MonoidSums
         FunctionLemmas
begin

```

### 4.1 Lemmas for simplification

The following are helpful in certain simplifications (esp. congruence rules). Warning: arbitrary use leads to looping.

```

lemma (in ring) coeff-in-ring:
   $\llbracket a \in A \rightarrow \text{carrier } R; x \in A \rrbracket \Longrightarrow a \ x \in \text{carrier } R$ 
  <proof>

```

```

lemma (in ring) coeff-in-ring2:
   $\llbracket x \in A; a \in A \rightarrow \text{carrier } R \rrbracket \Longrightarrow a \ x \in \text{carrier } R$ 
  <proof>

```

```

lemma ring-subset-carrier:
   $\llbracket x \in A; A \subseteq \text{carrier } R \rrbracket \Longrightarrow x \in \text{carrier } R$ 
  <proof>

```

```

lemma disj-if:
   $\llbracket A \cap B = \{\}; x \in B \rrbracket \Longrightarrow (\text{if } x \in A \text{ then } f \ x \ \text{else } g \ x) = g \ x$ 
  <proof>

```

```

lemmas (in module) sum-simp = ring-subset-carrier

```

### 4.2 Linear combinations

A linear combination is  $\sum_{v \in A} a_v v$ .  $(a_v)_{v \in S}$  is a function  $A \rightarrow K$ , where  $A \subseteq K$ .

