Knowledge Representation for Computers

- **The main problem with Semantic Networks is their appeal to human intuitions rather than to logic-based definitions.**
- **Human intuitions may differ according to cultural heritage, environmental conditioning, professional education, application context etc.**

  => misunderstandings, wrong inferences, inconsistent knowledge
- **Knowledge-based systems and large-scale knowledge management require that computers process knowledge unambiguously without human knowledge interpretation:**
  - **Assistance systems**
  - **Corporate knowledge management**
  - **Knowledge integration in a globalized economy**
  - **The Semantic Web**

1

Description Logics for Knowledge Representation

- **DLs are a family of knowledge-representation formalisms**
- **Decidable subset of FOL**
- **Object-centered, roles and features (binary relations)**
- **Necessary vs. sufficient attributes**
- **Inference services**
  - subsumption check
  - consistency check
  - classification
  - abstraction
  - default reasoning
  - spatial and temporal reasoning
- **Guaranteed correctness, completeness, and decidability properties**
- **Highly optimized implementations (e.g. RACER)**
- **Provides inferences for Semantic Web language OWL**

Development of Description Logics

There exist several experimental and commercial developments of DLs, among them

- KL-ONE first conception of a DL (1985)
- CLASSIC commercial implementation by AT&T
- LOOM experimental system at USC
- FaCT experimental and commercial system (Horrocks, Manchester)
- RACER experimental and commercial system (Haarslev & Moeller)

There is still active research on DLs:

- **Extending the expressivity of concept languages**
- **Decidability and tractability of inference services**
- **Integration of predicates over concrete domains (e.g. numbers)**
- **Adapting to Semantic Web requirements**
- **Highly optimized implementations**
- **Developing new inference services (e.g. for scene interpretation)**

Example of DL Expressions

TBox contains concept definitions and axioms ("terminology")

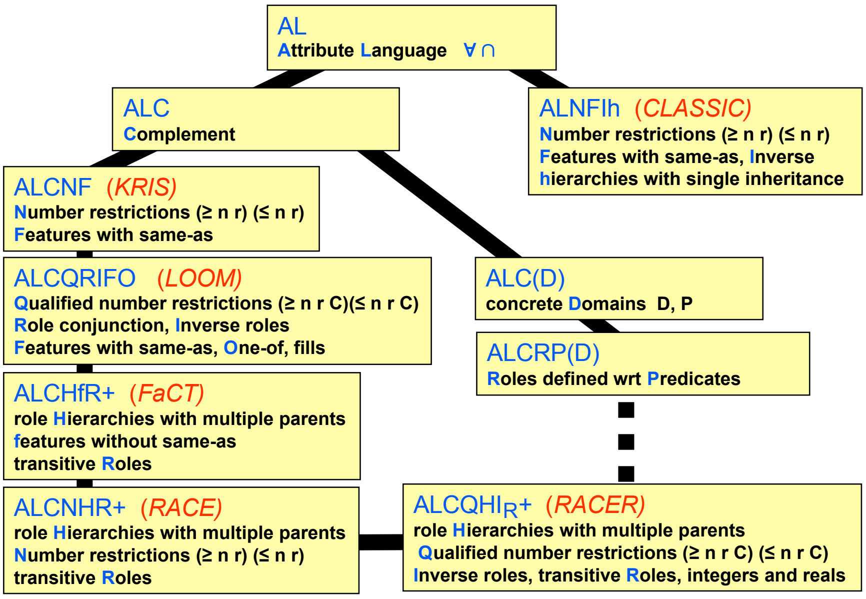
```
(equivalent celebrated-clean-car
        (and car
                (all has-drive (or electric-drive  hydrogen-drive))
                (not noisy-car)
                (has-speed (≥ 100))
                (at-least 1 possess award)))
```

ABox contains assertions about individuals based on TBox

```
(instance car013 (and car (not noisy-car)))
(related car013 best-car-award-2007 possess)
(instance best-car-award-2007 award)
```

