基于 Coq 的逆矩阵运算的形式化

**沈楠**1 **陈钢**1.2

1 南京航空航天大学计算机学院 南京 210000

2 北京京航计算与通信研究所 北京100074

（第一作者 Email 2257776063@qq.com 联系电话 15094319467）

**摘要：**矩阵是一种在计算机科学中应用广泛的数据结构，其运算正确性具有重要的意义。矩阵求逆在矩阵形式化工作当中，缺乏合理且实用的形式化工作。其原因在于，工程中现有的两种常见求逆方法的形式化均存在难点。第一种是基于伴随矩阵求解方法，难点在于无法形式化地表示n\*n矩阵的子矩阵，导致构建余子式组成的矩阵十分困难，因此难以实现伴随矩阵求解逆矩阵形式化；第二种称作高斯约旦初等变换求解法，难点在于构造初等矩阵及其操作函数。若使用Coq归纳结构设计操作函数，即采用行优先填充二维表的思想，将舍弃列维度对于二维表的描述信息，使得操作函数分支过多、需要设计复杂的归纳结构，导致后续形式化验证无法进行。本文提出了基于记录的矩阵函数构建法，使用行列两种维度同时描述矩阵，而使得构造并证明初等矩阵成为可能，在此基础上实现了在Coq系统中基于高斯约旦消元法的矩阵求逆的形式化工作。以一种代价更小且时间复杂度更低的方式，实现了首个形式化验证下的软件逆矩阵函数库。

**关键词：**形式化验证，形式化工程数学，逆矩阵形式化，Coq，软件安全

**Formalization of inverse matrix operation based on Coq**

Shen-nan1 ， Chen-gang1,2

1 College of Computer Science and Technology , Nanjing University of Aeronautics and Astronautics，Nanjing 210000，China

2 Beijing Jinghang Research Institute of Computing and Communication,Beijing 100074,China

**Abstract** Matrix is a data structure widely used in computer science, and its correctness of operation is of great significance. In matrix formalization, there is no reasonable and practical formalization work. The reason lies in the difficulty in formalizing the two common inverse methods in engineering. The first method is based on the adjoint matrix solution method. The difficulty lies in that the submatrices of N \* N matrix cannot be formally expressed, which makes it very difficult to construct the matrix composed of cosubformulas. Therefore, it is difficult to achieve the formalization of adjoint matrix inverse solution. The second method is called Gauss-Jordan elementary transformation method, the difficulty lies in the construction of elementary matrix and its operation function. If Coq is used to design the operation function of inductive structure, that is, the idea of filling two-dimensional table with rows first is adopted, the description information of two-dimensional table from column dimension will be discarded, so that the operation function branches too much and complex inductive structure needs to be designed, which leads to the failure of subsequent formal verification. In this paper, a record-based matrix function construction method is proposed, which describes the matrix in both column and column dimensions, making it possible to construct and prove the elementary matrix. On this basis, the formalization of matrix inversion in Coq system based on gaussian-Jordan elimination method is realized. And we implement the first software matrix inversion function library under formal verification in a way with lower cost and time complexity.

**Key words** Formal verification, Formal engineering mathematics, Inverse matrix formalization, Coq, Software security

引言

矩阵是线性代数中的主要研究对象与研究工具，相较于矩阵加法、转置、乘法等矩阵计算方式，矩阵求逆是矩阵运算当中最复杂的一种计算。近年来，矩阵求逆问题出现在诸多前沿领域之中。在机器学习领域，用矩阵逆方法求解二维线性回归问题。在信息安全领域，字符串经过代码子表转换后生成矩阵，利用矩阵的可逆过程，实现了信息的加密解密，并在此基础上发展出了众多优秀的信息加密技术。在飞行控制、火箭制导等安全攸关领域，卡尔曼滤波算法作为一种在仿真程序中的核心算法，需借助矩阵求逆,从而实现卡尔曼增益的更新。随着矩阵算法规模的扩大，矩阵的软件算法安全性日益重要。

工程数学领域中，通常非形式化方法会导致模型以及公式描述的偏差[1]。在安全攸关领域，传统的软件测试方法通常无法完全排除程序中的错误。特别是在航空航天、核电控制、智能制造、列车控制和医疗器械等安全攸关系统中，微小错误可能导致灾难性后果。形式化方法[2]具备精确描述数学模型以及公式推导的能力，以数学推导的方式验证软件符合设计的原则解决工程数学的安全问题。目前，许多定理证明器都已经建立起基础理论知识库，如实数、微积分、矩阵等，并验证其中的数学证明，这些研究工作为形式化工程数学的发展奠定了基础。定理证明器大体上可以分成两类，一类是基于自动证明的技术，另一类是人机交互式的证明技术[3]。前者的优势是自动化，缺点是只能证明一部分程序性质，随着问题的逐渐复杂，弊端也随之增大。后者的缺点是需要人工辅助，优点是能够更加全面完整的证明程序性质。当前，基于Coq定理证明器的交互式证明技术[4]是形式化方法的代表性成果之一，该技术已经广泛应用于数学定理证明和软硬件系统的安全性验证，以此来确保安全攸关系统的正确构造及可靠性。

