Proof of Dickson's Lemma Using the ACL2 Theorem Prover via an Explicit Ordinal Mapping

Mátyás Sustik

April 23, 2003

Abstract

In this paper we present the use of the ACL2 theorem prover to formalize and mechanically check a new proof of Dickson's lemma about monomial sequences. Dickson's lemma can be used to establish the termination of Büchberger's algorithm to find the Gröbner basis of a polynomial ideal. This effort is related to a larger project which aims to develop a mechanically verified computer algebra system.

1 Introduction

Dickson's lemma about monomial sequences [2] can be used to prove the termination of Büchberger's algorithm to find the Gröbner basis of a polynomial ideal [3]. This termination problem was the main incentive behind this proof attempt [10].

The classical proof of Dickson's lemma is not particularly complicated, however the non-constructive nature of the proof and the concept of infinite sequences make this theorem non-trivial to formalize and prove using the ACL2 theorem prover [5, 4].

Similar proofs were carried out using the Coq prover [12] and the ALF proof assistant [1]. Parallel to this proof attempt formalization and proof of Dickson's lemma using the multiset books [8, 11] distributed with ACL2 was carried out in [9]. [1]

Theorem (Dickson's lemma) If $m_1, m_2, m_3, \ldots$ is an infinite sequence of monomials of k variables, then there exist indices i, j such that $i < j$ and m_i divides m_j.

The above formulation is not directly applicable for a termination argument; to use it for that purpose we need to assume non-termination and arrive at a contradiction applying the lemma. In the case of Büchberger's algorithm this is done as follows: a monomial is assigned at each step of the algorithm to the polynomial set generating the ideal. The monomial sequence has the property that no monomial divides a subsequent one in this sequence. The application of Dickson's lemma will assert that the algorithm can only have a finite number of steps.

In the logical world of ACL2 a recursive function definition has to be provably terminating to ensure consistency. A measure function has the same signature as the function to be shown to be terminating and it returns an ordinal. The measure function is a witness to the termination of a recursive function if the ordinals assigned to each recursive call are smaller than the ordinal assigned to the invocation. This naturally suggests a way to formalize Dickson's lemma and the associated termination argument in terms of an ordinal embedding. We will assign ordinals to the initial segments of the monomial sequence in such a way that if no monomial divides a subsequent one then the ordinal sequence will be strictly decreasing. As it turns out this formulation could directly be used for the termination argument required in [10].

An ordinal arithmetic package ("book") was developed by Panagiotis Manolios and Daron Vroon [6, 7]. It is distributed with along the ACL2 sources starting with version 2.7. In these books they formulate and prove properties of ordinals written in Cantor normal form. They establish properties of ordinal addition, multiplication, exponentation and the ordering. They prove equivalence of their ordinal representation to

[1] The comparison of the two proofs is not discussed in this paper. It will be published separately.

that of the native ACL2 representation through a one-to-one embedding. A small collection of further definitions and lemmas about ordinals were necessary to carry out the presented proof attempt.

The full transcript of the ACL2 events and instructions on how to replay and certify the proof presented here are available as supporting material and be downloaded from the ACL2 home-page.

2 ACL2 formalized proof of Dickson's lemma

As indicated in the introduction, we will define a mapping from the sets of monomials to the ordinals (below ϵ_0) such that if a sequence of monomials does not have two elements where the former divides the latter, then the ordinals corresponding to the sets of monomials in the initial subsequences form a strictly decreasing sequence. This establishes that the sequence must be finite, and in turn can be used to prove termination of an ACL2 function.

Definition. A monomial of the $x_0, x_1, \ldots, x_{k-1}$ variables is a product in the form: $x_0^{u_0} x_1^{u_1} \ldots x_{k-1}^{u_{k-1}}$, where the exponents are natural numbers.