```

definition (in module) lincomb::['c  $\Rightarrow$  'a, 'c set]  $\Rightarrow$  'c
where lincomb a A =  $(\bigoplus_M v \in A. (a \ v \odot_M \ v))$ 

```

**lemma** (in module) *summands-valid*:  
**fixes**  $A a$   
**assumes**  $h2: A \subseteq \text{carrier } M$  **and**  $h3: a \in (A \rightarrow \text{carrier } R)$   
**shows**  $\forall v \in A. ((a v) \odot_M v) \in \text{carrier } M$   
 $\langle \text{proof} \rangle$

**lemma** (in module) *lincomb-closed* [*simp, intro*]:  
**fixes**  $S a$   
**assumes**  $h2: S \subseteq \text{carrier } M$  **and**  $h3: a \in (S \rightarrow \text{carrier } R)$   
**shows**  $\text{lincomb } a S \in \text{carrier } M$   
 $\langle \text{proof} \rangle$

**lemma** (in comm-monoid) *finprod-cong2*:  
 $[[ A = B;$   
 $!!i. i \in B ==> f i = g i; f \in B \rightarrow \text{carrier } G]] ==>$   
 $\text{finprod } G f A = \text{finprod } G g B$   
 $\langle \text{proof} \rangle$

**lemmas** (in abelian-monoid) *finsum-cong2 = add.finprod-cong2*

**lemma** (in module) *lincomb-cong*:  
**assumes**  $h2: A=B$  **and**  $h3: A \subseteq \text{carrier } M$   
**and**  $h4: \bigwedge v. v \in A \implies a v = b v$  **and**  $h5: b \in B \rightarrow \text{carrier } R$   
**shows**  $\text{lincomb } a A = \text{lincomb } b B$   
 $\langle \text{proof} \rangle$

**lemma** (in module) *lincomb-union*:  
**fixes**  $a A B$   
**assumes**  $h1: \text{finite } (A \cup B)$  **and**  $h3: A \cup B \subseteq \text{carrier } M$   
**and**  $h4: A \cap B = \{\}$  **and**  $h5: a \in (A \cup B \rightarrow \text{carrier } R)$   
**shows**  $\text{lincomb } a (A \cup B) = \text{lincomb } a A \oplus_M \text{lincomb } a B$   
 $\langle \text{proof} \rangle$

This is useful as a simp rule sometimes, for combining linear combinations.

**lemma** (in module) *lincomb-union2*:  
**fixes**  $a b A B$   
**assumes**  $h1: \text{finite } (A \cup B)$  **and**  $h3: A \cup B \subseteq \text{carrier } M$   
**and**  $h4: A \cap B = \{\}$  **and**  $h5: a \in A \rightarrow \text{carrier } R$  **and**  $h6: b \in B \rightarrow \text{carrier } R$   
**shows**  $\text{lincomb } a A \oplus_M \text{lincomb } b B = \text{lincomb } (\lambda v. \text{if } (v \in A) \text{ then } a v \text{ else } b v) (A \cup B)$   
**(is**  $\text{lincomb } a A \oplus_M \text{lincomb } b B = \text{lincomb } ?c (A \cup B)$   
 $\langle \text{proof} \rangle$

**lemma** (in module) *lincomb-del2*:  
**fixes**  $S a v$   
**assumes**  $h1: \text{finite } S$  **and**  $h2: S \subseteq \text{carrier } M$  **and**  $h3: a \in (S \rightarrow \text{carrier } R)$   
**and**  $h4: v \in S$

**shows**  $\text{lincomb } a \ S = ((a \ v) \odot_M v) \oplus_M \text{lincomb } a \ (S - \{v\})$   
 ⟨proof⟩

**lemma** (in module) *lincomb-insert*:  
**fixes**  $S \ a \ v$   
**assumes**  $h1$ : finite  $S$  **and**  $h2$ :  $S \subseteq \text{carrier } M$  **and**  $h3$ :  $a \in (S \cup \{v\}) \rightarrow \text{carrier } R$  **and**  $h4$ :  $v \notin S$  **and**  
 $h5$ :  $v \in \text{carrier } M$   
**shows**  $\text{lincomb } a \ (S \cup \{v\}) = ((a \ v) \odot_M v) \oplus_M \text{lincomb } a \ S$   
 ⟨proof⟩

**lemma** (in module) *lincomb-elim-if* [simp]:  
**fixes**  $b \ c \ S$   
**assumes**  $h1$ :  $S \subseteq \text{carrier } M$  **and**  $h2$ :  $\bigwedge v. v \in S \implies \neg P \ v$  **and**  $h3$ :  
 $c \in S \rightarrow \text{carrier } R$   
**shows**  $\text{lincomb } (\lambda w. \text{if } P \ w \ \text{then } b \ w \ \text{else } c \ w) \ S = \text{lincomb } c \ S$   
 ⟨proof⟩

**lemma** (in module) *lincomb-smult*:  
**fixes**  $A \ c$   
**assumes**  $h2$ :  $A \subseteq \text{carrier } M$  **and**  $h3$ :  $a \in A \rightarrow \text{carrier } R$  **and**  $h4$ :  
 $c \in \text{carrier } R$   
**shows**  $\text{lincomb } (\lambda w. c \otimes_R a \ w) \ A = c \odot_M (\text{lincomb } a \ A)$   
 ⟨proof⟩

### 4.3 Linear dependence and independence.

A set  $S$  in a module/vectorspace is linearly dependent if there is a finite set  $A \subseteq S$  and coefficients  $(a_v)_{v \in A}$  such that  $\text{sum}_{v \in A} a_v v = 0$  and for some  $v$ ,  $a_v \neq 0$ .

**definition** (in module) *lin-dep where*  
 $\text{lin-dep } S = (\exists A \ a \ v. (\text{finite } A \ \wedge \ A \subseteq S \ \wedge \ (a \in (A \rightarrow \text{carrier } R)) \ \wedge \ (\text{lincomb } a \ A = \mathbf{0}_M) \ \wedge \ (v \in A) \ \wedge \ (a \ v \neq \mathbf{0}_R)))$

**abbreviation** (in module) *lin-indpt::'c set  $\Rightarrow$  bool*  
**where**  $\text{lin-indpt } S \equiv \neg \text{lin-dep } S$

In the finite case, we can take  $A = S$ . This may be more convenient (e.g., when adding two linear combinations).

**lemma** (in module) *finite-lin-dep*:  
**fixes**  $S$   
**assumes**  $\text{fin } S$ : finite  $S$  **and**  $\text{ld}$ :  $\text{lin-dep } S$  **and**  $\text{in } C$ :  $S \subseteq \text{carrier } M$   
**shows**  $\exists a \ v. (a \in (S \rightarrow \text{carrier } R)) \ \wedge \ (\text{lincomb } a \ S = \mathbf{0}_M) \ \wedge \ (v \in S) \ \wedge \ (a \ v \neq \mathbf{0}_R)$   
 ⟨proof⟩



Criteria of linear dependency in a easy format to apply: apply  
(rule lin-dep-crit)

**lemma** (in module) *lin-dep-crit*:

**fixes**  $A S a v$   
**assumes** *fin*: finite  $A$  **and** *subset*:  $A \subseteq S$  **and** *h1*: ( $a \in (A \rightarrow \text{carrier } R)$ ) **and** *h2*:  $v \in A$   
**and** *h3*:  $a v \neq \mathbf{0}_R$  **and** *h4*: ( $\text{lincomb } a A = \mathbf{0}_M$ )  
**shows** *lin-dep*  $S$   
 $\langle \text{proof} \rangle$

If  $\sum_{v \in A} a_v v = 0$  implies  $a_v = 0$  for all  $v \in S$ , then  $A$  is linearly independent.

**lemma** (in module) *finite-lin-indpt2*:

**fixes**  $A$   
**assumes** *A-fin*: finite  $A$  **and** *AinC*:  $A \subseteq \text{carrier } M$  **and**  
 $lc0$ :  $\bigwedge a. a \in (A \rightarrow \text{carrier } R) \implies (\text{lincomb } a A = \mathbf{0}_M) \implies (\forall v \in A. a v = \mathbf{0}_R)$   
**shows** *lin-indpt*  $A$   
 $\langle \text{proof} \rangle$

Any set containing 0 is linearly dependent.

**lemma** (in module) *zero-lin-dep*:

**assumes**  $0$ :  $\mathbf{0}_M \in S$  **and** *nonzero*:  $\text{carrier } R \neq \{\mathbf{0}_R\}$   
**shows** *lin-dep*  $S$   
 $\langle \text{proof} \rangle$

**lemma** (in module) *zero-nin-lin-indpt*:

**assumes** *h2*:  $S \subseteq \text{carrier } M$  **and** *li*:  $\neg(\text{lin-dep } S)$  **and** *nonzero*:  $\text{carrier } R \neq \{\mathbf{0}_R\}$   
**shows**  $\mathbf{0}_M \notin S$   
 $\langle \text{proof} \rangle$

The *span* of  $S$  is the set of linear combinations with  $A \subseteq S$ .

**definition** (in module) *span::'c set  $\Rightarrow$  'c set*

**where**  $\text{span } S = \{\text{lincomb } a A \mid a A. \text{finite } A \wedge A \subseteq S \wedge a \in (A \rightarrow \text{carrier } R)\}$

The *span* interpreted as a module or vectorspace.

