



RULES FEST



Understanding Production Systems 101

Charles Young
Principal Consultant
Solidsoft Ltd.

Production Systems

- **Emil Post**
 - **'Canonical' System**
 - pattern-matching
 - string-rewriting
 - basis for BNF (Backus-Naur Form)
 - Turing-complete
 - Production rules



Production Systems

- **Extended Post system**
 - Facts, not just strings
 - Assertion of new facts
 - Retraction of existing facts



Why Production Systems?

- **Artificial Intelligence**
 - Knowledge-based processing
 - Mimicking human cognitive processes
- **What is knowledge?**
 - Expertise based on experience and understanding
 - Justified beliefs
 - Facts, truths, principles

Why Production Systems?

- Just another Turing machine...
 - Rules programming
 - Declarative (?)
 - Knowledge-based
 - Logic programming (?)
 - 'impure' model

Why Production Systems?

- **Commercial value?**
 - not in the engine...
 - in the tooling that exploits the engine.
- **Capture, analyse, verify and apply knowledge**
- **Align run-time logic to business view**
- **Separate and externalise declarative rules from procedural logic**
- **Deploy logic across distributed environments**

Why Production Systems?

- **Expert systems**
- **Business rules processing**
- **Process automation**
- **Complex event processing**
- **Cognitive and learning systems**

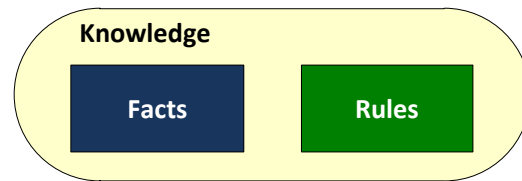
Facts

- **Relational tuples**
 - Collection of attribute values
 - May be ordered or unordered
 - May be triplets
- **May map to non-relational data**
 - XML/Xpath
 - OO / POJO / POCO
- **Fact Identity**

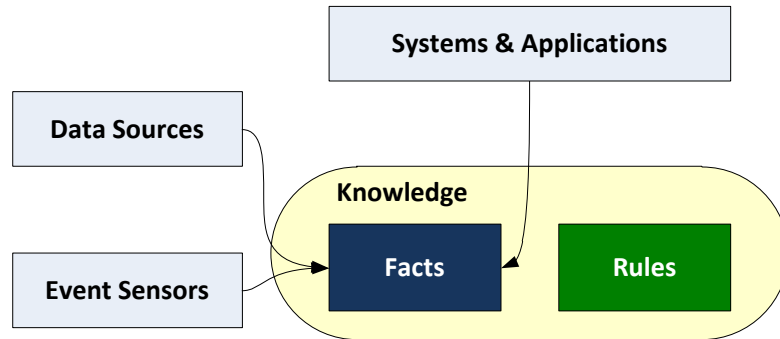
Production Rules

- **<conditions> → <actions>**
- **IF/WHEN/WHERE**
 <conjunction of logical expressions>
THEN / DO
 <list of actions>
- **Flexible**
- **Imprecise**

