Exercise 6-1. We have seen a number of protocols whose execution is partitioned in different phases. One example is the Needham-Schroeder protocol, where we have assumed that a session key does not remain secret indefinitely, but may be obtained by the attacker after some time.

Within the formalism of Horn clauses one can model different phases of a protocol using different predicates $\text{att}_0(x), \text{att}_1(x), \ldots$, where $\text{att}_i(x)$ represents the potential knowledge of the attacker in phase $i$ of the protocol.

Recall the original Needham-Schroeder protocol:

1. $A \rightarrow S : A, B, N_A$
2. $S \rightarrow A : \{K_{AB}, B, N_A, \{K_{AB}, A\} \}_{K_{BS}} K_{AS}$
3. $A \rightarrow B : \{K_{AB}, A\} \}_{K_{BS}}$  
4. $B \rightarrow A : \{(1, N_B)\} K_{AB}$
5. $A \rightarrow B : \{(0, N_B)\} K_{AB}$
6. $B \rightarrow A : \{\text{secret}\} K_{AB}$

An attacker in the possession of an old run of this protocol and the associated session key may obtain the secret.

a) Encode this protocol directly using Horn clauses. (Proverif has a mode for Horn clauses, which may help in testing the encoding.)

b) Modify the encoding so that there are two phases, where in the second phase all session keys from the first phase may have been leaked. Analyse the secrecy of the secret in this model using Proverif’s Horn clause mode.

Exercise 6-2. Construct a pi calculus process $P$ for which the translation to Horn clauses gives a false positive with respect to secrecy. That is, there is some secret $s$, such that the clause $\text{att}(s)$ is derivable from the Horn clauses, even though $P$ never outputs $s$ on a public channel.

Exercise 6-3. Construct a set of Horn clauses for which the saturation algorithm from the lectures does not terminate.