Final Report for Deliverable Nr. 4.3

Rohn's Theorem in Type Theory

Responsible: Yves Bertot, Yves.Bertot@sophia.inria.fr
Site: INRIA, France

Deliverable Date: July 2013

This report describes the current state for the formalized proofs of sufficient conditions for the regularity of matrices with interval coefficients, abbreviated in the project description as Rohn's theorem.

When addressing this example of formal description, we wanted to illustrate the use of formal proofs to study problems related to software used in robotics. This field has to cope with the difficulty of robustly controling physical objects, taking into account the problem that most of the measures performed by sensors have a limited precision. Everything has to be approximated and, even when theoretical problems have theoretical solutions, one must take into account the fact that these solutions may be extremely sensitive to errors introduced by the approximation process. In practice, the given robotics problems use linear algebra. Theoretically, problems are then represented by matrices whose properties must be studied. Because the coefficients may only be known up to some approximation, we end up having to consider matrices with interval coefficients. So we developed a formal library that considers matrices of that form.

1 The main theorems of Rohn's article

The most important statements from Rohn's articles [8, 9] are isolated in the formal library in one file[1]. One of the main feature is that these articles mention matrices with interval coefficients and matrices with plain coefficients. Obviously intervals represent sets of plain values and matrices with interval coefficients represent sets of matrices. the main results are expressed in the form "all matrices in the set represented by a given interval matrix are regular".

The formal development uses the `inSetm M N` predicate to express that some matrix with plain coefficients `N` belongs to the set represented by some matrix with interval coefficients `M`. In the following statements, `sigma_sol` is a predicate built using `inSetm` to express that there exists one matrix M inside the set represented by `A` and one vector V inside the set represented by `b` such that $M \times x = V$, as described in [7]. The function `mmuls_i` represents the obvious generalization of matrix multiplication of a matrix with interval coefficients by

[1] file `intSyst`

a matrix with plain coefficients. Last, the notation `setI` represents the intersection of two sets (the formalization customarily identifies predicates and sets of values satisfying these predicates.

```
Theorem Beeck1:
  forall x, sigma_sol A b x <->
     exists t , setI (inSetm (mmuls_i A x)) (inSetm b) t.
```

The ssreflect library of Coq already provides a comprehensive treatment of matrix multiplication and addition, but this treatment relies on the fact that the coefficient type is endowed with a ring structure, which is not the case here, because intervals, together with interval addition and multiplication, do not form a ring.

In the next example, theorem `thm31_midinva_specr`, `\rho` is the function that computes the spectral radius of a given matrix with plain coefficients. Last, when given an interval matrix `A`, `mmid A` obviously denotes the plain matrix whose coefficients are the centers of the intervals and `mrad A` denotes the matrix whose coefficients are the radii of the intervals.

```
Theorem thm31_midinva_specr:
forall X, specr (Mabs (1 - X * mmid A) + Mabs X * mrad A )  < 1 -> regular A.
```

This statement shows the importance of the spectral radius for the formal development. This has motivated most of the work done during the Formath project in task 4.3: contributing to a formal proof of the Perron-Frobenius theorem.

2 On complex coefficients

The results that we are mostly interested in rely on real number coefficients. The intervals have real numbers as bounds, and order is defined between real numbers. However, the spectrum of a matrix and the spectral radius are notions based on complex numbers, even if the spectral radius itself is a real value (that is, a complex value with a zero imaginary part). We use the definition of complex numbers as provided in the ssreflect library. In particular, when `F` is a real closed field the construction (similar to the construction of complex numbers) based on the cartesian product `F * F` and the relevant operations is shown to provide an algebraically closed field [2]. We proved that the type of real numbers, as provided by the Coq standard library, has the real closed field structure, as expressed in the ssreflect library. One thorny problem is the need to endow the type of real numbers with a `choiceType` structure. This structure ensures that one can associate an element of the type to any predicate, such that this element does satisfy the predicate if possible. Adding a `choiceType` structure is a step, somehow related to the axiom of choice, that is extremely uncomfortable with respect to constructive mathematics.

The concept of eigenvalue is already provided in the ssreflect library. An eigenvalue is an element of the base field such that the corresponding eigenspace is non-trivial. It is easy to relate this notion with the roots of the characteristic polynomial.

$$char(A) = det(A - XI)$$

We added this correspondence lemma, where `eigen_seq` is the sequence of roots of the characteristic polynomial and `M^%:C` denotes the natural injection of matrices with real coefficients in the type of matrices with complex coefficients.

```
Lemma eigenE n (M : 'M[R]_n) lam :
 (lam \in eigenvalue M^%:C) = (lam \in eigen_seq M^%:C).
```

Related to the notation `M^%:C`, which maps every real number to its complex counterpart, we also have the modulus function, which associates every complex value to its modulus, usually denoted $|v|$ in plain mathematical notation. Here the module is denoted as `` `|v| ``.

3 The Perron-Frobenius theorem

Because of the way it is defined (as the maximum modulus of an eigenvalue), the spectral radius of a matrix depends on the field structure in which this matrix is being considered. Thus, the same matrix with real coefficients will have a different spectral radius, whether it is seen as a matrix among the matrices with real coefficients, or as a matrix amon the matrices with complex coefficients. Of course, in type theory, the two matrices (with real coefficients and with complex coefficients that happen to be real) are distinct matrices, but the question arises because the Perron-Frobenius theorem that we will study in the next section actually states properties of the spectral radius when working in the field of complex numbers, while the formal proofs about Rohn's theorem have been proved using the spectral radius when working in the field of real numbers. We expect the Perron-Frobenius itself to provided ways of resolving this discrepancy, because the statement of the theorem actually states that under suitable conditions the modulus (a real value) is an eigenvalue. For now, while waiting for the theorem to be completely proved, we rely on the following axiom, where `specr` denotes the spectral radius in the field of real numbers, in other words the maximum absolute value of a real eigenvalue, and `\rho` denotes the spectral radius in the field of complex numbers.

```
Lemma corPF1 (m : nat) (A : 'M[R]_m): 0 <=m: A -> specr A = \rho A.
Proof. admit. Qed.
```