在矩阵求逆形式化工作当中，目前存在少量相关工作，主要集中在数学性质的验证，且不存在具备软件移植性的矩阵形式化库。原因在于，现有的两种在工程中常见的矩阵求逆方法，均存在形式化困难。在基于伴随矩阵求解方法中，很难形式化地表示n\*n矩阵的子矩阵，导致矩阵余子式组成的矩阵的形式化非常复杂和困难[5]，无法轻易构建伴随矩阵求解逆矩阵。在高斯约旦消元法中，主流的矩阵形式化表示方法基于嵌套一维列表，对向量和矩阵的操作本质上是对其列表或嵌套列表的操作,即采用行优先填充二维表的思想，舍弃了列维度对于二维表的描述信息。

**Inductive** List A: Type :＝

| nil: List A

| cons: A－＞List A－＞List A

嵌套列表在描述特殊矩阵二维表时，每出现一行需要k个非零值的情况，都需要设计相应的一维列表操作函数，并验证相关性质。这种方式不具备多态特性，复用性差；且函数构造子过多，难以形式化地表示初等变换矩阵以及矩阵求逆操作，导致后续性质证明无法进行（详细见3.1）。

为了解决上述问题，本文总结分析了已有矩阵形式化工作的利弊，提出了基于记录的矩阵函数构建法。克服了初等矩阵以及逆矩阵操作函数的形式化问题，从而系统地给出了初等矩阵的定义以及逆矩阵的形式化证明。同时继承了Coq中基于记录的矩阵形式化方法的良好软件移植性 [6]，实现了首个形式化Ocaml矩阵函数库。

本文第2节总结分析了定理证明器中矩阵形式化的相关工作；第3节对比基于Coq 记录的矩阵形式化方法，介绍了初等变换矩阵的形式化难点并提出了解决措施；第4节提出了基于高斯约旦消元法的逆矩阵形式化方法；第5节介绍了逆矩阵定义以及定理的证明过程以及证明成果；最后总结全文。

相关工作

在高阶定理证明器中，各个定理证明器社区已经存在了关于矩阵理论形式化的研究工作，目前,针对定长数据结构的形式化技术大体上有3种：

1. 函数表示法描述矩阵。首都师范大学施智平团队，使用该方法在HOL4当中开展了比较完善的矩阵形式化工作[7-9]。该方法用函数来代表矩阵，具有良好的通用性，关于矩阵数学性质的证明简单易懂。Coq中的mathcomp库沿用了使用函数表示矩阵的思路，把矩阵定义成从两个有限集到A的函数[10，11]，即表示为Fin -> Fin -> Type类型。也就是说，它的输入集合改成了高度和宽度下标取值的集合。

这种方式以函数而不是实际数据结构的方式描述矩阵，同矩阵在程序语言中的实现方式距离比较远，难以从这样的表示方式中直接抽取出程序语言可处理的矩阵表示以及矩阵代码。这样一种表示方式适合于把矩阵看成是一个数学概念，并完成矩阵的数学性质证明。而不是把矩阵看成是一种软件中的数据结构，所完成的形式化工作是数学概念、算法和性质的形式化，而不是矩阵软件的形式化。

1. 采用依赖类型进行描述, Nicolas Magaud提出了基于依赖类型的list的矩阵形式化方法，但只给出了关于向量的一些简单性质的证明，并没有给出其他基础性质的证明。Coq.Vectors标准库提供了定长数组的实现，可在此基础上实现数组的数组，即矩阵。这种技术同样能够定义出完整的矩阵类型并用于矩阵函数的类型检查。此外，它比基于函数的矩阵定义更接近于程序语言中的矩阵数据结构。然而，由于定义的复杂性，这一技术所产生的形式化描述可读性较差而且证明难度很高，当处理更为复杂的二维矩阵问题时将面临较大的困难[12]。另外，从基于依赖类型的矩阵函数抽取出的列表需要在每一层包含该层的表长，因此在表达上冗余度大，占用空间多，计算时间过长。

Coquelicot库采用基于依赖类型的对偶类型构建矩阵类型，但矩阵相关性质的较少[13]。

（3）基于记录的矩阵表示法[6]，这种方法定义了一种用于表示矩阵的记录类型，记录的一个域用嵌套list的方法存储二维表数据，记录的另外两个域分别保存二维表长宽的证明，即关于二维表的高度和宽度分别满足矩阵类型的高和宽的要求的证明。麻莹莹等人在此基础上，首次实现了基于Coq的分块矩阵运算的形式化[14]。该方法易于理解并且方便使用，同其他矩阵形式化方法相比，记录类型更容易抽取出合理的Ocaml代码，因此形式验证的性质等同于抽取的软件代码的性质。

据我们所知，高阶定理证明器的逆矩阵形式化，仅存在少量研究工作：首都师范大学施智平团队实现了在HOL中基于伴随矩阵方法的逆矩阵形式化[5]，该工作构造了特殊的n\*n矩阵来代替(n−1)\*(n−1)子矩阵，以便计算其余子矩阵，从而使形式上构造一个伴随矩阵成为可能。但这种矩阵算法计算复杂度高，达到O（），不适合作为大矩阵的求逆计算，特别是在飞行控制等具有海量数据的领域。在ssreflect库[10，11]中也有矩阵求逆的定义，其方法同样是基于伴随矩阵方法的逆矩阵形式化。这种实现被三层包装(Matrix/Finfun/Tuple)隐藏，定义了较为复杂的结构，且矩阵类型被视为是抽象的数学概念，无法进行实际应用。