Theorem (Dickson's lemma) If $m_1, m_2, m_3, \ldots$ is an infinite sequence of monomials of k variables, then there are $i < j$ indices such that m_i divides m_j.
Monomials of k variables can be represented by k-tuples, where each tuple consists of the exponents. The following partial order defined on k-tuples coincides with the divisibility of the corresponding monomials:

Definition. For the $u, v \in \mathbb{N}^k$ k-tuples $u \leq_k v$ if and only if $u_i \leq v_i$ for all $0 \leq i < k$, where $u = (u_0, u_1, \ldots, u_{k-1})$ and $v = (v_0, v_1, \ldots, v_{k-1})$.
The corresponding ACL2 functions realizing a recognizer for tuples and the $\leq_k$ relation:

```
(defun natural-tuplep (k x)
  (cond ((zp k) (null x))
        ((not (natp (first x))) nil)
        (T (natural-tuplep (1- k) (rest x)))))

(defun partial-tuple-<= (k x y)
  (cond ((zp k) t)
        ((< (car y) (car x)) nil)
        (t (partial-tuple-<= (1- k) (cdr x) (cdr y)))))
```

Let us denote the collection of the finite sets of k-tuples by $\mathcal{A}_k$:

Definition. $\mathcal{A}_k = \{A \subset \mathbb{N}^k : |A| < \omega\}$.

Definition. We define the $M_k : \mathcal{A}_k \to Ord$ function inductively. If $A \in \mathcal{A}_1$ then set

$$M_1(A) = \min A,$$

where min is defined to give the minimal number from a set of (natural) numbers[2] with the agreement that $M_1(\emptyset) = \omega$. Let us suppose now that $k > 1$ and that we have already defined M_{k-1}. For an arbitrary $A \in \mathcal{A}_k$ and $i \in \mathbb{N}$ define the $P_i^{(A)} \in \mathcal{A}_{k-1}$ sets and the $\alpha_i^{(A)}$ ordinals as follows:

$$P_i^{(A)} = \{(u_1, u_2, \ldots, u_{k-1}) : (u_0, u_1, \ldots, u_{k-1}) \in A, u_0 \leq i\},$$

$$\alpha_i^{(A)} = M_{k-1}(P_i^{(A)}).$$

If A is clear from the context then the (A) superscript will be omitted. Before we can define M_k we need the following lemma:

Lemma 0. The P_i, α_i sequences stabilize for every $A \in \mathcal{A}_k$.

Proof. Set $s^{(A)} = \max\{u_0 : (u_0, u_1, \ldots, u_{k-1}) \in A\}$ and notice that $P_i = P_{s^{(A)}}$ if $i \geq s^{(A)}$ and by definition $\alpha_i = \alpha_s$ for every $i \geq s$. □

[2]Note that $\mathbb{N}^1 = \mathbb{N}$.

Definition.

$$M_k(A) = \left(\sum_{i=0}^{s-1} \omega^{\alpha_i}\right) + \omega^{\alpha_s+1}.$$

To simplify the proof formalization we introduce the partial sums of M_k. We abuse the notation somewhat by denoting this function which takes an additional second argument by M_k as well:

Definition.

$$M_k(A,j) = \begin{cases} \min A & \text{if } k = 1 \\ \left(\sum_{i=j}^{s-1} \omega^{\alpha_i}\right) + \omega^{\alpha_s+1} & \text{if } k > 1 \end{cases}$$

The proof of Lemma 0 is implicit in the ACL2 definition of the mapping function:

```
(defun tuple-set->ordinal-partial-sum (k S i)
  (declare (xargs :measure (cons (1+ (nfix k))
                                 (nfix (- (tuple-set-max-first S) i)))))
  (cond ((or (not (natp k)) (not (natp i))) 0)
        ((zp k) 0)
        ((equal k 1)
         (tuple-set-min-first S))
        ((<= (tuple-set-max-first S) i)
         (o^ (omega)
             (o+ (tuple-set->ordinal-partial-sum
                  (1- k)
                  (tuple-set-projection S)
                  0)
                 1)))
        (T (o+
            (o^ (omega)
                (tuple-set->ordinal-partial-sum
                 (1- k)
                 (tuple-set-filter-projection S i)
                 0))
            (tuple-set->ordinal-partial-sum k S (1+ i))))))
```