4

Family of Description Logics



5

Description Logics Literature

The Description Logic Handbook
F. Baader, D. Calvanese, D. MacGuinness, D. Nardi, P. Patel-Schneider (eds.)
Cambridge University Press, 2003

OWL Web Ontology Language Guide
W3C Recommendation 10 February 2004
http://www.w3.org/TR/2004/REC-owl-guide-20040210

RacerPro Reference Manual Version 1.9
Racer Systems GmbH&Co. KG, December 8, 2005
http://www.racer-systems.com/products/racerpro/manual.phtml
=> RacerPro-User-Guide.pdf
=> RacerPro-Reference-Manual.pdf

6

The RACER DL-System

- **Highly expressive DL** $ALCQHI_{R^+}$
  - Role hierarchies with multiple parents
  - Qualified number restrictions (≥ n r C) (≤ n r C),
  - Inverse roles, transitive Roles
  - Integers and reals
- **Available as product RacerPro (http://www.racer-systems.com)**
  - Reasoner for the Semantic Web languages OWL/RDF
  - Evaluation copy for university research
  - Comprehensive manual
- **Developed in the Cognitive Systems Laboratory at Hamburg University**

  Research applications in
  - information management: TV-Assistant
  - content-based image retrieval
  - scene interpretation

RACER Concept Language

C concept term
CN concept name
R role term
RN role name

C -> *CN*
top
bottom
(not *C*)
(and *C1 ... Cn*)
(or *C1 ... Cn*)
(some *R C*)
(all *R C*)
(at-least *n R*)
(at-most *n R*)
(exactly *n R*)
(at-least *n R C*)
(at-most *n R C*)
(exactly *n R C*)
CDC

concept definition
(define-concept *CN C*)

concept axioms
(define-primitive-concept *CN C*)
(implies *C1 C2*)
(equivalent *C1 C2*)
(disjoint *C1 ... Cn*)

roles
R -> *RN*
(*RN role-props*)

role-props ->
((:transitive t)
(:feature t)
(:symmetric t)
(:reflexive t)
(:inverse RN)
(:domain C)
(:range C))

concept-domain concepts
AN *attribute name*
CDC -> (a *AN*)
(an *AN*)
(no *AN*)
(min *AN integer*)
(max *AN integer*)
(> *aexpr aexpr*)
(>= *aexpr aexpr*)
(< *aexpr aexpr*)
(<= *aexpr aexpr*)
(= *aexpr aexpr*)
aexpr -> *AN*
real
(+ *aexpr1 aexpr1**)
aexpr1
aexpr1 -> *real*
AN
(* *real AN*)

Primitive and Defined Concepts

Concept expressions of a DL describe classes of entities in terms of properties (unary relations) and roles (binary relations).

Main building blocks are primitive oder defined concepts.

Primitive concepts: concept => satisfied properties and relations

satisfied properties and relations are <u>necessary</u> conditions for an object to belong to a class

Defined concepts: concept <=> satisfied properties and relations

satisfied properties and relations are <u>necessary and sufficient</u> conditions for an object to belong to a class

Primitive concept "person":
(implies person (and mammal (some has-gender (or female male))))

Defined concept "parent":
(equivalent parent (and person (some has-child person)))

9

Roles

Roles describe binary relations beween objects

Features	*functional roles with at most one role filler for each individual*
Transitive roles	*roles are transitively closed for a transitive role r:* $r(i_1, i_2) \wedge r(i_2, i_3) \Rightarrow r(i_1, i_3)$
Role hierarchies	*Specification of super- and subroles*
Domain restriction	*simulated in RACER by* (implies (some RN *top*) C)
Range restriction	*simulated in RACER by* (implies *top* (all RN C))

In RACER there are only primitive roles, membership to a role cannot be deduced in general.

