



SWRL
RIF

Semantic Web
Rule Language
Rule Interchange
Format

Regole

- Regole formate da antecedente e conseguente
 - L'antecedente viene valutato su un determinato dataset
- Regole *dichiarative*
 - Hanno come conseguente una nuova asserzione (se x è presidente della Repubblica, x risiede al Quirinale)
- Regole di *produzione*
 - Hanno come conseguente un'azione (se hai l'influenza, prendi l'aspirina)

Specifica di SWRL

- Ufficiale dal 2004
- Utilizzo limitato a causa di problematiche di tipo computazionale (rende OWL non più decidibile)
- Supportato solo da alcuni reasoner (es. Pellet)
- Specifiche (submission): <https://www.w3.org/Submission/SWRL/>

Regole SWRL

Le regole SWRL hanno una forma IF-THEN, con antecedente e conseguente

- $\text{hasParent}(\text{?x1}, \text{?x2}) \wedge \text{hasBrother}(\text{?x2}, \text{?x3}) \Rightarrow \text{hasUncle}(\text{?x1}, \text{?x3})$

Le regole SWRL non possono creare individui, solo *asserire nuove relazioni/proprietà*

Struttura di antecedente e conseguente

Antecedente e conseguente contengono una congiunzione di atomi:

- **C(x)**
- **P(x,y),**
- **sameAs(x,y)**
- **differentFrom(x,y)**
- **operatore built-in(x, ...)**
- x e y sono variabili, individui o letterali

C è una classe

P è una proprietà

Built-ins: <https://www.w3.org/Submission/SWRL/#8>

Reasoner o Drools?

- E' possibile utilizzare i reasoner per interpretare le regole SWRL, con alcuni limiti:
 - Hermit non supporta gli operatori built-in
 - Pellet offre un supporto più ampio (ma non completo)
 - In entrambi i casi, le inferenze *non saranno aggiunte* all'ontologia a meno che non sia esportate (come nel caso standard)
- In alternativa, si può utilizzare Drools
 - Drools travasa nell'ontologia le inferenze effettuate
 - le inferenze effettuate *saranno aggiunte* alla A-box in modo permanente

SWRL Tab inProtégé

- Aggiungere il tab SWRL a Protégé (Window | Tabs | SWRLTab)
- Si basa su un engine esterno, chiamato Drools
- Implementazione migliorata dell'algoritmo RETE (*pattern matching*)
- Le regole vengono importate in Drools ed eseguite; i risultati possono essere importati in modo permanente nell'ontologia.

SWRL Tab in Protégé

The screenshot shows the Protégé application window with the 'cinema' ontology loaded. The 'SWRLTab' is active, displaying a table of rules. Below the table, there are buttons for 'New', 'Edit', 'Clone', and 'Delete'. At the bottom, there is a status window with tabs for 'Control', 'Rules', 'Asserted Axioms', 'Inferred Axioms', and 'OWL 2 RL'. The 'OWL 2 RL' tab is selected, showing a list of statistics and instructions. A blue arrow points to the 'Run Drools' button.

Name	Rule	Comment
<input checked="" type="checkbox"/> S1	cinema:Personaggio(?x) ^ cinema:personaggioDi(?x, ?y) ^ cinema:Commedia(?y) -> cinema:Person...	
<input checked="" type="checkbox"/> S2	cinema:haArgomento(?l, ?a) ^ cinema:Libro(?l) ^ cinema:Film(?f) ^ cinema:Argomento(?a) ^ cinema:...	
<input checked="" type="checkbox"/> S3	swrlb:greaterThan(?c, 300000) ^ cinema:haRuolo(?a, ?r) ^ cinema:compenso(?r, ?c) -> cinema:Att...	

Number of SWRL rules exported to rule engine: 3
Number of OWL class declarations exported to rule engine: 19
Number of OWL individual declarations exported to rule engine: 20
Number of OWL object property declarations exported to rule engine: 8
Number of OWL data property declarations exported to rule engine: 2
Total number of OWL axioms exported to rule engine: 156
The transfer took 92 millisecond(s).
Press the 'Run Drools' button to run the rule engine.
Successful execution of rule engine.
Number of inferred axioms: 147
The process took 142 millisecond(s).
Look at the 'Inferred Axioms' tab to see the inferred axioms.
Press the 'Drools->OWL' button to translate the inferred axioms to OWL knowledge.
Successfully transferred inferred axioms to OWL model.
The process took 12 millisecond(s).