To highlight the intuition behind the definition—especially the role of the last term—it is worth mentioning that M_k could be defined as a supremum:

$$M_k(A) = \sup_{n=1}^{\omega} \sum_{i=0}^{n} \omega^{\alpha_i}$$

however the finite sum form appeared to be more appropriate for the ACL2 formalization.
Some immediate properties that we can derive from the definition:

$$M_k(A) = M_k(A,0)$$

,

$$M_k(A,j) = \omega^{\alpha_j} + M_k(A,j+1).$$

The latter relies on lemmas of ordinal arithmetic and the "stabilization" of the α_i sequences.
The above defined $P_i^{(A)}$ sets are realized by the following ACL2 functions:

```
(defun tuple-set-filter (S i)
  (cond ((endp S) NIL)
        ((and (consp (first S)) (<= (first (first S)) i))
         (cons (first S) (tuple-set-filter (rest S) i)))
        (T (tuple-set-filter (rest S) i))))
```

```
(defun tuple-set-projection (S)
  (cond ((endp S) NIL)
        ((consp (first S)) (cons (rest (first S))
                                 (tuple-set-projection (rest S))))
        (T (tuple-set-projection (rest S)))))

(defun tuple-set-filter-projection (S i)
  (tuple-set-projection (tuple-set-filter S i)))
```

Basic properties of the above functions were automatically verified once stated.
To arrive to the promised proof of Dickson's lemma we need to establish certain properties of M_k and the corresponding α_i sequences used in the definition. The lemmas about M_k proved below can be generalized to refer to the partial sum version; but for succinctness and because they are obvious to derive we may omit the more general form of these lemmas. In the proofs usually we establish the more general property. (In the ACL2 proof the more general form had to be stated as a separate lemma.)

Lemma 1. If $A \subseteq B \in \mathcal{A}_k$ then $M_k(A) \geq M_k(B)$.

Proof. We prove the stronger $M_k(A, j) \geq M_k(B, j)$ (with $j \in N$) using induction on k. The statement is immediate from the definition if $k = 1$. Now suppose that $k > 1$ and that we have established the property for $k - 1$.
Notice that if $A \subseteq B$ as required then $P_i^{(A)} \subseteq P_i^{(B)}$ and so by the inductive hypothesis

$$\alpha_i^{(A)} = M_{k-1}(P_i^{(A)}) \geq M_{k-1}(P_i^{(B)}) = \alpha_i^{(B)},$$

for every $i \in \mathbb{N}$.
If $j \geq \max(s^{(A)}, s^{(B)})$ then $M_k(A, j) = \omega^{\alpha_j^{(A)}+1}$ and $M_k(B, j) = \omega^{\alpha_j^{(B)}+1}$ and we are done. A "downward" induction on j finishes the proof using the $M_k(A, j) = \omega^{\alpha_j^{(A)}} + M_k(A, j + 1)$ and $M_k(B, j) = \omega^{\alpha_j^{(B)}} + M_k(B, j + 1)$ equalities and properties of ordinal arithmetic. □
The ACL2 event for this lemma is as follows:

```
(defthm map-lemma-1
  (implies (and (tuple-setp k A)
                (tuple-setp k B)
                (tuple-set-subsetp A B)
                (natp k))
           (ord<= (tuple-set->ordinal k B)
                  (tuple-set->ordinal k A)))
```

The necessary hints required for the proof of the above and subsequent ACL2 theorems were omitted. Please refer to the supporting materials for the complete forms.

Lemma 2. For any $A \in A_k$, $k > 1$ the α_i sequence is monotone decreasing.

Proof. Notice that the P_i sets are monotone increasing (with regard to inclusion) and therefore by the previous Lemma 1 and the definition we can write:

$$\alpha_i = M_{k-1}(P_i) \geq M_{k-1}(P_{i+1}) = \alpha_{i+1}$$

finishing the proof. □
The corresponding ACL2 event is:

```
(defthm map-lemma-2
  (implies (and (tuple-setp k A)
                (natp k)
                (< 1 k)
                (natp i))
           (ord<= (tuple-set->ordinal (1- k)
                                      (tuple-set-filter-projection
```

```
                          A (1+ i)))
              (tuple-set->ordinal (1- k)
                                  (tuple-set-filter-projection
                                   A i)))))
```

Lemma 3. For any $A, B \in \mathcal{A}_k$, $k > 1$, $M_k(A) = M_k(B)$ holds if and only if $\alpha_i^{(A)} = \alpha_i^{(B)}$ is true for every $i \in \mathbb{N}$.

Proof. The non-obvious implication to prove is that if $M_k(A) = M_k(B)$ then the sequences are identical. We arrive at the proof through several lemmas. The letters i, j, k denote natural numbers and A, B denote sets of tuples from $\mathcal{A}_k$.

Lemma 3.1 If $k > 1$ then $M_k(A, i) \leq \omega^{\alpha_i^{(A)}+1}$.