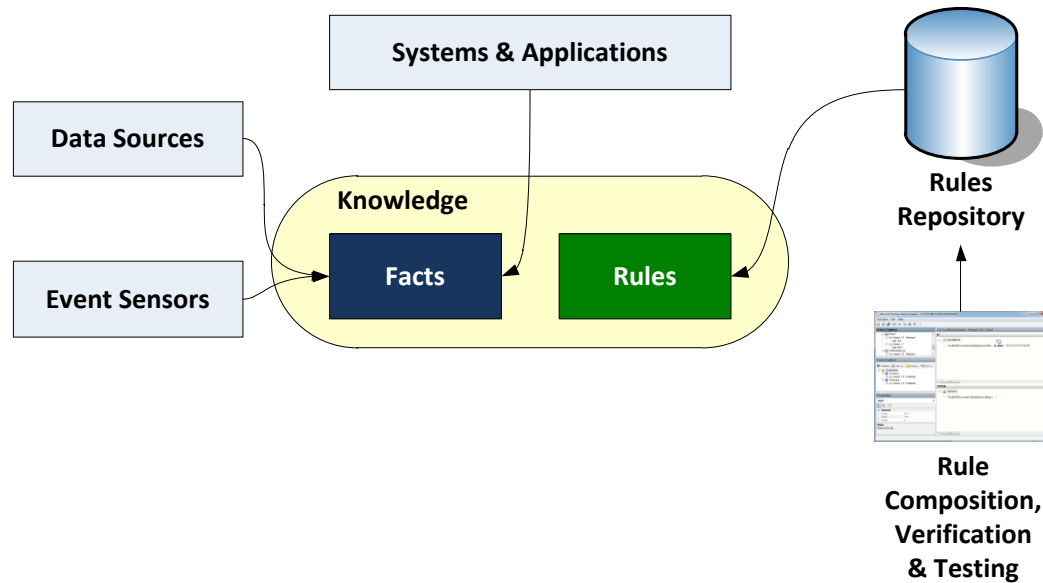
Production Systems



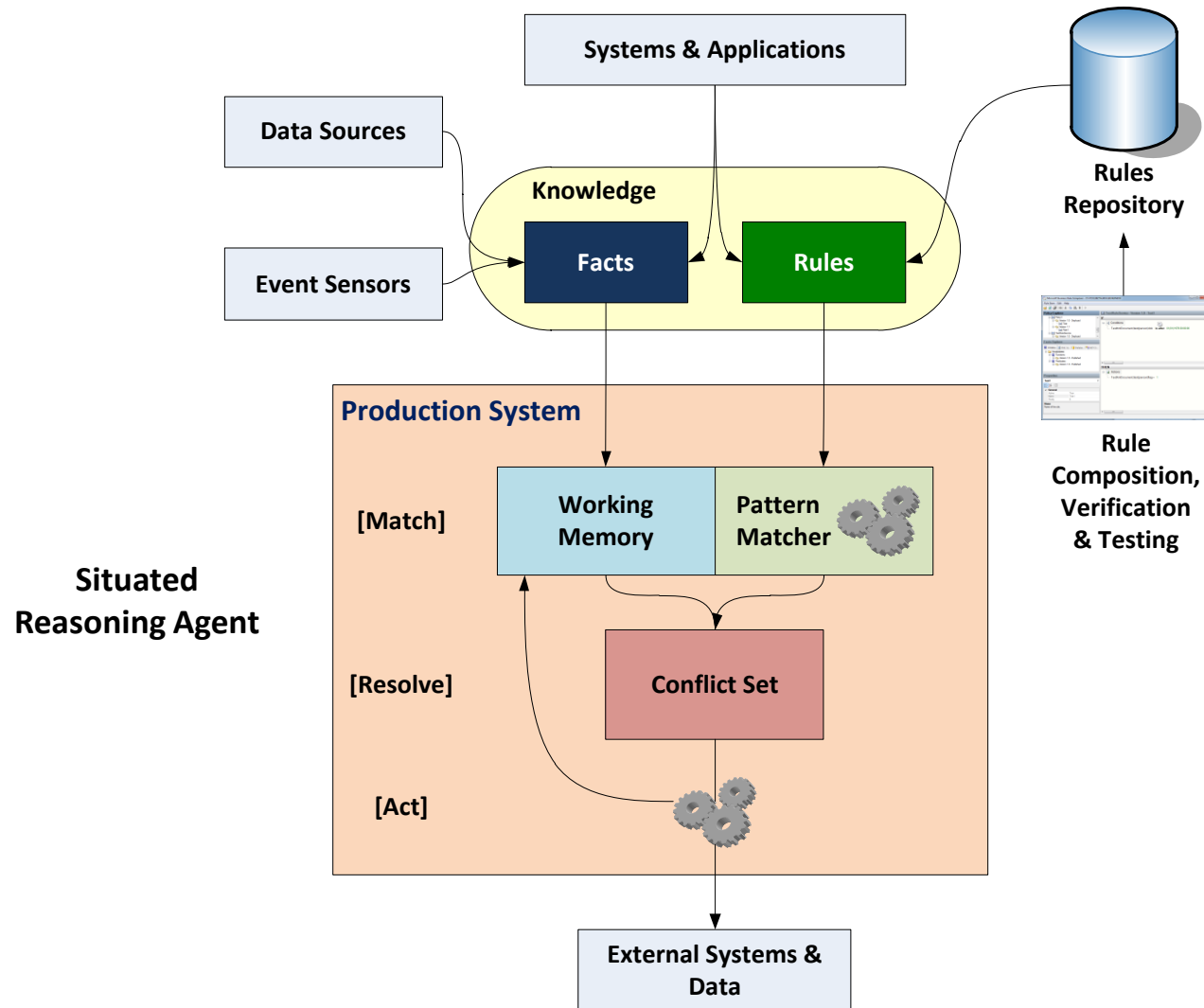
Production Systems



Production Systems



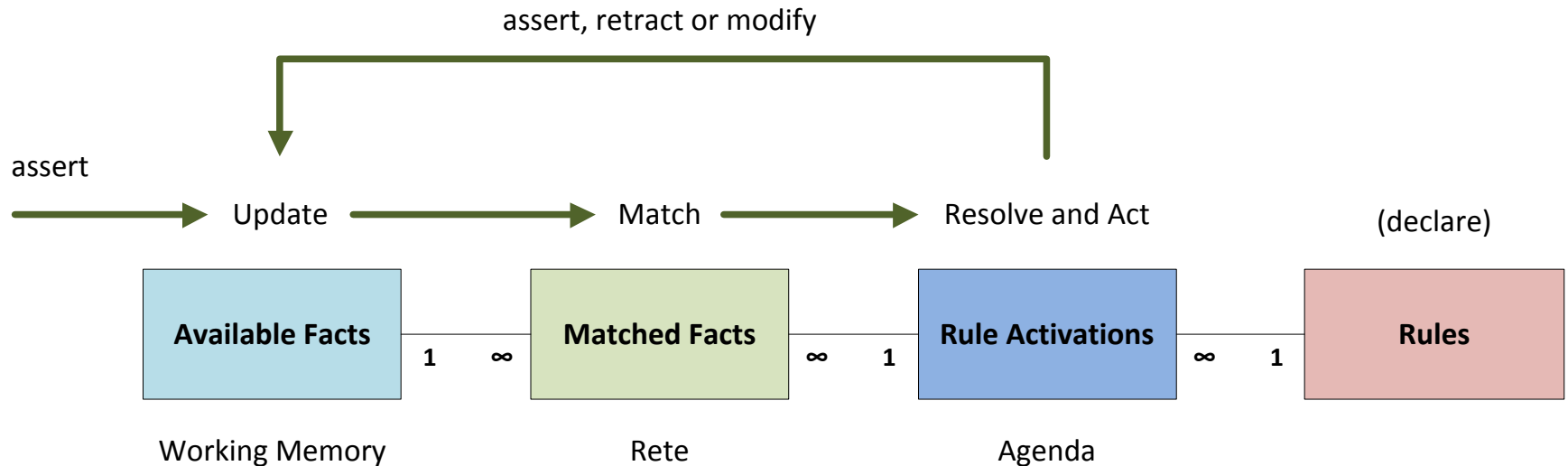
Production Systems



Matching

- **Filter**
 - by type
 - by attribute value
 - by time (CEP)
- **Aggregate**
 - Cartesian products
 - joins
 - quantification