**abbreviation** (in module) *span-vs::'c set  $\Rightarrow$  ('a,'c,'d) module-scheme*

**where**  $\text{span-vs } S \equiv M \ (\text{carrier} := \text{span } S)$

In the finite case, we can take  $A = S$  without loss of generality.

**lemma** (in module) *finite-span*:

**assumes** *fin*: finite  $S$  **and** *inC*:  $S \subseteq \text{carrier } M$   
**shows**  $\text{span } S = \{\text{lincomb } a S \mid a. a \in (S \rightarrow \text{carrier } R)\}$   
 $\langle \text{proof} \rangle$

If  $v \in \text{span } S$ , then we can find a linear combination. This is in an easy to apply format (e.g. obtain a  $A$  where...)

**lemma** (*in module*) *in-span*:  
**fixes**  $S v$   
**assumes**  $h2: S \subseteq \text{carrier } V$  **and**  $h3: v \in \text{span } S$   
**shows**  $\exists a A. (A \subseteq S \wedge (a \in A \rightarrow \text{carrier } R) \wedge (\text{lincomb } a A = v))$   
*<proof>*

In the finite case, we can take  $A = S$ .

**lemma** (*in module*) *finite-in-span*:  
**fixes**  $S v$   
**assumes**  $fin: \text{finite } S$  **and**  $h2: S \subseteq \text{carrier } M$  **and**  $h3: v \in \text{span } S$   
**shows**  $\exists a. (a \in S \rightarrow \text{carrier } R) \wedge (\text{lincomb } a S = v)$   
*<proof>*

If a subset is linearly independent, then any linear combination that is 0 must have a nonzero coefficient outside that set.

**lemma** (*in module*) *lincomb-must-include*:  
**fixes**  $A S T b v$   
**assumes**  $inC: T \subseteq \text{carrier } M$  **and**  $li: \text{lin-indpt } S$  **and**  $Ssub: S \subseteq T$   
**and**  $Ssub: A \subseteq T$   
**and**  $fin: \text{finite } A$   
**and**  $b: b \in A \rightarrow \text{carrier } R$  **and**  $lc: \text{lincomb } b A = \mathbf{0}_M$  **and**  $v-in: v \in A$   
**and**  $nz-coeff: b v \neq \mathbf{0}_R$   
**shows**  $\exists w \in A - S. b w \neq \mathbf{0}_R$   
*<proof>*

A generating set is a set such that the span of  $S$  is all of  $M$ .

**abbreviation** (*in module*) *gen-set::'c set  $\Rightarrow$  bool*  
**where**  $\text{gen-set } S \equiv (\text{span } S = \text{carrier } M)$

## 4.4 Submodules

**lemma** *module-criteria*:  
**fixes**  $R$  **and**  $M$   
**assumes**  $cring: \text{cring } R$   
**and**  $zero: \mathbf{0}_M \in \text{carrier } M$   
**and**  $add: \forall v w. v \in \text{carrier } M \wedge w \in \text{carrier } M \rightarrow v \oplus_M w \in \text{carrier } M$   
**and**  $neg: \forall v \in \text{carrier } M. (\exists \text{neg-}v \in \text{carrier } M. v \oplus_M \text{neg-}v = \mathbf{0}_M)$   
**and**  $smult: \forall c v. c \in \text{carrier } R \wedge v \in \text{carrier } M \rightarrow c \odot_M v \in \text{carrier } M$   
**and**  $comm: \forall v w. v \in \text{carrier } M \wedge w \in \text{carrier } M \rightarrow v \oplus_M w = w \oplus_M v$   
**and**  $assoc: \forall v w x. v \in \text{carrier } M \wedge w \in \text{carrier } M \wedge x \in \text{carrier } M \rightarrow (v \oplus_M w) \oplus_M x = v \oplus_M (w \oplus_M x)$   
**and**  $add-id: \forall v \in \text{carrier } M. (v \oplus_M \mathbf{0}_M = v)$   
**and**  $compat: \forall a b v. a \in \text{carrier } R \wedge b \in \text{carrier } R \wedge v \in \text{carrier } M \rightarrow (a \otimes_R b) \odot_M v = a \odot_M (b \odot_M v)$

**and smult-id:**  $\forall v \in \text{carrier } M. (\mathbf{1}_R \odot_M v = v)$   
**and dist-f:**  $\forall a \ b \ v. a \in \text{carrier } R \wedge b \in \text{carrier } R \wedge v \in \text{carrier } M \longrightarrow (a \oplus_R b) \odot_M v = (a \odot_M v) \oplus_M (b \odot_M v)$   
**and dist-add:**  $\forall a \ v \ w. a \in \text{carrier } R \wedge v \in \text{carrier } M \wedge w \in \text{carrier } M \longrightarrow a \odot_M (v \oplus_M w) = (a \odot_M v) \oplus_M (a \odot_M w)$   
**shows module**  $R \ M$   
 $\langle \text{proof} \rangle$

A submodule is  $N \subseteq M$  that is closed under addition and scalar multiplication, and contains 0 (so is not empty).

**locale submodule =**  
**fixes**  $R$  **and**  $N$  **and**  $M$  **(structure)**  
**assumes** *module:* *module*  $R \ M$   
**and** *subset:*  $N \subseteq \text{carrier } M$   
**and** *m-closed* [*intro*, *simp*]:  $\llbracket v \in N; w \in N \rrbracket \Longrightarrow v \oplus w \in N$   
**and** *zero-closed* [*simp*]:  $\mathbf{0} \in N$   
**and** *smult-closed* [*intro*, *simp*]:  $\llbracket c \in \text{carrier } R; v \in N \rrbracket \Longrightarrow c \odot v \in N$

**abbreviation (in module)**  $md::'c \text{ set} \Rightarrow ('a, 'c, 'd) \text{ module-scheme}$   
**where**  $md \ N \equiv M(\downarrow \text{carrier} := N)$

**lemma (in module)** *carrier-vs-is-self* [*simp*]:  
 $\text{carrier } (md \ N) = N$   
 $\langle \text{proof} \rangle$

**lemma (in module)** *submodule-is-module:*  
**fixes**  $N::'c \text{ set}$   
**assumes**  $0$ : *submodule*  $R \ N \ M$   
**shows** *module*  $R \ (md \ N)$   
 $\langle \text{proof} \rangle$

$$N_1 + N_2 = \{x + y \mid x \in N_1, y \in N_2\}$$

**definition (in module)** *submodule-sum*:: [*'c set*, *'c set*]  $\Rightarrow$  *'c set*  
**where** *submodule-sum*  $N1 \ N2 = (\lambda (x,y). x \oplus_M y) \ \{(x,y). x \in N1 \wedge y \in N2\}$

A module homomorphism  $M \rightarrow N$  preserves addition and scalar multiplication.

**definition** *module-hom*:: [*'a*, *'c0*] *ring-scheme*,  
 $(('a, 'b1, 'c1) \text{ module-scheme}, ('a, 'b2, 'c2) \text{ module-scheme}) \Rightarrow ('b1 \Rightarrow 'b2) \text{ set}$   
**where** *module-hom*  $R \ M \ N = \{f.$   
 $((f \in \text{carrier } M \rightarrow \text{carrier } N)$   
 $\wedge (\forall m1 \ m2. m1 \in \text{carrier } M \wedge m2 \in \text{carrier } M \longrightarrow f (m1 \oplus_M m2)$   
 $= (f \ m1) \oplus_N (f \ m2))$   
 $\wedge (\forall r \ m. r \in \text{carrier } R \wedge m \in \text{carrier } M \longrightarrow f (r \odot_M m) = r \odot_N (f \ m))\}$

**lemma** *module-hom-closed*:  $f \in \text{module-hom } R \ M \ N \implies f \in \text{carrier } M \rightarrow \text{carrier } N$

*<proof>*

**lemma** *module-hom-add*:  $\llbracket f \in \text{module-hom } R \ M \ N; m1 \in \text{carrier } M; m2 \in \text{carrier } M \rrbracket \implies f (m1 \oplus_M m2) = (f m1) \oplus_N (f m2)$

*<proof>*

**lemma** *module-hom-smult*:  $\llbracket f \in \text{module-hom } R \ M \ N; r \in \text{carrier } R; m \in \text{carrier } M \rrbracket \implies f (r \odot_M m) = r \odot_N (f m)$

*<proof>*

**locale** *mod-hom* =

*M*? : *module* *R* *M* + *N*? : *module* *R* *N*

**for** *R* **and** *M* **and** *N* +

**fixes** *f*

**assumes** *f-hom*:  $f \in \text{module-hom } R \ M \ N$

**notes** *f-add* [*simp*] = *module-hom-add* [*OF f-hom*]

**and** *f-smult* [*simp*] = *module-hom-smult* [*OF f-hom*]

Some basic simplification rules for module homomorphisms.

**context** *mod-hom*

**begin**

**lemma** *f-im* [*simp, intro*]:

**assumes**  $v \in \text{carrier } M$

**shows**  $f v \in \text{carrier } N$

*<proof>*

