

# Three Steps

Hans van Ditmarsch and Fernando Soler–Toscano

University of Sevilla, Spain  
{hvd,fsoler}@us.es

**Abstract.** Given is a deal of ten cards over three players, such that two players each get four cards and the remaining player (the ‘eavesdropper’) two cards. We show that there does not exist a protocol of two steps for the four-card players to inform each other safely of their hands of cards, and we then present a protocol of three steps that achieves that goal. We verify the properties of that protocol by combinatorial and, mainly, logical (model checking) means. No such three-step protocol for cards was known. The method can be generalized. This will advance the characterization of card deals for which such exchanges of secrets are possible.

## 1 Knowledge-based protocols for card players

*From a pack of seven known cards two players  $A$  and  $B$  each draw three cards and a third player  $C$  gets the remaining card. How can  $A$  and  $B$  openly (publicly) inform each other about their cards, without the third player learning from any of their cards who holds it?*

This problem is often known as the Russian Cards Problem [16] and goes back to [9]. Such issues in cards cryptography have been investigated in the logical and model checking community [19, 4, 17, 11], including related issues in epistemic puzzles, and also in combinatorics and theoretical computer science [7, 14, 10, 20], including related issues in bit-exchange protocols.

One solution for the riddle is as follows. Suppose that the actual deal of cards is that agent  $A$  has  $\{0, 1, 2\}$ ,  $B$  has  $\{3, 4, 5\}$  and  $C$  has  $\{6\}$ .

- $A$  says: My hand is one of 012, 046, 136, 145, 235.
- $B$  says:  $C$ 's card is 6.

After this, it is common knowledge to the three agents that  $A$  knows the hand of  $B$ , that  $B$  knows the hand of  $A$ , and that  $C$  is ignorant of the ownership of any card not held by herself.

We can also see these two sequences as the execution of a knowledge-based protocol. Given  $A$ 's hand of cards, there is a (non-deterministic) way to produce her announcement, and given her announcement,  $B$  always responds by announcing  $C$ 's card. The protocol is knowledge-based, because the agents initially only know their own hand of cards, and have public knowledge of the deck of cards and how many cards each agent has drawn from the pack. We can imagine agent  $A$  producing her announcement as follows from her initial knowledge (so, indeed, it is a function of her local state to her action):

Let my hand of cards be  $ijk$  and let the remaining cards be  $lmno$ . Choose one from  $ijk$ , say, w.l.o.g.  $i$ , and choose two from  $lmno$ , say  $lm$ . Three of the hands in my announcement are  $ijk$ ,  $ilm$ , and  $ino$ . From  $lm$  choose one, say  $l$ , and from  $no$  choose one, say  $n$ . The two remaining hands are  $jln$  and  $kmo$ . Announce these five hands in random order.

This first step of such a protocol extends to a trivial second step wherein  $B$  announces  $C$ 's card. It can be viewed as an *unconditionally secure* protocol, as  $C$  cannot learn any of the cards of  $A$  and  $B$ , no matter her computational resources. The security is therefore not conditional on the high complexity of some computation.

The Russian Card problem can be seen as the  $(3, 3, 1)$  instance of the general  $(a, b, c)$  case, where  $A, B, C$  hold  $a, b, c$  cards, respectively. When can  $A$  and  $B$  communicate their hands of cards to each other, and when not? And how many steps are needed in a protocol realizing this?

Some results are found in [7, 16, 1, 2]. An announcement by a player in a protocol is always equivalent to the announcement of a set of alternative hands including the actual hand. There are two-step protocols for a wide range of cases, for example, merely to mention some instantiations of more general patterns, for  $(4, 2, 1)$ , for  $(22, 22, 1)$ , and for  $(10, 18, 3)$  (an instantiation of [2, Theorem3] for a deck of  $p^2 + p + 1$  cards, for prime  $p = 5$ ).

The protocol above is a two-step protocol:  $B$ 's answer depends on hearing  $A$ 's announcement; he does not know what  $C$ 's card is initially. Surprisingly, there are 'one-step protocols' of a kind: a different way to produce the sequence 012, 046, 136, 145, 235 is to observe that the sum of the cards in every triple is 3 modulo 7. So  $A$  could execute the protocol wherein she announces the sum of her cards, modulo 7. Agent  $B$  could also have done that, and even *at the same time* as  $A$  (alternatively to announcing  $C$ 's card, *after*  $A$ 's announcement) [2]. We require that  $A$  and  $B$  make their announcement in some order and therefore keep calling that a two-step protocol as well. In [2] we have proved that for  $(n, n, 1)$  with  $n > 2$  there is a two-step protocol, with  $A$  announcing the sum of his cards modulo  $2n + 1$  and  $B$  announcing  $C$ 's card. Recently we have generalized this result to provide two-step protocols for  $(n, m, 1)$  with  $n, m > 2$ .

There does not always exist a protocol for  $(a, b, c)$ , for two players to exchange their hands of cards. For example, this is not possible when each player holds one card [7]. If there exists a protocol for  $(a, b, c)$ , is it always possible to make the exchange in two steps? Other works [7, 14, 10] give protocols of various length, three, four, or more steps—but without a proof of minimality. The answer to that question is: no. This contribution presents a three-step protocol for  $(4, 4, 2)$  and a proof that no shorter protocol exists.

## 2 Logical preliminaries

Protocols for card deals consist of public announcements. Public announcement logic [13, 3] is an extension of multi-agent epistemic logic. Its language, structures, and semantics are as follows.

Given are a finite set of agents  $\mathcal{A}$  and a countable set of propositional variables  $P$ . The *language of public announcement logic* is inductively defined as

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid K_A\varphi \mid C_{\mathcal{A}'}\varphi \mid [!\varphi]\psi$$

where  $p \in P$ ,  $A \in \mathcal{A}$ , and  $\mathcal{A}' \subseteq \mathcal{A}$ . For  $K_A\varphi$ , read ‘agent  $A$  knows formula  $\varphi$ ’. For  $C_{\mathcal{A}'}\varphi$ , read ‘group of agents  $\mathcal{A}'$  commonly know formula  $\varphi$ ’. For  $[!\varphi]\psi$ , read ‘after truthful public announcement of  $\varphi$ , formula  $\psi$  (is true)’.

