Racer

One description logic reasoning system
Where are we?

- XML
- RDF(S)/SPARQL
- PL/FOL
- OWL
- DL Extensions
- OWL Reasoning
- Scalability
- OWL in practice
- Practical Topics
Motivation

• Semantic Web needs
  • Meaningful, minimally redundant, richly axiomatized ontologies
  • Correct reasoning
  • Complete reasoning
  • (tractable reasoning)

• Development of several complex reasoning systems, using a lot of magic besides standard tableaux algorithms
  • Racer
  • Others: FaCT++. Hermit, Pellet. …
Racer

RACER

Renamed ABox And Concept Expression Reasoner

succeeding

RACE

Reasoner for ABoxes and Concept Expressions
People behind Racer

• Ralf Möller (TU Hamburg-Harburg)
• Michael Wessel (AI center, SRI International)
• Volker Haarslev (Concordia University)
• Kay Hidde

10 years later …?
Racer history

• 90es: RACE -> RACER
  • research project at University of Hamburg by Möller/Haarslev
• 2004: Racer Systems GmbH & CO. KG was founded
• 2005: Cooperation with Franz Inc.
  • Dynamic object-oriented development tools
  • Official distributor for America/Asia
  • Racer 1.9
• 2009: Racer 2.0 on the way; integration with AllegroGraph
• 2011: Plugin for Protégé
Racer 2.0

- Supported description logic SHIQ
  - ALC plus … ?
- In addition
  - concrete domains
  - More role constructors (=> SRIQ)
- Only approximations for nominals
Editions

- RacerPro
- RacerPorter
- RacerPlus
- RaerMaster
Editions: RacerPRO

• Kernel reasoner

• Applications as
  • Semantic Web reasoning system and information repository
    • OWL: Consistency checks, subclass-relationships, incremental QA
  • Description Logic reasoning system
    • Reasoning services for multiple TBoxes/ABoxes
    • Two standards for interconnection (among others)
      • HTTP/DIG (Description Logic Implementation Group)
      • nRQL (New Racer Query Language)
  • DL extensions with specific relational algebras
    • e.g. spatial and temporal relations

• Free preview until January 31, 2013 (hurry up!)
Editions: RacerPRO
Editions: RacerPorter

- Graphical user interface for RacerPro
- Uses TCP/IP connection to connect to one (or more) RacerPro instances
- Features (among many others)
  - Loading/Saving ontologies
  - Inspection of axioms
  - Visualization of (parts of) ontologies
- RacerPorter is free of charge
  - Included in the distribution of RacerPro
Editions: RacerPorter
Editions: RacerPlus

- RacerPro + RacerPorter in one application
- Avoids network overhead
- Maximum performance on a single workstation
Editions: RacerMaster

- RacerPro as a object code library
- Using Common Lisp, you can directly access Racer technology without need of a server running
- Available for many Lisp environments
License Fees

• Free trial
• Educational license: free for 180 days
• Standard license: 2.990 (32 Bit) – 9.990 Euro (64 Bit)
• Limited (Desktop) license: 990 Euro
• Site license: … 25.000+ Euro
Link for Racer

http://www.racer-systems.com
Preparing to work with Racer

• Racer is developed in LISP!

=> The (Shell-)interface makes heavy use of parentheses!

• General structure

  • (command parameter1 parameter2 … parameterN)
    is different to

  • (command (parameter1 parameter2) … )
Some terminology for Racer

- **Tell/told**: directly asserted facts or axioms
- **What else??**
Racer Shell: descriptions

• (Complex) concept descriptions
  • *bottom*, *top*, C, (not C), (and C D ...), (or C D ...),
    (some R D), (all R D), (at-most n R C), (at-least n R C),
    (exactly n R C)
  • Example: (and human (some has-gender (or female male)))

• (Complex) role descriptions
  • R, (inv R)
  • Example: (inv hasFather)
Racer Shell: declaring TBox assertions

- Concept subsumption
  - (implies C D)
  - Example: (implies woman (and person (some has-gender female)))

- Concept equivalence
  - (equivalent C D)
  - Example: (equivalent mother (and woman parent))

- Concept disjointness
  - (disjoint C D)
  - Example: (disjoint female male)
Racer Shell: declaring ABox assertions

• Concept assertion
  • (instance a C)
  • Example: (instance luis person)

• Role assertion
  • (related a b R)
  • Example: (related luis ronaldo has-child)

• Individual equality
  • (same-as a b)
  • Example: (same-as lisa smartest-girl-in-springfield)
Task

• Describe our usual Simpsons family ontology with RacerPro. using TBox and ABox axioms.
Racer Shell: retrieving knowledge I

