1. What are the five different ways of proving properties of systems that were presented in the slides and video content?

    - **Axiomatic proof: - proof using the rules of axiomatic sematic**
    - **Brute force: - Brute force the exploration of the model state place. We can brute force exploration of the program state which is to big unless we do it very small number of bits.**
    - **Transform to model: - Transforming the program to a simpler model.**
    - **Symbolic execution (abstract interpretation): keep track of symbolic values**
    - **Theorem proving**

2. Which way of proving does Spin use?

    - **As it is mentioned in the paper the basic structure of spin is model checker. The typical mode of working is to start with the specification of a high-level model of a concurrent system, or distributed algorithm.**

3. What is the formal system description language (modeling language) that Spin uses? What does the name stand for? What kinds of systems are the focus of Promela (and Spin)?

    - **Promela is a verification modelling language used by spin. The name stands for process meta language. Promela (spin) provides a vehicle for making distributed systems in general that suppress details that are unrelated to process interaction. The intended use of Spin is to verify fractions of process behaviour, that for one reason or another are considered suspect.**

4. Although some formal properties can be specified directly in Promela (e.g., with in-line assertions), Spin supports another property formalism. What is it?

    - **SPIN accepts correctness claims specified in the syntax of standard Linear Temporal Logic (LTL). LTL is a modal temporal logic with modalities referring to time [1], and it can specify the properties separate from the model.**

5. The paper references safety and liveness properties. What are two examples of each that the paper offers?

    - **absence of deadlock, unreachable code, unspecified receptions, network timeout… can be example of safety and liveness property.**

Reference

https://en.wikipedia.org/wiki/Linear_temporal_logic