Proof. We use a downward induction on i. When $i \geq s^{(A)}$ the inequality is the direct consequence of the definition of $M_k(A, i)$. The induction step follows from:

$$M_k(A,i) = \omega^{\alpha_i^{(A)}} + M_k(A, i+1) \leq \omega^{\alpha_i^{(A)}} + \omega^{\alpha_{i+1}^{(A)}+1} \leq \omega^{\alpha_i^{(A)}} + \omega^{\alpha_i^{(A)}+1} = \omega^{\alpha_i^{(A)}+1}.\square$$

The corresponding ACL2 theorem is the following:

```
(defthm map-lemma-3.1
  (implies (and (tuple-setp k A)
                (natp k)
                (< 1 k)
                (natp i))
           (ord<= (tuple-set->ordinal-partial-sum k A i)
                  (o^ (omega) (o+ (tuple-set->ordinal-partial-sum
                                   (1- k)
                                   (tuple-set-filter-projection A i)
                                   0)
                                  1)))))
```

The proof required the specification of the induction scheme, expansion the definition of the $M_k(A, i)$ function and instantiation of ordinal arithmetic lemmas corresponding to the inequalities establishing the induction step.

Lemma 3.2 If $k > 1$ then $\omega^{\alpha_i^{(A)}} < M_k(A, i)$.

Proof. This claim is immediate from $M_k(A, i) = \omega^{\alpha_i^{(A)}} + M_k(A, i + 1)$. $\square$
The corresponding ACL2 event is:

```
(defthm map-lemma-3.2
  (implies (and (tuple-setp k A)
                (natp k)
                (< 1 k)
                (natp i))
           (o< (o^ (omega) (tuple-set->ordinal-partial-sum
                            (1- k)
                            (tuple-set-filter-projection A i)
                            0))
               (tuple-set->ordinal-partial-sum k A i))))
```

Lemma 3.3 If $k > 1$ and $\alpha_i^{(A)} = M_{k-1}(P_i^{(A)}) < M_{k-1}(P_i^{(B)}) = \alpha_i^{(B)}$ then $M_k(A, i) < M_k(B, i)$.

Proof. The $\alpha_i^{(A)} < \alpha_i^{(B)}$ inequality implies $\alpha_i^{(A)} + 1 \leq \alpha_i^{(B)}$. Properties of ordinal exponentiation, Lemmas 3.1, 3.2 and $M_k(A, i) = \omega^{\alpha_i^{(A)}} + M_k(A, i + 1)$, $M_k(B, i) = \omega^{\alpha_i^{(B)}} + M_k(B, i + 1)$ finishes the proof. $\square$
The corresponding ACL2 theorem:

```
(defthm map-lemma-3.3
  (implies (and (tuple-setp k A)
                (tuple-setp k B)
                (natp k)
                (natp i)
                (< 1 k)
                (o< (tuple-set->ordinal-partial-sum
                     (1- k)
                     (tuple-set-projection (tuple-set-filter A i))
                     0)
                    (tuple-set->ordinal-partial-sum
                     (1- k)
                     (tuple-set-projection (tuple-set-filter B i))
                     0)))
           (o< (tuple-set->ordinal-partial-sum k A i)
               (tuple-set->ordinal-partial-sum k B i)))
```

A direct consequence of Lemma 3.3 and the totality of ordinal comparison:

Lemma 3.4 If $k > 1$ and $M_k(A, i) = M_k(B, i)$ then $\alpha_i^{(A)} = \alpha_i^{(B)}$.

Proof. Use Lemma 3.3. □
The ACL2 form:

```
(defthm map-lemma-3.4
  (implies (and (tuple-setp k A)
                (tuple-setp k B)
                (natp k)
                (natp i)
                (< 1 k)
                (equal (tuple-set->ordinal-partial-sum k A i)
                       (tuple-set->ordinal-partial-sum k B i)))
           (equal (equal (tuple-set->ordinal-partial-sum
                          (1- k)
                          (tuple-set-projection (tuple-set-filter A i))
                          0)
                         (tuple-set->ordinal-partial-sum
                          (1- k)
                          (tuple-set-projection (tuple-set-filter B i))
                          0))
                  T))
```

The following three lemmas rely on the above Lemma 3.4:

Lemma 3.5 If $k > 1$ and $M_k(A, i) = M_k(B, i)$ then $M_k(A, i + 1) = M_k(B, i + 1)$.