The property that is really used is then that the spectral radius can be used as a bound for coefficients that do not really act as eigenvalues.

```
Corollary corPF:
forall A a, Mle 0 A ->
   (exists u: 'cV_n,  0 <=m: u  /\ u <> 0 /\ (a *: u) <=m: A *m u) ->
  (a <= specr A)%Re.
```

Let's explain the notations: `*m` represents matrix multiplication and `*:` represents scalar multiplication, `A <=m: B` describes the comparison of two matrices A and B, this is simply a pointwise comparison of each coefficient. When considered as a matrix, `0` represents the matrix with all null coefficients, so that `0 <=m: A` expresses that all coefficients of `A` are non-negative. Last, the function `specr` describes the maximum of real eigenvalues.

The awkward characteristic of this corollary is that the witness provided in the hypothesis (named `u` in the existential statement) is not itself an eigen vector and it is therefore difficult to compare the value `a` with any eigenvalue. However, the relation with the spectral radius can be explained through the following statement, a key step in the proof of Perron-Frobenius theorem, which is also known as Gelfand's formula:

```
Lemma mx_cvg0P n (A :'M[R]_n.+1) :
  reflect ((fun k => A^+k) >->> 0) (\rho A < 1).
Proof. admit. Qed.
```

For matrices with positive coefficients, we proved that the following theorem was a consequence of `mx_cvg0P`: *for every positive matrix A and every eigenvalue λ such that $|\lambda| = \rho(A)$, if x is an associated eigenvector, then $|x|$ is also an eigenvector and the corresponding eigenvalue is $\rho(A)$.* This statements is written formally as follows:

```
Lemma Perron_gt0 n (A : 'M[R]_n) (x : 'cV_n) (lam : C) : 0 <m: A ->
  x != 0 -> (A *m x)^%:C = lam *: x^%:C -> `|lam| = (\rho (A))%:C ->
  [/\ A *m Mabs x = \rho (A) *: Mabs x, 0 <m: Mabs x & 0 < \rho A].
```

In this statement, `Mabs x` represents the vector where all coordinates are moduli of the coordinates of `x`. This statement naturally implies that $\rho(A)$ is an eigenvalue of A.

Going from matrices with positive coefficients to matrices with non-negative argument will rely on continuity arguments that we haven't been able to formalize yet.

The proof of Gelfand's formula made us embark on a long a difficult journey: the study matrices decomposed by blocks and various normal forms, among which the most important is the Jordan normal form.

4 Preliminary work on matrix structures

The lemma `mx_cvg0P` can benefit from results on equivalent matrices, because when $M = PNP^{-1}$ we also have $M^k = PN^kP^{-1}$. Then, we can use Jordan forms to reason, because powers of matrix in Jordan normal forms have nice properties.

4.1 Block diagonal matrices

As we saw in the previous sections, the type of matrices on a given ring is noted `'M[R]_(m,n)` in ssreflect. In particular, matrices have a dependent type, indexed by their size. This design choice has several important consequences. Among the advantages, it makes most statements more concise and explicit. If the size did not appear in the type, it would often be necessary to add explicit conditions as premises to most lemmas. This information guaranteed by types also makes it easier to construct abstract algebraic structures on top of matrices, for instance to show that square matrices are endowed with the structure of a ring.

However, this choice also has some drawbacks, most notably because typing algorithms consider sizes only modulo convertibility. Thus, `'M[R]_(m,n)` and `'M[R]_(m + 0, n)` are not recognized as the same type. In ssreflect this problem is particular sensitive because the

ring structure excludes the trivial ring that would have only one element (in ssreflect, it is required that the neutral element for addition, 0, should be distinct from the neutral element for multiplication, 1). Thus, square matrices form a ring only when their dimension is the successor of an integer (usually written `m.+1`).

Two functions from the library make it possible to work around problems due to type convertibility.

- The function `casmx`, given a matrix of type `'M[R]_(m1,n1)` and a proof that `m1 = m2` and `n1 = n2`, it produces a matrix of type `'M[R]_(m2,n2)` with the same coefficients.
- The function `conform_mx`, which takes two matrices `A` and `B` of type `'M[R]_(m1,n1)` and `'M[R]_(m2,n2)` and returns `B` when `m1=m2 n1=n2` and `A` otherwise.

The notion of block diagonal matrix only makes sense when the breakdown into blocks is specified. The interest of this kind of matrices is that many operations, like addition, can be expressed block by block, under the condition that both matrices have the same block structure.

A first idea would be to represent a block diagonal matrix with a sequence of dependent pairs (where each element in the sequence is a block indexed by its size). However, this would not make it possible to rely on typing to express that two matrices have compatible block structures.

Our definition of block diagonal matrices thus takes as first argument a list of natural numbers that describes the size of the blocks. Then we decide to represent each block with a function `F : forall (n : nat), nat -> 'M[R]_n`. Thus, if the i^{th} block has size n, it will be represented by `F n i`, or in a context where the size list is `s`, by `F s'_i i`.

To build a matrix from its blocks, we use the function `block_mx`. More precisely, if `A`, `B`, `C`, and `D` are matrices with suitable dimensions, then `block_mx A B C D` represents the matrix

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

A first try to describe block diagonal matrices relies on applying block_mx recursively:

```
Fixpoint diag_block_mx (s : seq nat) (F : forall (n : nat), nat -> M[R]_n) :=
if s is n :: s return M_(sumn s)
 then block_mx (F n 0) 0 0 (diag_block_mx s (fun n i => F n i.+1)) else 0.
```

But this definition does not fulfill the condition that we mentioned earlier that the size should be convertible to a success. Thus, it will be impossible to apply ring operations to block diagonal matrices, which was our initial motivation for defining them.

