Minimally informative protocol for the Russian Cards Problem: A Verification using Coq

Zoe Leyva-Acosta

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, UNAM

1 Introduction and preliminary notions

The problem we are dealing with, can be stated as an instance of the following *Generalized Russian Cards problem* [4]:

Let D be a deck of n cards labeled from 0 to $n-1$, which are distributed among three players A, B, C, so that A gets $\mathbf{a}$ cards, B gets $\mathbf{b}$ and C gets the remaining $\mathbf{c}$ cards. Show whether is possible for A and B to learn each other‘s cards via public communication while ensuring that C cannot know whether A or B has any particular card (except for the ones C owns).

A particular instance of this problem can be described by the signature $(\mathbf{a}, \mathbf{b}, \mathbf{c})$, with $n = \mathbf{a} + \mathbf{b} + \mathbf{c}$. In this project we are concerned with the *Classic Russian Cards problem* with signature $(3, 3, 1)$. In particular, we want to formally prove some properties about an *announcement protocol* for player A.

In this scenario, an *announcement* is a subset of $\mathscr{P}_3(D)$[1], which represents the set of alternative hands that A may hold. Thus, an *announcement protocol* for player A is a function $P_A : \mathscr{P}_3(D) \to \mathcal{M}$, where $\mathcal{M}$ is the set of possible messages A may send to B. We assume this function is surjective, thus, for a message $M \in \mathcal{M}$, $P_A^{-1}(M)$ is the *announcement* corresponding to M. Hence, we can also describe an m-message protocol, with $m = |\mathcal{M}|$, as a set of m announcements $P_A = \{P_A^{-1}(M) | M \in \mathcal{M}\}$.

Moreover, we are mostly concerned with the announcement's requirement or property known as *minimally informative*. Hence, the main propose of this project is to provide a formal verification of the this property for an announcement protocol for the Classic Russian Cards problem, given its definition or construction. Informally, an announcement for player A is said to be *minimally informative* for the *Classic Russian Card Problem* with signature $(3, 3, 1)$, if allows B to learn at least one (not necessarily all) of A's cards from her announcement. Also, we are interested in a verifying this property for a *two-message protocol*, for which $|\mathcal{M}| = 2$, i.e., there are only two possible announcements, which we denote as $P_A^{-1}(0)$ and $P_A^{-1}(1)$.

The fallowing is an example of a two-message minimally informative protocol χ for $(3, 3, 1)$:

$\chi^{-1}(0) = \{012, 013, 014, 015, 016, 023, 024, 025, 036, 046, 126, 134, 135, 156,$

[1] The notation $\mathscr{P}_{\mathbf{a}}(D)$ stands for all subsets of size $\mathbf{a}$ of the set D.

$$\begin{aligned}&234, 245, 246, 256, 345\}\\ \chi^{-1}(1) = &\{026, 034, 035, 045, 056, 123, 124, 125, 136, 145, 146, 235, 236, 346,\\ &356, 456\}\end{aligned}$$

The aim of this project is the formalization of the previous notions using the Coq proof assistant, in order to formally verify by this mean that the protocol we propose in Section 3 is indeed a *minimally informative announcement protocol* for player A for the Classic Russian Cards Problem.

In general, the terminology and formulations we use in this project are those from the work of Rajsbaum [2], which heavily relies on graph theory and in particular in Johnson Graphs.

2 Formalizing the problem

Let $D = \{0, \ldots, n-1\}$, $n > 1$, be the *deck* of n distinct cards. An element in the deck is a *card.* A subset a of cards is a *hand*, $a \in \mathscr{P}(D)$. For a hand a we denote $\bar{a}$ to be the set $D - a$, i.e., $\bar{a}$ is the complementary set of a with respect to D. We may say for short that a, $|a| = m$, is a m-set or m-hand, namely, if $a \in \mathscr{P}_m(D)$, the subsets of D of size m.

