



Universiteit
Leiden

The Netherlands

Formal models of software-defined networks

Feng, H.

Citation

Feng, H. (2024, December 3). *Formal models of software-defined networks*.
Retrieved from <https://hdl.handle.net/1887/4170508>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4170508>

Note: To cite this publication please use the final published version (if applicable).

Propositions

pertaining to the thesis

Formal models of Software-Defined Networks

by Hui Feng

1. Incorporating formal models and languages into software-defined network development enhances overall system reliability by enabling thorough verification of network behavior and policies. [Chapter 1 & 2]
2. The coordination language Reo can be used to model software-defined networks, allowing for compositional and formal specification of network components and their interactions. [Chapter 3 & 4]
3. By translating Reo models into Promela code, developers can utilize the SPIN model checker to perform exhaustive verification of software-defined networks for correctness and reliability. [Chapter 5]
4. Constraint automata can be used to provide extended formal semantics for concurrent NetKAT programs, enabling modeling and analysis of SDN policies. [Chapter 6]
5. OpenFlow is a protocol for controlling network devices, but it is not synonymous with software-defined networking. The latter is a broader paradigm that encompasses various architectures and technologies beyond just protocol-level control.
6. Testing software-defined networks remains difficult due to their dynamic nature and the complex interactions between the control and data planes, which can obscure subtle errors.
7. One of the primary challenges in the formal verification of software-defined networks is ensuring scalability as network size and complexity increase, requiring advanced algorithms and heuristics to manage the computational load.
8. Temporal logic and other formal logic systems can be effectively utilized to specify and verify dynamic properties of software-defined networks, such as flow consistency and packet forwarding correctness.

-
9. The PhD process, with its intense focus on critical thinking and deep analysis, remains the gold standard for intellectual mastery and is increasingly critical in today's fast-paced, skills-based economy.
 10. Engaging with new information has become a practical necessity, demanding constant adaptation to stay relevant. The true challenge lies in knowing what to prioritize and when to disengage.

Hui Feng
Leiden
December 3rd, 2024