For the complete matrix to be in a type with a ring structure, we need at least one of the blocks to be non-empty. Moreover, if we want to apply ring operations on each block, we even need to impose that all blocks should be non-empty. The type for `F` becomes `forall (n : nat), nat -> 'M[R]_n.+1` and we define the block diagonal matrices in two stages with the following functions:

```
Fixpoint size_sum_rec k (s : seq nat) : nat :=
  if s is x :: l then k + (size_sum_rec x l).+1 else k.

Fixpoint diag_block_mx_rec k (s : seq nat)
 (F : (forall n, nat -> 'M[R]_n.+1)) :=
  if s is x :: l return 'M_((size_sum_rec k s).+1)
   then block_mx (F k 0) 0 0 (diag_block_mx_rec x l (fun n i => F n i.+1))
   else F k 0.

Definition size_sum s := if s is x :: l then size_sum_rec x l else 0.

Definition diag_block_mx s F :=
  if s is x :: l return 'M_((size_sum s).+1)
  then diag_block_mx_rec x l F else 0.
```

We defined the function `size_sum` in such a way that the matrix built by `diag_block_mx s F` has the size `(size_sum s).+1`. With this new definition the sequence `s` describes the predecessors of the size of blocks (not directly the sizes).

This way, the size constraint makes our definition of block diagonal matrices less natural. Authorizing trivial rings in the definition of rings of ssreflect seems possible, but this would make the treatment of some theories on rings less comfortable. For instance, one-variable polynomials are defined as lists of coefficients where the last one is non-zero (the set of polynomials on the trivial ring would then be empty, and in particular it would not be a ring).

4.2 Equivalent and similar matrices

Equivalent and similar matrices are fundamental notions in matrix algebra. Two matrices A and B (not necessarily square) are equivalent if there exists two invertible matrices M and N so that $MAN = B$. This expresses that the linear systems represented by A and B accept the same solution space (up to isomorphism).

Two square matrices A and B are similar if there exists one invertible matrix P such that $PAP^{-1} = B$, or equivalently $PA = BP$. Being similar is a stronger property than equivalence and applies only to square matrices. It expresses a base change. if A represents a given endomorphism f in a base $\mathcal{B}$ and if A is similar to B, then B represents the same endomorphism f in another base.

The formal definition of similar matrices is given as follows.

```
Definition similar m n (A : 'M[R]_m) (B : 'M[R]_n) :=  m = n /\
  (exists P : 'M_m , P \in unitmx /\ P *m A = (conform_mx P B) *m P).
```

In this definition `unitmx` is a predicat that recognizes the invertible matrix and `*m` is a notation for matrix multiplication.

This definition relaxes the type of the matrices given as argument. The predicate `similar` may be applied to matrices that have non convertible types but `similar A B` is only provable when `A` and `B` have provably equal sizes. We use the same trick when defining equivalence:

```
Definition equivalent m1 n1 m2 n2 (A : 'M[R]_(m1,n1)) (B : 'M[R]_(m2,n2)) :=
  [/\ m1 = m2, n1 = n2 & exists M, exists N,
  [/\ M \in unitmx , N \in unitmx & M *m A *m N = conform_mx A B]].
```

An important link between these two notions is that two matrices similar if and only if their characteristic matrices are equivalent:

```
Theorem similar_fundamental m n (A: 'M[R]_m) (B : 'M[R]_n) :
  similar A B <-> equivalent (char_poly_mx A) (char_poly_mx B).
```

Here `char_poly_mx A` denotes the characteristic matrices of `A`, it the matrix `XI - A` where `X` is the indeterminate of the polynomial ring $R[X]$.

This result is sometimes called "fundamental theorem of similarity on a field". To establish it we follow the proof as described in [12]. One of the implications is easy, if A and B are similar, there exists an inversible matrix P such that $A = PBP^{-1}$ and thus $XI - A = P(XI - B)P$.

The other direction is harder. Let's suppose there exist two invertible matrices M and N such that

$$M(XI - A)N = XI - B$$

Until now, the objects we manipulated were matrices with polynomial coefficients. Now, we ned to see them as polynomial expressions with matrix coefficients. This is possible using the following isomorphism:

$$\phi : M(R[X]) \to M(R)[X]$$

This isomorphism already played a key role in the formal proof of the Cayley-Hamilton as described in [6]. It was defined in the ssreflect library in the following manner:

```
Definition phi n (A : 'M[{poly R}]_n.+1) :=
  \poly_(k < \max_i \max_j size (A i j)) \matrix_(i, j) (A i j)`_k.
```

The notations `\poly_(i < n) a i` and `\matrix_(i < m, j < n) M i j` make it possible to describe respectively a polynomial or a matrix by the general expression of their coefficients. In the ssreflect library, a polynomial `p` is seen as sequence of coefficient. Thus, `size p` denotes the size of this sequence. When `p` is non-zero, its degree is given by `(size p).-1`.

We can now define the matrix polynomials M_1, M_0 and N_1, N_0 respectively by division on the left-hand side of $\phi(M)$ and division on the right-hand side of $\phi(N)$ by $X - B$.

$$\phi(M) = (X - B)M_1 + M_0 \quad \text{with} \quad \deg M_0 = 0$$

$$\phi(N) = N_1(X - B) + N_0 \quad \text{with} \quad \deg N_0 = 0$$

The key step in the proof consists in establishing the following identity:

$$M_0(X - A)N_0 = (1 - (X - B)R_1)(X - B)$$

with $R_1 = M_1 * \phi(M^{-1}) + \phi(N^{-1}) * N_1 - M_1 * (X - A) * N_1$

This is don using elementary algebraic manipulations. Then, since the degree of the left-hand-side member is 1 (M_0 and N_0 are constants). the matrix R_1 must be 0 (otherwise the right-hand-side member would have degree at least 2). The previous identity then becomes:

$$M_0(X - A)N_0 = (X - B)$$

And then, by identifying coefficients:

$$M_0N_0 = 1$$

$$M_0AN_0 = B$$

From this we infer $M_0 = N_0^{-1}$ $N_0^{-1}AN_0 = B$ and thus A and B are similar. Even if this proof seems natural, the objects in the study demand a careful treatment: these objects are polynomials in a non-commutative and non-integral ring, and they obviously do not enjoy all the usual properties of polynomials. Using a proof assistant helps us avoid the temptation of hasty reasoning steps by analogy with conventional computations.