An *epistemic model*  $M = \langle S, \sim, V \rangle$  consists of a *domain*  $S$  of *states* (or ‘worlds’), an *accessibility function*  $R : \mathcal{A} \rightarrow \mathcal{P}(S \times S)$ , where each  $\sim_A$  is an equivalence relation, and a *valuation*  $V : P \rightarrow \mathcal{P}(S)$ . The accessibility relation  $\sim_{\mathcal{A}'}$  is defined as  $(\bigcup_{A \in \mathcal{A}'} \sim_A)^*$ . For  $s \in S$ ,  $(M, s)$  is an *epistemic state*, also known as a pointed Kripke model.

Assume an epistemic model  $M = \langle S, \sim, V \rangle$ .

$$\begin{aligned} M, s \models p & \quad \text{iff } s \in V(p) \\ M, s \models \neg\varphi & \quad \text{iff } M, s \not\models \varphi \\ M, s \models \varphi \wedge \psi & \quad \text{iff } M, s \models \varphi \text{ and } M, s \models \psi \\ M, s \models K_A\varphi & \quad \text{iff for all } t \in S : s \sim_A t \text{ implies } M, t \models \varphi \\ M, s \models C_{\mathcal{A}'}\varphi & \quad \text{iff for all } t \in S : s \sim_{\mathcal{A}'} t \text{ implies } M, t \models \varphi \\ M, s \models [!\varphi]\psi & \quad \text{iff } M, s \models \varphi \text{ implies } M|_{\varphi}, s \models \psi \end{aligned}$$

where the model restriction  $M|_{\varphi} = \langle S', \sim', V' \rangle$  is defined as  $S' = \{s' \in S \text{ such that } M, s' \models \varphi\}$ ,  $\sim'_A = \sim_A \cap (S' \times S')$  and  $V'(p) = V(p) \cap S'$ . Complete proof systems for this logic are presented in [13, 3].

Our definition of a knowledge-based protocol consisting of announcements is a special case of the *knowledge-based program* à la Fagin *et al.* [6]. Instead of each agent choosing an action conditional on her knowledge, each agent chooses an announcement conditional on her knowledge. Although as a concept it is fairly obvious, it has not been singled out in the literature so far.

**Definition 1 (Knowledge-based protocol).** *A knowledge-based protocol for public announcement logic is a finite sequence of instructions determining sequences of announcements. Each agent  $A$  chooses an announcement  $K_A\psi$  conditional on that agent’s knowledge  $K_A\varphi$ . The chosen announcements are uttered simultaneously, i.e., an  $|\mathcal{A}|$ -tuple of preconditions of form  $K_A\varphi_A$  determines an announcement  $!\bigwedge_{A \in \mathcal{A}} K_A\psi_A$ . The protocol is assumed common knowledge between all agents.*

In this work, we further assume that only one agent makes an announcement at the same time, that announcements are alternating between agents  $A$  and  $B$ , and that agent  $A$  starts the protocol. Given an initial epistemic model  $M$ , a *protocol execution* is a sequence of announcements determined by a knowledge-based protocol effecting successive model changes. The set of all such execution sequences is the extensional notion of protocols as in [12].

The protocol execution above consists of an announcement by  $A$  followed by announcement by  $B$  that can be modelled in public announcement logic as

$$\mathbf{a\_announce} = K_A(012_A \vee 046_A \vee 136_A \vee 145_A \vee 235_A) \quad (1)$$

$$\mathbf{b\_announce} = K_B6_C \quad (2)$$

where  $n_i$ , for  $0 \leq n \leq 6$  represents that agent  $i$  has the card  $n$ , and where  $nmk_i$  is an abbreviation for  $n_i \wedge m_i \wedge k_i$ . Note that  $A$ 's announcement is indeed a function of  $A$ 's knowledge, because  $K_A 012_A$  entails  $K_A(012_A \vee 046_A \vee 136_A \vee 145_A \vee 235_A)$ , and that also  $B$ 's announcement is a function of his knowledge.

We need some more card deal terminology in the continuation. Given players or agents  $A$ ,  $B$  and  $C$  and  $a$ ,  $b$  and  $c$  cards, with  $a + b + c = d$ . The cards are shuffled and dealt to the agents.  $A$  gets  $a$  cards, etc. This is a *card deal of size*  $(a, b, c)$ . A condition holds for a protocol if it holds after every execution sequence of the protocol. A protocol is safe if it preserves common knowledge of ignorance of  $C$ . A protocol is  $A$ -informative if after termination it is common knowledge that  $A$  knows the card deal. A protocol is  $B$ -informative if after termination it is common knowledge that  $B$  knows the card deal. These three conditions are:

$$\mathbf{b\_knows\_as} = \bigwedge_{n=0}^{d-1} (K_B n_A \vee K_B \neg n_A) \quad (3)$$

$$\mathbf{a\_knows\_bs} = \bigwedge_{n=0}^{d-1} (K_A n_B \vee K_A \neg n_B) \quad (4)$$

$$\mathbf{c\_ignorant} = \bigwedge_{n=0}^{d-1} (\neg K_C n_A \wedge \neg K_C n_B) \quad (5)$$

The protocol for  $(3, 3, 1)$  in the introductory section is safe,  $A$ -informative, and  $B$ -informative because after  $A$ 's public announcement of (1) it is true that

$$C_{ABC}(\mathbf{b\_knows\_as} \wedge \mathbf{c\_ignorant}) \quad (6)$$

and after  $B$ 's public announcement of (2) it is true that

$$C_{ABC}(\mathbf{b\_knows\_as} \wedge \mathbf{a\_knows\_bs} \wedge \mathbf{c\_ignorant}) \quad (7)$$

where we write  $C_{ABC}$  for  $C_{\{A,B,C\}}$ . Note that  $C_{ABC}(\mathbf{b\_knows\_as} \wedge \mathbf{c\_ignorant})$  holds whenever one of the hands in (1) is the actual hand. It is therefore sufficient to check that  $(\mathbf{b\_knows\_as} \wedge \mathbf{c\_ignorant})$  is a model validity after the announcement.