**premere i pulsanti
nell'ordine per ottenere le
inferenze in OWL**

OWL+SWRL->Drools Run Drools Drools->OWL

Reasoner state out of sync with active ontology ☒ Show Inferences

Regole: esempio

cinema:Personaggio(?x)

^

cinema:personaggioDi(?x, ?y)

^

cinema:Commedia(?y)

->

cinema:PersonaggioComico(?x)

Esempio 2

Se un film è ispirato a un libro, il suo argomento è lo stesso del libro

cinema:haArgomento(?l, ?a)

^

cinema:Libro(?l)

^

cinema:Film(?f)

^

cinema:Argomento(?a)

^

cinema:trattoDa(?f, ?l)

->

cinema:haArgomento(?f, ?a)

Esempio di operatore numerico

SWRL supporta un insieme di *operatori built-in*

Per esempio, `swrlb:multiply` o `swrlb:greaterThan`

- <https://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>
- <https://github.com/protegeproject/swrlapi/wiki/SWRLLanguageFAQ>
- Non sempre supportati: provare!
- Mutuati da XQuery e XPath

`cinema:haRuolo(?a, ?r)`

^

`cinema:compenso(?r, ?c)`

^

`swrlb:greaterThan(?c, 300000)`

->

`cinema:AttoreMoltoPagato(?a)`

Formato della regola

```
<rdf:Description rdf:about="http://www.semanticweb.org/rossana/ontologies/2017/10/cinema#x">
  <rdf:type rdf:resource="http://www.w3.org/2003/11/swrl#Variable"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/rossana/ontologies/2017/10/cinema#y">
  <rdf:type rdf:resource="http://www.w3.org/2003/11/swrl#Variable"/>
</rdf:Description>
<rdf:Description>
  <swrla:isRuleEnabled rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</swrla:isRuleEnabled>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"></rdfs:comment>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">S1</rdfs:label>
  <rdf:type rdf:resource="http://www.w3.org/2003/11/swrl#Imp"/>
  <swrl:body>
    <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2003/11/swrl#AtomList"/>
      <rdf:first>
        <rdf:Description>
          <rdf:type rdf:resource="http://www.w3.org/2003/11/swrl#ClassAtom"/>
          <swrl:classPredicate rdf:resource="http://www.semanticweb.org/rossana/ontologies/2017/10/cinema#Personaggio"/>
          <swrl:argument1 rdf:resource="http://www.semanticweb.org/rossana/ontologies/2017/10/cinema#x"/>
        </rdf:Description>
        ...
      </swrl:body>
    </swrl:head>
    <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2003/11/swrl#AtomList"/>
      <rdf:first>
        <rdf:Description>
          <rdf:type rdf:resource="http://www.w3.org/2003/11/swrl#ClassAtom"/>
          <swrl:classPredicate rdf:resource="http://www.semanticweb.org/rossana/ontologies/2017/10/cinema#PersonaggioComico"/>
          <swrl:argument1 rdf:resource="http://www.semanticweb.org/rossana/ontologies/2017/10/cinema#x"/>
        </rdf:Description>
      </rdf:first>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
    </rdf:Description>
  </swrl:head>
</rdf:Description>
</rdf:RDF>
```

Regola in formato TTL

```
[ swrl:isEnabled "true"^^xsd:boolean ;
  rdfs:comment ""^^xsd:string ;
  rdfs:label "S1"^^xsd:string ;
  rdf:type swrl:Imp ;
  swrl:body [ rdf:type swrl:AtomList ;
    rdf:first [ rdf:type swrl:ClassAtom ;
      swrl:classPredicate cinema:Personaggio ;
      swrl:argument1 cinema:x
    ] ;
    rdf:rest [ rdf:type swrl:AtomList ;
      rdf:first [ rdf:type swrl:IndividualPropertyAtom ;
        swrl:propertyPredicate cinema:personaggioDi ;
        swrl:argument1 cinema:x ;
        swrl:argument2 cinema:y
      ] ;
      rdf:rest [ rdf:type swrl:AtomList ;
        rdf:first [ rdf:type swrl:ClassAtom ;
          swrl:classPredicate cinema:Commedia ;
          swrl:argument1 cinema:y
        ] ;
        rdf:rest rdf:nil
      ]
    ]
  ] ;
  swrl:head [ rdf:type swrl:AtomList ;
    rdf:first [ rdf:type swrl:ClassAtom ;
      swrl:classPredicate cinema:PersonaggioComico ;
      swrl:argument1 cinema:x
    ] ;
    rdf:rest rdf:nil
  ]
] .
```

RIF

- Rule Interchange Format
- Standard del W3C (<https://www.w3.org/TR/rif-overview/>)
- Finalizzato a facilitare la sintesi e l'integrazione di sistemi di regole
- Le regole vengono usate anche nei Linked Data per effettuare ragionamenti ad hoc (in mancanza di un reasoner)

Esempio

(da <https://www.w3.org/TR/rif-primer/>)

- DBpedia esprime la relazione tra film e attore mediante la proprietà `starring`:

```
starring(?Film ?Actor)
```

- IMDB invece si avvale della nozione di ruolo:

```
playsRole(?Actor ?Role)
```

```
roleInFilm(?Role ?Film)
```

Esempio

- In RIF una regola per la corrispondenza tra IMDB e Dbpedia avrebbe la forma seguente:

```
Forall ?Actor ?Film ?Role (  
  If And(<http://example.com/imdbrelations#playsRole>( ?Actor ?Role)  
        <http://example.com/imdbrelations#roleInFilm>( ?Role ?Film))  
  Then <http://dbpedia.org/ontology/starring>( ?Film ?Actor)  
)
```

- Altri elementi di RIF
 - Si possono definire dei Rule Set
 - *rdf:type (a B)* dice che l'individuo *a* appartiene alla classe *B*