Exceptions: transitive roles, deductions in role hierarchies

10

Example of a TBox

(signature :atomic-concepts (person human female male woman man parent
mother father grandmother aunt uncle sister brother)
:roles ((has-child :parent has-descendant)
(has-descendant :transitive t)
(has-sibling)
(has-sister :parent has-sibling)
(has-brother :parent has-sibling)
(has-gender :feature t)))

Signature of TBox

(implies *top* (all has-child person))
(implies (some has-child *top*) parent)
(implies (some has-sibling *top*) (or brother sister))
(implies *top* (all has-sibling (or sister brother)))
(implies *top* (all has-sister (some has-gender female)))
(implies *top* (all has-brother (some has-gender male)))

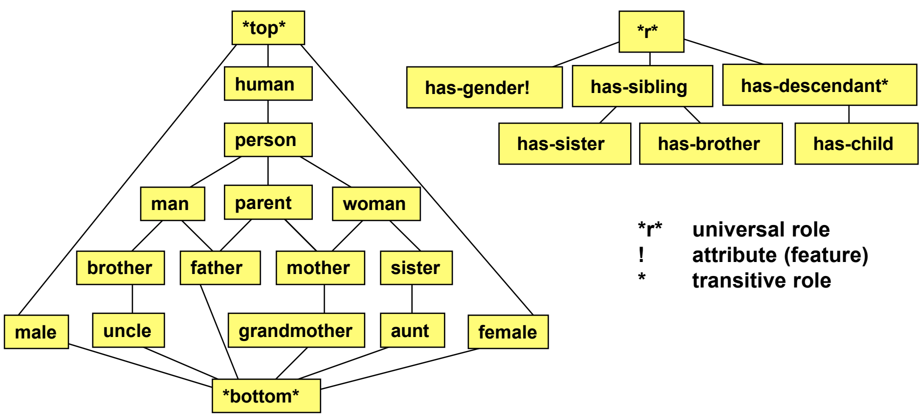
Domain and range restrictions for roles

(implies person (and human (some has-gender (or female male))))
(disjoint female male)
(implies woman (and person (some has-gender female)))
(implies man (and person (some has-gender male)))
(equivalent parent (and person (some has-child person)))
(equivalent mother (and woman parent))
(equivalent father (and man parent))
(equivalent grandmother (and mother (some has-child (some has-child person))))
(equivalent aunt (and woman (some has-sibling parent)))
(equivalent uncle (and man (some has-sibling parent)))
(equivalent brother (and man (some has-sibling person)))
(equivalent sister (and woman (some has-sibling person)))

Axioms for relating concept names

11

Concept and Role Hierachies Implied by TBox



r universal role
! attribute (feature)
* transitive role

12

TBox Inferences

A DL system offers several inference services. At the core is a consistency test:

$$C \overset{?}{\models} \text{*bottom*} \quad \text{(the empty concept)}$$

Example: (and (at-least 1 has-child) (at-most 0 has-child)) $\models$ *bottom*

Consistency checking is the basis for several other inference services:

- **subsumption**
  (implies C1 C2) <=> (and C1 (not C2)) $\models$ *bottom*
- **classification of a concept expression**
  searches the existing concept hierarchy for the most special concept which subsumes the concept expression

13

Formal Semantics of Concept Expressions

D	**Set of all possible domain objects**
$E[C] \subseteq D$	**Extension of a concept expression C (represents meaning of C)**
$E[RN] \subseteq D \times D$	**Extension of a role RN (represents meaning of RN)**

Formal semantics of concept operations:

$E[\text{*bottom*}] = \{ \}$

$E[\text{(and } C_1 \ldots C_n)] = E[C_1] \cap \ldots \cap E[C_n]$

$E[\text{(or } C_1 \ldots C_n)] = E[C_1] \cup \ldots \cup E[C_n]$

$E[\text{(all RN C)}] = \{x \mid \forall(x, y) \in E[RN] \Rightarrow y \in E[C]\}$