The model checker DEMO (for ‘a Demo of Epistemic MOdelling’) has been developed by van Eijck [18]. It is written in Haskell. DEMO allows the representation of epistemic models, performing updates with epistemic actions such as public announcements, and evaluating epistemic formulas in epistemic models or epistemic models resulting from update execution. The syntax of dynamic epistemic logic in DEMO is fairly similar to the standard language defined above, e.g.,  $K_B(8_B \wedge 9_B)$  is represented by  $K \mathbf{b} (\mathbf{Conj}[\mathbf{Prop}(\mathbf{R} \ 8), \mathbf{Prop}(\mathbf{R} \ 9)])$  (DEMO syntax only allows atoms called  $P$ ,  $Q$ , or  $R$ , with number labels) and  $C_{ABC} \mathbf{c\_ignorant}$  by  $(\mathbf{CK}[\mathbf{a}, \mathbf{b}, \mathbf{c}] \ \mathbf{c\_ignorant})$ . Instead of the single-pointed epistemic states  $(M, s)$  defined above, DEMO allows multi-pointed epistemic

states: the set of designated points of the model stands for the current uncertainty about the actual state. If all points of the model are designated, checking the truth of a formula in that model means checking if it is a model validity (we will use this in the continuation). In DEMO, the names of states must be natural numbers, starting with 0. Let the model  $M$  be `rus` and the actual state be 0, then the verification that  $M, s \models C_{ABC}c\_ignorant$  is represented in DEMO by

```
Main> isTrue (rus 0) (CK [a,b,c] c_ignorant)
True
```

The model resulting from updating (`rus 0`) with a public announcement of the formula `b_knows_as` is represented by (`upd (rus 0) (public b_knows_as)`). A multi-pointed model has a list of states instead of a single state, as in (`rus [0..224]`). The DEMO script employed in this paper is similar to the one in [17] that treats the (3, 3, 1) case (but the protocol is very different).

### 3 There is no two-step protocol for (4, 4, 2)

We show that there does not exist a protocol in two steps, consisting of one announcement by  $A$  and one announcement by  $B$ , for (4, 4, 2). We prove this using the upper and lower bounds for the number of hands in a safe announcement as in [1], namely by showing that the minimum number of hands is greater than the maximum number of hands. We recall that any announcement whatsoever must be equivalent to an announcement of alternative hands (see [16] — this is for the simple reason that the denotation of an announcement known to be true by the agent making it, is a union of equivalence classes for that agent; and an equivalence class is characterized by a hand of cards for that agent).

Consider an announcement consisting of alternative hands and containing the actual hand. An announcement is *more informative* if it consists of *fewer* hands. The most informative announcement consists of announcing the actual hand. The least informative announcement consists of announcing all hands. However, an announcement is *safer* if it consists of *more* hands. Clearly, the safest announcement consists of announcing all hands. But that announcement is uninformative. In [1] lower and upper bounds are given for the number of hands in an announcement. In the following, ‘good’ means ‘safe and  $B$ -informative’.

- [1, Prop.1] The number of hands in a good announcement is at least

$$\frac{(a + b + c)(c + 1)}{a}$$

- [1, Prop.2] The number of hands in a good announcement is at least

$$\frac{(a + b)(a + b + c)}{b(b + c)}$$

- [1, Prop.3] The number of hands in a good announcement is at most

$$\frac{(a+b+c)!(c+1)!}{(b+c)!(c+a+1)!} \left\lfloor \frac{a+c+1}{c+1} \right\rfloor$$

- [1, Prop.4] The number of hands in a good announcement is at most

$$\frac{(a+b+c)!(c+1)!}{a!(b+2c+1)!} \left\lfloor \frac{b+2c+1}{c+1} \right\rfloor$$

The two propositions for lower bounds should read ‘at least the ceiling of’, and those for higher bounds ‘at most the floor of’. We obviously need integers. Which of the two lower bounds is sharper, depends on the card deal  $(a, b, c)$ ; and similarly for the two upper bounds. For  $(4, 4, 2)$ , [1, Prop.2] delivers a lower bound of 4 and [1, Prop.3] a higher bound of 12. That is not problematic. However, the other two propositions prove that:

**Proposition 1.** *There is no two-step protocol for  $(4, 4, 2)$ .*

*Proof.* For a lower bound, apply [1, Prop.1]. For  $(a, b, c) = (4, 4, 2)$ , the number of hands required is at least

$$\left\lceil \frac{(a+b+c)(c+1)}{a} \right\rceil = \left\lceil \frac{10 \cdot 3}{4} \right\rceil = \lceil 7.5 \rceil = 8.$$

For a higher bound, apply [1, Prop.4]. For  $(a, b, c) = (4, 4, 2)$ , the number of hands required is at most

$$\left\lfloor \frac{(a+b+c)!(c+1)!}{a!(b+2c+1)!} \left\lfloor \frac{b+2c+1}{c+1} \right\rfloor \right\rfloor = \left\lfloor \frac{10!3!}{4!9!} \left\lfloor \frac{9}{3} \right\rfloor \right\rfloor = \left\lfloor \frac{10}{4} \cdot 3 \right\rfloor = \lfloor 7.5 \rfloor = 7.$$

As a safe and  $B$ -informative announcement for  $(4, 4, 2)$  should contain at least eight and at most seven hands, it cannot exist.

## 4 A safe and informative announcement

The first step in the three-step protocol that we propose employs a *block design* [15]. We recall the definition of a  $t$ -design with parameters  $(v, k, \lambda)$  (also called a  $t$ - $(v, k, \lambda)$  design). Relative to a  $v$ -set  $X$  (i.e., a set with  $|X| = v$  elements) this is a collection of  $k$ -subsets of  $X$  called *blocks* with the property that every  $t$ -subset of  $X$  is contained in exactly  $\lambda$  blocks.

Announcements in protocols for card deals may be, but do not have to be, block designs. Given a deal with parameters  $(a, b, c)$ , we take  $v = d = a + b + c$ , and  $k = a$  (the blocks are  $A$ -hands). For  $(3, 3, 1)$ , the announcement 012 034 056 135 246 is *not* a design. The announcement 012 034 056 135 146 236 245 is a 2- $(7, 3, 1)$  design—each number pair occurs once. And it is a 1- $(7, 3, 3)$  design—each of the numbers 0 to 6 occurs three times.

Now for the card deal of size  $(4, 4, 2)$ . Consider the following  $2$ – $(10, 4, 2)$  design  $\mathcal{L}_A$  listed in the database Design DB [5].

