DAML+OIL and Description Logic Reasoning

Ian Horrocks

horrocks@cs.man.ac.uk

University of Manchester
Manchester, UK

Talk Outline

- The Semantic Web and DAML+OIL
- Description Logics and Reasoning
  - Reasoning techniques
  - Implementing DL systems
- Research Challenges
- Summary

The Semantic Web and DAML+OIL

Semantic Web Ontology Languages

US **DAML** programme (in cooperation with W3C and a cast of thousands) aim to develop so-called **Semantic Web**

- ☞ Most existing Web resources only human understandable
  - Markup (HTML) provides **rendering information**
  - Textual/graphical information for **human consumption**
- ☞ Semantic Web aims at machine understandability
  - **Semantic** markup will be added to web resources
  - Markup will use **Ontologies** for shared understanding
- ☞ Requirement for a suitable ontology language
  - Compatible with existing Web standards (XML, RDF, RDFS)
  - Captures common KR idioms
  - Formally specified and of adequate expressive power
  - Can provide reasoning support
- ☞ DAML-ONT language developed to meet these requirements

OIL and DAML+OIL

Meanwhile, somewhere in darkest Europe...

- **OIL** language already developed to meet similar requirements
  - Extends existing Web standards (XML, RDF, RDFS)
  - Intuitive (frame) syntax plus high expressive power
  - Well defined semantics via mapping to $\mathcal{SHIQ}$ DL
  - Can use DL systems to reason with OIL ontologies
- Two efforts merged to produce single language, **DAML+OIL**
- Detailed specification agreed by **Joint EU/US Committee on Agent Markup Languages**
- Proposed W3C Ontology Language WG will take DAML+OIL as starting point (?)

DAML+OIL Language Overview

DAML+OIL is an **ontology** language

- Describes **structure** of the domain (i.e., a Tbox)
  - RDF used to describe specific **instances** (i.e., an Abox)
- Structure described in terms of **classes** (concepts) and **properties** (roles)
- Ontology consists of set of **axioms**
  - E.g., asserting class subsumption/equivalence
- Classes can be names or **expressions**
  - Various **constructors** provided for building class expressions
- **Expressive power** determined by
  - Kinds of axiom supported
  - Kinds of class (and property) constructor supported

DAML+OIL Overview: Class Constructors

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human $\sqcap$ Male
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor $\sqcup$ Lawyer
complementOf	$\neg C$	$\neg$Male
oneOf	$\{x_1 \ldots x_n\}$	$\{$john, mary$\}$
toClass	$\forall P.C$	$\forall$hasChild.Doctor
hasClass	$\exists P.C$	$\exists$hasChild.Lawyer
hasValue	$\exists P.\{x\}$	$\exists$citizenOf.$\{$USA$\}$
minCardinalityQ	$\geqslant n P.C$	$\geqslant 2$hasChild.Lawyer
maxCardinalityQ	$\leqslant n P.C$	$\leqslant 1$hasChild.Male
cardinalityQ	$= n\, P.C$	$=1$ hasParent.Female

- XMLS **datatypes** as well as classes
- Arbitrarily complex **nesting** of constructors
  - E.g., $\forall$hasChild.(Doctor $\sqcup$ $\exists$hasChild.Doctor)

DAML+OIL Overview: Axioms

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
sameClassAs	$C_1 \doteq C_2$	Man $\doteq$ Human $\sqcap$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
samePropertyAs	$P_1 \doteq P_2$	cost $\doteq$ price
sameIndividualAs	$\{x_1\} \doteq \{x_2\}$	{President_Bush} $\doteq$ {G_W_Bush}
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq$ $\neg$Female
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq$ $\neg${peter}
inverseOf	$P_1 \doteq P_2^-$	hasChild $\doteq$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
uniqueProperty	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant 1$hasMother
unambiguousProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant 1$isMotherOf$^-$

☞ Axioms (mostly) **reducible to subClass/PropertyOf**

DAML+OIL

- Is a **Description Logic** (but don't tell anyone)
- More precisely, DAML+OIL is $\mathcal{SHIQ}$
  - Plus **nominals**
  - Plus **datatypes** (simple concrete domains)
  - With RDFS based syntax
- $\mathcal{SHIQ}$/DAML+OIL was not built in a day (or even a year)
  - $\mathcal{SHIQ}$ is based on 15+ years of DL research
- Can use DL reasoning with DAML+OIL
  - Existing $\mathcal{SHIQ}$ implementations support (most of) DAML+OIL

Why Reasoning Services?

