TP4 MVFA

February 17, 2017

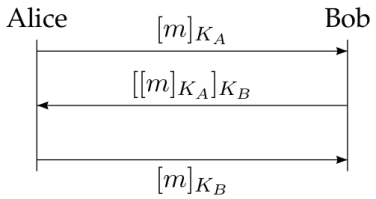
The aim of today's practical session is to verify some LTL properties with Spin.

1. Last time, we have seen the Dining philosophers' problem (`https://fr.wikipedia.org/wiki/D%C3%AEner_des_philosophes` (French), `https://en.wikipedia.org/wiki/Dining_philosophers_problem` (English)). You implemented this problem in Spin thanks to Petri nets. Implement the dining philosophers problem in Spin directly, without resorting to Petri nets. Verify whether the philosophers may starve to death.

2. We focus now on a cryptographic protocol. To share a secret, two agents use the following protocol. Each agent A has a pair of keys:

   - a key K_A, which is used as a lock. We denote by $[m]_{K_A}$ the message m encoded by K_A;
   - a key K_A^{-1}, used as a key, allowing one to decode messages encoded with K_A.

   We assume that the encoding is commutative, that is : $[[m]_{K_A}]_{K_B} = [[m]_{K_B}]_{K_A}$. Denoting by K_A the key of Alice, and by K_B that of Bob, the protocol used by Alice and Bob to share a secret message m is given as follows:

   Note that to compute the last message, Alice has used the fact that encoding is commutative.

   Consider a potential attack, by Charlie. We assume that Charlie can grab any message of the network, send a message he owns, encode a message with his key KC, and compute m from $[m]_{K_C}$, thanks to K_C^{-1}. But he cannot encode or decode a message using Alice's or Bob's keys: he does not know K_A, K_B, K_A^{-1}, K_B^{-1}.

   We want to model the protocol and an attacker using Spin. Model the net by a rendez-vous channel. We assume that the messages have 3 fields: the message content, and the encryption keys (2 keys maximum). We use the following Spin structure (to understand structures, read Section 2.1.7 of `http://spinroot.com/spin/Man/WhatsNew.html`):

```
typedef msg {
    mtype data;
    mtype key1;
    mtype key2;
};
```

The fields `key1` and `key2` of a message can be `KA`, `KB`, `KC` or `None`, which means that the message is not encrypted: a non encrypted message has both fields set to `None`. Similarly if `m.key1` is `None` and `m.key2` is `KB` (or conversely), this means that m is encrypted by K_B only. Finally, if `m.key1` is `KA` and `m.key2` is `KB`, this means that m is encrypted by both K_A and K_B.

3. Model Alice and Bob using Spin. You will need to replace `mtype` to any type you want to specify data and keys.

4. Model Charlie using Spin.

5. Show whether Charlie can intercept a uncrypted message.