0123 0145 0267 0389 0468 0579 1289 1367 1479 1568 2345 2478 2569 3469 3578

The set  $\mathcal{L}_A$  is a  $2$ – $(10, 4, 2)$  design, because each pair occurs twice. It is also a  $1$ – $(10, 4, 6)$  design: each card occurs 6 times; but it is not a  $3$ –design, e.g., triple 012 is in, but triple 018 is out. The block size 4 corresponds to the number of  $A$ 's cards. Would this make a suitable announcement for  $A$ ? Not every quadruple occurs in the design. What if  $A$ 's hand of cards is 0124? Consider the following protocol.

**Definition 2 (Protocol OneStep for  $A$ 's announcement).** *Let  $\{i, j, k, l\}$  be  $A$ 's hand of cards. Select a member  $\{m, n, o, p\}$  from  $\mathcal{L}_A$ . (We do not assume that  $i, j, k, l$  or  $m, n, o, p$  are ordered by size, as in 0123.) Select a permutation  $\pi$  of  $0..9$  such that  $\pi(i) = m$ ,  $\pi(j) = n$ ,  $\pi(k) = o$ ,  $\pi(l) = p$ . Let  $\pi(\mathcal{L}_A)$  be the collection induced by permutation  $\pi$ . Player  $A$  announces  $\pi(\mathcal{L}_A)$ . (I.e.,  $A$  says: "My hand of cards is one of  $\pi(\mathcal{L}_A)$ .")*

This is a knowledge-based protocol, because  $A$  knows her hand of cards. If 0123 is  $A$ 's actual hand, the selected element of  $\mathcal{L}_A$  is also 0123, and the selected permutation is the identity, then  $A$ 's announcement corresponds to the following public announcement.

$$K_a(0123_a \vee 0145_a \vee 0267_a \vee 0389_a \vee 0468_a \vee 0579_a \vee 1289_a \vee 1367_a \vee 1479_a \vee 1568_a \vee 2345_a \vee 2478_a \vee 2569_a \vee 3469_a \vee 3578_a) \quad (8)$$

We now show that this is a safe announcement. There are two ways to go about this: a combinatorial proof, and more logical proof, namely by means of dynamic epistemic model checking.

**Proposition 2.** *Protocol OneStep is safe.*

*Proof.* The combinatorial proof uses the two combinatorial safety requirements formulated in [1]. They are:

- **CA2.** For every  $c$ -set  $X$  the members of  $\mathcal{L}$  avoiding  $X$  have empty intersection.
- **CA3.** For every  $c$ -set  $X$  the members of  $\mathcal{L}$  avoiding  $X$  have union consisting of all cards except those of  $X$ .

For  $(4, 4, 2)$ , and  $\mathcal{L} = \mathcal{L}_A$ , CA2 guarantees that, if  $C$  holds the two cards in  $X$ , then after  $A$  announces  $\mathcal{L}_A$ ,  $C$  does not learn one of  $A$ 's cards (if a card occurs in all  $A$ -hands that  $C$  still considers possible, then  $A$  *must* have that card). CA3 guarantees that  $C$  does not learn one of  $B$ 's cards (if a card does not occur in all  $A$ -hands that  $C$  still considers possible, and  $C$  also does not have that card, then  $B$  *must* have that card).

Let  $X = \{0, 1\}$  ( $X = 01$ ). From the fifteen hands in  $\mathcal{L}_A$ , there are six containing 0 and six containing 1 (it's a  $1$ –design with  $\lambda = 6$ )—of which two contain

the pair 01 (it's a 2-design with  $\lambda = 2$ ): ten hands contain 0 or 1. Consider the five remaining hands 2345 2478 2569 3469 3578. Some numbers occur three times (2, 3, 4, and 5) and some twice only (6, 7, 8, and 9). But all numbers occur at least once—so they are contained in the union of these five hands—and at least once not—so they are not in the intersection of these five hands. Therefore, for  $X = 01$  the conditions CA2 and CA3 are satisfied.

This holds not just for 01 but for any  $ij$ . There are always five hands that do not contain  $i$  and  $j$ , because of the design properties. If a card  $k \neq i, j$  were to occur in all those five hands, it would only occur once in the remaining ten (as  $\mathcal{L}_A$  is a 1-design wherein each card occurs 6 times). But there must be two occurrences of the pair  $ik$ , as  $\mathcal{L}_A$  is a 2-design wherein each pair occurs twice. So no card  $k$  occurs in all remaining hands. However, if a card  $k \neq i, j$  were to occur in none of those five hands, it would occur six times in the ten hands containing  $i$  or  $j$ . One can easily see that the maximum number of allowed  $k$ -occurrences is four, otherwise there would be more than two  $ik$  or more than two  $jk$  pairs. So it can also not be the case that  $k$  does not occur in any of the remaining hands.

Alternatively, we can achieve this result by model checking. The requirement is then that after  $\mathcal{L}_A$  it is common knowledge, given the actual deal of cards, that  $C$  is ignorant. This requirement is satisfied if it is a model validity that  $C$  is ignorant. Therefore, the proof is:

```
Main> isTrue (rus [0..224]) c_ignorant
True
```

This succinct line hides the computation behind it. Also, the structure of the epistemic model resulting from  $\mathcal{L}_A$  will serve us in extending Protocol `OneStep` to a full protocol—we did not mention so far what  $B$  learns from  $\mathcal{L}_A$ . The next subsection presents this model, and some statistics on the model checking.

## 5 The model before and after the first announcement

The initial model has  $\binom{10}{4}\binom{6}{4} = 3150$  states/deals. After  $A$ 's announcement of 15 hands it is reduced to  $15 \cdot \binom{6}{4} = 225$  states, that are labeled from 0 to 224. Appendix A shows the equivalence classes for the three agents in this model. For example, it indicates that in the deals 0..14 agent  $A$  has the same hand (another part of the DEMO specification, not shown, nails down which hand).

Figure 1 depicts a part of the epistemic model after  $A$ 's announcement. This is a  $\sim_{BC}$ -equivalence class (we write  $\sim_{BC}$  for  $\sim_{\{B,C\}}$ ). All states in this class can be distinguished by  $A$ . The model consists of 15 such subgraphs (an  $A$ -class also consisting of 15 states, one in each of these subgraphs). In the picture, the solid lines represent accessibility for  $C$  and the dashed lines accessibility for  $B$ . Note that in 12 of the 15 states  $B$  knows the deal, and that in the remaining three ( $\{0, 59, 104\}$ )  $B$  considers three possible hands for  $C$ . The numbers in the points refer to the deals that appear in the equivalence classes of Appendix A. The state named 0 represents the deal 0123|4567|89, i.e., where  $A$ 's hand is  $\{0, 1, 2, 3\}$ ,  $B$ 's hand is  $\{4, 5, 6, 7\}$  and  $C$ 's hand is  $\{8, 9\}$ .