$E[\text{(some RN C)}] = \{x \mid \exists(x, y) \in E[RN] \wedge y \in E[C]\}$

14

Concrete Domain Example

(from RacerPro-User-Guide)

```
...
(signature
:atomic-concepts (... teenager)
:roles (...)
:attributes ((integer age)
(real temperature-celsius)
(real temperature-fahrenheit)))
...
(equivalent teenager (and human (min age 16)))
(equivalent old-teenager (and human (min age 18)))
(equivalent human-with-fever (and human (>= temperature-celsius 38.5))
(equivalent seriously-ill-human (and human (>= temperature-celsius 42.0)))
...
```

RACER concludes that
- the concept "old-teenager" is subsumed by "teenager"
- the concept "seriously-ill" is subsumed by "human-with-fever"

15

ABox of a Description Logic System

TBox = terminological knowledge (concepts and roles)
ABox = assertional knowledge (facts)

An ABox contains:

- concept assertions (instance IN C)
  individual IN is instance of a concept expression C
- role assertions (related IN_1 IN_2 RN)
  individual IN_1 is related to IN_2 by role RN

- An ABox always refers to a particular TBox.
- An ABox requires unique names.
- ABox facts are assumed to be incomplete (OWA).

  OWA = Open World Assumption
  (new facts may be added, hence inferences are restricted)
  CWA = Closed World Assumption
  (inference assumes that all facts are in ABox)

16

ABox Inferences

ABox inferences = inferring facts about ABox individuals

Typical queries:

- consistency — *Is ABox consistent?*
- retrieval — *Which individuals satisfy a concept expression?*
- classification — *What are the most special concept names which describe an individual?*

ABox consistency checking is in general more complicated than TBox consistency checking.

ABox consistent <=> there exists a "model" for ABox and TBox

All ABox inferences are based on the ABox consistency check.

17

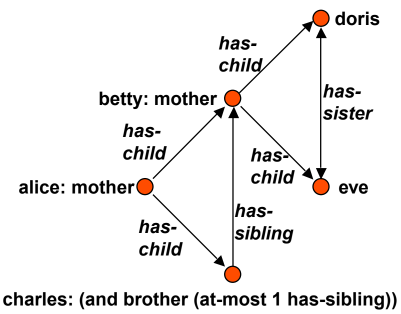
Example of ABox Queries

Contents of ABox

(instance alice mother)
(related alice betty has-child)
(related alice charles has-child)

(instance betty mother)
(related betty doris has-child)
(related betty eve has-child)

(instance charles brother)
(related charles betty has-sibling)
(instance charles (at-most 1 has-sibling))
(related doris eve has-sister)
(related eve doris has-sister)



Questions and answers

(individual-instance? doris woman) T	*Is* doris *instance of the concept* woman*?*
(individual-types eve) ((sister) (woman) (person) (human) (*top*))	*Of which concept names is* eve *an instance?*
(individual-fillers alice has-descendant) (doris eve charles betty)	*What are the descendants of* eve*?*
(concept-instances sister) (doris betty eve)	*Which instances has the concept* sister*?*

18

Abstraction with Description Logics

Abstraction = omission of properties or relations, extending a concept, generalization

Examples:

- **Superordinate concept name of a concept expression (= concept classification)**

  (and person (some has-size tall)) → person

- **Generalization of concept expressions**

  (and (some has-occupation professor) (at-least 3 has-child))

  ⬇

  (and (some has-occupation civil-servant) (at-least 1 has-child))

- **Concept expression which subsumes several individuals**

  1. classify individuals
  2. determine least common subsumer (LCS)
     - for RACER: trivial solution in terms of (OR C_1 ... C_n)
     - for DLs without OR: special abstraction operator LCS

19

ABox Retrieval by TV Assistant

TV assistant selects program items based on conceptual description of user preferences represented in DL - prototype developed by LKI

