OWL, DL and rules
Rule based systems are an important and useful way to represent and reason with knowledge.

Adding rules to OWL has proved to be fraught with problems.

We’ll look at the underlying issues and two approaches:
- SWRL: failed standard that has become widely used
- RIF: a successful standard that’s not yet widely used
The Semantic Web is grounded in logic

But what logic?

- OWL Full = Classical first order logic (FOL)
- OWL-DL = Description logic
- N3 rules \(\sim=\) logic programming (LP) rules
- SWRL \(\sim=\) DL + LP
- Other choices are possible, e.g., default logic, fuzzy logic, probabilistic logics, ...

How do these fit together and what are the consequences
We need both structure and rules

- **OWL’s ontologies** based on DL (and thus in FOL)
  - The Web is an open environment
  - Reusability / interoperability
  - An ontology is a model easy to understand

- **Many rule systems** based on logic programming
  - To achieve decidability, ontology languages don’t offer the expressiveness we want. Rules do it well
  - Efficient reasoning support already exists
  - Rules are well-known and often more intuitive
Description Logics vs. Horn Logic

- Neither is a subset of the other
- Impossible in OWL DL: people who study & live in same city are local students,
- Easily done with a rule
  \[
  \text{studiesAt}(X,U), \text{loc}(U,L), \text{lives}(X,L) \rightarrow \text{localStud}(X)
  \]
- Impossible in horn rules: every person is either a man or a woman
- Easily done in OWL DL:
  \[
  
  :\text{Person} \text{owl:disjointUnionOf} (:\text{Man} :\text{Woman}).
  \]
What’s Horn clause logic

- Prolog and most ‘logic’-oriented rule languages use horn clause logic
  - Defined by UCLA mathematician Alfred Horn
- Horn clauses are a subset of FOL where every sentence is a disjunction of literals (atoms) where at most one is positive
  \[ \sim P \lor \sim Q \lor \sim R \lor S \]
  \[ \sim P \lor \sim Q \lor \sim R \]
- Atoms are propositional variables (isRaining) or predicates (married(alice, ?x))
Horn clauses can be re-written using the implication operator

\[ \neg P \lor Q = P \rightarrow Q \]
\[ \neg P \lor \neg Q \lor R = P \land Q \rightarrow R \]
\[ \neg P \lor \neg Q = P \land Q \rightarrow \]

What we end up with is ~ “pure prolog”

- Single positive atom as the rule conclusion
- Conjunction of positive atoms as the rule antecedents (conditions)
- No **not** operator
- Atoms can be predicates (e.g., mother(X,Y))
We can relax this a bit

- Head can contain a conjunction of atoms
  - $P \land Q \leftarrow R$ is equivalent to $P \leftarrow R$ and $Q \leftarrow R$

- Body can have disjunctions
  - $P \leftarrow R \lor Q$ is equivalent to $P \leftarrow R$ and $P \leftarrow Q$

- But something are just not allowed:
  - No disjunction in head
  - No negation operator, i.e. NOT
Where are the quantifiers?

- Quantifiers (forall, exists) are implicit
  - Variables in \textit{rule head} are universally quantified
  - Variables only \textit{in rule body} are existentially quantified

- Example:
  - isParent(X) \iff hasChild(X,Y)
  - forall X: isParent(X) if Exisits Y: hasChild(X,Y)
Facts & rule conclusions are definite

- Definite means *not a disjunction*
- Facts are rule with the trivial true condition
- Consider these true facts:
  - \( P \lor Q \)  # either \( P \) or \( Q \) (or both) are true
  - \( P \rightarrow R \)  # if \( P \) is true, then \( R \) is true
  - \( Q \rightarrow R \)  # if \( Q \) is true, then \( R \) is true
- What can you conclude?
- Can this be expressed in horn logic?
Facts & rule conclusions are definite

- Consider these true facts where *not* is classical negation rather than “negation as failure”
  
  \[
  \text{not}(P) \Rightarrow Q, \quad \text{not}(Q) \Rightarrow P \quad \# \text{i.e. } P \lor Q
  \]
  
  \[
  P \Rightarrow R, \quad Q \Rightarrow R
  \]

- A horn clause reasoner can’t prove that either P or Q is necessarily true or false so can’t show that R must be true

- Treating *not* as negation as failure yields a loop
The LP-semantics is defined in terms of minimal Herbrand model, i.e., sets of ground facts. Because of this, Horn clause reasoners cannot derive rules, so that they cannot do general subsumption reasoning.
Decidability

- The largest obstacle!
  Tradeoff between expressiveness and decidability

- Facing decidability issues from
  - In LP: Finiteness of the domain
  - In classical logic (and thus in DL): combination of constructs