**definition** *im*:: 'e set

**where**  $im = f'(\text{carrier } M)$

**definition** *ker*:: 'c set

**where**  $ker = \{v. v \in \text{carrier } M \ \& \ f v = \mathbf{0}_N\}$

**lemma** *f0-is-0*[*simp*]:  $f \ \mathbf{0}_M = \mathbf{0}_N$

*<proof>*

**lemma** *f-neg* [*simp*]:  $v \in \text{carrier } M \implies f (\ominus_M v) = \ominus_N f v$

*<proof>*

**lemma** *f-minus* [*simp*]:  $\llbracket v \in \text{carrier } M; w \in \text{carrier } M \rrbracket \implies f (v \ominus_M w) =$

$f v \ominus_N f w$

*<proof>*

**lemma** *ker-is-submodule*: *submodule* *R* *ker* *M*

*<proof>*

**lemma** *im-is-submodule*: *submodule* *R* *im* *N*

*<proof>*

**lemma** (in *mod-hom*) *f-ker*:

$v \in \text{ker} \implies f v = \mathbf{0}_N$

*<proof>*

**end**

We will show that for any set  $S$ , the space of functions  $S \rightarrow K$  forms a vector space.

**definition** (in *ring*) *func-space*::  $'z \text{ set} \Rightarrow ('a, ('z \Rightarrow 'a)) \text{ module}$

**where** *func-space*  $S = (\text{carrier} = S \rightarrow_E \text{carrier } R,$

$\text{mult} = (\lambda f g. \text{restrict } (\lambda v. \mathbf{0}_R) S),$

$\text{one} = \text{restrict } (\lambda v. \mathbf{0}_R) S,$

$\text{zero} = \text{restrict } (\lambda v. \mathbf{0}_R) S,$

$\text{add} = (\lambda f g. \text{restrict } (\lambda v. f v \oplus_R g v) S),$

$\text{smult} = (\lambda c f. \text{restrict } (\lambda v. c \otimes_R f v) S))$

**lemma** (in *cring*) *func-space-is-module*:

**fixes**  $S$

**shows** *module*  $R$  (*func-space*  $S$ )

*<proof>*

Note: one can define  $M^n$  from this.

A linear combination is a module homomorphism from the space of coefficients to the module,  $(a_v) \mapsto \sum_{v \in S} a_v v$ .

**lemma** (in *module*) *lincomb-is-mod-hom*:

**fixes**  $S$

**assumes**  $h$ : *finite*  $S$  **and**  $h2$ :  $S \subseteq \text{carrier } M$

**shows** *mod-hom*  $R$  (*func-space*  $S$ )  $M$  ( $\lambda a. \text{lincomb } a S$ )

*<proof>*

**lemma** (in *module*) *lincomb-sum*:

**assumes**  $A$ -*fin*: *finite*  $A$  **and**  $A$  in  $C$ :  $A \subseteq \text{carrier } M$  **and**  $a$ -*fun*:  $a \in A \rightarrow \text{carrier } R$  **and**

$b$ -*fun*:  $b \in A \rightarrow \text{carrier } R$

**shows** *lincomb*  $(\lambda v. a v \oplus_R b v) A = \text{lincomb } a A \oplus_M \text{lincomb } b A$

*<proof>*

The negative of a function is just pointwise negation.

**lemma** (in *cring*) *func-space-neg*:

**fixes**  $f$

**assumes**  $f \in \text{carrier}$  (*func-space*  $S$ )

**shows**  $\ominus_{\text{func-space } S} f = (\lambda v. \text{if } (v \in S) \text{ then } \ominus_R f v \text{ else undefined})$

*<proof>*

Ditto for subtraction. Note the above is really a special case, when  $a$  is the 0 function.

**lemma** (in module) *lincomb-diff*:  
**assumes** *A-fin*: finite *A* **and** *AinC*:  $A \subseteq \text{carrier } M$  **and** *a-fun*:  $a \in A \rightarrow \text{carrier } R$  **and**  
*b-fun*:  $b \in A \rightarrow \text{carrier } R$   
**shows**  $\text{lincomb } (\lambda v. a \ v \oplus_R b \ v) \ A = \text{lincomb } a \ A \oplus_M \text{lincomb } b \ A$   
 $\langle \text{proof} \rangle$

The union of nested submodules is a submodule. We will use this to show that span of any set is a submodule.

**lemma** (in module) *nested-union-vs*:  
**fixes**  $I \ N \ N'$   
**assumes** *subm*:  $\bigwedge i. i \in I \implies \text{submodule } R \ (N \ i) \ M$   
**and** *max-exists*:  $\bigwedge i \ j. i \in I \implies j \in I \implies (\exists k. k \in I \wedge N \ i \subseteq N \ k \wedge N \ j \subseteq N \ k)$   
**and** *uni*:  $N' = (\bigcup i \in I. N \ i)$   
**and** *ne*:  $I \neq \{\}$   
**shows**  $\text{submodule } R \ N' \ M$   
 $\langle \text{proof} \rangle$

**lemma** (in module) *span-is-monotone*:  
**fixes**  $S \ T$   
**assumes** *subs*:  $S \subseteq T$   
**shows**  $\text{span } S \subseteq \text{span } T$   
 $\langle \text{proof} \rangle$

**lemma** (in module) *span-is-submodule*:  
**fixes**  $S$   
**assumes** *h2*:  $S \subseteq \text{carrier } M$   
**shows**  $\text{submodule } R \ (\text{span } S) \ M$   
 $\langle \text{proof} \rangle$

A finite sum does not depend on the ambient module. This can be done for monoid, but "submonoid" isn't currently defined. (It can be copied, however, for groups...) This lemma requires a somewhat annoying lemma `foldD-not-depend`. Then we show that linear combinations, linear independence, span do not depend on the ambient module.

**lemma** (in module) *finsum-not-depend*:  
**fixes**  $a \ A \ N$   
**assumes** *h1*: finite  $A$  **and** *h2*:  $A \subseteq N$  **and** *h3*:  $\text{submodule } R \ N \ M$  **and**  
*h4*:  $f: A \rightarrow N$   
**shows**  $(\bigoplus_{(md \ N)} v \in A. f \ v) = (\bigoplus_M v \in A. f \ v)$   
 $\langle \text{proof} \rangle$

**lemma** (in module) *lincomb-not-depend*:  
**fixes**  $a \ A \ N$   
**assumes** *h1*: finite  $A$  **and** *h2*:  $A \subseteq N$  **and** *h3*:  $\text{submodule } R \ N \ M$  **and**

*h4*:  $a:A \rightarrow \text{carrier } R$   
**shows**  $\text{lincomb } a \ A = \text{module.lincomb } (\text{md } N) \ a \ A$   
 $\langle \text{proof} \rangle$

**lemma** (**in** *module*) *span-li-not-depend*:  
**fixes**  $S \ N$   
**assumes**  $h2: S \subseteq N$  **and**  $h3: \text{submodule } R \ N \ M$   
**shows**  $\text{module.span } R \ (\text{md } N) \ S = \text{module.span } R \ M \ S$   
**and**  $\text{module.lin-dep } R \ (\text{md } N) \ S = \text{module.lin-dep } R \ M \ S$   
 $\langle \text{proof} \rangle$

**lemma** (**in** *module*) *span-is-subset*:  
**fixes**  $S \ N$   
**assumes**  $h2: S \subseteq N$  **and**  $h3: \text{submodule } R \ N \ M$   
**shows**  $\text{span } S \subseteq N$   
 $\langle \text{proof} \rangle$

**lemma** (**in** *module*) *span-is-subset2*:  
**fixes**  $S$   
**assumes**  $h2: S \subseteq \text{carrier } M$   
**shows**  $\text{span } S \subseteq \text{carrier } M$   
 $\langle \text{proof} \rangle$

**lemma** (**in** *module*) *in-own-span*:  
**fixes**  $S$   
**assumes**  $\text{in } C: S \subseteq \text{carrier } M$   
**shows**  $S \subseteq \text{span } S$   
 $\langle \text{proof} \rangle$

**lemma** (**in** *module*) *supset-ld-is-ld*:  
**fixes**  $A \ B$   
**assumes**  $ld: \text{lin-dep } A$  **and**  $\text{sub}: A \subseteq B$   
**shows**  $\text{lin-dep } B$   
 $\langle \text{proof} \rangle$

**lemma** (**in** *module*) *subset-li-is-li*:  
**fixes**  $A \ B$   
**assumes**  $li: \text{lin-indpt } A$  **and**  $\text{sub}: B \subseteq A$   
**shows**  $\text{lin-indpt } B$   
 $\langle \text{proof} \rangle$

**lemma** (**in** *mod-hom*) *hom-sum*:  
**fixes**  $A \ B \ g$   
**assumes**  $h2: A \subseteq \text{carrier } M$  **and**  $h3: g: A \rightarrow \text{carrier } M$   
**shows**  $f \left( \bigoplus_M a \in A. g \ a \right) = \left( \bigoplus_N a \in A. f \ (g \ a) \right)$   
 $\langle \text{proof} \rangle$

end

## 5 The direct sum of modules.