本文期望找到一种矩阵表示方案，一方面要求同矩阵的软件实现有更紧密的关联，另一方面要方便矩阵性质的证明。我们在马振威、麻莹莹的工作基础上增加了更多的矩阵性质证明，特别是初等矩阵函数性质的补充证明，在此基础上实现逆矩阵的形式化并证明相关性质。因此本文提出Record矩阵函数描述法，结合了两种典型方案的特性，构建实用矩阵形式化的基础库。

矩阵形式化

矩阵类型定义

Coq提供了Record命令来将多个对象聚合为单个对象的数据结构。Record中的每一个域，都作为描述该数据结构的一种属性，具有良好的可读性以及便利的代码抽取特性。马振威等人借助记录类型构建了一个矩阵类型的定义：

**Record** Mat (A:Set)(m n:nat): Set:=mkMat

{

mat: List (List A);

mat\_height: height mat=m;

mat\_width: width mat n

}.

基于记录的矩阵表示法中，Mat类型分别使用三个域来描述二维矩阵，第一是存储元素的二维表，由嵌套list构成，其他两个分别是这个二维表高和宽的证明，即指定列表的长度以及嵌套列表的长度[6]。在这个定义中,A表示矩阵内部元素的类型,是一种多态性质的变量，m和n分别描述矩阵的行数以及列数,mat\_height用于保存一个关于二维表mat的高度为m的证明,mat\_width则用于保存一个关于二维表mat宽度为n的证明。

矩阵构造函数

3.2.1嵌套list归纳构造法

Record矩阵的二维表采用嵌套list归纳构造法，向量和矩阵的操作本质上是对其中的列表或嵌套列表的操作,再将操作的结果转换成对应的向量或矩阵类型。

嵌套列表在描述特殊矩阵二维表时，每出现一行需要k个非零值的情况，都需要设计相应的一维列表操作函数，并验证相关性质，且不具备复用性。以描述n阶上三角矩阵为例，该方案存在n种不同的一维列表情况（含有k个元素的一维列表，1<=k<=n），那么需要分别构建n种一维列表操作函数list\_i（下面表示描述上三角矩阵的伪代码，省去了部分参数）。

**Inductive** Upper m n i1 i2… in j (v1 v2… : A):=

match m,i with

| O,\_ => List.nil

| S m', S i1' => List.cons list\_1 Upper

| S m', S i2' => List.cons list\_2 Upper

| S m', S i3' => List.cons list\_3 Upper

……

end.

以这种方式难以构造复杂矩阵，最核心的问题在于，该方案采用了行优先填充二维表的思想，舍弃了列维度对于二维表的描述信息，造成函数分支数量骤增。

Record矩阵性质证明采用induction机制，对每种出现的分支分别进行展开证明，证明的分支数量随着展开呈现指数型的上升。因此构造矩阵时复杂的归纳结构，导致高度宽度证明以及特别是后续矩阵性质引理证明困难。以下仅证明一维列表的长度性质时，含有1~3个元素的代码量已截然不同。随着矩阵规模的扩大，证明代码量也将以指数级的速度增长，显然不符合后续证明的需求。

