

Unified Rule Engine

Usage and Control

Nil Geisweiller

SingularityNetathon 2018

OpenCog Foundation

URE: Unified Rule Engine

The URE is a tool to **evolve Inference Trees**

A $A \rightarrow B$

----- (MP)

B

$A \rightarrow B$ $B \rightarrow C$

----- (DED)

A $A \rightarrow C$

----- (MP)

C

ForAll X P(X) $\cup(P(X), A)$

----- (INS)

A

$A \rightarrow B$ $B \rightarrow C$

----- (DED)

$A \rightarrow C$

----- (MP)

C

- Leaves are **premises**
- Roots are **conclusions**

URE: Unified Rule Engine

Inference Trees are constructed by composing **Rules**

- **forward**: expand conclusion



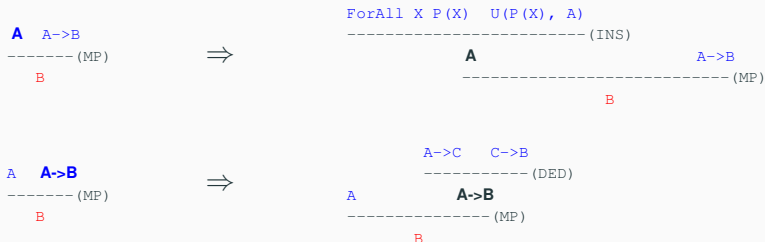
URE: Unified Rule Engine

Inference Trees are constructed by composing **Rules**

- **forward**: expand conclusion



- **backward**: expand premises



URE: Unified Rule Engine

Inference Trees are **Atomese** Programs

A A→B

----- (MP)

B

```
(BindLink
  (VariableList
    (TypedVariableLink
      (VariableNode "$A")
      (TypeChoice
        (TypeNode "LambdaLink")
        (TypeNode "PredicateNode")
      )
    )
    (TypedVariableLink
      (VariableNode "$B")
      (TypeChoice
        (TypeNode "LambdaLink")
        (TypeNode "PredicateNode")
      )
    )
  )
  (AndLink
    (ImplicationLink
      (VariableNode "$A")
      (VariableNode "$B")
    )
    (EvaluationLink
      (GroundedPredicateNode "scm: true-enough")
      (ImplicationLink
        (VariableNode "$A")
        (VariableNode "$B")
      )
    )
  )
  (EvaluationLink
    (GroundedPredicateNode "scm: true-enough")
    (VariableNode "$A")
  )
)
(ExecutionOutputLink
  (GroundedSchemaNode "scm: modus-ponens-formula")
  (ListLink
    (VariableNode "$B")
    (VariableNode "$A")
    (ImplicationLink
      (VariableNode "$A")
      (VariableNode "$B")
    )
  )
)
```

URE: Unified Rule Engine

Inference Trees are Atomese Programs

A A→B
----- (MP)
 B

```
(ExecutionOutputLink
  (GroundedSchemaNode "scm: modus-ponens-formula")
  (ListLink
    (VariableNode "$B")
    (VariableNode "$A")
    (ImplicationLink
      (VariableNode "$A")
      (VariableNode "$B")
    )
  )
)
```

Inference Trees are **Atomese** Programs

$A \rightarrow C$ $C \rightarrow B$
 ----- (DED)
 A $A \rightarrow B$
 ----- (MP)
 B

```
(ExecutionOutputLink
  (GroundedSchemaNode "scm: modus-ponens-formula")
  (ListLink
    (VariableNode "$B")
    (VariableNode "$A")
  )
  (ExecutionOutputLink
    (GroundedSchemaNode "scm: deduction-formula")
    (ListLink
      (ImplicationLink
        (VariableNode "$A")
        (VariableNode "$B")
      )
      (ImplicationLink
        (VariableNode "$A")
        (VariableNode "$C")
      )
      (ImplicationLink
        (VariableNode "$C")
        (VariableNode "$B")
      )
    )
  )
)
```

Algorithm

1. Select an inference tree to expand
2. Select a node from that tree to expand
3. Select a rule to expand with
4. Expand the inference tree and place it back to the pool of inference trees. Repeat till termination.

Combinatorial Explosion

Inference Control

Delegate the hard decisions to a **cognitive process**

Cognitive Schematics:

Context & **Action** \Rightarrow **Goal**

Atomese:

Implication <TV>

And

<Context>

<Action>

<Goal>

Which rule to choose?