```
theory SumSpaces
imports Main
  HOL-Algebra.Module
  HOL-Algebra.Coset
  RingModuleFacts
  MonoidSums
  FunctionLemmas
  LinearCombinations
begin
```

We define the direct sum  $M_1 \oplus M_2$  of 2 vector spaces as the set  $M_1 \times M_2$  under componentwise addition and scalar multiplication.

```
definition direct-sum:: ('a,'b, 'd) module-scheme  $\Rightarrow$  ('a, 'c, 'e) module-scheme  $\Rightarrow$  ('a, ('b $\times$ 'c)) module
  where direct-sum M1 M2 = ( $\downarrow$ carrier = carrier M1  $\times$  carrier M2,
    mult = ( $\lambda$  v w. ( $\mathbf{0}_{M1}$ ,  $\mathbf{0}_{M2}$ )),
    one = ( $\mathbf{0}_{M1}$ ,  $\mathbf{0}_{M2}$ ),
    zero = ( $\mathbf{0}_{M1}$ ,  $\mathbf{0}_{M2}$ ),
    add = ( $\lambda$  v w. (fst v  $\oplus_{M1}$  fst w, snd v  $\oplus_{M2}$  snd w)),
    smult = ( $\lambda$  c v. (c  $\odot_{M1}$  fst v, c  $\odot_{M2}$  snd v)))
```

**lemma** *direct-sum-is-module:*

```
fixes R M1 M2
assumes h1: module R M1 and h2: module R M2
shows module R (direct-sum M1 M2)
<proof>
```

```
definition inj1:: ('a,'b) module  $\Rightarrow$  ('a, 'c) module  $\Rightarrow$  ('b $\Rightarrow$ ('b $\times$ 'c))
  where inj1 M1 M2 = ( $\lambda$ v. (v,  $\mathbf{0}_{M2}$ ))
```

```
definition inj2:: ('a,'b) module  $\Rightarrow$  ('a, 'c) module  $\Rightarrow$  ('c $\Rightarrow$ ('b $\times$ 'c))
  where inj2 M1 M2 = ( $\lambda$ v. ( $\mathbf{0}_{M1}$ , v))
```

**lemma** *inj1-hom:*

```
fixes R M1 M2
assumes h1: module R M1 and h2: module R M2
shows mod-hom R M1 (direct-sum M1 M2) (inj1 M1 M2)
<proof>
```

**lemma** *inj2-hom:*

```
fixes R M1 M2
assumes h1: module R M1 and h2: module R M2
shows mod-hom R M2 (direct-sum M1 M2) (inj2 M1 M2)
```



*<proof>*

For submodules  $M_1, M_2 \subseteq M$ , the map  $M_1 \oplus M_2 \rightarrow M$  given by  $(m_1, m_2) \mapsto m_1 + m_2$  is linear.

**lemma** (*in module*) *sum-map-hom*:

**fixes**  $M1\ M2$

**assumes**  $h1$ : submodule  $R\ M1\ M$  **and**  $h2$ : submodule  $R\ M2\ M$

**shows**  $mod\text{-}hom\ R\ (direct\text{-}sum\ (md\ M1)\ (md\ M2))\ M\ (\lambda\ v.\ (fst\ v)\ \oplus_M\ (snd\ v))$

*<proof>*

**lemma** (*in module*) *sum-is-submodule*:

**fixes**  $N1\ N2$

**assumes**  $h1$ : submodule  $R\ N1\ M$  **and**  $h2$ : submodule  $R\ N2\ M$

**shows** submodule  $R\ (submodule\text{-}sum\ N1\ N2)\ M$

*<proof>*

**lemma** (*in module*) *in-sum*:

**fixes**  $N1\ N2$

**assumes**  $h1$ : submodule  $R\ N1\ M$  **and**  $h2$ : submodule  $R\ N2\ M$

**shows**  $N1 \subseteq submodule\text{-}sum\ N1\ N2$

*<proof>*

**lemma** (*in module*) *msum-comm*:

**fixes**  $N1\ N2$

**assumes**  $h1$ : submodule  $R\ N1\ M$  **and**  $h2$ : submodule  $R\ N2\ M$

**shows**  $(submodule\text{-}sum\ N1\ N2) = (submodule\text{-}sum\ N2\ N1)$

*<proof>*

If  $M_1, M_2 \subseteq M$  are submodules, then  $M_1 + M_2$  is the minimal subspace such that both  $M_1 \subseteq M$  and  $M_2 \subseteq M$ .

**lemma** (*in module*) *sum-is-minimal*:

**fixes**  $N\ N1\ N2$

**assumes**  $h1$ : submodule  $R\ N1\ M$  **and**  $h2$ : submodule  $R\ N2\ M$  **and**  $h3$ : submodule  $R\ N\ M$

**shows**  $(submodule\text{-}sum\ N1\ N2) \subseteq N \iff N1 \subseteq N \wedge N2 \subseteq N$

*<proof>*

$span\ A \cup B = span\ A + span\ B$

**lemma** (*in module*) *span-union-is-sum*:

**fixes**  $A\ B$

**assumes**  $h2$ :  $A \subseteq carrier\ M$  **and**  $h3$ :  $B \subseteq carrier\ M$

**shows**  $span\ (A \cup B) = submodule\text{-}sum\ (span\ A)\ (span\ B)$

*<proof>*

**end**

## 6 Basic theory of vector spaces, using locales

```
theory VectorSpace
imports Main
  HOL-Algebra.Module
  HOL-Algebra.Coset
  RingModuleFacts
  MonoidSums
  LinearCombinations
  SumSpaces
begin
```

### 6.1 Basic definitions and facts carried over from modules

A *vectorspace* is a module where the ring is a field. Note that we switch notation from  $(R, M)$  to  $(K, V)$ .

```
locale vectorspace =
  module?: module K V + field?: field K
for K and V
```

A *subspace* of a vectorspace is a nonempty subset that is closed under addition and scalar multiplication. These properties have already been defined in submodule. Caution:  $W$  is a set, while  $V$  is a module record. To get  $W$  as a vectorspace, write  $vs\ W$ .

```
locale subspace =
  fixes K and W and V (structure)
  assumes vs: vectorspace K V
  and submod: submodule K W V
```

```
lemma (in vectorspace) is-module[simp]:
  subspace K W V  $\implies$  submodule K W V
<proof>
```

We introduce some basic facts and definitions copied from module. We introduce some abbreviations, to match convention.

```
abbreviation (in vectorspace) vs::'c set  $\Rightarrow$  ('a, 'c, 'd) module-scheme
where vs W  $\equiv$  V(carrier := W)
```

```
lemma (in vectorspace) carrier-vs-is-self [simp]:
  carrier (vs W) = W
<proof>
```

