The Semantic Web is grounded in logic

But what logic?
- OWL Full = Classical first order logic (FOL)
- OWL-DL = Description logic
- N3 rules ~= logic programming (LP) rules
- SWRL ~= DL + LP
- Other choices are possible, e.g., default logic, Markov logic, …

How do these fit together?
What are the consequences

We need both structure and rules

- OWL’s ontologies are based on Description Logics (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 in practice and often more intuitive

A common approach
LP and classical logic overlap

- FOL: (All except (6)), (2)+(3)+(4): DLs
- (4): Description Logic Programs (DLP), (3): Classical Negation
- (4)+(5): Horn Logic Programs,
- (4)+(5)+(6): LP
- (6): Non-monotonic features (like NAF, etc.)
- (7): ^head and, Vbody

Description Logics vs. Horn Logic

- Neither of them is a subset of the other
- It is impossible to assert that persons who study and live in the same city are “local students” in OWL
  - This can be done easily using rules:
    `studies(X,Y), lives(X,Z), loc(Y,U), loc(Z,U) → localStudent(X)`
- Rules cannot assert the information that a person is either a man or a woman
  - This information is easily expressed in OWL using disjoint union

Basic Difficulties

- Classical Logic vs. Logic Programming
  - Monotonic vs. Non-monotonic Features
    - Open-world vs. Closed-world assumption
    - Negation-as-failure vs. classical negation
  - Non-ground entailment
  - Strong negation vs. classical negation
  - Equality
  - Decidability

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
  - `~P V ~Q V ~R V S`
  - `~P V ~Q V ~R`
- Atoms are propositional variables (isRaining) or predicates (married(alice, ?x))
An alternate formulation

- 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 \textasciitilde\textasciitilde “pure prolog”
  - Single positive atom as the rule conclusion
  - Conjunction of positive atoms as the rule antecedents (conditions)
  - No \texttt{not} operator
  - Atoms can be predicates (e.g., mother(X,Y))

Where are the quantifiers?

- Quantifiers (forall, exists) are implicit
  - Variables in rule head (i.e., conclusion or consequent) are universally quantified
  - Variables only in rule body (i.e., condition or antecedent) are existentially quantified
- Example:
  - isParent(X) \leftarrow hasChild(X,Y)
  - forAll X: isParent(X) if Exists Y: hasChild(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

Facts & rule conclusions are definite

- Definite means \textit{not a disjunction}
- Facts are rule with the trivial true condition
- Consider these true facts:
  \[ P \lor Q \]
  \[ P \Rightarrow R \]
  \[ Q \Rightarrow R \]
- What can you conclude?
- Can this be expressed in horn logic?
Facts & rule conclusions are definite

- Consider these true facts where not is Prolog’s “negation as failure” operator:
  - not(P) → Q, not(Q) → P
  - P → R
  - Q → R
- A horn clause reasoner is unable to prove that either P or Q is necessarily true or false.
- And can not show that R must be true.

Open- vs. closed-world assumption

- Logic Programming – CWA
  - If KB |= a, then KB = KB ∪ ¬a
- Classical Logic – OWA
  - It keeps the world open.
  - KB:
    - Man ⊑ Person, Woman ⊑ Person
    - Bob ∈ Man, Mary ∈ Woman
  - Query: “find all individuals that are not women”

Non-ground entailment

- The LP-semantics is defined in terms of minimal Herbrand model, i.e. sets of ground facts.
- Because of this, Horn clause reasoners can not derive rules, so that can not do general subsumption reasoning.

Decidability

- The largest obstacle!
  - Tradeoff between expressiveness and decidability.
- Facing decidability issues from 2 different angles
  - 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)
Rules + Ontologies

- Still a challenging task!
- A number of different approaches exist: SWRL, DLP (Grosof), dl-programs (Eiter), DL-safe rules, Conceptual Logic Programs (CLP), AL-Log, DL+log
- Two main strategies:
  - Tight Semantic Integration (Homogeneous Approaches)
  - Strict Semantic Separation (Hybrid Approaches)

Homogeneous Approach

- Interaction with tight semantic integration
- Both ontologies and rules are embedding in a common logical language
- No distinction between rule predicates and ontology predicates
- Rules may be used for defining classes and properties of the ontology
- Example: SWRL, DLP

Hybrid Approach

- Integration with strict semantic separation between the two layers
- Ontology used to conceptualize the domain
- Rules can’t define ontology classes and properties, but some application-specific relations
- Communication via a “safe interface”
- Example: answer set programming (ASP)

The Essence of DLP

- Simplest approach for combining DLs with Horn logic: their intersection
  - the Horn-definable part of OWL, or equivalently
  - the OWL-definable part of Horn logic
- The OWL 2 RL profile is the DLP part of OWL
Advantages of DLP

- **Modeling**: Freedom to use either OWL or rules
  - and their associated tools and methodologies
- **Implementation**: use either description logic reasoners or deductive rule systems
  - extra flexibility, interoperability with a variety of tools
- **Expressivity**: existing OWL ontologies frequently use few constructs outside DLP

---

OWL in Horn Logic

<table>
<thead>
<tr>
<th>Rule</th>
<th>Horn Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>C sameClassAs D</td>
<td>C(X) → D(X)</td>
</tr>
<tr>
<td>P samePropertyAs Q</td>
<td>P(X,Y) → Q(X,Y)</td>
</tr>
</tbody>
</table>

