Matching Logic = Operational Semantics + FOL

- A logic for reasoning about configurations
- Formulae
  - FOL over configurations, called patterns
  - Configurations are allowed to contain variables
- Models
  - Ground configurations
- Satisfaction
  - Matching for configurations, plus FOL for the rest

Examples of Patterns

- \( x \) points to sequence \( A \), and the reversed sequence \( A \) has been output
  \( \exists n \exists \pi \exists \omega \ (x \mapsto \pi) \mapsto \omega \mapsto \{ \text{init}(n, A), \omega \} \mapsto \{ \text{read}, \text{rev}(A) \} \mapsto \omega \)
- untrusted() can only be called from trusted()
  \( \exists n \exists \omega \ (\text{untrusted}(n) \mapsto \{ n, \text{trusted}(n), \omega \} \mapsto \omega \)
- Read/Write data race (simplified)
  \( \exists X \exists a \ (X \mapsto a \mapsto \_ ) \)

Partial Correctness

- We have two rewrite relations on configurations
  \( \rightarrow \) given by the language operational semantics; \( \text{safe} \)
  \( \rightarrow \) given by specifications; \( \text{unsafe} \), has to be proved
- Idea (simplified for deterministic languages):
  - Pick \( \text{left} \rightarrow \text{right} \). Show that always \( \text{left} \rightarrow \{ \ldots \rightarrow \text{right} \}
  - \text{modulo matching logic reasoning (between rewrite steps)}
- Theorem (soundness):
  - If \( \text{left} \rightarrow \text{right} \) and \( \text{config} \) matches \( \text{left} \) such that \( \text{config} \)
    has a normal form for \( \rightarrow \), then \( \text{nf(config) matches right} \)