Proof. Use lemma 3.4 and that $M_k(A, i) = \omega^{\alpha_i^{(A)}} + M_k(A, i + 1)$, $M_k(B, i) = \omega^{\alpha_i^{(B)}} + M_k(B, i + 1)$. □

Lemma 3.6 If $k > 1, i \leq j$ and $M_k(A, i) = M_k(B, i)$ then $M_k(A, j) = M_k(B, j)$.

Proof. Use induction on j and lemma 3.5. □

Lemma 3.7 If $k > 1, i \leq j$ and $M_k(A, i) = M_k(B, i)$ then $\alpha_j^{(A)} = \alpha_j^{(B)}$.

Proof. Instantiate Lemmas 3.4 and 3.6. □

The corresponding ACL2 theorems for the above lemmas are:

```
(defthm map-lemma-3.5
  (implies (and (tuple-setp k A)
                (tuple-setp k B)
                (natp k)
                (natp i)
                (< 1 k)
                (equal (tuple-set->ordinal-partial-sum k A i)
                       (tuple-set->ordinal-partial-sum k B i)))
           (equal (tuple-set->ordinal-partial-sum k A (1+ i))
                  (tuple-set->ordinal-partial-sum k B (1+ i)))))

(defthm map-lemma-3.6
  (implies (and (tuple-setp k A)
                (tuple-setp k B)
                (natp k)
                (< 1 k)
                (natp i)
                (natp j)
                (<= i j)
                (equal (tuple-set->ordinal-partial-sum k A i)
                       (tuple-set->ordinal-partial-sum k B i)))
           (equal (equal (tuple-set->ordinal-partial-sum k A j)
                         (tuple-set->ordinal-partial-sum k B j))
                  T))

(defthm map-lemma-3.7
  (implies (and (tuple-setp k A)
                (tuple-setp k B)
                (natp k)
                (< 1 k)
                (natp i)
                (natp j)
                (<= i j)
                (equal (tuple-set->ordinal-partial-sum k A i)
                       (tuple-set->ordinal-partial-sum k B i)))
           (equal (equal (tuple-set->ordinal-partial-sum
                          (1- k)
                          (tuple-set-projection (tuple-set-filter A j))
                          0)
                         (tuple-set->ordinal-partial-sum
                          (1- k)
                          (tuple-set-projection (tuple-set-filter B j))
                          0))
                  T))
```

The proof of Lemma 3 follows from Lemma 3.7 when $i = 0$. □

The following theorem is the key step in establishing that the ordinal sequence in question is strictly decreasing (under the right assumptions), by providing a link between M_k and $\leq_k$.

Lemma 4. If $A \subseteq B \in \mathcal{A}_k$, $v \in B \setminus A$ and $M_k(A) = M_k(B)$ then there is a $w \in A$ such that $w \leq_k v$.

Proof. We prove the claim by induction on k. The statement for $k = 1$ is a trivial consequence of the definition of M_1 and $\leq_1$. Suppose that $k > 1$ and that we have the claim established for $k - 1$ and $A, B, v = (v_0, v_1, \ldots, v_{k-1})$ are as described above.

By Lemma 3 we can conclude that $\alpha_i^{(A)} = \alpha_i^{(B)}$ for any $i \in \mathbb{N}$. In particular for $i = v_0$ we have:

$$\alpha^{(A)}_{v_0} = M_{k-1}(P^{(A)}_{v_0}) \text{ where } P^{(A)}_{v_0} = \{(u_1, u_2, \ldots, u_{k-1}) : (u_0, u_1, \ldots, u_{k-1}) \in x, u_0 \leq v_0\},$$

$$\alpha^{(B)}_{v_0} = M_{k-1}(P^{(A)}_{v_0}) \text{ where } P^{(B)}_{v_0} = \{(u_1, u_2, \ldots, u_{k-1}) : (u_0, u_1, \ldots, u_{k-1}) \in y, u_0 \leq v_0\}).$$