```
lemma (in vectorspace) subspace-is-vs:
fixes W::'c set
```

**assumes**  $0$ : *subspace*  $K$   $W$   $V$   
**shows** *vectorspace*  $K$  (*vs*  $W$ )  
 $\langle$ *proof* $\rangle$

**abbreviation** (**in** *module*) *subspace-sum*:: [*'c set*, *'c set*]  $\Rightarrow$  *'c set*  
**where** *subspace-sum*  $W1$   $W2 \equiv$  *submodule-sum*  $W1$   $W2$

**lemma** (**in** *vectorspace*) *vs-zero-lin-dep*:  
**assumes**  $h2$ :  $S \subseteq$  *carrier*  $V$  **and**  $h3$ : *lin-indpt*  $S$   
**shows**  $\mathbf{0}_V \notin S$   
 $\langle$ *proof* $\rangle$

A *linear-map* is a module homomorphism between 2 vectorspaces over the same field.

**locale** *linear-map* =  
 $V?$ : *vectorspace*  $K$   $V$  +  $W?$ : *vectorspace*  $K$   $W$   
+ *mod-hom?*: *mod-hom*  $K$   $V$   $W$   $T$   
**for**  $K$  **and**  $V$  **and**  $W$  **and**  $T$

**context** *linear-map*  
**begin**  
**lemmas**  $T$ -*hom* =  $f$ -*hom*  
**lemmas**  $T$ -*add* =  $f$ -*add*  
**lemmas**  $T$ -*smult* =  $f$ -*smult*  
**lemmas**  $T$ -*im* =  $f$ -*im*  
**lemmas**  $T$ -*neg* =  $f$ -*neg*  
**lemmas**  $T$ -*minus* =  $f$ -*minus*  
**lemmas**  $T$ -*ker* =  $f$ -*ker*

**abbreviation** *imT*:: *'e set*  
**where** *imT*  $\equiv$  *mod-hom.im*

**abbreviation** *kerT*:: *'c set*  
**where** *kerT*  $\equiv$  *mod-hom.ker*

**lemmas**  $T0$ -*is-0*[*simp*] =  $f0$ -*is-0*

**lemma** *kerT-is-subspace*: *subspace*  $K$  *ker*  $V$   
 $\langle$ *proof* $\rangle$

**lemma** *imT-is-subspace*: *subspace*  $K$  *imT*  $W$   
 $\langle$ *proof* $\rangle$   
**end**

**lemma** *vs-criteria*:  
**fixes**  $K$  **and**  $V$   
**assumes** *field*: *field*  $K$   
**and** *zero*:  $\mathbf{0}_V \in$  *carrier*  $V$   
**and** *add*:  $\forall v w. v \in$  *carrier*  $V \wedge w \in$  *carrier*  $V \longrightarrow v \oplus_V w \in$  *carrier*

$V$   
**and neg:**  $\forall v \in \text{carrier } V. (\exists \text{ neg-}v \in \text{carrier } V. v \oplus_V \text{neg-}v = \mathbf{0}_V)$   
**and smult:**  $\forall c v. c \in \text{carrier } K \wedge v \in \text{carrier } V \longrightarrow c \odot_V v \in \text{carrier } V$   
 $V$   
**and comm:**  $\forall v w. v \in \text{carrier } V \wedge w \in \text{carrier } V \longrightarrow v \oplus_V w = w \oplus_V v$   
**and assoc:**  $\forall v w x. v \in \text{carrier } V \wedge w \in \text{carrier } V \wedge x \in \text{carrier } V \longrightarrow (v \oplus_V w) \oplus_V x = v \oplus_V (w \oplus_V x)$   
**and add-id:**  $\forall v \in \text{carrier } V. (v \oplus_V \mathbf{0}_V = v)$   
**and compat:**  $\forall a b v. a \in \text{carrier } K \wedge b \in \text{carrier } K \wedge v \in \text{carrier } V \longrightarrow (a \otimes_K b) \odot_V v = a \odot_V (b \odot_V v)$   
**and smult-id:**  $\forall v \in \text{carrier } V. (\mathbf{1}_K \odot_V v = v)$   
**and dist-f:**  $\forall a b v. a \in \text{carrier } K \wedge b \in \text{carrier } K \wedge v \in \text{carrier } V \longrightarrow (a \oplus_K b) \odot_V v = (a \odot_V v) \oplus_V (b \odot_V v)$   
**and dist-add:**  $\forall a v w. a \in \text{carrier } K \wedge v \in \text{carrier } V \wedge w \in \text{carrier } V \longrightarrow a \odot_V (v \oplus_V w) = (a \odot_V v) \oplus_V (a \odot_V w)$   
**shows vectorspace**  $K V$   
 $\langle \text{proof} \rangle$

For any set  $S$ , the space of functions  $S \rightarrow K$  forms a vector space.

**lemma (in vectorspace) func-space-is-vs:**  
**fixes**  $S$   
**shows vectorspace**  $K (\text{func-space } S)$   
 $\langle \text{proof} \rangle$

**lemma direct-sum-is-vs:**  
**fixes**  $K V1 V2$   
**assumes**  $h1: \text{vectorspace } K V1$  **and**  $h2: \text{vectorspace } K V2$   
**shows vectorspace**  $K (\text{direct-sum } V1 V2)$   
 $\langle \text{proof} \rangle$

**lemma inj1-linear:**  
**fixes**  $K V1 V2$   
**assumes**  $h1: \text{vectorspace } K V1$  **and**  $h2: \text{vectorspace } K V2$   
**shows linear-map**  $K V1 (\text{direct-sum } V1 V2)$  ( $\text{inj1 } V1 V2$ )  
 $\langle \text{proof} \rangle$

**lemma inj2-linear:**  
**fixes**  $K V1 V2$   
**assumes**  $h1: \text{vectorspace } K V1$  **and**  $h2: \text{vectorspace } K V2$   
**shows linear-map**  $K V2 (\text{direct-sum } V1 V2)$  ( $\text{inj2 } V1 V2$ )  
 $\langle \text{proof} \rangle$

For subspaces  $V_1, V_2 \subseteq V$ , the map  $V_1 \oplus V_2 \rightarrow V$  given by  $(v_1, v_2) \mapsto v_1 + v_2$  is linear.

**lemma (in vectorspace) sum-map-linear:**  
**fixes**  $V1 V2$

**assumes**  $h1$ : subspace  $K$   $V1$   $V$  **and**  $h2$ : subspace  $K$   $V2$   $V$   
**shows** linear-map  $K$  (direct-sum (vs  $V1$ ) (vs  $V2$ ))  $V$  ( $\lambda$   $v$ . (fst  $v$ )  
 $\oplus_V$  (snd  $v$ ))  
 $\langle$ proof $\rangle$

**lemma** (in *vectorspace*) *sum-is-subspace*:  
**fixes**  $W1$   $W2$   
**assumes**  $h1$ : subspace  $K$   $W1$   $V$  **and**  $h2$ : subspace  $K$   $W2$   $V$   
**shows** subspace  $K$  (subspace-sum  $W1$   $W2$ )  $V$   
 $\langle$ proof $\rangle$

If  $W_1, W_2 \subseteq V$  are subspaces,  $W_1 \subseteq W_1 + W_2$

**lemma** (in *vectorspace*) *in-sum-vs*:  
**fixes**  $W1$   $W2$   
**assumes**  $h1$ : subspace  $K$   $W1$   $V$  **and**  $h2$ : subspace  $K$   $W2$   $V$   
**shows**  $W1 \subseteq$  subspace-sum  $W1$   $W2$   
 $\langle$ proof $\rangle$

**lemma** (in *vectorspace*) *vsum-comm*:  
**fixes**  $W1$   $W2$   
**assumes**  $h1$ : subspace  $K$   $W1$   $V$  **and**  $h2$ : subspace  $K$   $W2$   $V$   
**shows** (subspace-sum  $W1$   $W2$ ) = (subspace-sum  $W2$   $W1$ )  
 $\langle$ proof $\rangle$

If  $W_1, W_2 \subseteq V$  are subspaces, then  $W_1 + W_2$  is the minimal subspace such that both  $W_1 \subseteq W$  and  $W_2 \subseteq W$ .

**lemma** (in *vectorspace*) *vsum-is-minimal*:  
**fixes**  $W$   $W1$   $W2$   
**assumes**  $h1$ : subspace  $K$   $W1$   $V$  **and**  $h2$ : subspace  $K$   $W2$   $V$  **and**  $h3$ :  
subspace  $K$   $W$   $V$   
