A toolchain that allows the apple-to-apple comparison of
Implementation Level (frameworks):

A1. JFA: an FA variant that uses state variables to avoid SE, and

It's unclear the advantages and disadvantages of different
tools, especially on the many-core (e.g., GPU) devices due to:

- CPU, GPU, FPGA, Automata Processor, etc
- can't directly handle the out-of-order packets.
- this is vulnerable to denial-of-service (DoS) attacks.
- it has to buffer the packets and then process flow only
- only work on small and simple datasets.

Automata-based processing has been applied to diverse areas
even on the many-core (GPU) devices due to:

- the many-core (GPU) devices due to:
- tight dependencies between every two computation steps.
- constructing memory access patterns.
- is unclear the advantages and disadvantages of different
automata processing accelerators and the innovation space

CPU, GPU, FPGA, Automata Processor, etc.

The above design

F1. Platform Comparison Toolchain

Out-of-Order Packets issue in NIDS

- cannot handle the out-of-order packets.
- due to different pattern densities.
- can't handle the out-of-order packets.

A2. O-FA

- NFA: cannot directly handle out-of-order packets.
- DFA: can't directly handle the out-of-order packets.

A3. O-FA

- cannot handle the out-of-order packets.
- due to different pattern densities.
- can't handle the out-of-order packets.