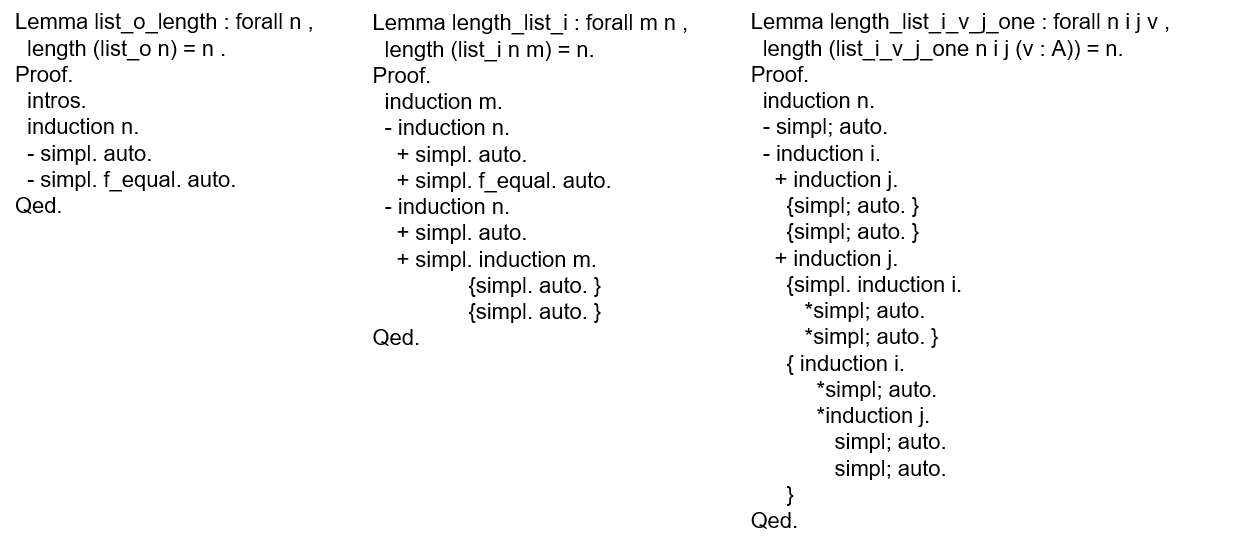


图3.1 1~3个元素证明代码量对比

Fig.3.1 1~3 element proof code quantity comparison

3.2.2 Record矩阵函数描述法

本文提出了基于Record矩阵的函数描述法，使用矩阵函数描述法，可根据函数内容定义不同矩阵，不同矩阵创建，只需要改变描述函数f: nat -> nat -> A。通过建立描述函数(f: nat -> nat -> A)和generate\_matrix函数产生的二维list之间的联系，合理利用了行参数i、列参数j对于二维列表的描述信息，以多态的思想解决了上述代码量爆炸的问题，使得构造复杂矩阵成为可能。该方法通过Mfill函数实现，核心为二维列表构造函数generate\_matrix以及generate\_row。

**Definition** Mfill (m n: nat) (f: nat -> nat -> A) :=

let dl := generate\_matrix f m n m in

mkMat A m n dl (dlist\_o\_height f m n) (dlist\_o\_width f m n).

函数generate\_row，根据函数f的内容，利用下标i，j，分别赋值产生相应的一维列表。相应二维列表，由一维列表构造函数generate\_row的结果依次赋值得到。

**Fixpoint** generate\_row (f: nat -> nat -> A) (j n i: nat) :=

match j with

| 0 => nil

| S j' =>

f i (n - j):: generate\_row f j' n i

end.

**Fixpoint** generate\_matrix (f: nat -> nat -> A) (m n i: nat) :=

match i with

| 0 => nil

| S i' =>

generate\_row f n n (m-i):: generate\_matrix f m n i'

end.

同时通过引理dlist\_o\_height，dlist\_o\_width证明所有由该函数创建的二维表高度，宽度均符合要求，免去了Record原有反复证明矩阵高度宽度的步骤，进一步压缩代码量。

以初等倍乘矩阵为例，借助行数x和值c两个参数即可创建n\*n维度的初等倍乘矩阵。

**Definition** multi\_mult (x: nat) (c: A) :=

@Mfill n n (fun i j => (if beq\_nat i j then (if beq\_nat i x then c else M.One) else M.Zero)).

这种方法克服了矩阵表示困难的问题，给其他复杂矩阵的构造提供了可能性，具备良好的扩展性。以函数的方式替代Fixpoint归纳结构，克服了在证明时遇到复杂矩阵难以展开归纳结构的问题[14]。

逆矩阵形式化

高斯约旦消元法

逆矩阵存在多种求解方法，最常见的称为基于伴随矩阵的求逆方法，采用构建行列式，伴随矩阵的定义 。该方法易于在命令式语言中表示，是实际应用中最常见的方法，但在高阶定理证明器当中，难点在于无法形式化地表示n\*n矩阵的子矩阵，导致构建余子式组成的矩阵十分困难[5]；需要额外构建行列式定义，证明行列式相关性质，导致引理数量变得庞大，给证明带来更多成本。

另一种常见方法叫做初等变换法求逆，又名高斯约旦消元法求逆。矩阵A经讨有限次初等行变换变为单位矩阵,则单位矩阵经过同样的初等行变换变为。相较于上述方法时间复杂度为，该方法计算复杂度为，降低了求逆计算量。

**1**

在高阶定理证明器当中，初等变换矩阵具有后进先出的栈性质，逻辑清晰,但由于初等变换存在多种求解可能性，需要确定一个合理且固定的变换顺序，作为求解的方法。

初等矩阵形式化

任何将单位矩阵做一次初等变换所得到的矩阵，称之为初等矩阵。本文定义了三种初等矩阵函数:

（1）初等倍乘矩阵multi\_mult：将单位矩阵第i行（或列）乘c，得到矩阵;

（2）初等倍加矩阵multi\_add：将单位矩阵第i行乘c加到第j行，或将第j列乘c加到第i列，得到矩阵;

（3）初等对换矩阵swap：将单位矩阵第i，j行（或列）对换，得到矩阵;

（为方便求逆，下文中定义的multi\_mult \_n，multi\_add\_n，swap\_n，分别表示维度是n\*n的三种初等变换矩阵。）

行阶梯矩阵形式化

对于任一个m \*n型矩阵，如果它的第一列元素不全为零，则可以对它作若干次初等行变换，将其化为行简化阶梯型矩阵，把所作的初等行变换所对应的初等矩阵依次记为， ，…，。

**2**

行阶梯函数row\_echelon\_form，输入参数为n\*n大小的方阵MA和负责控制列的参数i。函数正向求解所需的初等变换后的结果矩阵，另外逆序存储相应初等变换矩阵，作为求逆的条件。最终返回一个含有两个n\*n大小的方阵的对偶，第一个方阵存储变换矩阵，第二个方阵存储结果（即行阶梯矩阵）。