Reasoning is important for:

- Ontology **design**
  - Check class consistency and (unexpected) implied relationships
  - Particularly important with large ontologies/multiple authors
- Ontology **integration**
  - Assert inter-ontology relationships
  - Reasoner computes integrated class hierarchy/consistency
- Ontology **deployment**
  - Determine if set of facts are consistent w.r.t. ontology
  - Determine if individuals are instances of ontology classes

"**The Semantic Web needs a logic on top**" (Henry Thompson)

Why Decidable Reasoning?

Set of operators/axioms restricted so that reasoning is **decidable**

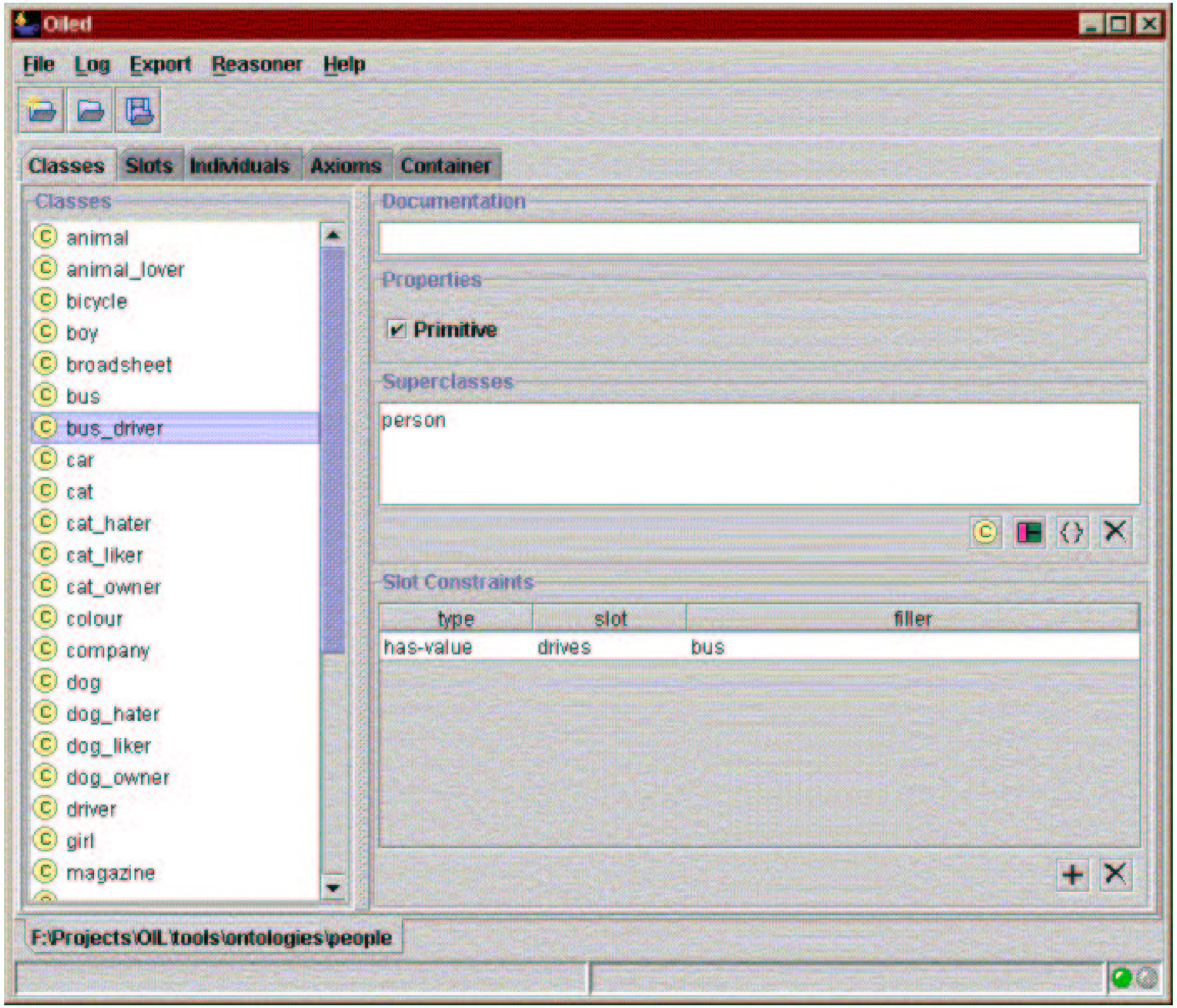
- ☞ Consistent with Semantic Web's **layered architecture**
  - XML provides syntax transport layer
  - RDF provides basic relational language
  - RDFS provides basic ontological primitives
  - DAML+OIL provides (decidable) logical layer
  - Further layers (e.g., **rules**) will extend DAML+OIL
    - ➜ Extensions will almost certainly be **undecidable**
- ☞ Facilitates provision of **reasoning services**
  - Known algorithms
  - Implemented systems
  - Evidence of **empirical tractability**