For instance, reasoning on the degree of this kind of polynomials requires specific lemmas for unitary polynomials, or more generally, for the case where the dominant coefficient is a regular element in the base ring. Fortunately, the ssreflect library provides well-suited abstraction levels for this.

4.3 Smith Normal form

A matrix is in Smith normal form whe it has the following shape:

$$\begin{pmatrix} d_1 & 0 & \dots & \dots & & \dots & 0 \\ 0 & d_2 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \dots & 0 & d_k & 0 & \dots & 0 \end{pmatrix}$$

with the extra particularity that $\forall i, 1 \leq i < k, d_i \mid d_{i+1}$.

Every matrix on a principal domain is equivalent to a matrix in Smith normal form. This result can be viewed a a generalization of Gaussian elimination for a matrix on a field (which makes it possible to obtain a diagonal formw tih only 1s and was already known in China in the second century AD [1].

In this section we use a pre-existing formal description of an algorithm to produce Smith normal forms [3] to ensure its existence and we establish, independently from the algorithm, the uniqueness of the normal form, so that it characterizes the equivalence between two matrices on a principal domain. Last, by applying these tools to characteristic matrices, we can obtain the invariant factors of a matrix on a field and their properties.

4.3.1 Existence

To represent formally the Smith normal form, we use a function `diag_ms_seq` which denotes a diagonal matrix where all elements on the diagonal are given by a sequence given as argument. It is defined as follows:

```
Definition diag_mx_seq m n s := \matrix_(i < m, j < n) s`_i *+ (i == j :> nat).
```

The notation `x *+ x` when `x` is an element of a ring and `n` is a natural number describes `x` $+\ldots+$ `x` where `x` is repeated n times. In the expression of the general coefficient for the matrix given above, `i` and `j` are ordinals (i.e. numbers bounded respectively by `m` and `n`), they have distinct types (written respectively `'I_m`, `'I_n`). The formula `i == j :> nat` makes it possible to specify that these numbers must be compared as natural numbers and that the result should be a boolean value. Then a coercion maps this boolean value to a natural number (it maps `true` to `1` and `false` to `0`). Thus, `s`_i *+ (i == j :> nat)` denotes the element of rank `i` in `s` if `i` and `j` have the same value, and `0` otherwise.

The developments in the Formath project provide a function `Smith` which, when applied to a matrix `A` returns a list `s` and the matrices `L0` and R0 so that

- The sequence `s` is sorted for the divisibility relation,
- The matrix tt diag_mx_seq m n s is equivalent to `A` and the transfer matrices are `L0` and `R0`.

This is translated formally with the help of an inductive predicate:

```
CoInductive Smith_spec {m n} M : 'M[R]_m * seq R * 'M[R]_n -> Type :=
  SmithSpec L0 d R0 of L0 *m M *m R0 = diag_mx_seq m n d
    & sorted (@dvdr R) d
    & L0 \in unitmx & R0 \in unitmx : Smith_spec M (L0, d, R0).
```

and the correctness lemma that follows:

```
Lemma SmithP : forall (m n : nat) (M : 'M_(m,n)), Smith_spec M (Smith M).
```

4.3.2 Unicity

The algorithm to compute Smith normal forms relies on gcd computation, whose result is unique only modulo "association" (in other words, modulo the relation $\sim$ defined by $a \sim b$ if and only iff $a|b$— a divides b— and $b|a$—b divides a). We can thus not hope to associate to each matrix a Smith normal form that is strictly unique, but we prove unicity modulo $\sim$ for the diagonal coefficients. Since we are in an integral domain, this tantamounts to stating that the Smith normal form is unique modulo multiplication of the coefficients by invertible elements of the ring.

To establish this unicity result, we prove that the coefficients of the Smith normal form of an arbitrary matrix A can be expressed with the help of the gcd of various minors of A (i.e. determinants of square sub-matrices of A). More precisely, if we note $\wedge$ the gcd and $|A|_k$ the set of all minors of k order in A we have the following identity:

$$\prod_{i=1}^{k} d_i \sim \bigwedge_{x \in |A|_k} x.$$

To express this result formally, we use notions of sub-matrices and minors defined with the hep of re-indexing functions, as in [10].

The identity that we want to prove can then be expressed as follows for a sequence `s` and a matrix `A` that safisfies the specification `Smith_spec`:

```
Lemma Smith_gcdr_spec :
  \prod_(i < k) s‘_i  %=  \big[gcdr/0]_f \big[gcdr/0]_g minor k f g A .
```

where `%=` denotes the relation $\sim$ and the `big` notation indicates that we are repeating the gcd operation.

As a first step, we prove the theorem for the matrix `diag_mx_seq n n s` (instead of `A`). Since it is a diagonal matrix, the only non-zero minors of k order are products of k elements from the sequence `s`. Since chaque element of this sequence divides the next one, the gcd of all the minors of order k is the product of the k first elements of the `s`.

In the second stage, we only need to prove that the gcd of minors of order k in the matrix `A` is the same as the gcd of `diag_mx_seq m n s`

```
\big[gcdr/0]_f \big[gcdr/0]_g minor k f g (diag_mx_seq m n s)
   %= \big[gcdr/0]_f \big[gcdr/0]_g minor k f g A
```

We thus have to show that the two members of the equality divide eachother. Since the two proofs work exactly in the same manner, we describe here only how to show that the right-hand side divides the left-hand side.