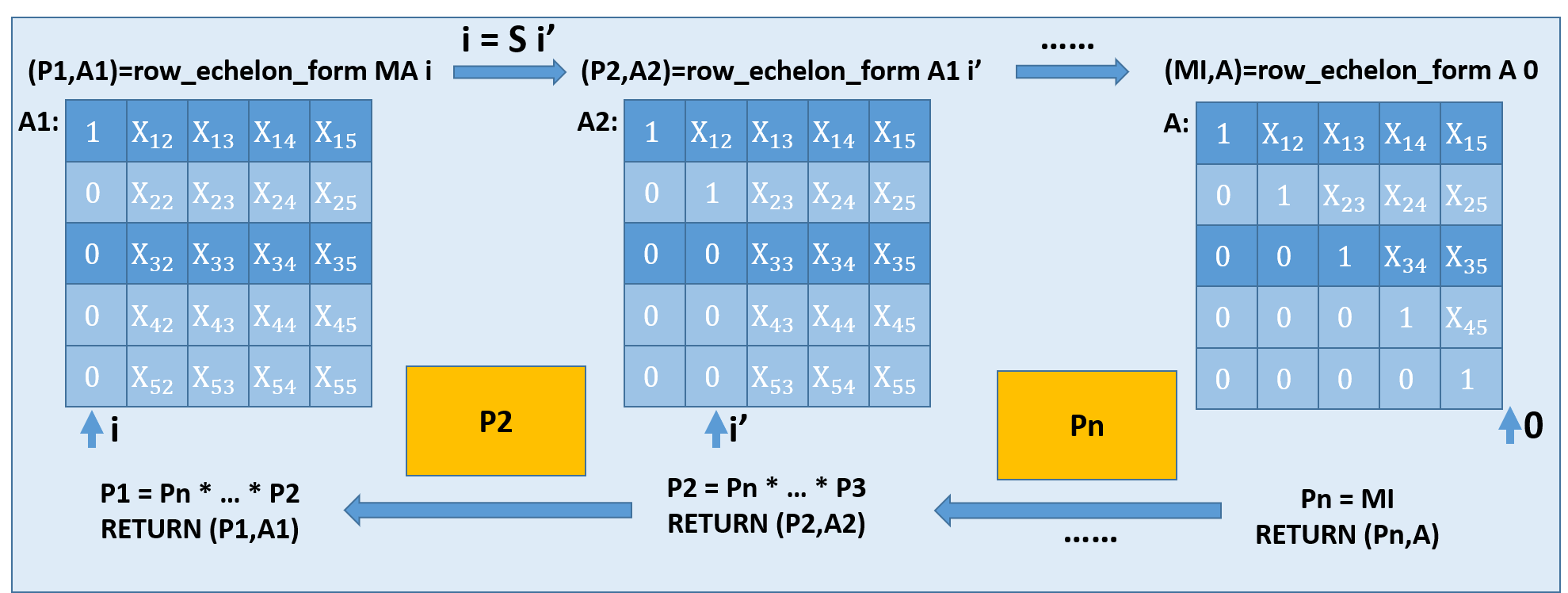


图4.1 row\_echelon\_form示意图

Fig.4.1 Diagram of fuction row\_echelon\_form

为匹配Coq中归纳变量i从n到0的归纳机制，统一使用(n - i)表示当前列。其中，M\*表示矩阵相乘，M.inv表示取倒数，A&[i，j]表示取下标为i j的元素值。

**Fixpoint** row\_echelon\_form (MA: Mat A n n) (i: nat) :=

match i with

| 0 => (MI, MA)

| S i’ =>

(\*找出当前列的主元（第一个值非0的元素）所在行，交换到第cur行\*）

let cur := (n-i) in

let r := first\_none\_zero MA i cur in

let P1:= (swap\_n cur r) in

let A0 := P1 M\* MA in

(\*求出对角线当前行，也就是第A&[cur，cur]个元素的逆元（即取它的倒数）\*）

let P2 := (multi\_mult\_n cur (M.inv A0&[cur，cur])) in

(\*更改当前行乘以逆元，第（i，i）个元素直接就是1\*）

let A1 := P2 M\* A0 in

(\*随后减去1\*该列大于主元行数的其他元素的值，依次消元\*）

let (P3, A2) := minus\_down A1 cur i’ in

let (P4, A3) := row\_echelon\_form A2 i' in

(P4 M\* P3 M\* P2 M\* P1, A3)

end.

