1. DLs for reasoning about (database) conceptual models
2. DLs for ontology engineering
State of the art systems are highly optimised versions of the tableau algorithm for $SHIQ$: FaCT and Racer

Optimisations implemented in FaCT and Racer

- are known from other “logic implementations”,
- include minimisation of subsumption tests for the classification of all concepts defined in the TBox
- include SAT-heuristics, e.g., semantic branching or Davis-Putnam,
- include dependency-directed backtracking (non-determinism is implemented through search with backtracking)
- include strong pre-processing of TBox (localisation/absorption of CEs)
- include optimised double-blocking conditions (otherwise, blocking occurs dramatically too late)
Optimised blocking for the $SHIQ$ tableau algorithm

**Acyclic blocking:**

\[ B_3.c \leq (n - 1) S\text{-succ with } C \]
\[ B_4.a \geq m T\text{-succ with } E \]

\[ \Rightarrow B_3.c \leq (n - 1) S\text{-succ with } C \]
\[ \Rightarrow B_4.a \geq m T\text{-succ with } E \]

\[ \Rightarrow B_3.b \Rightarrow C \]
\[ \Rightarrow B_4.b \Rightarrow E \]

\[ \Rightarrow B_3.a \text{ no } Inv(S) \]
\[ \Rightarrow B_4.b \text{ Inv}(T) \]

\[ w' \Rightarrow B_3.b (\leq n S C) \]
\[ B_4. (\geq m T E) \]

\[ \Rightarrow B_2. \forall S.C \]

\[ \Rightarrow B_2.a \Rightarrow B_2.b \forall R.C \]

\[ v \Rightarrow B_2.a C, \Rightarrow \forall R.C \]

\[ \Rightarrow B_1 L(w) \subseteq L(w') \]
Cyclic blocking:

> B6: $\forall S.C$
> $w' \Rightarrow B7: (\leq n \ T \ E)$

$\Rightarrow_{B6.a} C, \Rightarrow_{B6.b} \forall R.C$

$\Rightarrow_{B7} \neg E$

$\Rightarrow_{B7. no Inv(T)} \Rightarrow_{B6} Inv(S)$

B6.b: $Inv(R), R \sqsubseteq S, Trans(R)$

B8: $U$

$\Rightarrow_{B6.a} \Rightarrow_{B5} L(w) \subseteq L(w')$

$\Rightarrow_{B8} \neg F$
DLs for conceptual modelling

**Idea:** Translate EER/UML database conceptual model $E$ into $SHIQ$ TBox $T_E$:

- each model of $E$ is a model of $T_E$ and vice versa.

use DL system to infer inconsistencies and implicit subsumption relations useful for integrating databases/in datawarehousing

**Technicalities:** If only binary relations are present in $E$, translation into $SHIQ$ is rather straightforward

If $n$-ary relations are present in $E$, translation into $SHIQ$ requires reification of relations
Example:

- Client
- Banquet
- Celebration
- Restaurant
- Menu
- Dish
- Person
- Firm
- Place in
- Served in
- Order
- Related
- Offer m
- Offer s
- Characteristic
- Speciality
- Related
- Offered in
- Offered by
- Is comprised
- Me of
- Re of
- Characteristic
- Spe of

TU Dresden
Germany
**SHIQ TBox for the restaurant UML**

client \( \equiv (\geq 1 \text{order}. \top) \sqcap (\forall \text{order}. \text{banquet}) \)

banquet \( \equiv (= 1 \text{order}^-. \top) \sqcap (\forall \text{order}^- . \text{client}) \sqcap \\
(= 1 \text{placed-in}. \top) \sqcap (\forall \text{placed-in} . \text{restaurant}) \sqcap \\
(= 1 \text{served-in}^- . \top) \sqcap (\forall \text{served-in}^- . \text{menu}) \\
(= 1 \text{related-to}. \top) \sqcap (\forall \text{related-to} . \text{celebration}) \)

celebration \( \equiv (= 1 \text{related}^- . \top) \sqcap (\forall \text{related-to}^- . \text{banquet}) \)

person \( \sqsubseteq \) client \( \sqcap \neg \text{firm} \)

firm \( \sqsubseteq \) client \( \sqcap \neg \text{person} \)

\( \vdots \)

\( \vdots \)
System demo of i.com,

a tool for intelligent conceptual modeling,

implemented by Enrico Franconi and Gary Ng
using FaCT by Ian Horrocks
Ontologies are formalization of a conceptualization
fix the relevant concepts of an application domain (like a TBox)
are widely used in enterprise knowledge management systems
will widely be used in the semantic web

Useful ontologies

- should be consistent and not redundant
- are built possibly by several knowledge engineers
- evolve over time
- are possibly integrated with other ontologies
DLs for ontology engineering

An ontology language

- should have a clear semantics \(\iff\) logic
- should allow for tractable inference services to support the design, evolution, maintenance, and integration of ontologies

DAML+OIL, OWL, and OIL are DL-based ontology language

System demo of OilEd,

a tool for ontology engineering,

implemented by Sean Bechhofer and Gary Ng

using FaCT by Ian Horrocks