Reasoning Support for Ontology Design: OilEd

OilEd is a DAML+OIL **ontology editor** with DL reasoning support

- **Frame based** interface (inspired by Protégé)
  - Classes defined by superclass(es) plus slot constraints
- Extended to **clarify semantics** and capture whole language
  - Primitive ($\sqsubseteq$) and defined ($\doteq$) classes
  - Explicit $\exists$ (hasClass), $\forall$ (toClass) and cardinality restrictions
  - Boolean connectives ($\sqcap$, $\sqcup$, $\neg$) and nesting
  - Transitive, symmetrical and functional properties
  - Disjointness, inclusion ($\sqsubseteq$) and equality ($\doteq$) axioms
  - Fake individuals
- **Reasoning support** provided by FaCT system
  - Ontology translated into $\mathcal{SHIQ}$ DL
  - Communicates with FaCT via CORBA interface
  - Indicates inconsistencies and implicit subsumptions

OilEd



Oiled
File Log Export Reasoner Help
Classes Slots Individuals Axioms Container
Classes
animal
animal_lover
bicycle
boy
broadsheet
bus
bus_driver
car
cat
cat_hater
cat_liker
cat_owner
colour
company
dog
dog_hater
dog_liker
dog_owner
driver
girl
magazine
Documentation
Properties
Primitive
Superclasses
person
Slot Constraints
type
slot
filler
has-value
drives
bus
F:\Projects\OIL\tools\ontologies\people

Description Logics and Reasoning

What are Description Logics?

- Based on **concepts** (classes) and **roles**
  - Concepts (classes) are interpreted as sets of objects
  - Roles are interpreted as binary relations on objects
- Descendants of **semantic networks** and **KL-ONE**
- **Decidable fragments** of FOL
  - Many DLs are fragments of L2, C2 or the **Guarded Fragment**
- Closely related to **propositional modal logics**
- Also known as terminological logics, concept languages, etc.
- Key features of DLs are
  - Well defined **semantics** (they are logics)
  - Provision of **inference services**

Short History of Description Logics

Phase 1:

- Incomplete systems (Back, Classic, Loom, . . .)
- Based on **structural algorithms**

Phase 2:

- Development of **tableau algorithms** and complexity results
- Tableau-based systems (Kris, Crack)
- Investigation of optimisation techniques

Phase 3:

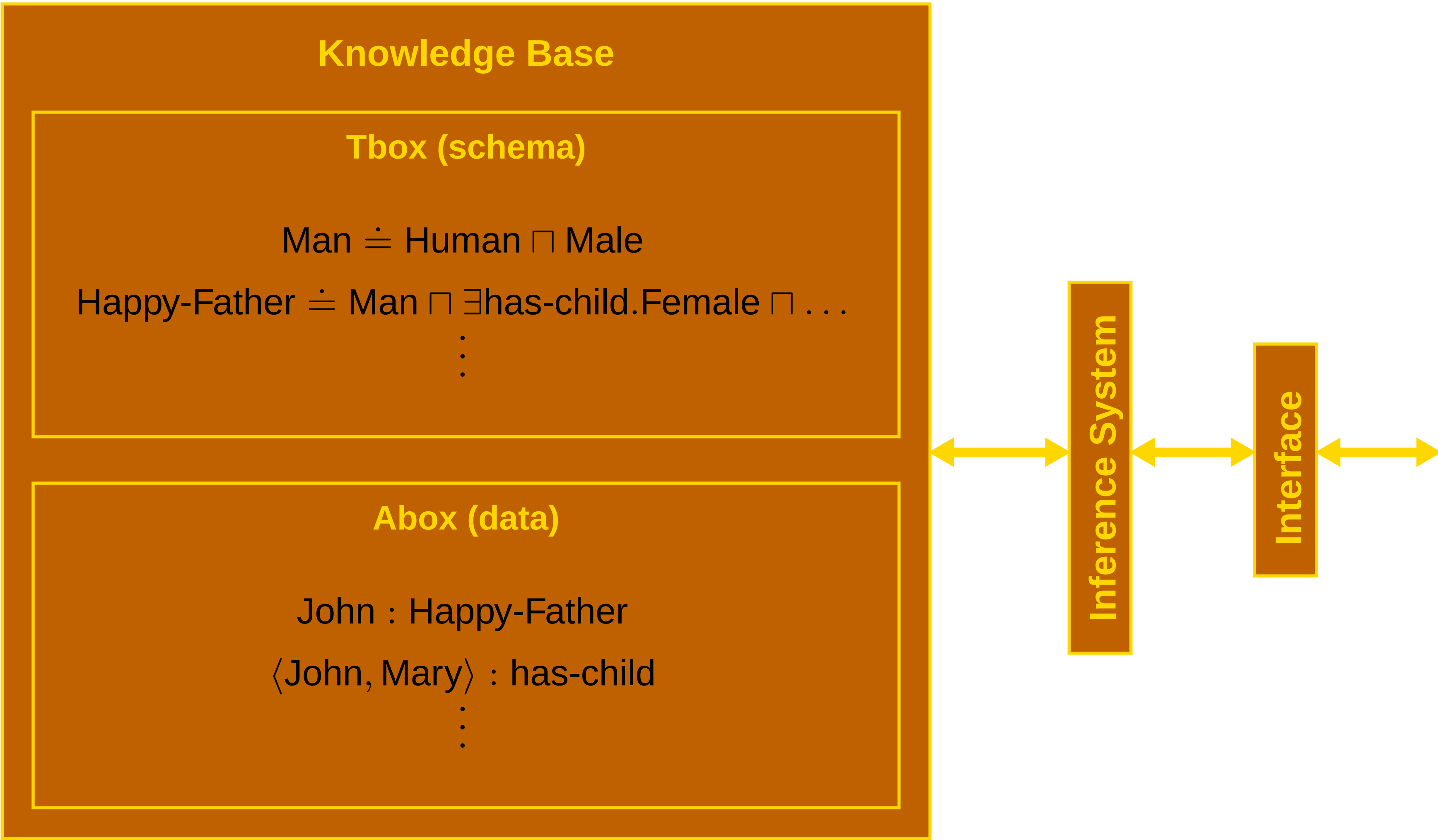
- Tableau algorithms for **very expressive** DLs
- **Highly optimised** tableau systems (FaCT, DLP, Racer)
- Relationship to modal logic and decidable fragments of FOL

Latest Developments

Phase 4:

- Mature **implementations**
- Mainstream **applications** and Tools
  - Databases
    - Consistency of conceptual schemata
    - Schema integration
    - Query subsumption (w.r.t. a conceptual schema)
  - Ontologies and Semantic Web
    - Design and Maintenance
    - Integration
    - Deployment
- **Commercial** implementations
  - Cerebra system from Network Inference Ltd

DL System Architecture



DL Constructors

Particular DLs characterised by **set of constructors** provided for building complex concepts and roles from simpler ones

☞ Usually include at least:

- Conjunction ($\sqcap$), disjunction ($\sqcup$), negation ($\neg$)
- Restricted (guarded) forms of quantification ($\exists$, $\forall$)

☞ This basic DL is known as $\mathcal{ALC}$

DL Syntax and Semantics

Semantics given by **interpretation** $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

Constructor	Syntax	Example	Semantics
atomic concept	A	Human	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
atomic role	R	has-child	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
and for C, D concepts and R a role name			
conjunction	$C \sqcap D$	Human $\sqcap$ Male	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	Doctor $\sqcup$ Lawyer	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	$\neg$Male	$\Delta^{\mathcal{I}} \setminus C$
exists restr.	$\exists R.C$	$\exists$has-child.Male	$\{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
value restr.	$\forall R.C$	$\forall$has-child.Doctor	$\{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \implies y \in C^{\mathcal{I}}\}$

Other DL Constructors

Many different DLs/DL constructors have been investigated, e.g.