To show that some value divides a gcd, it is enough to show that the value divides all the elements that are used as input to the gcd. We want to show that for any functions `f` and `g` the right-hand side divides `minor k f g (diag_mx_seq m n s)`. On the other hand, the matrix `diag_mx_seq m n s` is equivalent to `A` and there exists two matrices `M` and `N` so that `diag_mx_seq m n s = M *m A *m N`. Thus, we have to observe the determinant of a product. At this stage, we use the formal proof of the Binet-Cauchy formula [10], which can be stated as follows:

$$\det(AB) = \sum_{\substack{I \in \mathcal{P}(\{1,\ldots,l\}) \\ \#|I|=k}} \det(A_I)\det(B_I)$$

where A is a matrix of size $k \times l$ and B is a matrix of size $l \times k$, A_I (resp. B_I) is the matrix obtained with the K columns (resp. lines) of A (resp. B) whose indices are in I. This theorem makes it possible to transform the expression `minor k f g (M *m A *m N)` into a sum of minors. To divide a sum, it is enough to divide all its terms, so it is enough to prove that for any `h` and `i` we have:

```
\big[gcdr/0]_f \big[gcdr/0]_g minor k f g A %| minor k h i A
```

This is true by definition of the gcd operation.

The previous lemma used with $k = 1$ makes it possible to determine uniquely (modulo the equivalence relation $\sim$) the first diagonal element of the Smith normal form. It then imposes progressively all the other elements of the diagonal. Among the matrices that are equivalent modulo $\sim$ to the Smith normal form of `A`, we choose a representant that has the same determinant as `A`.

```
Definition Smith_form m n (A : 'M[R]_(m,n)) := diag_mx_seq m n (Smith_seq A).

Lemma det_Smith n (A :'M[R]_n) : \det (Smith_form A) = \det A.
```

4.3.3 Invariant factors

Let F be a field and A be a matrix with coefficients in F. We apply the Smith algorithm to matrix $XI - A$, the characteristic of A. According to the lmma `det_Smith` mentioned above the determinant of the Smith normal is the characterisc polynomial of A. This ensures that no diagonal element of the Smith normal form is zero, for the following two reasons :

- The determinant of the Smith normal form is the product of its diagonal coefficients,
- The characteristic polynomial of matrix is never zero.

As a consequence, The diagonal coefficients of the Smith normal form are non-zero polynomials with coefficients in the field, and it is possible to divide each of these polynomials by its leading coefficient to obtain monic polynomials.

```
Definition Frobenius_seq n (A : 'M[F]_n) :=
  [seq (lead_coef p)^-1 *: p | p <- (take n (Smith_seq (char_poly_mx A)))].
```

where `take n s` is the sequence of the first `n` elements of the sequence `s`. The Smith algorithm gives us a sequence, but no information on its size. Using the function `take` makes it possible to ensure that `size (Frobenius_seq A) = n`. This result will be useful in the formal development. This use of the `take` function does not modify the Smith normal form, as shown by the following result:

```
Lemma diag_mx_seq_take n s : diagmx_seq n n s = diag_mx_seq n n (take n s).
```

The invariant factors are the non-constant polynomials from `Frobenius_seq`:

```
Definition invariant_factors n (A : 'M[F]_n) :=
  [seq p : {poly R} <- (Frobenius_seq A) | 1 < size p].
```

We defined these invariant factors in such a way that they are monic. Later, we will work on the companion matrices of these polynomials, but we shall see that the usual properties of the companion matrices only hold when the polynomials are monic.

5 Frobenius normal forms

The Frobenius normal form of a matrix M is a block diagonal matrix where the blocs are companion matrices of the invariant factors of M. We shall now describe the our formalization of companion matrices before we give a precise definition of the Frobenius normal form. We then give a sketch of the formal proof that a matrix and its Frobenius normal form are similar.

5.1 Companion matrices

The companion matrix of a polynomial $p = X^n + a_{n-1}X^{n-1} + \ldots + a_1X + a_0$ is the following matrix.

$$\mathcal{C}_p = \begin{pmatrix} 0 & \ldots & 0 & 0 & -a_0 \\ 1 & \ddots & \vdots & \vdots & -a_1 \\ 0 & \ddots & 0 & \vdots & \vdots \\ \vdots & \ddots & 1 & 0 & \vdots \\ 0 & \ldots & 0 & 1 & -a_{n-1} \end{pmatrix}$$

Such a matrix is interesting because p is its characteristic polynomial and its minimal polynomial.

Normally, if `p` is non-constant polynomial, the dimension of the companion matrix is `(size p).-1`, but as we already stated, the size of a companion matrix needs to be convertible with the success of some number, so we define these matrices in such a way that their size is `(size p).-2.+1`, which is convertible with `(size p).-1` as soon as we work with non-constant polynomials.

```
Definition companion_mx (p : {poly R}) :=
  \matrix_(i,j < (size p).-2.+1)
      ((i == j.+1 :> nat)%:R + p`_i *+ ((size p).-2 == j)).
```

The notation `n%:R` denotes `1*+n` (i.e. `1 + ...+ 1` with n terms). This definition is valid but it does not make it possibl to construct block diagonal matrices where the blocs are companion matrices, because the type of `diag_block_mx` imposes to the function that describes the blocks to have the type `forall n, nat -> 'M_n.+1` but the type of `companion_mx` is `forall (p : {poly R}), 'M_((size p).-2.+1)`.

To solve this problem, we introduce an intermediate definition `companion_mxn` which relaxes the size of the result matrix.

```
Definition companion_mxn n (p : {poly R}) :=
  \matrix_(i,j < n) ((i == j.+1 :>nat )%:R + p`_i *+ ((size p).-2 == j)).

Definition companion_mx (p : {poly R}) := companion_mxn (size p).-2.+1 p.
```

Thus, `diag_block_mx s companion_mxn` is well type and has the exected properties if `s` contains sizez of the form `(size p).-2`. Lemmas about companion matrices will be expressed on `companion_mx`.