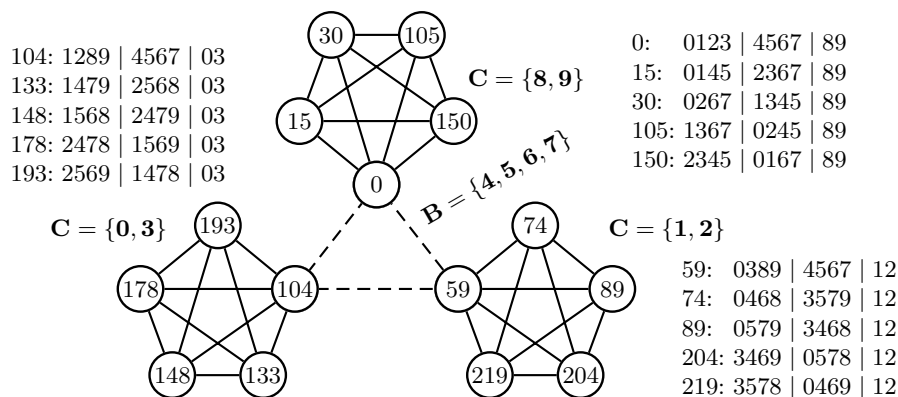


Fig. 1. The epistemic model for  $B$  and  $C$  after  $A$ 's announcement

The verification in epistemic logic of Proposition 2 was represented by

```
Main> isTrue (rus [0..224]) c_ignorant
True
```

and to similar effect we can check

```
Main> isTrue (rus 0) CK [a,b,c] c_ignorant
True
```

The computational cost of this check depends on some inessential but for practical purposes important parameters. Table 1 shows the runtime required to check Proposition 2 with DEMO in a 2.26 GHz processor, running GHCi, version 6.12.3 over Ubuntu Linux 9.10. Two formulas that represent the ignorance of  $C$  about card  $n$  are provided. The first is the one used in (5). The second checks  $C$ 's ignorance only for cards that  $C$  does not hold. Both formulas are equivalent in the model. There are also two ways for checking  $C$ 's ignorance. The first one (central column) is by checking that ignorance of  $C$  is common knowledge at some point in the model (0 in this case). The second one (right) is by checking that  $C$ 's ignorance is a model validity. As the model is connected, both ways are equivalent in the model. The minimum time corresponds to checking that (5) is a model validity. The improvement for the lower, conditional formula for  $C$ 's ignorance is only when checking common knowledge. We have also checked `c_ignorant` (5) in the (initial) model prior to  $A$ 's announcement, and the cost of announcing  $\mathcal{L}_A$  (8) in that initial model. This takes 13.2 seconds.

We have used the Haskell script in Appendix B to demonstrate that the epistemic model after executing Protocol `OneStep` indeed consists of fifteen  $\sim_{BC}$ -connected subgraphs like the one in Figure 1, where all points in one such subgraph are all different for  $A$ . Function `(goodGraph n)` checks that every point  $n$  in the epistemic model after the execution of Protocol `OneStep` belongs to a  $\sim_{BC}$ -graph like that of Figure 1.

<i>C's ignorance about n</i>	$(\text{rus}[0]) \models C_{ABC}c\_ignorant$	$(\text{rus}) \models c\_ignorant$
$\neg K_C n_A \wedge \neg K_C n_B$	629	256
$\neg n_C \rightarrow \neg K_C n_A \wedge \neg K_C n_B$	544	399

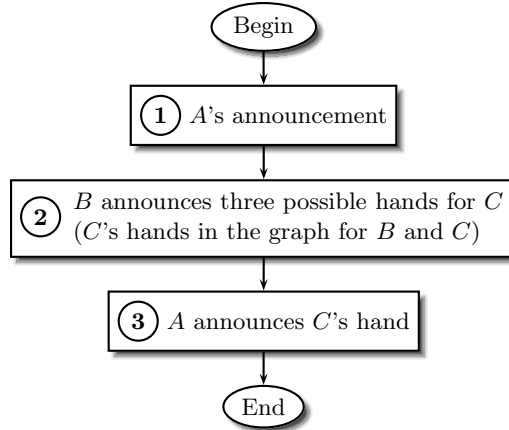
**Table 1.** DEMO's runtime in milliseconds to check Proposition 2 in several ways

```
Main> foldl1 (&&) (map goodGraph [0..224])
True
```

By showing the structure of this model we have demonstrated (or rather corroborated) two aspects of Protocol **OneStep**. Firstly, given an actual hand of  $A$ , the actual announcement is not  $\mathcal{L}_A$  but  $\pi(\mathcal{L}_A)$ . That does not change the model structure, as this merely involves renaming atoms. In fact, we have done this anyway: ‘ $B$  holds 8’ is not an atom  $8_B$  in DEMO, but some  $\mathbf{R} \ 8$  (see the previous section). So this involves only further renaming. Secondly, the point of the structure need not be card deal 0123|4567|89, as here, but can be another deal of cards wherein  $A$  holds another hand of cards in  $\mathcal{L}_A$ . But the model looks the same from any such different perspective. This is as expected: for any two hands  $X$  and  $Y$  of the design, there are (many) permutations  $\pi$  such that  $\pi(X) = Y$  and  $\pi(\mathcal{L}_A) = \mathcal{L}_A$ , in other words, these permutations induce automorphisms of the design. The remaining sections will not refer to design theory phenomena — the further extensions of the protocol are not designs.

## 6 A three step protocol for (4, 4, 2)

We now present a three-step protocol for (4, 4, 2) that extends Protocol **OneStep**.