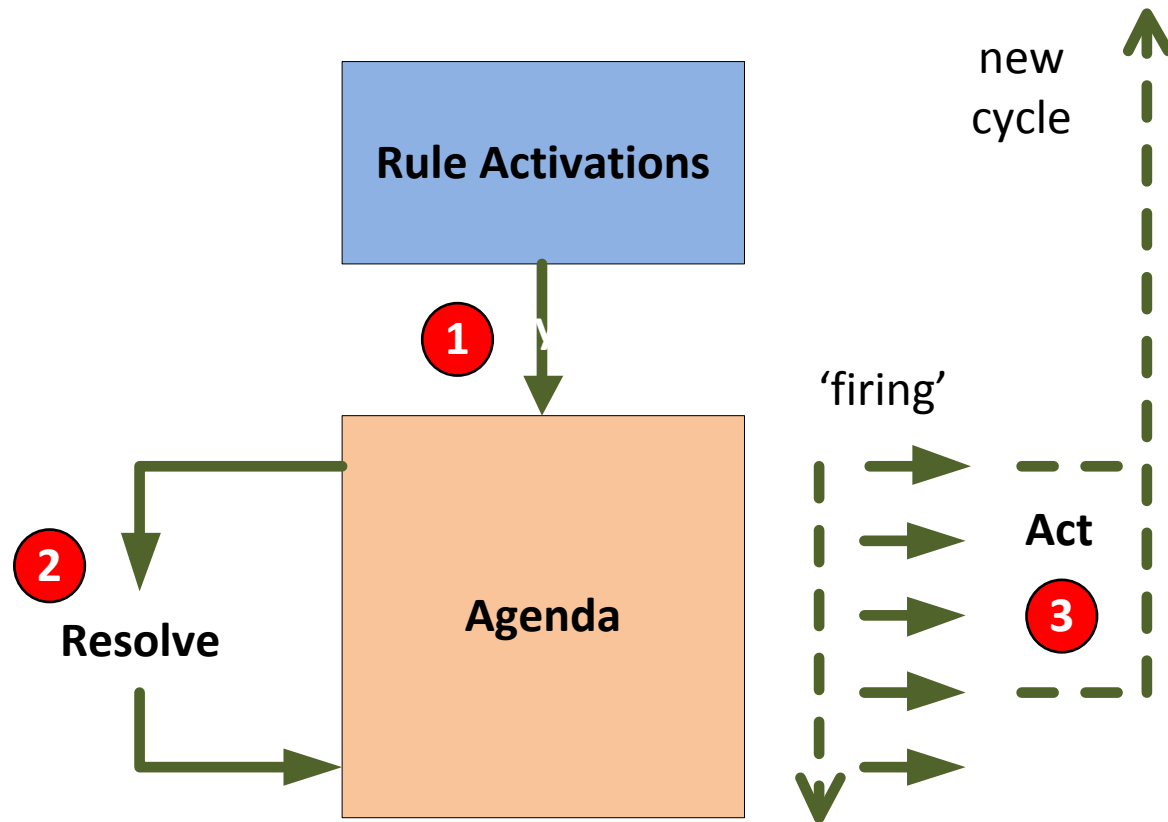
Matches and Activations



Resolving

- **Problem Spaces**
 - Search strategies
 - MEA / LEX / Depth-First/ Breadth-First / Random
- **Salience**

Agenda



Inference & Chaining

- **Basic Inference**
 - May be possible in single step (rule)
 - IF age < 18 THEN set status = minor
- **Chained Inference**
 - Multiple, interim steps
 - Enabled through recursive match-resolve-act
 - Supports non-monotonic reasoning