The Frobenius normal form of a matrix A (with coefficients in a field) is the following matrix:

$$\begin{pmatrix} \mathcal{C}_{p_1} & & & 0 \\ & \mathcal{C}_{p_2} & & \\ & & \ddots & \\ 0 & & & \mathcal{C}_{p_k} \end{pmatrix}$$

where the polynomials p_i are the invariant factors of the matrix A. Formally, it can be defined as follows:

```
Definition Frobenius_form n (A : 'M[R]_n) :=
  let sizes :=  [seq (size p).-2 | p : {poly R} <- (invariant_factors A)] in
  let blocks n i := companion_mxn n.+1 (nth 0 (invariant_factors A) i) in
   diag_block_mx sizes blocks.
```

5.2 From Smith to Frobenius

We will now show that every matrix on a field `F` is similar to its Frobenius normal form:

```
Lemma Frobenius n (A : 'M[F]_n.+1) : similar A (Frobenius_form A).
```

The theorem `similar_fundamental` given in section 4.2 shows that to prove this result, it is enough to show that the characteristic matrices of `A` and `Frobenius_form A` are equivalent, for any `A`.

We will start from the matrix $XI - A$ and by transitivity of equivalence, we will arrive at the characteristic matrix of `Frobenius_form A`.

We know that $XI - A$ is equivalent to its Smith normal form, which, considering monic polynomials, has the following shape:

$$\begin{pmatrix} 1 & & & & \\ & \ddots & & 0 & \\ & & 1 & & \\ & & & p_1 & \\ & 0 & & & \ddots \\ & & & & p_n \end{pmatrix}$$

where the polynomials p_i are the invariant factors of the matrix A.

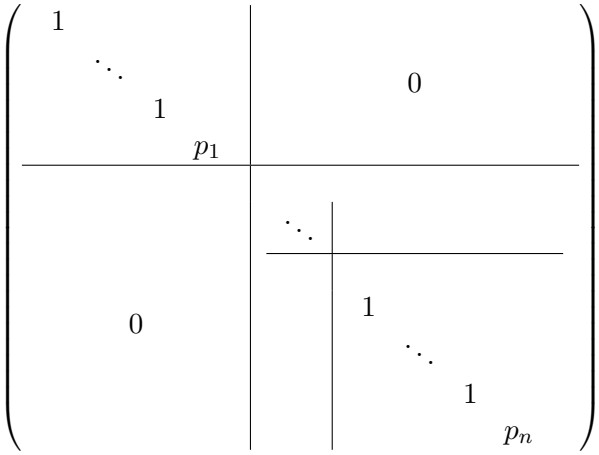
Permuting the diagonal elements preserves equivalence, because we can use permutation matrices, which are invertible, as transfer matrices. The formal statement of this reasoning step can be written as follows:

```
Lemma similar_diag_mx_seq m n s1 s2 :
  m = n -> size s1 = m -> perm_eq s1 s2 ->
  similar (diag_mx_seq m m s1) (diag_mx_seq n n s2).
```

where `perm_eq s1 s2` expresses that the sequences `s1` and `s2` are the same modulo permutation. We can permute the elements of the previous matrix:

$$\begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & 0 & & \\ & & 1 & & & & & \\ & & & p_1 & & & & \\ & & & & \ddots & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & 0 & & & & & & 1 \\ & & & & & & & & p_n \end{pmatrix}$$

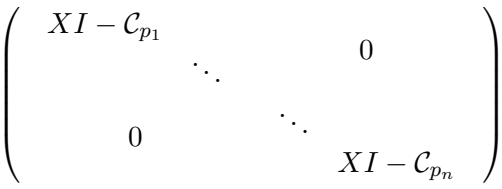
If `pi` represents the polynomial p_i, then the number of ones before p_i is `(size pi).-2`. This matrix can be seen as a block diagonal matrix:



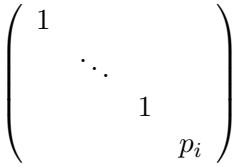
Now, we have to prove that this matrix is equivalent to the characteristic matrix of the Frobeinius normal form. We use first the fact that the characteristic matrix of block diagonal matrix is is block diagonal matrix, where each block is is characteristic to the corresponding block in the previous matrix:

```
Lemma char_diag_block_mx s (F : forall n, nat -> 'M[R]_n.+1) :
  s != [::] ->
  char_poly_mx (diag_block_mx s F) =
  diag_block_mx s (fun n i => char_poly_mx (F n i)).
```

We want to show that the previous matrix is equivalent with the following one:



We work block by block. For each index i we want to establish that $XI - \mathcal{C}_{p_i}$ is equivalent to:



This matrix is the Smith normal form of the matrix $XI - \mathcal{C}_{p_i}$. To show this last result, we use the lemma `Smith_gcdr_spec` from section 4.3. For every `k` such that `k < (size pi).-2`

we can find a sub-matrix of matrix $XI - \mathcal{C}_{p_i}$ that only has -1 on the diagonal:

$$XI - \mathcal{C}_{p_i} = \begin{pmatrix} X & \dots & 0 & 0 & a_0 \\ -1 & \ddots & \vdots & \vdots & a_1 \\ 0 & \ddots & X & \vdots & \vdots \\ \vdots & \ddots & -1 & X & \vdots \\ 0 & \dots & 0 & -1 & X + a_{n-1} \end{pmatrix}$$

In other words, for any `k` this matrix has a minor of order `k` associated to 1 (because $(-1)^k \sim 1$) and thus the gcd of all these minors is itself associated to 1. It is then possible to choose the `(size pi).-2` first diagonal elements of the Smith normal form of $XI - \mathcal{C}_{p_i}$ so that they are all equal to 1. For the last diagonal element the only possible choice is the polynomial p_i, because the product of all the elements on the diagonal is the determinant of the Smith normal form and ith is also the determinant of the matrix $XI - \mathcal{C}_{p_i}$.