**Fig. 2.** Protocol ThreeSteps for (4,4,2)

**Definition 3 (Protocol ThreeSteps).**

**Step 1.** *A announces (a permutation of) (8). Call this  $\mathcal{L}_A$ , as before.*

**Step 2.** *B announces that C has one of three hands. These are the three C hands in the  $\sim_{BC}$ -connected part of the model  $M|\mathcal{L}_A$ .*

**Step 3.** *A announces the hand of C.*

**Proposition 3.** *Protocol ThreeSteps is safe.*

*Proof.* The safety of Step 1 is guaranteed by Proposition 2. The effect of Step 2 is to reduce the epistemic model to just one  $\sim_{BC}$ -class like the one depicted in Figure 1. As it contains three complete  $\sim_C$ -classes, the ignorance of C remains common knowledge. Step 3 reduces the model to a single  $\sim_C$ -class. So, again, common knowledge of ignorance is preserved.

**Proposition 4.** *Protocol ThreeSteps is A-informative.*

*Proof.* Agent A knows the deal after Step 2. All  $\sim_A$ -classes are then singletons.

**Proposition 5.** *Protocol ThreeSteps is B-informative.*

*Proof.* Agent B knows the deal after Step 3: a  $\sim_C$ -class consists of five singleton  $\sim_B$ -classes.

## 7 A probabilistic protocol

Consider a 15-state  $\sim_{BC}$ -class, as in Figure 1. It seems appealing for B to announce the card deal in the twelve cases where he knows the card deal, and only to continue the protocol in the three cases where he does not know that. This is not safe (Figure 3), but we can make it safe (Figure 4 and Definition 4).

Look at Figure 3. If B knows the deal after A's announcement at Step 1, B announces C's cards and the protocol finishes in just two steps. It is Step 2' in Figure 3. Otherwise, the protocol continues as in Figure 2 and takes three steps.

After every step in Figure 3 agent C remains ignorant about any card of A and B, but this becomes different when C also knows the protocol. Kerckhoffs's Principle [8] says that the protocol should be common knowledge: the design of a security protocol should be done under the assumption that an intruder knows everything about the protocol except the key. For cards communication this means that we may assume that C knows everything about the protocol except the hands of A and B; i.e., C knows the flowchart in Figure 3. If B announces C's cards in Step 2 he implicitly announces `b_knows_as` and if B announces some alternative hands for C he implicitly announces the negation of that. Fortunately, we have that

```
Main> isTrue (upd (rus [0..224]) (public b_knows_as)) c_ignorant
True
```

This is not trivial (there remain four indistinguishable card deals for C, and that is enough in this case to keep her ignorant) but unfortunately, we also have that

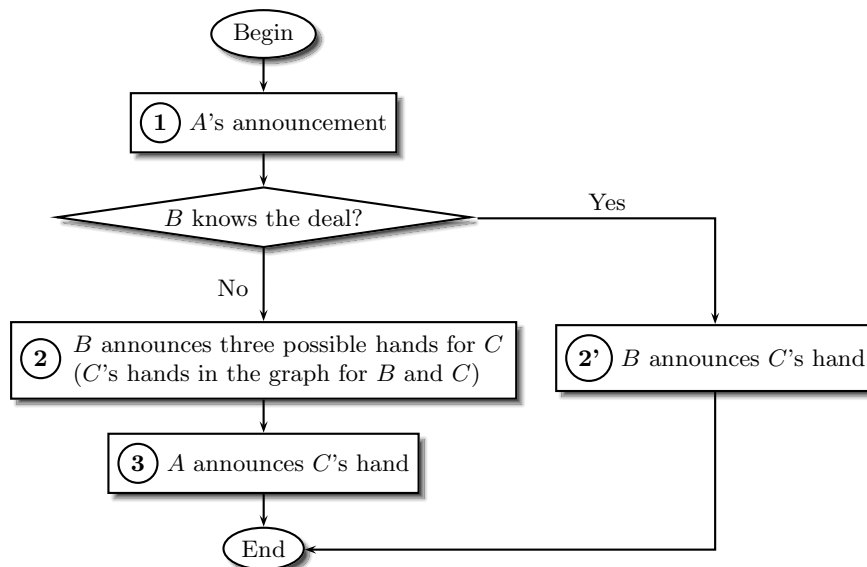


Fig. 3. An unsafe protocol for (4,4,2)

```

Main> isTrue (upd (rus [0..224]) (public (Neg b_knows_as)))
      (Neg c_ignorant)
True
  
```

When  $C$  is informed about  $B$ 's ignorance, she learns that the actual deal is the only one in  $C$ 's pentagon (see Figure 1) where  $B$  does not know the deal:  $A$  gains full knowledge of the card deal.

The protocol in Figure 3 is unsafe. The problem is that  $B$  performs Step 2 *only* if he does not know the card deal. The link is broken in Protocol ProbSteps—see Figure 4.

**Definition 4 (Protocol ProbSteps).**

**Step 1.**  $A$  announces (a permutation of)  $\mathcal{L}_A$ .

**Step 2'.** If  $B$  knows the deal then with probability  $p < 1$   $B$  announces  $C$ 's hand.

**Step 2.** If  $B$  does not know the deal after Step 1 or if  $B$  did not execute Step 2', then  $B$  announces the three  $C$ -hands in the  $\sim_{BC}$ -class.

**Step 3.** If Step 2 was executed,  $A$  announces the hand of  $C$ .

**Proposition 6.** Protocol ProbSteps is safe.

*Proof.* In case that the protocol follows the sequence of Steps 1, 2, 3, it is like Protocol ThreeSteps, so it is safe (Proposition 3). Note that when  $B$  performs Step 2, he is not implicitly announcing his ignorance of  $C$ 's cards, because there is non-zero probability that  $B$  performs Step 2 when he knows the deal. When the protocol consists of Steps 1 and 2' it is also safe, because (as before)