Constructor	Syntax	Example	Semantics
number restr.	$\geqslant nR$	$\geqslant 3$ has-child	$\{x \mid \lvert\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}\rvert \geqslant n\}$
	$\leqslant nR$	$\leqslant 1$ has-mother	$\{x \mid \lvert\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}\rvert \leqslant n\}$
inverse role	R^-	has-child$^-$	$\{\langle x, y\rangle \mid \langle y, x\rangle \in R^{\mathcal{I}}\}$
trans. role	R^*	has-child*	$(R^{\mathcal{I}})^*$
concrete domain	$f_1, \ldots, f_n.P$	earns spends $<$	$\{x \mid P(f_1^{\mathcal{I}}, \ldots, f_n^{\mathcal{I}})\}$

⋮

DL Knowledge Base (Tbox)

Terminological part (**Tbox**) is set of axioms describing **structure** of domain

Definition axioms introduce macros/names for concepts

$A \doteq C$, $A \sqsubseteq C$

Father $\doteq$ Man $\sqcap$ $\exists$has-child.Human

Human $\sqsubseteq$ Animal $\sqcap$ Biped

Inclusion (GCI) axioms assert subsumption relations

$C \sqsubseteq D$ (note $C \doteq D$ equivalent to $C \sqsubseteq D$ and $D \sqsubseteq C$)

$\exists$has-degree.Masters $\sqsubseteq$ $\exists$has-degree.Bachelors

DL Knowledge Base (Abox)

Assertional part (**Abox**) is set of axioms describing **concrete situation**

Concept assertions

$a : C$

$\text{John} : \text{Man} \sqcap \exists \text{has-child}.\text{Female}$

Role assertions

$\langle a, b \rangle : R$

$\langle \text{John}, \text{Mary} \rangle : \text{has-child}$

Basic Inference Problems

Subsumption (structure knowledge, compute taxonomy)

$C \sqsubseteq D$? Is $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations?

Subsumption w.r.t. Tbox $\mathcal{T}$

$C \sqsubseteq_{\mathcal{T}} D$? Is $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all models of $\mathcal{T}$?

Consistency

Is C consistent w.r.t. $\mathcal{T}$? Is there a model $\mathcal{I}$ of $\mathcal{T}$ s.t. $C^{\mathcal{I}} \neq \emptyset$?

KB Consistency

Is $\langle \mathcal{T}, \mathcal{A} \rangle$ consistent? Is there a model $\mathcal{I}$ of $\langle \mathcal{T}, \mathcal{A} \rangle$?