- Problem:
  Combination of “simple” DLs and Horn Logic are undecidable. (Levy & Rousset, 1998)
**SWRL: Semantic Web Rule Language**

- SWRL is the **union** of DL and horn logic + many built-in functions (e.g., for math)
- Submitted to the W3C in 2004, but failed to become a recommendation
  - W3C pursued a more general solution: **RIF**
- Problem: full SWRL specification leads to **undecidability** in reasoning
- SWRL is well specified and subsets are widely supported (e.g., in Pellet, HermiT)
SWRL

- OWL classes are unary predicates, properties are binary ones
  \[
  \text{Person}(?p) \land \text{sibling}(?p, ?s) \land \text{Man}(?s) \rightarrow \text{brother}(?p, ?s)
  \]

- As in Prolog, built-ins can be booleans or do a computation and unify the result to a variable
  - \text{swrlb:greaterThan}(?age2, ?age1) \quad \# \text{age2}>\text{age1}
  - \text{swrlb:subtract}(?n1, ?n2, ?diff) \quad \# \text{diff}=\text{n1}-\text{n2}

- SWRL predicates for OWL axioms and data tests
  - \text{differentFrom}(?x, ?y), \text{sameAs}(?x, ?y), \text{xsd:int}(?x), [3, 4, 5](?x), ...
The Essence of SWRL

- Combines OWL DL (and thus OWL Lite) with function-free Horn logic
- Thus it allows Horn-like rules to be combined with OWL DL ontologies
Rules in SWRL

$B_1, \ldots, B_n \rightarrow A_1, \ldots, A_m$

$A_1, \ldots, A_m, B_1, \ldots, B_n$ have one of the forms:

- $C(x)$
- $P(x,y)$
- $\text{sameAs}(x,y)$ $\text{differentFrom}(x,y)$

where $C$ is an OWL description, $P$ is an OWL property, and $x,y$ are variables, OWL individuals or OWL data values
SWRL Built-Ins

- SWRL defines a set of built-in predicate that allow for comparisons, math evaluation, string operations and more
- See [here](#) for the complete list
- Examples
  - Person(?p), hasAge(?p, ?age), `swrlb:greaterThan(?age, 18)` -> Adult(?p)
  - Person(?p), bornOnDate(?p, ?date), `xsd:date(?date)`, `swrlb:date(?date, ?year, ?month, ?day, ?timezone)` -> bornInYear(?p, ?year)
- Some reasoners (e.g., Pellet) allow you to define new built-ins in Java
Drawbacks of SWRL

- Main *source of complexity*: arbitrary OWL expressions, such as restrictions, can appear in the head or body of a rule.
- Adds significant expressive power to OWL, but causes *undecidability*.
  - there is no inference engine that draws exactly the same conclusions as the SWRL semantics.
SWRL Sublanguages

- SWRL adds the expressivity of DLs and function-free rules
- One challenge: identify sublanguages of SWRL with right balance between expressivity and computational viability
- A candidate OWL DL + *DL-safe rules*
  - every variable must appear in a non-description logic atom in the rule body
Standard reasoners support only DL-safe rules
  - Rule variables bind only to known individuals (i.e., owl2 owl:NamedIndividual)

Example

:Vehicle(?v) ^ :Motor(?m) ^ :hasMotor(?v,?m) -> :MotorVehicle(?v)

Where

:Car = :Vehicle and some hasMotor Motor
:x a :Car

The reasoner will not bind ?m to a motor since it is not a known individual
Protégé 5 had SWRLTab

Add/edit rules and optionally run a separate rules engine

Using the Drools rule engine.

Press the 'OWL+SWRL→Drools' button to transfer SWRL rules and relevant OWL knowledge to the rule engine.
Press the 'Run Drools' button to run the rule engine.
Press the 'Drools→OWL' button to transfer the inferred rule engine knowledge to OWL knowledge.

The SWRLAPI supports an OWL profile called OWL 2 RL and uses an OWL 2 RL–based reasoner to perform reasoning. See the 'OWL 2 RL' sub-tab for more information on this reasoner.
SWRL limitations

SWRL rules do not support many useful features of some rule-based systems:
- Default reasoning
- Rule priorities
- Negation as failure (e.g., for closed-world semantics)
- Data structures
- ...

The limitations gave rise to RIF
Summary

- Horn logic is a subset of predicate logic that allows efficient reasoning, orthogonal to description logics.
- Horn logic is the basis of monotonic rules.
- DLP and SWRL are two important ways of combining OWL with Horn rules.
  - DLP is essentially the intersection of OWL and Horn logic.
  - SWRL is a much richer language.
Nonmonotonic rules are useful in situations where the available information is incomplete. They are rules that may be overridden by contrary evidence. Priorities are sometimes used to resolve some conflicts between rules. Representation XML-like languages is straightforward.