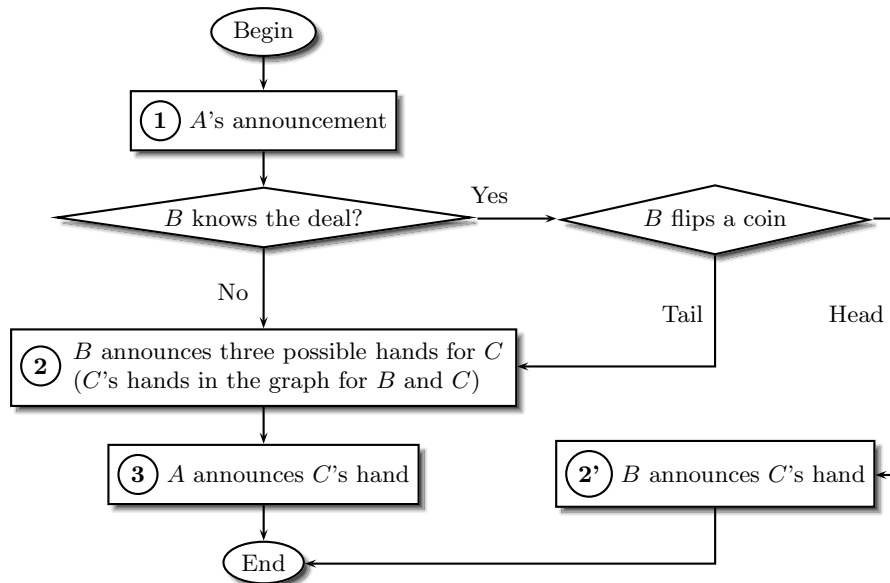


Fig. 4. A probabilistic version of the protocol

```

Main> isTrue (upd (rus [0..224]) (public b_knows_as)) c_ignorant
True
  
```

The following two propositions are obvious.

**Proposition 7.** *Protocol ProbSteps is A-informative.*

**Proposition 8.** *Protocol ProbSteps is B-informative.*

**Proposition 9.** *The average length of Protocol ProbSteps is  $3 - 0.8p$ .*

*Proof.* Given that the permutation of (8) announced by  $A$  at Step 1 is randomly chosen, then on the further assumption that all cards deals are equally likely (random probability distribution), the probability that  $B$  knows the deal after  $A$ 's announcement is 0.8. ( $B$  knows the deal in four out of five states in every  $\sim_C$ -pentagon.) As  $p$  is the probability of then announcing the deal, the protocol has length 2 with probability  $0.8p$ , and it has length 3 with probability  $1 - 0.8p$ . So the average length of the protocol is

$$3(1 - 0.8p) + 2(0.8p) = 3 - 0.8p$$

When  $p$  approaches 1, this probability approaches 2.2. (It cannot be 1, as the protocol is then unsafe again.)

The value of  $p$  in Proposition 9 is common knowledge. Agent  $C$  can apply Bayes' Theorem when  $B$  performs Step 2, to calculate the probability for  $B$  not

knowing the deal:

$$\begin{aligned} P(\neg b\_knows\_all | Step\ 2) &= \frac{P(Step\ 2 | \neg b\_knows\_all) P(\neg b\_knows\_all)}{P(Step\ 2)} = \\ &= \frac{1 \times 0.2}{0.2 + 0.8(1 - p)} = \frac{0.2}{1 - 0.8p} \end{aligned}$$

This is the probability for  $C$  to correctly guess the deal when Step 2 is performed with  $B$  is still ignorant.

It remains safe ‘in principle’ to have a large  $p$  value but that also increases the probability that the eavesdropper  $C$  correctly *guesses* the deal. We can reduce the probability of guessing if we consider a parallel execution of  $n$  instances of the protocol, where the secret is the conjunction of the  $n$  deals. The probability of guessing correctly is then multiplied to the power of  $n$ , and can therefore be reduced as much as we want.

## 8 Conclusion

For a card deal of size  $(4, 4, 2)$  we have shown that there does not exist a protocol of two steps for the four-card players to inform each other safely of their hands of cards. We have presented a three-step Protocol `ThreeSteps` that achieves that goal. We verified the properties of that protocol by combinatorial means, using properties of block designs, and also by model checking in DEMO. Future work involves investigating other designs, in order to find protocols of at least three steps for card deals of size  $(a, b, c)$  for  $c \geq 2$  (for  $c = 1$  and two steps, a full characterization is known). This will advance the characterization of card deals for which such exchanges of secrets are possible.

## Acknowledgment

We thank three anonymous CLIMA reviewers for their comments. Hans van Ditmarsch is also affiliated to IMSC (Institute of Mathematical Sciences Chennai), India, as associated researcher. We thank the participants of the Sevilla logic seminar for their interaction.

## References

1. M.H. Albert, R.E.L. Aldred, M.D. Atkinson, H. van Ditmarsch, and C.C. Handley. Safe communication for card players by combinatorial designs for two-step protocols. *Australasian Journal of Combinatorics*, 33:33–46, 2005.
2. M.H. Albert, A. Cerdón, H. van Ditmarsch, D. Fernández, J.J. Joosten, and F. Soler. Secure communication of local states in interpreted systems. In Ajith Abraham, Juan Corchado, Sara González, and Juan De Paz Santana, editors, *International Symposium on Distributed Computing and Artificial Intelligence*, Advances in Intelligent and Soft Computing, Vol. 91, pages 117–124. Springer, 2011.

3. A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In I. Gilboa, editor, *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98)*, pages 43–56, 1998.
4. C. Dixon. Using temporal logics of knowledge for specification and verification—a case study. *Journal of Applied Logic*, 4(1):50–78, 2006.
5. P. Dobcsányi. Design db, 2011. <http://batman.cs.dal.ca/~peter/designdb/>.
6. R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge MA, 1995.
7. M.J. Fischer and R.N. Wright. Bounds on secret key exchange using a random deal of cards. *Journal of Cryptology*, 9(2):71–99, 1996.
8. A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX:5–38 and 161–191, 1883.
9. T. Kirkman. On a problem in combinations. *Camb. and Dublin Math. J.*, 2:191–204, 1847.
10. K. Koizumi, T. Mizuki, and T. Nishizeki. Necessary and sufficient numbers of cards for the transformation protocol. In K.-Y. Chwa and J. Ian Munro, editors, *Computing and Combinatorics, 10th Annual International Conference (COCOON 2004)*, LNCS 3106, pages 92–101. Springer, 2004.
11. X. Luo, K. Su, A. Sattar, and Y. Chen. Solving sum and product riddle via bdd-based model checking. In *Web Intelligence/IAT Workshops*, pages 630–633. IEEE, 2008.
12. R. Parikh and R. Ramanujam. A knowledge based semantics of messages. *Journal of Logic, Language and Information*, 12:453–467, 2003.
13. J.A. Plaza. Logics of public communications. In M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems: Poster Session Program*, pages 201–216. Oak Ridge National Laboratory, 1989.
14. A. Stiglic. Computations with a deck of cards. *Theoretical Computer Science*, 259(1–2):671–678, 2001.
15. D.R. Stinson. *Combinatorial Designs – Constructions and Analysis*. Springer, 2004.
16. H. van Ditmarsch. The Russian cards problem. *Studia Logica*, 75:31–62, 2003.
17. H. van Ditmarsch, W. van der Hoek, R. van der Meyden, and J. Ruan. Model checking russian cards. *Electronic Notes in Theoretical Computer Science*, 149:105–123, 2006. Presented at MoChArt 05 (Model Checking in Artificial Intelligence).
18. J. van Eijck. DEMO — a demo of epistemic modelling. In J. van Benthem, D. Gabbay, and B. Löwe, editors, *Interactive Logic — Proceedings of the 7th Augustus de Morgan Workshop*, pages 305–363. Amsterdam University Press, 2007. Texts in Logic and Games 1.
19. S. van Otterloo, W. van der Hoek, and M. Wooldridge. Model checking a knowledge exchange scenario. *Applied Artificial Intelligence*, 18(9-10):937–952, 2004.
20. Y. Wang. *Epistemic Modelling and Protocol Dynamics*. PhD thesis, Universiteit van Amsterdam, 2010.