A *deal* $= (a, b, c)$ consists of three disjoint hands, meaning that cards in a are dealt to A, cards in b to B, and cards in c to C. We say that the hand is the *input* of the agent. We call $\gamma = (\mathbf{a}, \mathbf{b}, \mathbf{c})$ the *signature* of the *deal* $= (a, b, c)$ if $|a| = \mathbf{a}$, $|b| = \mathbf{b}$ and $|c| = \mathbf{c}$. Hence, for the instance of the problem with signature $(\mathbf{a}, \mathbf{b}, \mathbf{c})$, the inputs of A, B and C, are the hands $a \in \mathscr{P}_{\mathbf{a}}(D)$, $b \in \mathscr{P}_{\mathbf{b}}(D)$ and $c \in \mathscr{P}_{\mathbf{c}}(D)$, respectively.

Given a $\mathbf{b}$-set $b \in \mathscr{P}_{\mathbf{b}}(D)$, we use the notation $K_p(\bar{b})$, with $|K_p(\bar{b})| = p = \binom{n-\mathbf{b}}{\mathbf{a}}$ for denoting the subset of $\mathscr{P}_{\mathbf{a}}(D)$ whose elements are all disjoints sets with respect to b. In other words, $K_p(\bar{b})$ consist of all elements $a \in \mathscr{P}_{\mathbf{a}}(D)$, such that $a \subset \bar{b}$. This notation is due to the fact that the elements in $K_p(\bar{b})$ induce a click in the distance $\mathbf{c}$ Johnson graph, $J^{\mathbf{c}}(n, \mathbf{a})$. Intuitively, $K_p(\bar{b})$ the set of hands that B considers possible for A, provided that B holds the hand b.

We now instantiate the *minimally informative* characterization from [2, Theorem 2], for the case of the Classic Russian Cards problem, with signature $(3, 3, 1)$:

Theorem 1. ***Minimally informative announcement protocol for*** $(3, 3, 1)$. *Let $P_A : \mathscr{P}_3(D) \to \mathcal{M}$ be a protocol for the Classic Russian Cards Problem, then P_A is minimally informative if and only if for each $b \in \mathscr{P}_3(D)$, there are some $a, a' \in K_4(\bar{b})$ such that $P_A(a) \neq P_A(a')$.*

3 A two-message minimally informative protocol construction for (3,3,1)

The protocol construction we present here is based on Singer difference sets (or perfect difference sets) [3] and is inspired in the *good announcement* construction proposed in [1, Theorem 3].

First, we present the notions that we use for the protocol construction and then, some results that will be useful for proving that such construction yields a deterministic minimally informative protocol for $(3, 3, 1)$, using two messages.

Definition 1. *A set S of size $m+1$, is a perfect difference set if the differences $s_i - s_j$ module $m(m+1)+1$, with $i \neq j$, $s_i, s_j \in S$, are all the different integers from 1 to $m(m+1)$.*

In the following, the notation $x + S$ for a set S stands for the set $\{x + s \mod v \mid s \in S\}$.

3.1 Protocol construction

For a prime power m there is a perfect difference set of size $m + 1$ [3], with all elements between 0 and $m(m + 1)$. Thus, we know there is a perfect difference set S of size 3, such that $S \subseteq \mathbb{Z}_7$ which is what we need for the following protocol construction.

Let S be a perfect difference set of size 3 and S' a 3-*set* such that $S' \subseteq D - S$. Let L and L' be defined as follows:

$$L = \{x + S \mid x \in \mathbb{Z}_7\} \tag{1}$$

$$L' = \{x + S' \mid x \in \mathbb{Z}_7\} \tag{2}$$

Then, the protocol $\chi_S : \mathscr{P}_3(D) \to \mathbb{Z}_2$ is defined by,

$$\chi_S(0)^{-1} = L \cup L',$$
$$\chi_S(1)^{-1} = \mathscr{P}_3(D) - \chi_S(0)^{-1}.$$

3.2 Minimally informative protocol verification

The proof of the following lemma is similar to the one presented in [1, Theorem 3] for verifying that their announcement construction is informative.