**shows** (subspace-sum  $W1$   $W2$ )  $\subseteq$   $W \iff W1 \subseteq W \wedge W2 \subseteq W$   
 $\langle$ proof $\rangle$

**lemma** (in *vectorspace*) *span-is-subspace*:  
**fixes**  $S$   
**assumes**  $h2$ :  $S \subseteq$  carrier  $V$   
**shows** subspace  $K$  (span  $S$ )  $V$   
 $\langle$ proof $\rangle$

### 6.1.1 Facts specific to vector spaces

If  $av = w$  and  $a \neq 0$ ,  $v = a^{-1}w$ .

**lemma** (in *vectorspace*) *mult-inverse*:  
**assumes**  $h1$ :  $a \in$  carrier  $K$  **and**  $h2$ :  $v \in$  carrier  $V$  **and**  $h3$ :  $a \odot_V v =$   
 $w$  **and**  $h4$ :  $a \neq \mathbf{0}_K$   
**shows**  $v =$  (inv $_K$   $a$ )  $\odot_V w$   
 $\langle$ proof $\rangle$

If  $w \in S$  and  $\sum_{w \in S} a_w w = 0$ , we have  $v = \sum_{w \notin S} a_w^{-1} a_w w$

**lemma** (in *vectorspace*) *lincomb-isolate*:

**fixes**  $A v$

**assumes**  $h1$ : finite  $A$  **and**  $h2$ :  $A \subseteq \text{carrier } V$  **and**  $h3$ :  $a \in A \rightarrow \text{carrier } K$  **and**  $h4$ :  $v \in A$

**and**  $h5$ :  $a v \neq \mathbf{0}_K$  **and**  $h6$ : *lincomb a A=0 V*

**shows**  $v = \text{lincomb } (\lambda w. \ominus_K(\text{inv}_K(a v)) \otimes_K a w) (A - \{v\})$  **and**  $v \in \text{span } (A - \{v\})$

*<proof>*

The map  $(S \rightarrow K) \mapsto V$  given by  $(a_v)_{v \in S} \mapsto \sum_{v \in S} a_v v$  is linear.

**lemma** (in *vectorspace*) *lincomb-is-linear*:

**fixes**  $S$

**assumes**  $h$ : finite  $S$  **and**  $h2$ :  $S \subseteq \text{carrier } V$

**shows** *linear-map K (func-space S) V* ( $\lambda a. \text{lincomb } a S$ )

*<proof>*

## 6.2 Basic facts about span and linear independence

If  $S$  is linearly independent, then  $v \in \text{span } S$  iff  $S \cup \{v\}$  is linearly dependent.

**theorem** (in *vectorspace*) *lin-dep-iff-in-span*:

**fixes**  $A v S$

**assumes**  $h1$ :  $S \subseteq \text{carrier } V$  **and**  $h2$ : *lin-indpt S* **and**  $h3$ :  $v \in \text{carrier } V$  **and**  $h4$ :  $v \notin S$

**shows**  $v \in \text{span } S \iff \text{lin-dep } (S \cup \{v\})$

*<proof>*

If  $v \in \text{span } A$  then  $\text{span } A = \text{span } (A \cup \{v\})$

**lemma** (in *vectorspace*) *already-in-span*:

**fixes**  $v A$

**assumes**  $\text{in } C$ :  $A \subseteq \text{carrier } V$  **and**  $\text{in span}$ :  $v \in \text{span } A$

**shows**  $\text{span } A = \text{span } (A \cup \{v\})$

*<proof>*

## 6.3 The Replacement Theorem

If  $A, B \subseteq V$  are finite,  $A$  is linearly independent,  $B$  generates  $W$ , and  $A \subseteq W$ , then there exists  $C \subseteq V$  disjoint from  $A$  such that  $\text{span}(A \cup C) = W$  and  $|C| \leq |B| - |A|$ . In other words, we can complete any linearly independent set to a generating set of  $W$  by adding at most  $|B| - |A|$  more elements.

**theorem** (in *vectorspace*) *replacement*:

**fixes**  $A B$

**assumes**  $h1$ : finite  $A$

**and**  $h2$ : finite  $B$

**and**  $h3: B \subseteq \text{carrier } V$   
**and**  $h4: \text{lin-indpt } A$   
**and**  $h5: A \subseteq \text{span } B$   
**shows**  $\exists C. \text{finite } C \wedge C \subseteq \text{carrier } V \wedge C \subseteq \text{span } B \wedge C \cap A = \{\} \wedge \text{int}$   
 $(\text{card } C) \leq (\text{int } (\text{card } B)) - (\text{int } (\text{card } A)) \wedge (\text{span } (A \cup C) = \text{span } B)$   
**(is**  $\exists C. ?P A B C)$

$\langle \text{proof} \rangle$

## 6.4 Defining dimension and bases.

Finite dimensional is defined as having a finite generating set.

**definition** (in *vectorspace*)  $\text{fin-dim}:: \text{bool}$   
**where**  $\text{fin-dim} = (\exists A. ((\text{finite } A) \wedge (A \subseteq \text{carrier } V) \wedge (\text{gen-set } A)))$

The dimension is the size of the smallest generating set. For equivalent characterizations see below.

**definition** (in *vectorspace*)  $\text{dim}:: \text{nat}$   
**where**  $\text{dim} = (\text{LEAST } n. (\exists A. ((\text{finite } A) \wedge (\text{card } A = n) \wedge (A \subseteq \text{carrier } V) \wedge (\text{gen-set } A))))$

A *basis* is a linearly independent generating set.

**definition** (in *vectorspace*)  $\text{basis}:: 'c \text{ set} \Rightarrow \text{bool}$   
**where**  $\text{basis } A = ((\text{lin-indpt } A) \wedge (\text{gen-set } A) \wedge (A \subseteq \text{carrier } V))$

From the replacement theorem, any linearly independent set is smaller than any generating set.

**lemma** (in *vectorspace*)  $\text{li-smaller-than-gen}$ :  
**fixes**  $A B$   
**assumes**  $h1: \text{finite } A$  **and**  $h2: \text{finite } B$  **and**  $h3: A \subseteq \text{carrier } V$  **and**  
 $h4: B \subseteq \text{carrier } V$   
**and**  $h5: \text{lin-indpt } A$  **and**  $h6: \text{gen-set } B$   
**shows**  $\text{card } A \leq \text{card } B$   
 $\langle \text{proof} \rangle$

The dimension is the cardinality of any basis. (In particular, all bases are the same size.)

**lemma** (in *vectorspace*)  $\text{dim-basis}$ :  
**fixes**  $A$   
**assumes**  $\text{fin}: \text{finite } A$  **and**  $h2: \text{basis } A$   
**shows**  $\text{dim} = \text{card } A$   
 $\langle \text{proof} \rangle$

A *maximal* set with respect to  $P$  is such that if  $B \supseteq A$  and  $P$  is also satisfied for  $B$ , then  $B = A$ .

**definition** *maximal*:: 'a set  $\Rightarrow$  ('a set  $\Rightarrow$  bool)  $\Rightarrow$  bool  
**where** *maximal* A P = ((P A)  $\wedge$  ( $\forall$  B. B  $\supseteq$  A  $\wedge$  P B  $\longrightarrow$  B = A))

A *minimal* set with respect to P is such that if  $B \subseteq A$  and P is also satisfied for B, then  $B = A$ .

**definition** *minimal*:: 'a set  $\Rightarrow$  ('a set  $\Rightarrow$  bool)  $\Rightarrow$  bool  
**where** *minimal* A P = ((P A)  $\wedge$  ( $\forall$  B. B  $\subseteq$  A  $\wedge$  P B  $\longrightarrow$  B = A))

A maximal linearly independent set is a generating set.