Reasoning Techniques

Subsumption and Satisfiability

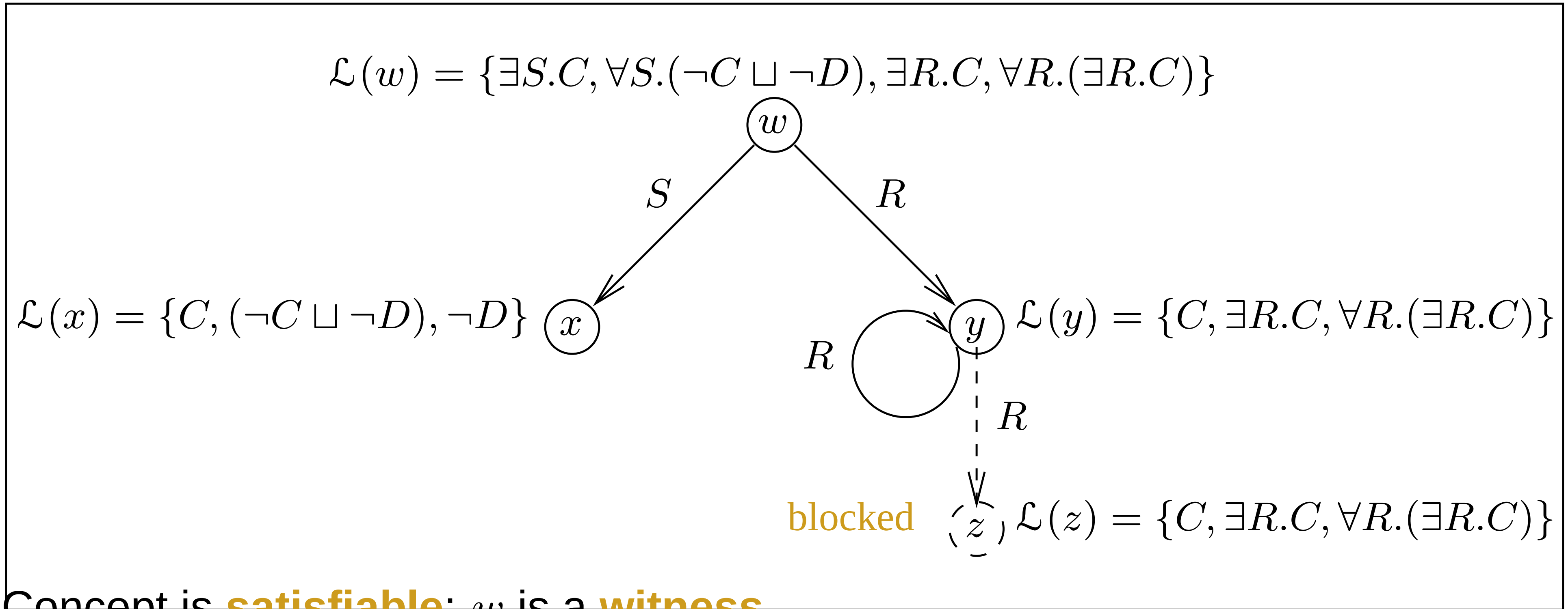
Subsumption transformed into **satisfiability**

Tableaux algorithm used to test satisfiability

- ☞ Try to build **model** (witness) of concept C
- ☞ Model represented by **tree** $\mathbf{T}$
  - Nodes in $\mathbf{T}$ correspond to individuals in model
  - Nodes labeled with sets of subconcepts of C
  - Edges labeled with role names in C
- ☞ Start from **root node** labeled $\{C\}$
- ☞ Apply **expansion rules** to node labels until
  - Rules correspond with language constructs
  - Expansion completed (tree represents valid model)
  - Contradictions prove there is no model
- ☞ Non-deterministic expansion $\longrightarrow$ **search** (e.g., $C \sqcup D$)
- ☞ **Blocking** ensures termination (with expressive DLs)

Tableaux Expansion

Test satisfiability of $\exists S.C \sqcap \forall S.(\neg C \sqcup \neg D) \sqcap \exists R.C \sqcap \forall R.(\exists R.C)\}$ where R is a **transitive** role



Concept is **satisfiable**: w is a **witness**

More Advanced Techniques

Satisfiability w.r.t. a Terminology

- For each GCI $C \sqsubseteq D \in \mathcal{T}$, add $\neg C \sqcup D$ to **every** node label

More expressive DLs

- Basic technique can be extended to deal with
  - Role inclusion axioms (role hierarchy)
  - Number restrictions
  - Inverse roles
  - Concrete domains
  - Aboxes
  - etc.
- Extend **expansion rules** and use more sophisticated **blocking** strategy
- Forest instead of Tree (for Aboxes)

Implementing DL Systems

Naive Implementations

Problems include:

- ☞ **Space** usage
  - Storage required for tableaux datastructures
  - Rarely a serious problem in practice
  - But problems can arise with inverse roles and cyclical KBs
- ☞ **Time** usage
  - Search required due to non-deterministic expansion
  - **Serious** problem in practice
  - Mitigated by:
    - ➜ Careful **choice of algorithm**
    - ➜ Highly **optimised implementation**

Careful Choice of Algorithm

- **Transitive roles** instead of transitive closure
  - Deterministic expansion of $\exists R.C$, even when $R \in \mathbf{R}_+$
  - (Relatively) simple blocking conditions
  - Cycles **always** represent (part of) valid cyclical models
- **Direct algorithm**/implementation instead of encodings
  - GCI axioms can be used to "encode" additional operators/axioms
  - Powerful technique, particularly when used with FL closure
  - Can encode cardinality constraints, inverse roles, range/domain, ...
    - ➜ E.g., $(\text{domain } R.C) \equiv \exists R.\top \sqsubseteq C$
  - (FL) encodings introduce (large numbers of) axioms
  - **BUT** even simple domain encoding is **disastrous** with large numbers of roles