6 Jordan normal forms

The Jordan normal form of a matrix A is an upper triangular matrix where the diagonal elements are roots of the characteristic polynomial of the matrix.

For this form to exist, it is enough that the characteristic polynomial splits in the coefficient field. To ensure this condition, we choose to work in an algebraically closed field `F`.

We first restate the elements of the theore of algebraically closed field that are used in our work [2]. Then we define the Jordan normal form and we show how to obtain it from the Frobenius normal form.

6.1 Polynomials with coefficients in an algebraically closed field

If p is a polynomial with coefficients in an algebraically closed field such that

$$p = \prod_{i=1}^{m} (X - \lambda_i)^{\mu_i}$$

then

- we name `root_seq p` the sequance containing all λ_i,
- we name `root_mu_seq p` the sequence of pairs (μ_i, λ_i),
- we name `linear_factor_seq p` the sequence of polynomials $(X - \lambda_i)^{\mu_i}$,
- if `s` is a sequence of polynomials, then `roo_seq_poly s` is the concatenation of all sequences `root_mu_seq` for each of the polynomials in `s`.

6.2 Definitions

We call a Jordan block a matrix with the following shape:

$$J(\lambda, n) = \begin{pmatrix} \lambda & 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \dots & \dots & 0 & \lambda \end{pmatrix}$$

This is written formally in the following manner:

```
Definition Jordan_block lam n : 'M[F]_n :=
  \matrix_(i,j) (lam *+ (i == j) + (i.+1 == j)%:R).
```

The Jordan normal form is a block diagonal matrix where each block is Jordan:

```
Definition Jordan_form  n (A : 'M[R]_n.+1) :=
  let sp := root_seq_poly (invariant_factors A) in
  let sizes := [seq x.1 | x <- sp] in
  let blocks n i :=  Jordan_block (nth (0,0) sp i).2 n.+1 in
  diag_block_mx sizes blocks.
```

In what follows, we explain how to go from a Frobenius normal form to a Jordan normal form. This makes it possible to understand the definition given above.

6.3 From Frobenius to Jordan

When A is a matrix with coefficients in an algebraically closed field, we have alread proved that this matrix is similar to its Frobenius normal form. We will now show that the Frobenius normal form is similar to the Jordan normal form. This will make it possible to obtain the following result by transitivity:

```
Lemma Jordan n (A : 'M[F]_n.+1) : similar A (Jordan_form A).
```

Without loss of generality, we can consider only one of the blocs in the Frobenius normal form. We fix an index i and work on the matrix $\mathcal{C}_{p_i}$ where p_i is the i^{th} invariant factor of A. Let's first show that if $q = q_1 \dots q_m$ and the polynomials q_i are pairwise coprime, then the matrix $\mathcal{C}_q$ is similar to:

$$\begin{pmatrix} \mathcal{C}_{q_1} & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & \mathcal{C}_{q_m} \end{pmatrix}$$

By induction on m it is enough to show that $\mathcal{C}_q$ is similar to:

$$\begin{pmatrix} \mathcal{C}_{q_1} & 0 \\ 0 & \mathcal{C}_{q_2 \dots q_m} \end{pmatrix}$$

Our first attempt to prove this equivalence was to construct explicitly the transfer matrix. This attempt achieves the result, but this proof is rather long and consists in reproducing the proof of the Smith algorithm. We finally decided to use a proof that uses the theorem `similar_fundamental` from section 4.2. This boils down to proving an equivalence between matrices that contain only Smith normal forms of companion matrices. To show this equivalence we use the lemma `Smith_gcdr_spec` in the same manner as in section 5.2. We obtain that $q_1 * q_2 \dots q_m = q$ is the only invariant factor for the small matrix above. The two matrices have the same invariant factors and are therefore equivalent. This is enough to conclude.

Since the field `F` is algebraically closed, we can decompose the invariant factor:

$$p_i = \prod_{j=1}^{m_i} (X - \lambda_{ij})^{\mu_{ij}}$$

where the λ_{ij} are the roots of p_i and the mu_{ij} their multiplicities. The previous result makes it possible to establish that the companion matrix $\mathcal{C}_{p_i}$ is similar to:

$$\begin{pmatrix} \mathcal{C}_{(X-\lambda_{i1})^{\mu_{i1}}} & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & \mathcal{C}_{(X-\lambda_{im_i})^{\mu_{im_i}}} \end{pmatrix}$$

We will now prove that for every λ and n the Jordan block $J(\lambda, n)$ is similar to the companion matrix $\mathcal{C}_{(X-\lambda)^n}$. Here again, we first proved this result by giving explicitely the transfer matrix, but this verification is long and makes uses of heavy computations on binomial coefficients. We finally use the same method as previously, using theorem `similar_fundamental` to transform the problem into an equivalence and then the lemma `Smith_gcdr_spec` to state the invariant factors of the matrices. We thus show that the only invariant of $J(\lambda, n)$ is the polynomial $(X - \lambda)^n$.

We can now establish that the matrix $\mathcal{C}_{p_i}$ is similar to:

$$\begin{pmatrix} J_{i1} & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & J_{im_i} \end{pmatrix}$$

with $J_{ij} = J(\lambda_{ij}, \mu_{ij})$. Thus, each block of the Frobenius normal form of A is similar to a matrix like the one above where (λ_{ij}, μ_{ij}) are the pairs composed of roots of the invariant factors and their multiplicity. This explains the definition of the Jordan normal form stated earler and proves that the Frobenius noral form of A is similar to the Jordan normal form.

6.4 Diagonalization

We have seen that in an algebraically closed field, every matrix is similar to its Jordan normal form, which is upper triangular. This directly gives a theorem for trigonalization. We will now study the conditions for matrix to be diagonalizable.

