CWI

# Rascal for DSL construction

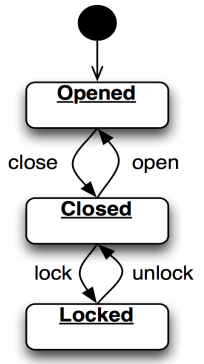
- **Goal**: power tools for DSL construction
- **Challenges**:
  - Modular syntax definition
  - Semantic analysis
  - Transformation
  - Code generation
  - IDE support
- **Rascal provides:**
  - Declarative context-free grammars
  - Built-in relational calculus
  - Pattern matching and traversal
  - Auto-indenting string templates
  - Hooks into Eclipse IDE services

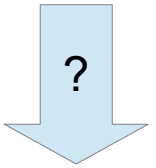
DSL Code

```
state Opened
  close => Closed

state Closed
  open => Opened
  lock => Locked

state Locked
  unlock => Closed
```




Parse, check, transform, generate

Java Code

```
while (true) {
    event = input.next();
    switch (current) {
    case "Opened":
        if (event.equals("close"))
            current = "Closed";
        break;
    case "Closed":
        if (event.equals("open"))
            current = "Opened";
        if (event.equals("lock"))
            current = "Locked";
        break;
    case "Locked":
        if (event.equals("unlock"))
            current = "Closed";
        break;
    }
}
```

www.rascal-mpl.org