ARD	ZDF	RTL	SAT.1
20.15	20.15	20.15	20.00
Fußball-WM	China heute	Galactica	Dragonheart
21.45	21.15	21.35	21.00
Sissi	Wetten, daß...	Braveheart	Stirb langsam 2
22.30	22.00	22.45	22.15
Tagesthemen	Heute	Sexshow	Rolling Stones
23.00	22.30	23.30	23.00
The Rock	Terminator 2	Speed	Alien

user selects examples →

Braveheart

Stirb langsam 2

Terminator 2

system determines similarity of contents

↓

- Action/Horror
- Cinema Highlights
- Known actors
- New films

← system proposes program items with similar contents

ARD	N3	RTL	PRO 7
20.15	20.15	20.15	20.00
Schatzinsel	Eiskunstlauf	Goldfinger	Psycho II
21.45	21.00	21.30	21.00
Lindenstraße	Sterbehilfe	Dallas	Deep Impact
22.30	22.00	22.15	22.15
Tagesthemen	Extra 3	Titanic	Killerwale
23.00	22.30	23.30	23.00
Armageddon	Achterbahn	Robocop	Arabella

20

Useful Extensions

Feature chains: (compose F1 ... Fn)   short: (F1 o ... o Fn)

The composition of features F1 ... Fn is a feature whose fillers are the fillers of Fn applied to the fillers of Fn-1 applied to ... the fillers of F1.

Feature (chain) agreement: (same-as F1 F2) short: (= F1 F2)

Concept expression for elements which possess the same fillers for features F1 and F2.

Example: (same-as (has-plate o has-colour) (has-saucer o has-colour))

Requirement for a cover that plate and saucer have the same colour

Cannot be combined with expressive DLs without jeopardising decidability!

Instead of features, also roles may be composed, and a subset operator relates role-fillers similar to same-as for features.

Role-value map: (subset R1 R2)

Concept expression of elements where the fillers of role R1 are a subset of the fillers of role R2.

Causes undecidability even in DLs with low expressivity (e.g. CLASSIC).

21

RACER Query Language

Interface language for retrieving patterns from an ABox

Basic retrieval command: (retrieve <list-of-objects> <query-body>)

Example:
Smart-room environment, ABox contains description of table top, system wants to find out where a cover has been laid.

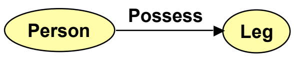
```
(retrieve (?x ?y ?z) (and   (?x plate)
                            (?y saucer)
                            (?z cup)
                            (?x ?y near)
                            (?z ?y on)))

→   (((?x plate1) (?y saucer1) (?z cup1))
     ((?x plate2) (?y saucer2) (?z cup2)))
```

Note: Query language retrieval commands allow to retrieve patterns for which no individuals have been introduced.

22

Description Logics vs. Semantic Networks



Unclear semantics:

- A Specification of person concept restricted by role possess?
- B Specification of role possess restricted by domain and range concepts?
- C Specification of leg concept restricted by role possess?

+ in all cases unclear cardinalities

With DLs:

(and person (some possess leg))
"objects which are persons and each of which possesses at least one leg"

(and person (all possess leg))
"objects which are persons and each of which only possesses legs"

Each DL concept expression describes a set of objects!

Description Logics vs. Frames (1)

ID:	squirrel
ISA:	rodent
colour:	brown
weight:	[300 500]

Unclear semantics (inference directions) of frames:

- *A "All squirrels are rodents with colour brown and weight between 300 and 500"*
- *B "All rodents with colour brown and weight between 300 and 500 are squirrels"*

With DLs:

A (implies squirrel (and rodent (all colour brown)
(all weight (≥ 300))
(all weight (≤ 500))))

B (equivalent squirrel (and rodent (all colour brown)
(all weight (≥ 300))
(all weight (≤ 500))))

Description Logics vs. Frames (2)

ID:	squirrel
ISA:	rodent
colour:	brown
weight:	[300 500]

ID:	jackie
INSTANCE:	squirrel
colour:	pink
weight:	350
age:	7

Frame inheritance mechanisms may allow overwriting of properties.

DL TBox:

(implies squirrel (and rodent (all colour brown)
(all weight (≥ 300))
(all weight (≤ 500))))

DL ABox:

(instance jackie squirrel)
(related jackie pink colour)
(related jackie 350 weight)
(relared jackie 7 age)

DL system determines:
ABox is inconsistent with TBox