A $A \rightarrow B$
----- (MP)
B

1. Modus Ponens
2. Universal Instantiation

Look for:

Implication <TV>

And

<inference-tree-pattern>

<node-pattern>

<rule-pattern>

<produce-good-inference>

Examples of actual URE Cognitive Schematics

```
(ImplicationScopeLink (stv 0.45945946 0.04625)
  (VariableList
    (VariableNode "$T")
    (TypedVariableLink
      (VariableNode "$A")
      (TypeNode "DontExecLink")
    )
    (VariableNode "$I")
    (TypedVariableLink
      (VariableNode "$B")
      (TypeNode "DontExecLink")
    )
  )
  (AndLink
    (EvaluationLink
      (PredicateNode "URE:BC:preproof-of")
      (ListLink
        (VariableNode "$A")
        (VariableNode "$T")
      )
    )
  )
  (ExecutionLink
    (SchemaNode "URE:BC:expand-and-BIT")
    (ListLink
      (VariableNode "$A")
      (VariableNode "$I")
      (DontExecLink
        (DefinedSchemaNode "deduction-inheritance-rule")
      )
    )
    (VariableNode "$B")
  )
)
(EvaluationLink
  (PredicateNode "URE:BC:preproof-of")
  (ListLink
    (VariableNode "$B")
    (VariableNode "$T")
  )
)
```

```
(ImplicationScopeLink (stv 1 0.00625)
  (VariableList
    (VariableNode "$T")
    (TypedVariableLink
      (VariableNode "$A")
      (TypeNode "DontExecLink")
    )
    (VariableNode "$X")
    (TypedVariableLink
      (VariableNode "$B")
      (TypeNode "DontExecLink")
    )
  )
  (AndLink
    (EvaluationLink
      (PredicateNode "URE:BC:preproof-of")
      (ListLink
        (VariableNode "$A")
        (VariableNode "$T")
      )
    )
    (ExecutionLink
      (SchemaNode "URE:BC:expand-and-BIT")
      (ListLink
        (VariableNode "$A")
        (InheritanceLink
          (ConceptNode "a")
          (VariableNode "$X")
        )
      )
      (DontExecLink
        (DefinedSchemaNode "conditional-full-instantiation-implication-scope-meta-rule")
      )
    )
    (VariableNode "$B")
  )
)
(EvaluationLink
  (PredicateNode "URE:BC:preproof-of")
  (ListLink
    (VariableNode "$B")
    (VariableNode "$T")
  )
)
```

How to get Cognitive Schematics?

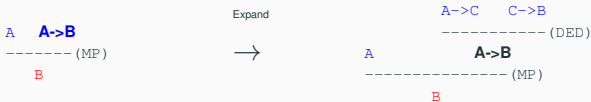
Answer: **Learning**

Record a **trace** of all decisions
the URE takes and **learn** from it

How to learn Cognitive Schematics?

1. URE
2. Pattern Mining
3. MOSES
4. Others

Inference Control Learning: URE Trace



Atome:

Inference Control Learning: URE Trace



Atomese:

```
(evalatomlink (stv 0.0001 0.00075)
  (predicatemode "DRE:st-go-proof-of-c")
  (listlink
    (inputatomlink
      (inputlink
        (typevariablelink
          (variablemode "A-25df17a5")
          (typechoice
            (typechoice "ConceptMode")
            (typechoice "AxiomMode")
            (typechoice "UtilMode")
            (typechoice "UtilMode")
          )
        )
      )
      (typevariablelink
        (variablemode "B-2bdc1299")
        (typechoice
          (typechoice "ConceptMode")
          (typechoice "AxiomMode")
          (typechoice "UtilMode")
          (typechoice "UtilMode")
        )
      )
      (typevariablelink
        (variablemode "D-6a776a5c")
        (typechoice
          (typechoice "ConceptMode")
          (typechoice "AxiomMode")
          (typechoice "UtilMode")
          (typechoice "UtilMode")
        )
      )
    )
    (axiomlink
      (evalatomlink
        (roundpredicatemode "sum: qt-sevo-confidence")
        (inputatomlink
          (inputlink
            (typevariablelink
              (variablemode "A-25df17a5")
              (variablemode "B-2bdc1299")
            )
            (inputatomlink
              (stv 1 1)
              (conceptmode "u" (stv 0.84615385 1))
              (conceptmode "u")
            )
          )
        )
      )
      (inputatomlink
        (variablemode "D-6a776a5c")
        (variablemode "B-2bdc1299")
      )
      (inputatomlink
        (variablemode "A-25df17a5")
        (variablemode "B-2bdc1299")
      )
      (inputatomlink
        (stv 1 1)
        (conceptmode "u" (stv 0.84615385 1))
        (conceptmode "u")
      )
    )
  )
  (utillink
    (inputatomlink
      (variablemode "B-2bdc1299")
      (variablemode "A-25df17a5")
    )
  )
  (inputatomlink
    (variablemode "A-25df17a5")
    (variablemode "D-6a776a5c")
  )
)
```