Forward & Backward Reasoning

- **Forward Reasoning**

- IF value > \$1,000
THEN set discount = 10%

- **Backward Reasoning**

- IF GOAL = 10% discount
THEN send order of value > \$1,000

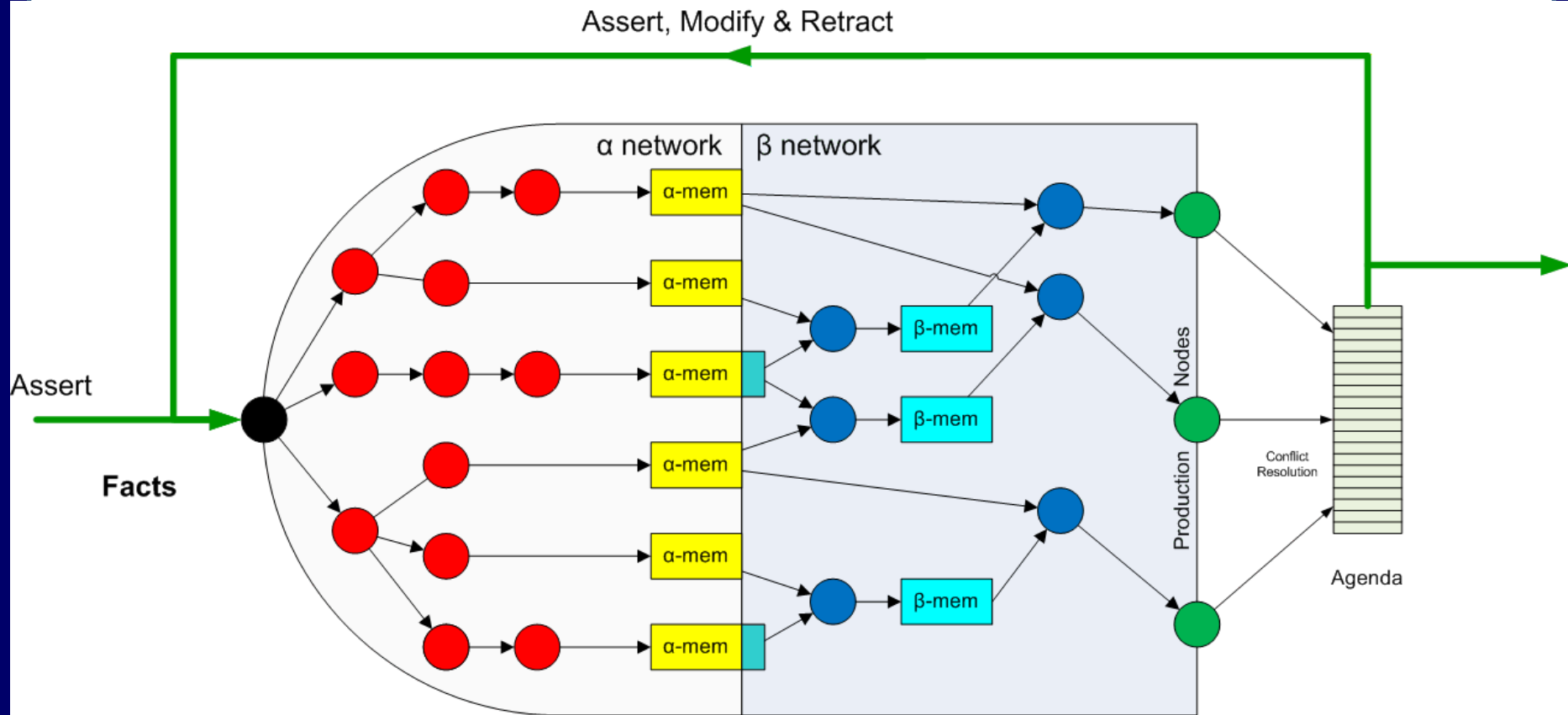
Forward & Backward Chaining

- **Chained Inference**
 - Forward & Backward reasoning
- **Same Match-Resolve-Act cycle**
- **Sub-goaling**
- **Hybrid models**

The Rete Algorithm

- **An algorithm for efficient pattern matching**
 - **Dr. Charles Forgy**
 - **Small, incremental changes between M-R-A cycles**
 - **Eliminate redundancies**
 - **Recalculate delta changes to conflict set**

The Rete Algorithm



Other Issues

- **Discrete vs. Continuous Execution**
- **Temporal Logic**
- **Truth Maintenance**
- **Justification**
- **Debugging and Tracing**