Lemma 1. *Let S be a perfect difference set of size $m+1$ and $v = m(m+1)+1$, then for any two distinct elements $l_1, l_2 \in \{x+S \mid x \in \mathbb{Z}_v\}$, it holds that $|l_1 \cap l_2| = 1$.*

Proof. Let l_1 be $x + S$ and l_2 be $y + S$ with $x \neq y$. Assume for contradiction that $|l_1 \cap l_2| \neq 1$, then $|l_1 \cap l_2| = 0$ or $|l_1 \cap l_2| > 1$.

If it is the case that $|l_1 \cap l_2| = 0$, as S is a perfect difference set there are two elements $s_1, s_2 \in S$ such that $s_1 - s_2 = x - y \mod v$, then $y + s_1 = x + s_2 \mod v$, that is, an element from l_1 equal to one from l_2, being a contradiction to $|l_1 \cap l_2| = 0$.

In the other case, any element in the intersection of l_1 and l_2 is equal to both $x + s_1$ and $y + s_2$, module v, for some $s_1, s_2 \in S$. Then $x - y = s_2 - s_1 \mod v$ and, as S is a perfect difference set, this uniquely define the pair s_1, s_2 so there is no more than one element in the intersection of l_1 and l_2, which contradicts $|l_1 \cap l_2| > 1$.

Lemma 2. *The sets of cliques $K_p(\bar{a})$ of $J(7,3)$, $a \in L$ is a partition of $\mathscr{P}_3(D) - L$.*

Proof. For any $a \in L$, $K_p(\bar{a}) \subseteq \mathscr{P}_3(D) - L$, given that any element $a' \in L$ intersects with a by Lemma 1, it cannot be part of $K_p(\bar{a})$.

Let a and a' be two distinct elements of L, so by Lemma 1 $|a \cap a'| = 1$, then $|\bar{a} \cap \bar{a'}| = 2$. Thus, any 3-*set* in $K_p(\bar{a})$ intersects with any 3-*set* in $K_p(\bar{a'})$ in at most two elements, which means that $K_p(\bar{a})$ and $K_p(\bar{a'})$ are disjoint sets.

Finally, as $|\mathscr{P}_3(D) - L| = \binom{7}{3} - 7 = 28$ and $|K_p(\bar{a})| = 4$ for any 3-*set* a, we have that the union of the seven cliques $K_p(\bar{a})$ of $J(7,3)$ with $a \in L$, is the set $\mathscr{P}_3(D) - L$.

The main result of this section is stated in the following Theorem:

Theorem 2. *The protocol χ_S is minimally informative for* $(3,3,1)$.

Proof. By Lemma 2, for any $b \in \mathscr{P}_3(D)$ we have two cases, namely $b \in L$ or $b \in K_p(\bar{a})$ for some $a \in L$.

Suppose $b \in L$, then there is $x \in \mathbb{Z}_7$, such that $b = x + S$. Therefore $x + S' \in K_p(\bar{b})$, otherwise if $x + S$ and $x + S'$ were to have common elements, it would mean that S and S' are not disjoint. Thus, as $|L| = |L'|$, for any $b \in L$ there is exactly one element $a \in L'$ such that $a \in K_p(\bar{b})$, given that all the cliques $K_p(\bar{b})$, $b \in L$ are disjoints by Lemma 2. Finally, let $a \in K_p(\bar{b})$ be $x + S'$ and a' be any element in $K_p(\bar{b}) - a$, then $\chi_S(a) = 0$ and $\chi_S(a') = 1$.

Now suppose $b \in K_p(\bar{a})$ for some $a \in L$. Then $a \in K_p(\bar{b})$ and $\chi_S(a) = 0$. Let a' be any element in $K_p(\bar{b}) - \{a\}$, then $dist(a, a') = 1$, i.e. $|a \cap a'| = 2$ and therefore $a' \notin L$, otherwise it would contradict Lemma 1. Now let a_1, a_2, a_3 be the three elements in $K_p(\bar{b}) - \{a\}$, and assume for contradiction that $\chi_S(a_1) = \chi_S(a_2) = \chi_S(a_3) = 0$, i.e. $a_1, a_2, a_3 \in L'$. Let $\bar{b} = \{x, y, z, k\}$, then w.l.o.g. $a_1 = \{x, y, z\}$, $a_2 = \{x, y, k\}$ and $a_3 = \{x, k, z\}$. Thus, there are only three different ways in which the elements of these sets could be obtained according to (2), from $S' = \{s'_1, s'_2, s'_3\}$ and different $i, j, l \in \mathbb{Z}_7$:

$$\begin{array}{cccc|cccc|cccc}
 & s'_1 & s'_2 & s'_3 & & s'_1 & s'_2 & s'_3 & & s'_1 & s'_2 & s'_3 \\
+i & x & y & z & +i & x & y & z & +i & x & y & z \\
+j & y & x & k & +j & k & x & y & +j & y & k & x \\
+l & \text{-} & \text{-} & \text{-} & +l & z & k & x & +l & z & x & k
\end{array}$$

Thus, for any of the previous scenarios to hold we would need respectively that,

$$x - y \equiv y - x \mod 7$$
$$x - z \equiv z - x \mod 7$$
$$k - x \equiv x - k \mod 7$$

Given that 7 is prime and x, y, z, k are all different module 7, it's a simple consequence of Fermat's little theorem that any of the previous statements is impossible. Then, we have arrived to a contradiction, and therefore the theorem follows.

Regarding the final argument from the previous proof, the reader can verify that we can always arrive to a contradiction of the form $x - y \equiv y - x \mod 7$, for different $x, y \in \mathbb{Z}_7$ or the more trivial $x + i \equiv x + j \mod 7$, for different $x, i, j \in \mathbb{Z}_7$ (which we discarded from the beginning in the proof).

4 Mechanic verification using Coq

As we previously remarked, the main goal of this project is to provide a formal verification of Theorem 2 from the previous section, using the Coq proof assistant. To this end, we tried to translate, in the most similar way possible, the proof of the theorem presented previously to the Coq scripts. The result of this effort is distributed throughout three Coq files. All definitions in the scripts, as well as most properties and lemmas, are commented in detail (in the corresponding files) using the terminology and notation previously introduced.

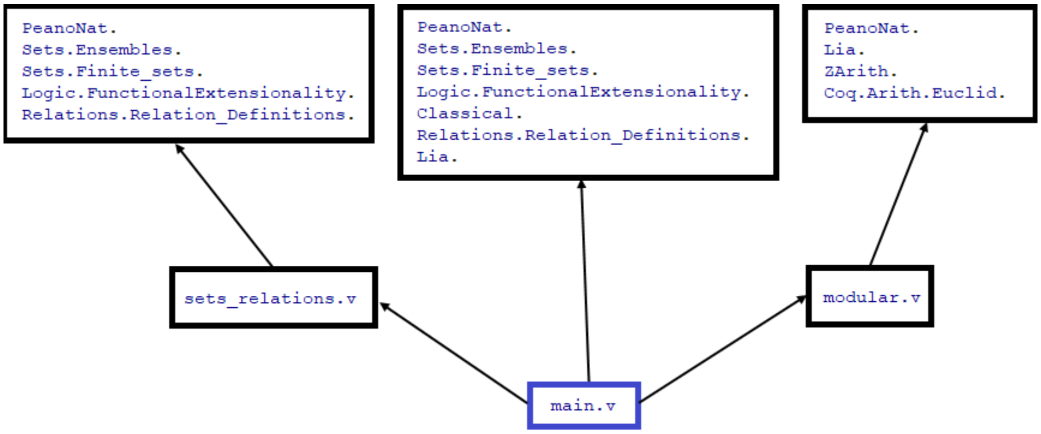
4.1 Scripts organization

We used three script files, which are described in the following:

- `modular.v`
  Includes all Number Theory properties required, in particular, related to modular arithmetic.
- `sets_relations.v`
  Includes all required definitions and general properties about sets and binary relations. Also, it is included a definition of Combinations, i.e. $\binom{n}{k}$, in terms of the criminality of the set consisting of all k-sets included in a set of n elements.
- `main.v`
  This is the main script of the project. Includes all definitions and properties about the problem representation, some of which are compatibility axioms about alternative representation choices. Additionally, it includes all problem-specific definitions, including the minimally informative definition according to Theorem 1, instantiated for two-message protocols. It also provides the definition of the two-message protocol χ_S, from the sets S and S', by the name of **MyProtocol**. Finally, this file contains all properties, lemmas and the main theorem from the previous section. The theorem statement can be found in the final part of the script and it is worth noting that it may take various seconds for Coq to mechanically check the whole proof.