We saw that the Jordan normal form of a matrix A is composed of Jordan blocks $J(\lambda, k)$ where λ is a root of an invariant factor of A and k its multiplicity. Moreover, k is also the size of the block $J(\lambda, k)$. If $k = 1$, the Jordan normal form is diagonal. Since invariant factors divide each other, it is enough that the last only has simple roots for all invariant factors to only have simple roots. The last invariant factor is the minimal polynomial of A. So it is enough that the minimal polynomial of the matrix A only has simple roots for A to be diagonalizable.

```
Lemma diagonalization n (A : 'M[R]_n.+1) : uniq (root_seq (mxminpoly A)) ->
  similar A (diag_mx_seq n.+1 n.+1 (root_seq (char_poly A))).
```

Here `uniq` is a predicate expressing that the sequence has no duplicate values.

7 Conclusion

In this work, we chose to deduce the existence of Frobenius and Jordan normal forms from the theory of invariant factors for a matrix on a principal domain. This point of view is quite natural, but is not systematically used in the literature. Some other approaches rely on an *ad hoc* theory of cyclic endomorphisms.

The theorem of existence for Frobenius normal forms is important because it gives means to handle the structure of endomorphisms for a vector space of finite dimension on any field. To our knowledge, the work presented in this report is the first formal proof of this result.

An extra outcome of our development is that it presents definition and properties for block diagonal matrices and companion matrices, which can have many reuses, since these are basic notions of matrix algebra. In the same manner, even though our development handles mathematical notions that are somewhat elementary, it already rests on several pre-existing formal libraries and it provides a testbed for these libraries. For instance, the existence of the Smith normal [3], algebraically closed fields [2], or the Binet-Cauchy formals and the definition of submatrices and corresponding minors [10] were developed by other authors.

In most cases, the existing libraries were easy to adapt to our context. The proof of the fundamental theorem of similarity on a field, as described in section 4.2 notably took advantage of the modularity of the theory of polynomial division provided in ssreflect.

However, our work was made harder by the fact the general definition of rings excludes the trivial ring, as we explained in section 4.1. Modifying this definition might have bad consequences, for example provoking a desastrous change in the size of the global hierarchy of algebraic structures, which might put to much stress on the implementation of Coq. An other point that could be improved and is independent from the previous is that our definition cannot be iterated. Indeed, it is not possible to define a block diagonal matrix whose blocks are themselves block diagonal matrices. The main technical difficulty is germane to the problem described in section 5.2 for companion matrices.

Our next objective is to use the notions defined here as specifications for efficient implementation of an algorithm to compute the Frobenius form. For instance, the algorithm described in [11] has a complexity $O(n^\omega \log n \log\log n)$ assuming an algorithm for matrix

product in complexity ω and is deterministic. A subroutine of this work is known as Keller-Gehrig's algorihm [5] and has an interest of its own because it makes it possible to compute the characteristic polynomial of a matrix of size n in $O(n^\omega \log n)$.

To implement and prove the correction of these algorithms we plan to use the technique based on refinements and the formally verified implementation of Strassen's algorithm for matrix multiplication as described in [4].

The complete development for canonical forms of matrices can be found at the following address: http://www-sop.inria.fr/members/Maxime.Denes/canonical_forms.

Concerning Rohn's theorems, we intend to use Jordan normal forms to establish Gelfand's formula in the case of matrices with positive coefficients. Then, the remaining work will consist in generalizing the result to matrices with non-negative coefficients by using a density argument (every matrix with non-negative coefficients is arbitrarily close to a matrix with positive coefficients and arbitrarily close eigenvalues).

Other theorems from Rohn's collection rely on rayleigh coefficients. This subject has not been considered yet.

References

[1] K. Chemla, G. Shuchun, G. E. R. Lloyd, and T. Yasumoto. *Les neuf chapitres le classique mathmatique de la Chine ancienne et ses commentaires.* Dunod, Paris, 2004.

[2] C. Cohen. *Formalized algebraic numbers: construction and first order theory.* PhD thesis, École Polytechnique, 2012.

[3] C. Cohen, M. Dénès, A. Mörtberg, and V. Siles. Smith Normal form and executable rank for matrices, 2012. http://wiki.portal.chalmers.se/cse/pmwiki.php/ForMath/ProofExamples.

[4] M. Dns, A. Mrtberg, and V. Siles. A refinement-based approach to computational algebra in coq. In *Interactive Theorem Proving*, volume 7406 of *LNCS*, pages 83–98, 2012.

[5] W. Keller-Gehrig. Fast algorithms for the characteristics polynomial. *Theoretical Computer Science*, 36:309–317, January 1985.

[6] S. Ould Biha. Formalisation des mathématiques : une preuve du théorème de Cayley-Hamilton. In *JFLA (Journées Francophones des Langages Applicatifs)*, pages 1–14, Etretat, France, 2008.

[7] Ioana Pasca. Formally Verified Conditions for Regularity of Interval Matrices. In *Symposium on the Integration of Symbolic Computation and Mechanised Reasoning, Calculemus*, volume 6167 of *LNAI*, pages 219–233, Paris, France, June 2010. Springer. The final publication is available at www.springerlink.com.

[8] Georg Rex and Jiri Rohn. Sufficient conditions for regularity and singularity of interval matrices. *SIAM Journal on Matrix Analysis and Applications*, 20:437–445, 1998.

[9] Jiri Rohn. Forty necessary and sufficient conditions for regularity of interval matrices: A survey. *Electronic Journal of Linear Algebra*, 18:500–512, 2009.

[10] V. Siles. A formal proof of the Cauchy-Binet formula, 2012. http://wiki.portal.chalmers.se/cse/pmwiki.php/ForMath/ProofExamples.

[11] A. Storjohann. Deterministic computation of the frobenius form. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 368–377, 2001.

[12] J.H.M. Wedderburn. *Lectures on Matrices*. Colloquium Publications. American Mathematical Society, 2008.