求逆函数形式化

对于任何一个可逆矩阵A，都可以作若干次初等含变换将其化为单位矩阵E，即存在初等矩阵，，…，，使得

**3**

同理，单位化函数fst\_to\_I，输入参数为n\*n大小的行阶梯方阵MA和负责控制列的参数i。函数正向求解所需的初等变换后的结果矩阵，逆序存储相应初等变换矩阵，作为求逆的条件。最终返回一个含有两个n\*n大小的方阵的对偶，第一个方阵存储变换矩阵，第二个方阵存储结果（即单位矩阵）。

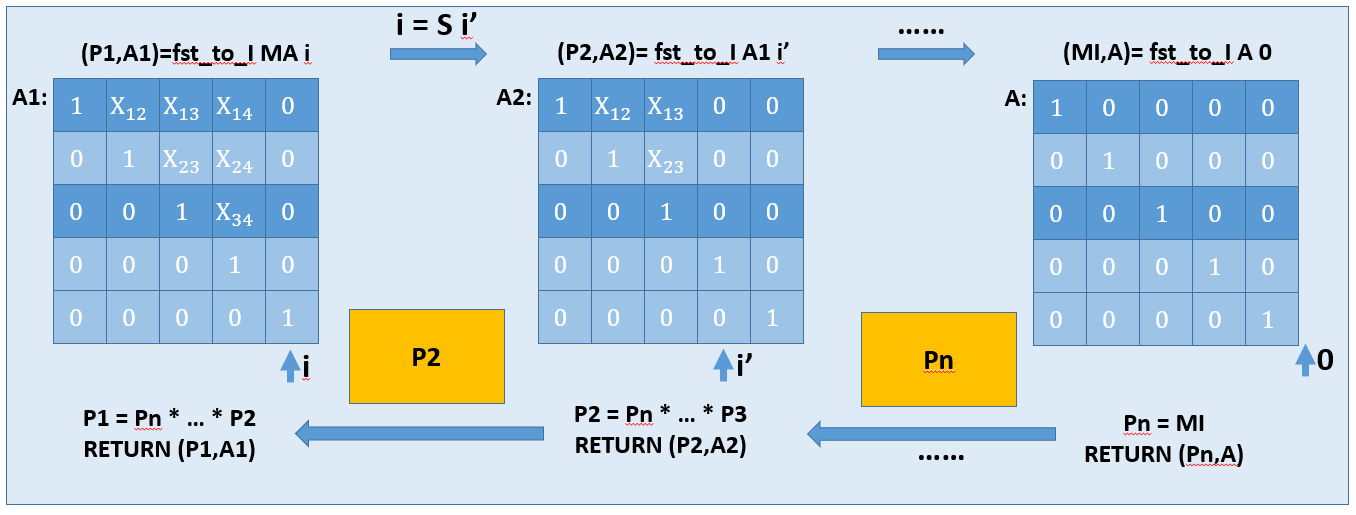


图4.2 fst\_to\_I函数示意图

Fig.4.2 Diagram of function fst\_to\_I

设计Inversion，输入参数为n\*n大小的行阶梯方阵。作用是合并上述两个函数，取出row\_echelon\_form对偶的第一个元素，即，并取出当前fst\_to\_I对偶的第一个元素,相乘得到。

**Definition** Inversion (MA: Mat A n n) :=

let (P1, A1) := row\_echelon\_form MA n in

let (P2, A2) := fst\_to\_I A1 n in

P2\* P1

end.