## A Equivalence classes after $A$ 's announcement

$c_A = [[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29], [30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44], [45, 46,$

47,48,49,50,51,52,53,54,55,56,57,58,59], [60,61,62,63,64,65,66,67,68,69,70,71,72,73,74], [75,76,77,78,79,80,81,82,83,84,85,86,87,88,89], [90,91,92,93,94,95,96,97,98,99,100,101,102,103,104], [105,106,107,108,109,110,111,112,113,114,115,116,117,118,119], [120,121,122,123,124,125,126,127,128,129,130,131,132,133,134], [135,136,137,138,139,140,141,142,143,144,145,146,147,148,149], [150,151,152,153,154,155,156,157,158,159,160,161,162,163,164], [165,166,167,168,169,170,171,172,173,174,175,176,177,178,179], [180,181,182,183,184,185,186,187,188,189,190,191,192,193,194], [195,196,197,198,199,200,201,202,203,204,205,206,207,208,209], [210,211,212,213,214,215,216,217,218,219,220,221,222,223,224]]

cB = [[0,59,104], [1], [2], [3], [4], [5,44,119], [6], [7], [8], [9], [10], [11], [12], [13], [14,29,164], [15], [16,87,132], [17], [18], [19,72,147], [20], [21], [22], [23], [24], [25], [26], [27], [28], [30], [31,82,191], [32], [33], [34,67,176], [35], [36], [37], [38], [39], [40], [41], [42], [43], [45], [46,78,220], [47], [48], [49,63,205], [50], [51], [52], [53], [54], [55], [56], [57], [58], [60], [61], [62], [64], [65], [66], [68], [69], [70], [71], [73], [74], [75], [76], [77], [79], [80], [81], [83], [84], [85], [86], [88], [89], [90], [91], [92,141,186], [93,126,171], [94], [95], [96], [97], [98], [99], [100], [101], [102], [103], [105], [106], [107,139,217], [108,124,202], [109], [110], [111], [112], [113], [114], [115], [116], [117], [118], [120], [121], [122], [123], [125], [127], [128], [129], [130], [131], [133], [134], [135], [136], [137], [138], [140], [142], [143], [144], [145], [146], [148], [149], [150], [151], [152,170,215], [153,185,200], [154], [155], [156], [157], [158], [159], [160], [161], [162], [163], [165], [166], [167], [168], [169], [172], [173], [174], [175], [177], [178], [179], [180], [181], [182], [183], [184], [187], [188], [189], [190], [192], [193], [194], [195], [196], [197], [198], [199], [201], [203], [204], [206], [207], [208], [209], [210], [211], [212], [213], [214], [216], [218], [219], [221], [222], [223], [224]]

cC = [[0,15,30,105,150], [1,16,60,135,151], [2,17,152,180,195], [3,18,153,165,210], [4,19,75,120,154], [5,20,45,90,155], [6,31,61,106,166], [7,32,107,121,196], [8,46,62,91,197], [9,47,92,122,167], [10,33,108,136,211], [11,34,76,109,181], [12,48,93,137,182], [13,49,77,94,212], [14,35,50,95,110], [21,36,63,138,168], [22,37,78,123,183], [23,64,96,139,184], [24,79,97,124,169], [25,66,111,141,213], [26,81,112,126,198], [27,51,67,142,199], [28,52,82,127,214], [29,69,84,129,144], [38,65,98,125,170], [39,80,99,140,185], [40,70,156,171,216], [41,85,157,186,201], [42,57,72,173,203], [43,58,87,188,218], [44,73,88,174,189], [53,68,113,128,200], [54,83,114,143,215], [55,71,158,187,202], [56,86,159,172,217], [59,74,89,204,219], [100,146,162,191,206], [101,131,163,176,221], [102,117,132,177,207], [103,118,147,192,222], [104,133,148,178,193], [115,145,160,175,220], [116,130,161,190,205], [119,134,149,208,223], [164,179,194,209,224]]

## B A Haskell script

```
import List
goodGraph :: Integer -> Bool
goodGraph n =
  let
```

```

-- cc1: C-class with n (a pentagon)
cc1 = head (take 1 (filter (\x -> (elem n x)) cC))
-- bc1: points B-accessible with cc1 (4 singlet. + 1 triang.)
bc1 = foldl1 (++)
      (take 5 (filter (\x -> (intersect x cc1) /= []) cB))
-- cc2: C-classes with points in bc1 (three pentagons)
cc2 = filter (\x -> (intersect x bc1) /= []) cC
-- cc3: points in the three pentagons
cc3 = foldl1 (++) cc2
-- bc2: B-classes with points in cc2 (12 singlet. + 1 triang.)
bc2 = filter (\x -> (intersect x cc3 /= [])) cB
-- na: number of A-classes with elements in the graph
na = length (filter (\x -> (intersect x cc3 /= [])) cA)
in
-- 1: all the A-classes are represented (1 point per class)
na == 15 &&
-- 2: in the B-classes there is just one triangle
length (filter (\x -> (length x) == 3) bc2) == 1 &&
-- 3: in the B-classes there are 12 singletons
length (filter (\x -> (length x) == 1) bc2) == 12 &&
-- 4: in the C-classes there are 15 points (3 pentagons)
length cc3 == 15

```