1. Motivation

- The formal verification of complex industrial designs often entails checking a large number of properties.
- **Equivalence checking** compares pairwise equality of each design output: distinct property per output,
- **Functional verification** checks low-level assertions to high-level encompassing properties, and
- **Design-space exploration** via model checking verifies properties against competing system designs.

- Most research and development efforts address the problem of single-property verification, multiple properties are verified concurrently, or one-at-a-time.

Possible inter-property relationships, and shared sub-problems are typically ignored. 
Opportunity to save verification resource.

2. Multiple Property Verification

- Develop efficient and scalable techniques for automatic verification of multiple properties.
- **Inter-property relationships** – utilize logical dependencies to minimize model-checking runs.
- **Information reuse** – learned state-space information is reused across various property verification tasks.
- **Improved orchestration** – properties with nearly identical cone-of-influence are verified concurrently.

3. Inter-Property Relationships

- Proprocess the set of properties to find pairwise logical dependencies; LTL satisfiability checking.

\[
\varphi_1 = \Box p \\
\varphi_2 = \Box (p \land q) \\
\varphi_3 = \Box (p \lor q)
\]

- \(M \models \varphi_2\) then \(M \models \varphi_1\)
- \(\varphi_3\) and \(\varphi_2\) are dependent

- Results:

\[
\begin{array}{c|c|c|c|c|c|c}
\hline
\text{keys} & \varphi_1 & \varphi_2 & \varphi_3 & \varphi_4 & \varphi_5 & \varphi_6 \\
\hline
\varphi_1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\varphi_2 & 0 & 1 & 0 & 0 & 0 & 0 \\
\varphi_3 & 0 & 0 & 1 & 0 & 0 & 0 \\
\varphi_4 & 0 & 0 & 0 & 1 & 0 & 0 \\
\varphi_5 & 0 & 0 & 0 & 0 & 1 & 0 \\
\varphi_6 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
\end{array}
\]

- Property Table

- Conclusion: Several checks, but only one produces a result.

Few minutes to find dependencies between properties, and <10% properties checked for each design.

4. Information Reuse

- Sequentially check properties by reusing information; state approximations, counterexamples, and invariants.
- Stored information is repaired before reuse; add “just enough” extra information to enable reuse.

Adapt IC3/PDR for multi-property model checking by reusing frames to enable 4.5× faster verification.

5. Improved Orchestration

- Property grouping saves substantial verification resource by concurrent verification of high-affinity properties.

- Partition properties into provably high-affinity groups based on cone-of-influence (COI); ~linear runtime.
- Two-level orchestration; structural property grouping followed by semantic property-group refinement.

Improved multiple property verification offering 4.8× end-to-end speedup; advance state-of-the-art localization.

6. Ongoing and Future Work

- When to use structural vs. semantic grouping? Difficult to discern what COI subset is relevant to what property.

- Sequential equivalence checking (SEC) is a prevalent multiple property verification application; several miters.
- Improve SEC by intelligently discharging non-inductive provable miters by improved property orchestration.

---