4.2 Scripts dependencies diagram

The following is the dependencies diagram for the project scripts, displaying all the required libraries. This is, for executing the main script, the other two must be previously compiled.



4.3 Fundamental definitions (script listing)

The fallowing are some of the most important problem-specific definitions:

```
(****************************************************************)
Definition setCards:= nat -> Prop. (* Tipo conjunto de cartas   *)
(****************************************************************)

(****************************************************************)
(* Deck D: Conjunto de n cartas de la 0 a la n-1                *)
(****************************************************************)
Definition Deck (n: nat): setCards:=
  fun x => x < n.

(****************************************************************)
(* Predicado para decidir si un número es una carta para |D|=7  *)
(****************************************************************)
(* Definition Card (x: nat): Prop:= Deck 7 x.                   *)

(****************************************************************)
(* Definición del tipo Terna o Tripleta                         *)
(****************************************************************)
Inductive Triple : Type :=
  | triple (x y z: nat).

(****************************************************************)
(* Predicado para decidir si n pertenece a la Terna H           *)
(****************************************************************)
Definition inTriple (n : nat) (H: Triple): Prop :=
  match H with
  | triple x y z => n = x \/ n = y \/ n = z
  end.

(****************************************************************)
(* Conjunto de tres elementos (no necesariamente distintos), es *)
(* una definicion alternativa a la de Triple de la biblioteca   *)
(* Ensembles, pero se define a partir de un elemento de tipo    *)
(* Terna, en lugar de a partir de tres elementos naturales      *)
(****************************************************************)
Definition MyTriple (H : Triple): setCards:=
  fun x => inTriple x H.

(****************************************************************)
(* Predicado para decidir si una Terna es una 3-hand            *)
(****************************************************************)
Definition HandT (H: Triple): Prop:=
  Included (nat) (MyTriple H) (Deck 7) /\
  cardinal (nat) (MyTriple H) 3.

(****************************************************************)
(* Propiedad de corrección del tipo Terna (Triple)              *)
(****************************************************************)
(* Equivalencia entre la definicion propia del conjunto          *)
(* MyTriple y la definición del conjunto (Ensemble) Triple      *)
(* de la biblioteca Ensembles, para cualquier terna x y z.      *)
(* Esta propiedad confirma que para cualquier (triple x y z),   *)
(* i.e., dato de tipo Terna (Triple), todas las operaciones de  *)
(* conjuntos definidas en la biblioteca Ensembles pueden        *)
(* aplicarse de manera compatible sobre el conjunto definido    *)
(* por MyTriple (triple x y z).                                 *)
(****************************************************************)
Lemma myTripleEqualTriple: forall (x y z: nat),
  MyTriple (triple x y z) = (Ensembles.Triple nat x y z).
Proof.
  intros. apply Extensionality_Ensembles. unfoldAll. split.
  { unfold MyTriple. unfold inTriple. intros.
    destruct H. { rewrite H. apply Triple_l. }
    destruct H. { rewrite H. apply Triple_m. }
                { rewrite H. apply Triple_r. } }
  { unfold MyTriple. unfold inTriple. intros.
    destruct H. { left. reflexivity. }
                { right. left.  reflexivity. }
                { right. right. reflexivity. } }
Qed.
```

```
(****************************************************************)
(*        AXIOMAS DE LA REPRESENTACION  DEL PROBLEMA            *)
(****************************************************************)
(* El lema anterior "myTripleEqualTriple" justifica los axiomas *)
(* siguientes:                                                  *)
(****************************************************************)

(****************************************************************)
(*      Axioma de Equivalencia entre Ternas (Triple)            *)
(****************************************************************)
(* Esto es, dos ternas son iguales si y solo si los conjuntos   *)
(* que definen sus elementos son iguales.                       *)
(* Este axioma permite el empleo las definiciones de la biblio- *)
(* teca Ensembles y en particular el axioma de Extensionality   *)
(* para comparar Ternas (Triples) u operar sobre ternas iguales *)
(****************************************************************)
Axiom Triple_eqv_MyTriple: forall (H1 H2: Triple),
  MyTriple (H1) = MyTriple (H2)  -> H1 = H2.

(****************************************************************)
(*              Axioma de Conteo para Ternas                    *)
(****************************************************************)
(* En cualquier conjunto de conjuntos de naturales, la cantidad *)
(* de elementos (conjuntos) de tamaño (cardinalidad) 3 es igual *)
(* a la cantidad de Ternas de 3 elementos diferentes que perte- *)
(* necen al conjunto.                                           *)
(* Esto se debe a la equivalencia entre el tipo Terna (Triple) y*)
(* los conjuntos de naturales (Ensemble nat) con igual cantidad *)
(* de elementos.                                                *)
(****************************************************************)
Axiom countingTriple: forall (E: (Ensemble nat) -> Prop)(n : nat),
  cardinal (Triple)       ( fun T => E (MyTriple T)  /\