Highly Optimised Implementation

Modern systems include **MANY** optimisations, e.g.:

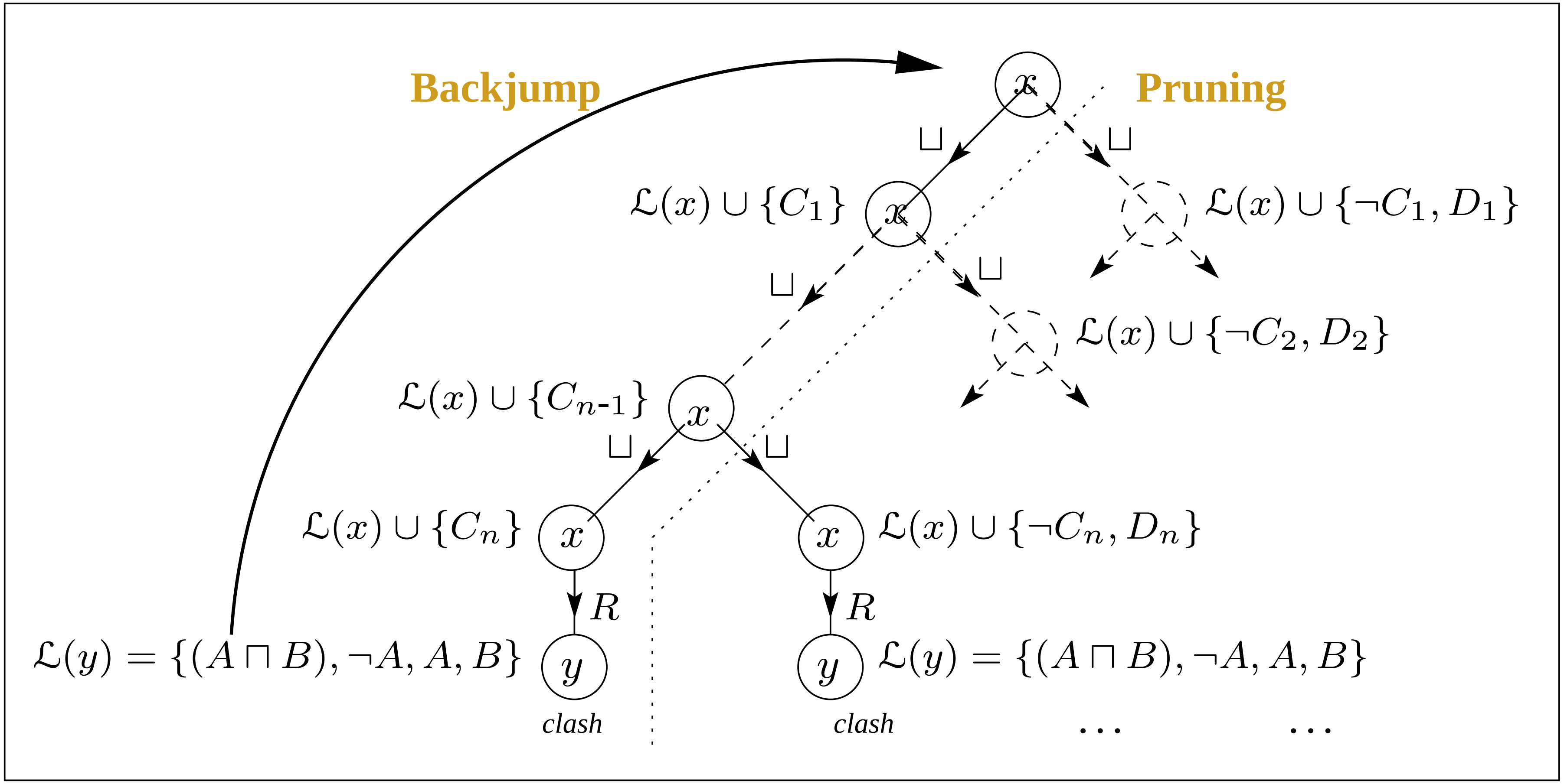
- ☞ Optimised **classification**
  - Use enhanced traversal (exploit information from previous tests)
  - Use structural information to select classification order
- ☞ Optimised **subsumption** testing
  - Normalisation and simplification of concepts
  - Absorption (simplification) of general axioms
  - Davis-Putnam style semantic branching search
  - Dependency directed backtracking
  - Caching
  - Heuristic ordering of propositional and modal expansion

