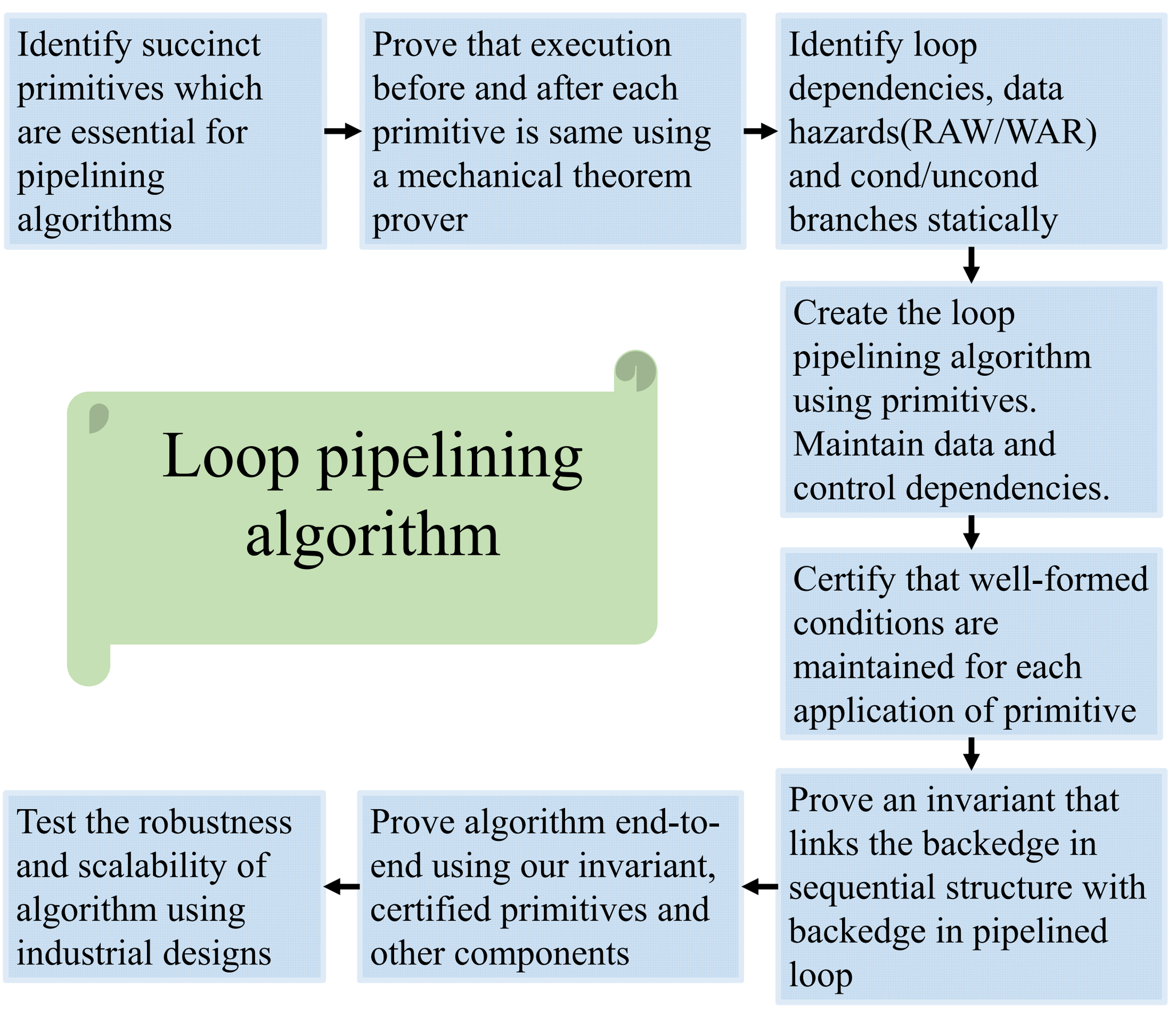
Identify succinct primitives which are essential for pipelining algorithms
Prove that execution before and after each primitive is same using a mechanical theorem prover
Identify loop dependencies, data hazards(RAW/WAR) and cond/uncond branches statically
Create the loop pipelining algorithm using primitives. Maintain data and control dependencies.
Certify that well-formed conditions are maintained for each application of primitive
Prove an invariant that links the backedge in sequential structure with backedge in pipelined loop
Prove algorithm end-to-end using our invariant, certified primitives and other components
Test the robustness and scalability of algorithm using industrial designs
Loop pipelining algorithm