**lemma** (in *vectorspace*) *max-li-is-gen*:  
**fixes** A  
**assumes** h1: *maximal* A ( $\lambda$ S. S  $\subseteq$  carrier V  $\wedge$  *lin-indpt* S)  
**shows** *gen-set* A  
 $\langle$ proof $\rangle$

A minimal generating set is linearly independent.

**lemma** (in *vectorspace*) *min-gen-is-li*:  
**fixes** A  
**assumes** h1: *minimal* A ( $\lambda$ S. S  $\subseteq$  carrier V  $\wedge$  *gen-set* S)  
**shows** *lin-indpt* A  
 $\langle$ proof $\rangle$

Given that some finite set satisfies P, there is a minimal set that satisfies P.

**lemma** *minimal-exists*:  
**fixes** A P  
**assumes** h1: *finite* A **and** h2: P A  
**shows**  $\exists$  B. B  $\subseteq$  A  $\wedge$  *minimal* B P  
 $\langle$ proof $\rangle$

If V is finite-dimensional, then any linearly independent set is finite.

**lemma** (in *vectorspace*) *fin-dim-li-fin*:  
**assumes** fd: *fin-dim* **and** li: *lin-indpt* A **and** inC: A  $\subseteq$  carrier V  
**shows** *fin*: *finite* A  
 $\langle$ proof $\rangle$

If V is finite-dimensional (has a finite generating set), then a finite basis exists.

**lemma** (in *vectorspace*) *finite-basis-exists*:  
**assumes** h1: *fin-dim*  
**shows**  $\exists$   $\beta$ . *finite*  $\beta$   $\wedge$  *basis*  $\beta$   
 $\langle$ proof $\rangle$

The proof is as follows.

1. Because V is finite-dimensional, there is a finite generating set (we took this as our definition of finite-dimensional).



2. Hence, there is a minimal  $\beta \subseteq A$  such that  $\beta$  generates  $V$ .
3.  $\beta$  is linearly independent because a minimal generating set is linearly independent.

Finally,  $\beta$  is a basis because it is both generating and linearly independent.

Any linearly independent set has cardinality at most equal to the dimension.

**lemma** (in *vectorspace*) *li-le-dim*:  
**fixes**  $A$   
**assumes**  $fd$ : *fin-dim* **and**  $c$ :  $A \subseteq \text{carrier } V$  **and**  $l$ : *lin-indpt*  $A$   
**shows** *finite*  $A$   $\text{card } A \leq \text{dim}$   
 $\langle \text{proof} \rangle$

Any generating set has cardinality at least equal to the dimension.

**lemma** (in *vectorspace*) *gen-ge-dim*:  
**fixes**  $A$   
**assumes**  $fa$ : *finite*  $A$  **and**  $c$ :  $A \subseteq \text{carrier } V$  **and**  $l$ : *gen-set*  $A$   
**shows**  $\text{card } A \geq \text{dim}$   
 $\langle \text{proof} \rangle$

If there is an upper bound on the cardinality of sets satisfying  $P$ , then there is a maximal set satisfying  $P$ .

**lemma** *maximal-exists*:  
**fixes**  $P B N$   
**assumes**  $maxc$ :  $\bigwedge A. P A \implies \text{finite } A \wedge (\text{card } A \leq N)$  **and**  $b$ :  $P B$   
**shows**  $\exists A. \text{finite } A \wedge \text{maximal } A P$   
 $\langle \text{proof} \rangle$

Any maximal linearly independent set is a basis.

**lemma** (in *vectorspace*) *max-li-is-basis*:  
**fixes**  $A$   
**assumes**  $h1$ : *maximal*  $A$  ( $\lambda S. S \subseteq \text{carrier } V \wedge \text{lin-indpt } S$ )  
**shows** *basis*  $A$   
 $\langle \text{proof} \rangle$

Any minimal linearly independent set is a generating set.

**lemma** (in *vectorspace*) *min-gen-is-basis*:  
**fixes**  $A$   
**assumes**  $h1$ : *minimal*  $A$  ( $\lambda S. S \subseteq \text{carrier } V \wedge \text{gen-set } S$ )  
**shows** *basis*  $A$   
 $\langle \text{proof} \rangle$

Any linearly independent set with cardinality at least the dimension is a basis.

**lemma** (in *vectorspace*) *dim-li-is-basis*:  
**fixes**  $A$   
**assumes**  $fd$ : *fin-dim* **and**  $fa$ : *finite A* **and**  $ca$ :  $A \subseteq \text{carrier } V$  **and**  $li$ :  
*lin-indpt A*  
**and**  $d$ :  $\text{card } A \geq \text{dim}$   
**shows** *basis A*  
 $\langle \text{proof} \rangle$

Any generating set with cardinality at most the dimension is a basis.

**lemma** (in *vectorspace*) *dim-gen-is-basis*:  
**fixes**  $A$   
**assumes**  $fa$ : *finite A* **and**  $ca$ :  $A \subseteq \text{carrier } V$  **and**  $li$ : *gen-set A*  
**and**  $d$ :  $\text{card } A \leq \text{dim}$   
**shows** *basis A*  
 $\langle \text{proof} \rangle$

$\beta$  is a basis iff for all  $v \in V$ , there exists a unique  $(a_v)_{v \in S}$  such that  $\sum_{v \in S} a_v v = v$ .

**lemma** (in *vectorspace*) *basis-criterion*:  
**assumes**  $A$ -*fin*: *finite A* **and**  $A$ -*inC*:  $A \subseteq \text{carrier } V$   
**shows** *basis A*  $\longleftrightarrow (\forall v. v \in \text{carrier } V \longrightarrow (\exists! a. a \in A \rightarrow_E \text{carrier } K \wedge \text{lincomb } a \ A = v))$   
 $\langle \text{proof} \rangle$

**lemma** (in *linear-map*) *surj-imp-imT-carrier*:  
**assumes** *surj*:  $T'(\text{carrier } V) = \text{carrier } W$   
**shows**  $\text{im } T = \text{carrier } W$   
 $\langle \text{proof} \rangle$

## 6.5 The rank-nullity (dimension) theorem

If  $V$  is finite-dimensional and  $T : V \rightarrow W$  is a linear map, then  $\dim(\text{im}(T)) + \dim(\text{ker}(T)) = \dim V$ . Moreover, we prove that if  $T$  is surjective linear-map between  $V$  and  $W$ , where  $V$  is finite-dimensional, then also  $W$  is finite-dimensional.

**theorem** (in *linear-map*) *rank-nullity-main*:  
**assumes**  $fd$ :  $V$ -*fin-dim*  
**shows**  $(\text{vectorspace.dim } K (W.\text{vs im } T)) + (\text{vectorspace.dim } K (V.\text{vs ker } T)) = V.\text{dim}$   
 $T'(\text{carrier } V) = \text{carrier } W \implies W.\text{fin-dim}$   
 $\langle \text{proof} \rangle$

**theorem** (in *linear-map*) *rank-nullity*:  
**assumes**  $fd$ :  $V$ -*fin-dim*  
**shows**  $(\text{vectorspace.dim } K (W.\text{vs im } T)) + (\text{vectorspace.dim } K (V.\text{vs ker } T)) = V.\text{dim}$   
 $\langle \text{proof} \rangle$

end