Inference Control Learning: URE

Call the URE with a Control Rule Template as target

```
(cog-bc icr-rb (QuoteLink
  (ImplicationScopeLink
    (UnquoteLink
      (VariableNode "$impl-vardecl")
    )
    (AndLink
      (EvaluationLink
        (PredicateNode "URE:BC:preproof-of")
        (UnquoteLink
          (VariableNode "$preproof-A-args")
        )
      )
      (ExecutionLink
        (SchemaNode "URE:BC:expand-and-BIT")
        (UnquoteLink
          (VariableNode "$expand-inputs")
        )
        (UnquoteLink
          (VariableNode "$expand-output")
        )
      )
    )
  )
  (EvaluationLink
    (PredicateNode "URE:BC:preproof-of")
    (UnquoteLink
      (VariableNode "$preproof-B-args")
    )
  )
)
)
```


Inference Control Learning: URE

Call the URE with a Control Rule Template as target

```
(cog-bc icr-rb (QuoteLink
  (ImplicationScopeLink
    (UnquoteLink
      (VariableNode "$impl-vardecl")
    )
    (AndLink
      (EvaluationLink
        (PredicateNode "URE:BC:preproof-of")
        (UnquoteLink
          (VariableNode "$preproof-A-args")
        )
      )
      (ExecutionLink
        (SchemaNode "URE:BC:expand-and-BIT")
        (UnquoteLink
          (VariableNode "$expand-inputs")
        )
        (UnquoteLink
          (VariableNode "$expand-output")
        )
      )
    )
  )
  (EvaluationLink
    (PredicateNode "URE:BC:preproof-of")
    (UnquoteLink
      (VariableNode "$preproof-B-args")
    )
  )
)
)
```

```
(ImplicationScopeLink (stv 0.45945946 0.04625)
  (VariableList
    (VariableNode "$T")
    (TypedVariableLink
      (VariableNode "$A")
      (TypeNode "DontExecLink")
    )
    (VariableNode "$L")
    (TypedVariableLink
      (VariableNode "$B")
      (TypeNode "DontExecLink")
    )
  )
  (AndLink
    (EvaluationLink
      (PredicateNode "URE:BC:preproof-of")
      (ListLink
        (VariableNode "$A")
        (VariableNode "$T")
      )
    )
    (ExecutionLink
      (SchemaNode "URE:BC:expand-and-BIT")
      (ListLink
        (VariableNode "$A")
        (VariableNode "$L")
        (DontExecLink
          (DefinedSchemaNode "deduction-inheritance-rule")
        )
      )
    )
  )
  (EvaluationLink
    (PredicateNode "URE:BC:preproof-of")
    (ListLink
      (VariableNode "$B")
      (VariableNode "$T")
    )
  )
)
```

Inference Control Learning: URE + Pattern Miner

Call the URE with a Control Rule Template as target + Pattern Miner

```
(cog-bc icr-rb (QuoteLink
  (ImplicationScopeLink
    (UnquoteLink
      (VariableNode "$impl-vardecl")
    )
    (AndLink
      (EvaluationLink
        (PredicateNode "URE:BC:preproof-of")
        (UnquoteLink
          (VariableNode "$preproof-A-args")
        )
      )
      (ExecutionLink
        (SchemaNode "URE:BC:expand-and-BIT")
        (UnquoteLink
          (VariableNode "$expand-inputs")
        )
        (UnquoteLink
          (VariableNode "$expand-output")
        )
      )
    )
  )
  (EvaluationLink
    (PredicateNode "URE:BC:preproof-of")
    (UnquoteLink
      (VariableNode "$preproof-B-args")
    )
  )
)
)
```

```
(ImplicationScopeLink (stv 1 0.00625)
  (VariableList
    (VariableNode "$T")
    (TypedVariableLink
      (VariableNode "$A")
      (TypeNode "DontExecLink")
    )
  )
  (VariableNode "$X")
  (TypedVariableLink
    (VariableNode "$B")
    (TypeNode "DontExecLink")
  )
)
(AndLink
  (EvaluationLink
    (PredicateNode "URE:BC:preproof-of")
    (ListLink
      (VariableNode "$A")
      (VariableNode "$T")
    )
  )
  (ExecutionLink
    (SchemaNode "URE:BC:expand-and-BIT")
    (ListLink
      (VariableNode "$A")
      (InheritanceLink
        (ConceptNode "a")
        (VariableNode "$X")
      )
    )
    (DontExecLink
      (DefinedSchemaNode "conditional-full-instantiation-implication-scope-meta-rule")
    )
  )
  (VariableNode "$B")
)
)
(EvaluationLink
  (PredicateNode "URE:BC:preproof-of")
  (ListLink
    (VariableNode "$B")
    (VariableNode "$T")
  )
)
)
```