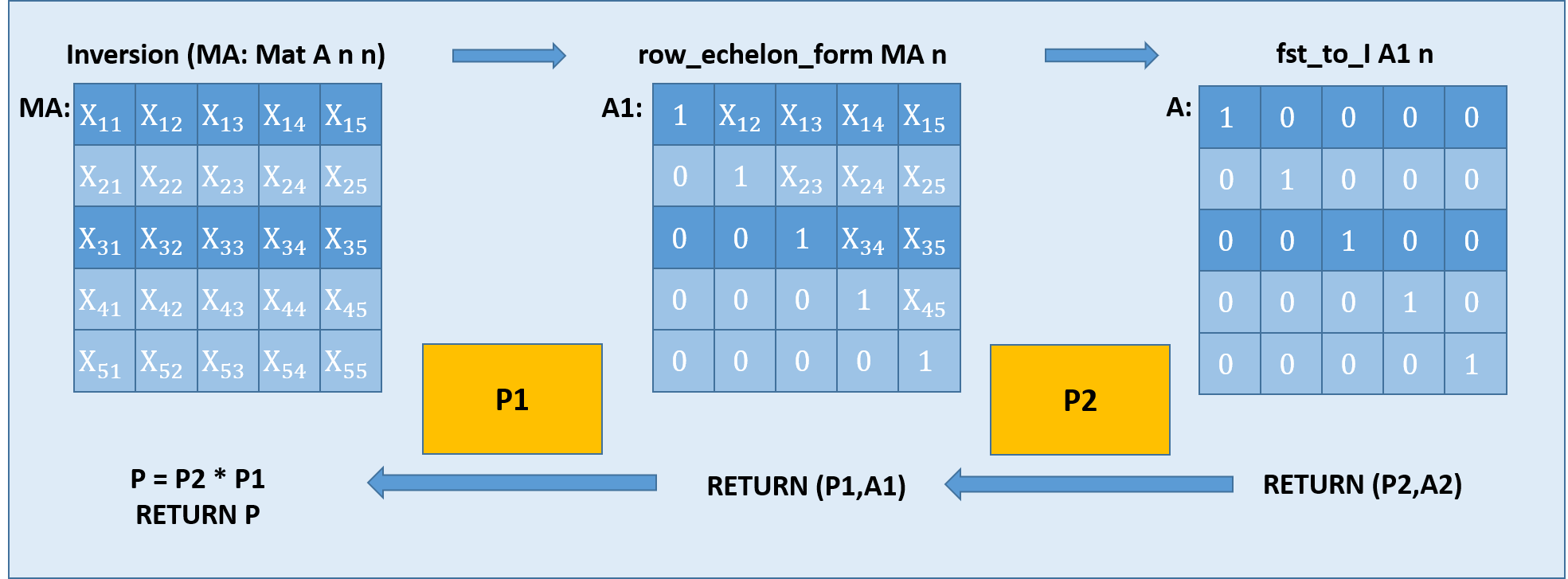


图4.3 Inversion函数示意图

Fig.4.3 Diagram of function Inversion

本次研究采取Module Type的方式定义矩阵元素的性质，使得所有矩阵函数具有多态性质，提取出了形式化整数矩阵、实数矩阵等基础库，可直接应用于Ocaml代码抽取。

**Module** Type MType.

……

End MType.

**Module** Matrix (M : MType).

逆矩阵证明

矩阵类型等价

在文献[6]中,矩阵类型的等价，采用Coq标准库中“eq”函数的方式定义矩阵间的等价关系，即莱布尼兹等式。

**Axiom** M\_eq : forall (A:Set) (m n:nat) (m1 m2:Mat A m n) , mat A m n m1 = mat A m n m2 <-> m1 = m2.

由于Record类型的等价，要求其中的每一个属性相等，即要求二维表的高度以及宽度证明引理相等。且该公理存在漏洞，前后条件并非存在充分必要关系，二维表相等无法得到Record内二维表的高度以及宽度证明引理相等。我们知道，对于两个矩阵，只需要满足二维列表的相等即可认为两个矩阵相等。而对于各自行、列数的证明，不必要要求相等。为了解决上述公理产生的问题,我们提出了新的矩阵等价判断函数 Meq,定义如下：

**Definition** Meq

{m n} (m1: Mat A m n) (m2 : Mat A m n) :=

forall i j，i < m -> j < n -> Mget m1 i j = Mget m2 i j.

Mget采用下标i，j的表示方法，用来解决Record类型的相等要求，而且能够充分利用下标访问的便利性，完成文献[6]中无法证明的性质。

矩阵与逆矩阵形式化性质证明

本文证明的矩阵函数性质，可看作是对文献[6]已有性质的基础上的补充。补充证明性质包括但不限于：（1）单位矩阵转置不变性质（2）左乘相同矩阵性质（3）右乘相同矩阵性质。证明逆矩阵性质有：（1）五大逆矩阵性质（2）矩阵乘法消去律（3）转置仍可逆性质（4）矩阵两次求逆相等性质等。

表1 矩阵性质引理

Table 1 Lemma of properties of matrix

|  |  |  |
| --- | --- | --- |
| Lemma\_name |  | lemma |
|  | trans |  |
| Inv\_trans\_trans |  | = |
| I\_trans\_I |  | = I |
|  | mult |  |
| mult\_equal\_r |  | A = B->A \* C = B \* C |
| mult\_equal\_l |  | A = B->C \* A = C \* B |
| matrix\_cancellation |  | -> A \* C = B \* C -> A = B |
|  | inv |  |
| row\_mul\_c\_inv |  | A \* B = I ->  A \* C = I -> C = B |
| AB\_inv |  | \* \* A\*B =I. |
| trans\_mult\_inv\_correct |  | \* = I |
| inv\_inv\_correct |  | = A |
| row\_mul\_c\_inv |  | = \* |

以单位矩阵转置不变性质I\_trans\_I为例，借助新等价定义，以destruct结构分解矩阵各行各列元素，可绕过复杂的list归纳结构直接证明正确性，从而证明转置仍可逆性质\* = I：

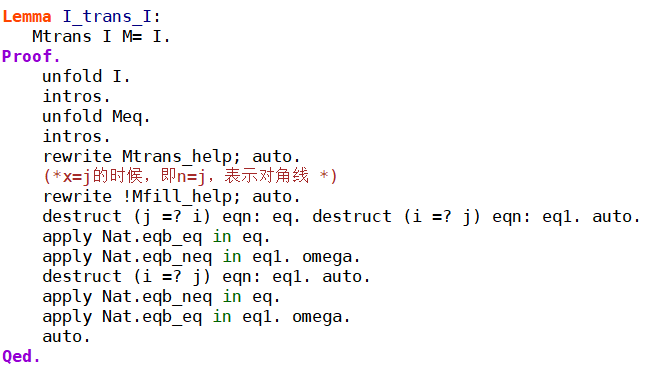


图5.1 单位矩阵转置不变性的验证

Fig.5.1 Verification of identity matrix transpose invariance

对于转置仍可逆性质证明如下：

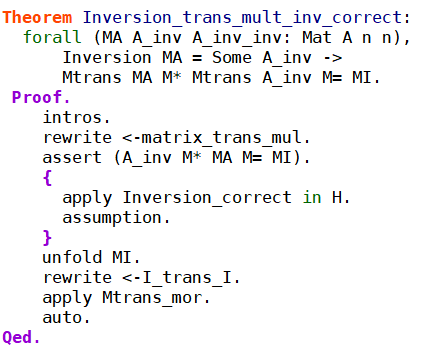


图5.2 转置可逆性的验证

Fig.5.2 Verification of the reversibility of transpose

从上述例子当中，可以发现新矩阵等价定义对于引理证明的简化，并且该等价定义亦然能够作用于更复杂的矩阵性质证明，以及逆矩阵性质证明，它能够很大程度上地化简逆矩阵函数性质证明过程，减少证明工作。