- **Instance check**
  - (individual-instance? a C)
  - Example: (instance marge person)

- **Individual realization**
  - (individual-types a)
  - Example: (individual-types a)

- **Concept realization**
  - (concept-instances C)
  - Example: (concept-instances person)
• nRQL
  • A very powerful query language
  • Allows formulation of conjunctive queries
  • Similarities to SPARQL
  • Concept query atoms: (?X C)
  • Role query atoms: (?X ?Y R)
  • Complex query
    • (retrieve (?X1 … ?Xn) (?X1 C1) … (?X1 ?Xj R) )

• Please note:
  1. All the concept and role descriptions can be complex
  2. This all involves real (=complex) reasoning in the background, opposed to SPARQL
Racer Shell: retrieving knowledge II

- nRQL
  - There is much more about nRQL, just check the manual
  - (more than 50% of the manual are about NRQL!)
Task

• Solve the following puzzle:

• There are three chairs on a stage
• On the left chair sits a woman
• On the right chair sits a man
• Assuming that any person sits on the middle chair, does a man sit next to a woman on the stage?

Use description logics to model the scenario and then Racer to solve the puzzle!
Task
Task

(equivalent Person (or Male Female))
(disjoint Male Female)

(instance a Female)
(instance c Male)
(instance b Person)
(related a b nextto)
(related b a nextto)
(related b c nextto)
(related c b nextto)
(instance s Stage)
(related a s onstage)
(related b s onstage)
(related c s onstage)

(concept-instances (and Stage (some (inv onstage) (and Male (some nextto Female))))))
Task

- It works, because all models agree
- Please note that we cannot retrieve “the” man sitting next to a woman.
Task

• Solve the following Sudoku:

Use description logics to model the scenario and then Racer to solve the puzzle!
Task

• General idea for one group of numbers:

(set-unique-name-assumption t)
(equivalent Field (or N1 N2 N3 N4))
(disjoint N1 N2 N3 N4)
(equivalent Group (and (exactly 1 contains N1) (exactly 1 contains N2) (exactly 1 contains N3) (exactly 1 contains N4) (exactly 4 contains Field)))
(instance f11 N1)
(instance f12 N2)
(instance f13 Field)
(instance f14 N4)
(instance g1 Group)
(related g1 f11 contains)
(related g1 f12 contains)
(related g1 f13 contains)
(related g1 f14 contains)

(individual-types f13) => N2(f13)
## Task

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

- This works reasonably well for 4*4, how about 9*9, or even bigger? You will see in your homework!
Be careful

• Although solving these puzzles with description logics and Racer sounds nice, it usually means that you have to truly understand and often solve the actual problem beforehand!
RacerPro: explanation

• Often it is interesting to find out why a knowledge base is inconsistent
• Inspection of (annotated) tableaux can help to figure out reasons
• Command: (check-abox-coherence)
• Example session with explanation …
  • (implies male person)
  • (instance bart male)
  • (instance bart (not person))
  • (check-abox-coherence)
RacerPro: event recognition

• Assumption:
  • Assertions only hold for a period of time

• Idea:
  • Identify patterns in the set of changing (usually ABox) assertions

• Example session with event recognition …
RacerPro: what else?

- Simple retraction/update mechanism for ABox assertions
- Abduction support by using rules
- Reasoning with triples on external memory
- Publish-Subscribe Mechanism
- OWL interface
- Non-standard inferences
  - LCS, MSC-k
RacerPorter

• All the other tabs are quite intuitive
• Special attention to
  • ABox graph (snapshots of views on the ABox)
  • Query IO (formatted result of the bindings)
Programming Racer

- LRacer for Lisp
- JRacer for Java
JRacer – very basic example

```java
import com.racersystems.jracer.*;

...  
RacerClient racer = new RacerClient("127.0.0.1",8088);
racer.openConnection();  
racer.sendRaw("(instance lisa Person)");
...
res2 =racer.racerAnswerQuery$("(?x)", "(?x Person)");

for (RacerList<RacerList<RacerSymbol>> bindings : res2)
    for (RacerList<RacerSymbol> binding : bindings)
        System.out.println(binding.value.get(0)+"="+ binding.value.get(1));
```
Examination

• Oral examination
• How about having the examination in the last week of the lecture period (11.02.2013)?
Next week?

- Semantic Web topic of students' choice
  or
- Quiz/competition about the lecture so far
  - with small prices to win
  - Might help to reduce the time for exam preparation later on

Which one do you prefer?
Acknowledgements

• Michael Wessel: The RacerPro Environment for Lisp-based Semantic Web Applications