---

RDFS and Horn Logic

<table>
<thead>
<tr>
<th>Rule</th>
<th>Horn Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement(a,P,b)</td>
<td>P(a,b)</td>
</tr>
<tr>
<td>type(a,C)</td>
<td>C(a)</td>
</tr>
<tr>
<td>C subClassOf D</td>
<td>C(X) → D(X)</td>
</tr>
<tr>
<td>P subPropertyOf Q</td>
<td>P(X,Y) → Q(X,Y)</td>
</tr>
<tr>
<td>domain(P,C)</td>
<td>P(X,Y) → C(X)</td>
</tr>
<tr>
<td>range(P,C)</td>
<td>P(X,Y) → C(Y)</td>
</tr>
</tbody>
</table>

---

OWL in Horn Logic (2)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Horn Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>transitiveProperty(P)</td>
<td>P(X,Y), P(Y,Z) → P(X,Z)</td>
</tr>
<tr>
<td>inverseProperty(P,Q)</td>
<td>Q(X,Y) → P(Y,X)</td>
</tr>
<tr>
<td>functionalProperty(P)</td>
<td>P(X,Y), P(X,Z) → Y=Z</td>
</tr>
</tbody>
</table>
**OWL in Horn Logic (3)**

\[(C_1 \cap C_2) \text{ subClassOf } D\]
- \(C_1(X), C_2(X) \rightarrow D(X)\)

\(C \text{ subClassOf } (D_1 \cap D_2)\)
- \(C(X) \rightarrow D_1(X)\)
- \(C(X) \rightarrow D_2(X)\)

**OWL in Horn Logic (4)**

\[(C_1 \cup C_2) \text{ subClassOf } D\]
- \(C_1(X) \rightarrow D(X)\)
- \(C_2(X) \rightarrow D(X)\)

\(C \text{ subClassOf } (D_1 \cup D_2)\)
- Translation not possible!
- \(C \rightarrow D_1 \text{ or } D_2\)

**OWL in Horn Logic (5)**

\(C \text{ subClassOf } \text{AllValuesFrom}(P, D)\)
- \(C(X), P(X,Y) \rightarrow D(Y)\)

\(\text{AllValuesFrom}(P,D) \text{ subClassOf } C\)
- Translation not possible!

**OWL in Horn Logic (6)**

\(C \text{ subClassOf } \text{SomeValuesFrom}(P, D)\)
- Translation not possible!
- \(C(X) \rightarrow P(X,Y), D(Y)\)

\(\text{SomeValuesFrom}(P,D) \text{ subClassOf } C\)
- \(D(X), P(X,Y) \rightarrow C(Y)\)
OWL in Horn Logic (7)

- MinCardinality cannot be translated due to existential quantification
- MaxCardinality 1 may be translated if equality is allowed
- Complement cannot be translated, in general

SWRL

- **Semantic Web Rule Language**
- SWRL is the union of DL and horn logic + many built-in functions (e.g., 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 in Protege

- Protégé 4.x has minimal support for SWL
  You add/edit rules, some reasoners (Pellet, HermiT) use them
- Protégé 3.x has Jess, an internal rules engine
  Jess is a production rule system with a long ancestry
- And good tools for editing, managing and using rules
- See the SWRL tab

SWRL architecture for Protégé 3.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.

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
DL-safe rules

- (all?) reasoners support only DL-safe rules
  - Rule variables bind only to known individuals
- Example (mixing syntaxes):

```
:Vehicle(?v) ^ :Motor(?m) ^ :hasMotor(?v,?m) -> :MotorVehicle(?v)
:Car = :Vehicle and some hasMotor Motor
: a :Car
```

- The reasoner will not bind ?m to a motor since it is not a known individual

Protégé SWRL-Tab

**Protégé SWRL-Tab**

**SWRL in Protégé 4.2**
### Non-monotonic rules

- Non-monotonic rules use an “unprovable” operator
- This can be used to implement default reasoning, e.g.,
  - assume $P(X)$ is true for some $X$ unless you can prove that it is not
  - Assume that a bird can fly unless you know it can not

### Non-monotonic

<table>
<thead>
<tr>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>canFly(X) :- bird(X), + not(canFly(X))</code></td>
</tr>
<tr>
<td><code>bird(X) :- eagle(X)</code></td>
</tr>
<tr>
<td><code>bird(X) :- penguin(X)</code></td>
</tr>
<tr>
<td><code>not(canFly(X)) :- penguin(X)</code></td>
</tr>
<tr>
<td><code>eagle(sam)</code></td>
</tr>
<tr>
<td><code>penguin(tux)</code></td>
</tr>
</tbody>
</table>

### Rule priorities

- This approach can be extended to implement systems where rules have priorities
- This seems to be intuitive to people – used in many human systems
  - E.g., University policy overrules Department policy
  - The “Ten Commandments” can not be contravened
Two Semantic Webs?

- Rule inference support not integrated with OWL classifier
  - New assertions by rules may violate existing restrictions in ontology
  - New inferred knowledge from classification may produce knowledge useful for rules

Limitations

- Existing solution: solve possible conflicts manually
- Ideal solution: a single module for both ontology classification and rule inference
- What if we want to combine non-monotonic features with classical logic?
- Partial Solutions:
  - Answer set programming
  - Externally via appropriate rule engines

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
<table>
<thead>
<tr>
<th>Summary (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Nonmonotonic rules are useful in situations where the available information is incomplete</td>
</tr>
<tr>
<td>• They are rules that may be overridden by contrary evidence</td>
</tr>
<tr>
<td>• Priorities are sometimes used to resolve some conflicts between rules</td>
</tr>
<tr>
<td>• Representation XML-like languages is straightforward</td>
</tr>
</tbody>
</table>