From $A \subseteq B$ it follows that $P^{(A)}_{v_0} \subseteq P^{(B)}_{v_0}$ and that $v' = (v_1, v_2, \ldots, v_{k-1}) \in P^{(B)}_{v_0}$. Suppose first that $v' \in P^{(A)}_{v_0}$ also holds. This means that there is a $u = (u_0, u_1, \ldots, u_{k-1})$ k-tuple in A such that $u_0 \leq v_0$ (in fact no equality is possible) while $u_i = v_i$ for $i = 1, 2, \ldots, k-1$ and so $w = u$ will be a satisfying choice. If $v' \notin P^{(A)}_{v_0}$ then by the induction hypothesis there is a $w' \in P^{(A)}_{v_0}$ such that $w' \leq_{k-1} v'$. By the definition of $P^{(A)}_{v_0}$ there must be a $w = (w_0, w_1, \ldots, w_{k-1}) \in A$ for which $w' = (w_1, w_2, \ldots, w_{k-1})$ and it is obvious that $w \leq_k v$ as required. □

To formalize the above argument in ACL2 we had to explicitly define witnesses, which capture the required properties of the filtering and projecting functions. For example the witness function shown below called *exists-projection-filter-inverse-witness* finds a tuple from a tuple set which is "projected" onto a specified element of the projected filtered set of tuples. Precisely: if $S \in \mathcal{A}_k$ and $u \in P^{(S)}_i$ then we get an element $x \in S$ which maps to u.

The following functions were defined to serve witness purposes:

```
(defun exists-partial-tuple-<=-set-witness (k S x)
  (cond ((endp S) nil)
        ((partial-tuple-<= k (first S) x) (first S))
        (t (exists-partial-tuple-<=-set-witness k (rest S) x))))

(defun exists-projection-filter-inverse-witness (S v i)
  (cond ((endp S) nil)
        ((and (equal v (rest (first S)))
              (<= (first (first S)) i))
         (first S))
        (T (exists-projection-filter-inverse-witness (rest S) v i))))

(defun exists-partial-tuple-<=-set (k S x)
  (let ((w (exists-partial-tuple-<=-set-witness k S x)))
    (and (natural-tuplep k w)
         (tuple-in-set w S)
         (partial-tuple-<= k w x))))
```

Several lemmas had to be proved to enable ACL2 to reason about these functions. These lemmas are very similar to the lemmas obtained by using the *defun-sk* (macro) definition which allows existential quantification to be formalized in ACL2 by defining a witness function. In this proof the sets requiring witnesses are all from a finite set (represented by a list) and therefore these witnessing functions can be and were made explicit (and executable). A theorem about the projection witness function:

```
(defthm exists-projection-filter-inverse-impl
  (implies (and (tuple-setp k S)
                (natural-tuplep (1- k) v)
                (<= 1 k)
                (exists-projection-filter-inverse S v i))
           (and (equal (natural-tuplep
                        k
                        (exists-projection-filter-inverse-witness S v i))
                       T)
                (equal (tuple-in-set
                        (exists-projection-filter-inverse-witness S v i)
                        S)
                       T)
```

```
                (equal (rest (exists-projection-filter-inverse-witness
                              S v i))
                       v)
                (<= (first (exists-projection-filter-inverse-witness
                            S v i))
                    i))))
```

A main auxiliary lemma required for the proof of Lemma 4 was one that described the above mentioned property of the witness function:

```
(defthm map-lemma-4.2
  (implies (and (tuple-setp k S)
                (natp k)
                (<= 2 k)
                (natural-tuplep k v)
                (tuple-in-set v S))
           (tuple-in-set
            (cdr v)
            (tuple-set-projection (tuple-set-filter S (car v))))))
```

And here is the ACL2 event describing Lemma 4. Please refer to the supporting materials for additional definitions referred to.

```
(defthm map-lemma-4
  (implies (and (tuple-setp k A)
                (tuple-setp k B)
                (tuple-set-subsetp A B)
                (natural-tuplep k v)
                (tuple-in-set v B)
                (equal (tuple-set->ordinal-partial-sum k A 0)
                       (tuple-set->ordinal-partial-sum k B 0))
                (natp k)
                (<= 1 k))
           (exists-partial-tuple-<=-set k A v)))
```

And finally the theorem that can be used for a termination argument implementing the Büchberger algorithm:

Theorem (Dickson's lemma equivalent) If $s_1, s_2, \ldots, s_n$ is a finite sequence of k-tuples of natural numbers such that for any $1 \leq i < j \leq n$ we have $s_i \not\leq_k s_j$ then the $M_k(t_1), M_k(t_2), \ldots, M_k(A_n)$ sequence of ordinals is strictly decreasing where A_j denotes an initial segment of s_i: $A_j = \{s_i : 1 \leq i \leq j\}$.