                            cardinal nat (MyTriple T) 3 ) n    <->
  cardinal (Ensemble nat) ( fun T => E T /\ cardinal nat T  3 ) n.

(****************************************************************)
(*       OTRAS DEFINICIONES ESPECIFICAS DEL PROBLEMA            *)
(****************************************************************)

(****************************************************************)
(* \bar{b}: Conjunto complemento de una Terna b, respecto       *)
(* al universo de cartas del mazo de 7 cartas (Deck 7)          *)
(****************************************************************)
Definition b_barra (b: Triple): setCards:=
  Setminus nat (Deck 7) (MyTriple b).

(****************************************************************)
(* Predicado para decidir si la Terna a es una mano y pertenece *)
(* al clique de la Terna b, i.e. a Kp(\bar{b}), o sea, si a es  *)
(* una mano que es subconjunto del complmento de b, i.e. \bar{b}*)
(****************************************************************)
Definition disjoint_hands (b a : Triple): Prop :=
  Included nat (MyTriple a) (b_barra (b)) /\
  cardinal nat ((MyTriple a)) 3.

(****************************************************************)
Definition TripleSet:= Triple -> Prop. (*Tipo conjunto de Ternas*)
(****************************************************************)

(****************************************************************)
(* Kp(\bar{b}): Conjunto de Ternas en el clique de la Terna b   *)
(****************************************************************)
Definition clique_b_barra (b : Triple): TripleSet :=
  fun a => disjoint_hands b a.

(****************************************************************)
(* b inCliqueR a: Relacion de pertenencia de a al clique de b.  *)
(****************************************************************)
Definition inCliqueRelation: relation Triple:=
  fun b a => (clique_b_barra b) a.

(****************************************************************)
(* Devuelve la Terna (Triple) x + T, T: Triple                  *)
(****************************************************************)
Definition xPlus7TripleT (x: nat) (T: Triple): Triple:=
  match T with
  | triple t1 t2 t3 => triple ((x + t1) mod 7)
                              ((x + t2) mod 7)
                              ((x + t3) mod 7)
  end.

(****************************************************************)
(* Predicado para decidir si una Terna pertenece al conjunto    *)
(* {x + T| x <= 7, T: Triple}. Este predicado usa como auxiliar *)
(* al predicado xPlus7TripleT                                   *)
(****************************************************************)
Definition inXplusTripleSetT (T H : Triple): Prop:=
  exists x: nat, x < 7 /\ H = xPlus7TripleT (x) (T).
```

```
(****************************************************************)
(* {x + T| x <= 7, T: Triple}: Conjunto de Ternas de forma x + T*)
(****************************************************************)
Definition L_or_L'_isXplusTripleSet (T: Triple): TripleSet:=
  fun x => inXplusTripleSetT T x.

(****************************************************************)
(* Para el PROBLEMA DE LAS CARTAS RUSAS:                        *)
(****************************************************************)
(* Un protocolo de dos mensajes o anuncios (0,1), es el conjunto*)
(* de ternas (que son manos) que conforman uno de los mensajes  *)
(* (el mensaje 1 en este caso).                                 *)
(****************************************************************)
Definition Two_msg_protocol:= TripleSet.

(****************************************************************)
(* Un protocolo de dos mensajes para el Problema de las Cartas  *)
(* Rusas es MINIMAMENTE INFORMATIVO si, para cualquier terna b, *)
(* que sea una mano, existen otras ternas a y a' que sean manos *)
(* disjuntas respecto a b, tales que P(a) != P(a'), o           *)
(* (equivalenetementemanos en el caso de un protocolo de dos    *)
(* mensajes) P(a) /\ ~ P(a').                                   *)
(****************************************************************)
Definition TwoMsgMinmlyInfrPrtcl (P: Two_msg_protocol): Prop :=
  forall b: Triple, HandT b ->
  (exists a : Triple, In Triple (clique_b_barra b) a  /\  P a ) /\
  (exists a': Triple, In Triple (clique_b_barra b) a' /\ ~ P a').

(****************************************************************)
(* Conjunto de Singer (Singer set) o de diferencias perfectas   *)
(* (perfect difference set) de 3 elementos (y consecuentemente  *)
(* módulo 7).                                                   *)
(****************************************************************)
Definition SingerTripleModN (T : Triple) (n: nat): Prop :=
  HandT T /\
  forall x:nat, x < n -> exists s1 s2: nat, inTriple s1 T /\
                                            inTriple s2 T /\
                                         (n + s1 - s2) mod n = x.

(****************************************************************)
(* Nuestra propuesta de construcción de un protocolo de dos     *)
(* mensajes para el Problema de las Cartas Rusas a partir de dos*)
(* conjuntos de tres elementos cada uno.                        *)
(****************************************************************)
Inductive MyProtocol (T T' : Triple): Two_msg_protocol:=
 (* introL: Define el conjunto de ternas L  *)
  | introL : forall H : Triple, HandT H /\
                                inXplusTripleSetT T H  ->
                                (MyProtocol T T' (H))
 (* introL: Define el conjunto de ternas L' *)
  | introL': forall H : Triple, HandT H /\
                                inXplusTripleSetT T' H ->
                                (MyProtocol T T' (H)).
```