**结束语**

本文分析并对比了现有高阶定理证明器中的矩阵形式化工作，相较而言，利用函数表示法的矩阵形式化方法是一种较成熟的形式化方式，可以方便直观地表示复杂矩阵，具备表示复杂矩阵的可能性，但缺乏实际应用能力以及软件移植性；利用Record类型的矩阵形式化方法易于理解并且方便使用，记录类型更容易抽取出合理的Ocaml代码，具备良好的工程数学基础。本文提出基于Record的矩阵函数表示法，完成初等矩阵的形式化，首次实现了高斯约旦消元法求解逆矩阵的形式化，并实现了首个形式化验证下的逆矩阵软件函数库。相较于现有逆矩阵形式化工作，以代价更小且时间复杂度更低的方式，完成了逆矩阵形式化函数库的构建，并在实际应用中做出了初步探索。之后将继续完善相关引理，将函数库整合入飞行控制系统的形式化工作，是我们未来的研究重点。

**参考文献**

[1] Chen Gang, Qiu Zongyan, Song Xiaoyu, et al. A Report from Head Start: Formalizing Engineering Mathematics [J]. Communications of China Computer Society, 2017.

陈钢, 裘宗燕, 宋晓宇,等. 来自启智会的报告:形式化工程数学[J]. 中国计算机学会通讯, 2017.

[2] Almeida JB, Frade MJ, Pinto JS, Sousa SMD. An overview of formal methods tools and techniques[C]. in Rigorous Software Development, London: Springer, 2011: 15-44.

[3] Formalizing the safety of Java, the Java virtual machine, and Java card[J]. ACM Computing Surveys (CSUR),2001,33(4).

[4] Bertot Y, Casteran P. Interactive Theorem Proving and Program Development[M]// Interactive theorem proving and program development. Coq'Art: The Calculus of inductive constructions. Springer, 2004.

[5] Liming Li, Zhiping Shi, Yong Guan, Jie Zhang, Hongxing Wei. Formalizing the Matrix Inversion Based on the Adjugate Matrix in HOL4. 8th International Conference on Intelligent Information Processing (IIP), Oct 2014, Hangzhou, China. pp.178-186, ff10.1007/978-3-662-44980-6\_20ff. ffhal-01383331ff

[6] Ma ZW, Chen G. Matrix Formalization Based on Coq Record[J]. Computer Science, 2019. DOI: 10.11896/j.issn.1002-137X.2019.07.022

马振威,陈钢.基于 Coq 记录的矩阵形式化方法[J].计算机科学, 2019, 46(007):139-145.

[7] Shi ZP, Liu ZK, Guan Y, Ye SW, Zhang J, Wei HX, Luo GM. Formalization of Function Matrix Theory in HOL[J]. Journal of Applied Mathematics,2014,2014. DOI:10.1155/2014/201214

[8] Kang XN, Shi ZP, Ye SW, Guan Y. Formalization of Matrix Transformation Theory in HOL4[J]. Computer Simulation, 2014. DOI: 10.1080/19443994.2014.944221

[9] Yang XM, Guan Y, Shi ZP, Wu AX, Zhang QY, Zhang J. Higher-order Logic Formalization of Function Matrix and its Calculus[J]. Computer Science, 2016. DOI: 10.11896/j.issn.1002-137X.2016.11.005

[10] Sidi Ould Biha. Finite groups representation theory with Coq. 8th International Conference on Mathematical Knowledge Management 2009.

[11] Jónathan Heras, María Poza, Maxime Dénès, Laurence Rideau. Incidence Simplicial Matrices Formalized in Coq/SSReflect. Calculemus/MKM 2011: 30-44.

[12] DENES M,BERTOT Y.Experiments with computable matricesin the Coq system[R].France:INRIA,2011:2-27.

[13] Boldo S, Lelay C, Melquiond G. Coquelicot: A User-Friendly Library of Real Analysis for Coq[J]. Mathematics in Computer ence, 2015, 9(1):41-62. DOI: 10.1007/s11786-014-0181-1

[14] Ma YY, Ma ZW, Chen G. Formalization of Operations of Block Matrix Based on Coq. Ruan Jian Xue Bao/Journalof Software.

麻莹莹, 马振威, 陈 钢.基于Coq的分块矩阵运算的形式化.软件学报.2021

沈楠，1998年出生，研究生，南京航空航天大学学生。主要研究方向为形式化方法。

陈刚，1958年出生，博士，教授，中国计算机学会高级会员。主要研究方向为形式化方法。

SHEN Nan, born in 1998, postgraduate, is a student member of Nanjing University of Aeronautics and Astronautics. His main research interests include formal methods and so on．

CHEN Gang,born in 1958 ,Ph.D, professor, is a senior member of China Computer Federation．His main research interests include formal methods and so on．

自校负责人姓名、电话及邮箱

沈楠

15094319467

2257776063@qq.com