Proof. We will prove that $M_k(A_{j-1}) > M_k(A_j)$ for an arbitrary $1 \leq j \leq n-1$. It cannot be the case that $s_j \in A_{j-1}$ because we would have $s_j = s_l$ for some $l \leq j-1$ which implies $s_l \leq_k s_j$. Apply lemma 4 with $A = A_{j-1}$, $B = A_j$, and $v = s_j$ and notice from lemma 1 that $M_k(A_{j-1}) \geq M_k(A_j)$. If we had equality here then there need to exist an $i < j$ such that $s_i \leq_k s_j$ contradicting our assumption. □
The above theorem in ACL2 looks like:

```
(defthm dickson-map-thm
  (implies (and (tuple-setp k S)
                (consp S)
                (natp k)
                (<= 1 k)
                (not (exists-partial-tuple-<=-set
                      k (rest S) (first S))))
           (o< (tuple-set->ordinal k S)
               (tuple-set->ordinal k (rest S)))))
```

For technical reasons—to operate on the native ordinal representation of ACL2—another theorem immediately derivable from the above was actually used in the termination proof.

The mapping function returns ordinals below ϵ_0 and this fact can be proved using induction on the number of variables. In fact the necessary inequalities were needed as supporting lemmas in the proof of Dickson's lemma, please see the proof script for details and also note that the ordinal representation allows ordinals below ϵ_0 only.

Conclusion

The need for a machine verified proof of Dickson's lemma lead to a new proof of the theorem. The explicit construction of the ordinal mapping which asserts that there is no infinite chain of 'offending' sequence of monomials (tuples) is not directly derivable from the classical proofs which are non-constructive in nature.

The proof effort enabled the verification of the termination of the Büchberger algorithm implemented in ACL2 and also motivated the development of an ordinal book which may benefit other proof attempts in the future.

Acknowledgments

The author thanks Sandip Ray for sharing his valuable comments and insights which helped to succeed and simplify this proof attempt. Robert Krug and Matt Kaufmann helped with ACL2 problems and questions. The author was glad to cooperate with I. Medina-Bulo and Panagiotis Manolios while working on this project, and to receive the support and encouragement of J Strother Moore.

References

[1] Thiéry Coquand and Henrik Persson. Integrated development of algebra in type theory. Preliminary version, May 1998.

[2] L. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *Am. J. Math.*, 35:113–122, 1913.

[3] Keith O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Kluwer Academic Publishers Group, Norwell, MA, USA, and Dordrecht, The Netherlands, 1992.

[4] Matt Kaufmann, Panagiotis Manolios, and J Strother Moore. *Computer-Aided Reasoning: ACL2 Case Studies*. Kluwer Academic Publishers, June 2000.

[5] Matt Kaufmann, Panagiotis Manolios, and J Strother Moore. *Computer-Aided Reasoning: An Approach*. Kluwer Academic Publishers, June 2000.

[6] Panagiotis Manolios and Daron Vroon. Algorithms for ordinal arithmetic. In *19th International Conference on Automated Deduction (CADE)*, 2003.

[7] Panagiotis Manolios and Daron Vroon. Ordinal arithmetic in acl2. In *ACL2 Workshop*, 2003. Submitted.

[8] F.J. Martin-Mateos, J.A. Alonso, M.J. Hidalgo, and J.L. Ruiz-Reina. A generic instantiation tool and a case study: A generic multiset theory. In *ACL2 Workshop*, 2002.

[9] F.J. Martín-Mateos, J.A. Alonso, M.J. Hidalgo, and J.L. Ruiz-Reina. Formalization and proof of Dickson's lemma in ACL2. Private communication, 2003.

[10] I. Medina-Bulo. Phd thesis. Private communication.

[11] J.L. Ruiz-Reina, J.A. Alonso, and M.J. Hidalgo F.J. Martin. Multiset relations: A tool for proving termination. In *ACL2 Workshop*, 2000.

[12] Laurent Théry. A machine-checked implementation of Buchberger's algorithm. *Journal of Automated Reasoning*, 26:107–137, 2001.