4.4 Fundamental theorem statement (script listing)

```
(****************************************************************)
(*                     TEOREMA PRINCIPAL                        *)
(****************************************************************)
(* La demostración del siguiente teorema es el objetivo general *)
(* del presente trabajo:                                        *)
(****************************************************************)
Theorem myProtocolIsMinmlyInfr: forall T T': Triple,
  SingerTripleModN T 7 /\ disjoint_hands T T'
  -> TwoMsgMinmlyInfrPrtcl (MyProtocol T T').
```

5 Conclusions

As it was the main goal of this project, we provided a formal and mechanically verified proof of the minimally informative property of the protocol χ_S for the Classic Russian Cards problem. Most of the auxiliary properties and lemmas were also mechanically checked in Coq and can be found in the project scripts. Among the auxiliary lemmas which were not verified in Coq, the most important ones were proved in the present document.

References

1. Albert, M.H., Aldred, R.E.L., Atkinson, M.D., van Ditmarsch, H., Handley, C.C.: Safe communication for card players by combinatorial designs for two-step protocols. Australas. J Comb. **33**, 33–46 (2005)
2. Rajsbaum, S.: A distributed computing perspective of unconditionally secure information transmission in russian cards problems (2020), draft version
3. Singer, J.: A theorem in finite projective geometry and some applications to number theory. Transactions of the American Mathematical Society **43**(3), 377–385 (1938)
4. Swanson, C.M., Stinson, D.R.: Combinatorial solutions providing improved security for the generalized russian cards problem. Des. Codes Cryptography **72**(2), 345–367 (Aug 2014). https://doi.org/10.1007/s10623-012-9770-7, https://doi.org/10.1007/s10623-012-9770-7