Dependency Directed Backtracking

- Allows **rapid recovery** from bad branching choices
- Most commonly used technique is **backjumping**
  - Tag concepts introduced at **branch points** (e.g., when expanding disjunctions)
  - Expansion rules combine and **propagate tags**
  - On discovering a clash, **identify** most recently introduced concepts involved
  - **Jump back** to relevant branch points **without exploring** alternative branches
  - Effect is to **prune** away part of the search space
- **Highly effective** — essential for usable system
  - E.g., GALEN KB, 30s (with) $\longrightarrow$ months++ (without)

Backjumping

E.g., if $\exists R.\neg A \sqcap \forall R.(A \sqcap B) \sqcap (C_1 \sqcup D_1) \sqcap \ldots \sqcap (C_n \sqcup D_n) \subseteq \mathcal{L}(x)$



Research Challenges

Research Challenges

☞ **Increased expressive power**
- Datatypes
- Nominals
- Extensions to DAML+OIL

☞ **Performance**
- Inverse roles and qualified number restrictions
- Very large KBs
- Reasoning with individuals

☞ **Tools and Infrastructure**
- Support for large scale ontological engineering and deployment

☞ **New reasoning tasks**
- Querying
- Lcs/matching
- Sanctioning
- ...

Increased Expressive Power: Datatypes

DAML+OIL extends $\mathcal{SHIQ}$ with datatypes and nominals

Datatypes

- ☞ DAML+OIL has simple form of datatypes
  - Unary predicates plus disjoint abstract/datatype domains
- ☞ **Theoretically** not particularly challenging
  - Existing work on concrete domains [Baader & Hanschke, Lutz]
  - Algorithm already known for $\mathcal{SHOQ}(\mathbf{D})$ [Horrocks & Sattler]
- ☞ May be **practically** challenging
  - All XMLS datatypes supported
- ☞ Already seeing some (limited) **implementations**
  - E.g., Cerebra system (Network Inference)

Increased Expressive Power: Nominals

Nominals

- DAML+OIL has **oneOf** constructor
  - Extensionally defined concepts, e.g., $\{Mary\}^{\mathcal{I}} = \{Mary\}$
  - Equivalent to nominals in modal logic
- Theoretically **very challenging**
  - Resulting logic has known high complexity (NExpTime)
  - No known “practical” algorithm
  - Not obvious how to extend tableax techniques in this direction
    - ➜ Loss of tree model property
    - ➜ Spy-points: $\top \sqsubseteq \exists R.\{Spy\}$
    - ➜ Finite domains: $\{Spy\} \sqsubseteq \leqslant n R^{-}$
- Relatively straightforward (in theory) without **inverse roles**
  - Algorithm for $\mathcal{SHOQ}(\mathbf{D})$ deals with nominals
  - Practical implementation still to be demonstrated

Increased Expressive Power: Extensions

- DAML+OIL **not expressive enough** for all applications
- Extensions **wish list** includes:
  - Feature chain (path) agreement, e.g., output of component of composite process equals input of subsequent process
  - Complex roles/role inclusions, e.g., a city located in part of a country is located in that country
  - Rules—proposal(s) already exist for “datalog/LP style rules”
  - Temporal and spatial reasoning
  - ...
- May be impossible/undesirable to resist such extensions
- Extended language sure to be **undecidable**
- How can extensions best be **integrated** with DAML+OIL?
- How can reasoners be developed/adapted for extended languages
  - Some existing work on language **fusions** and **hybrid** reasoners

Performance Problems

- Evidence of **empirical tractability** mostly w.r.t. $\mathcal{SHF}$— problems can arise when systems extended to $\mathcal{SHIQ}$
- Important **optimisations** no longer (fully) work
  - E.g., problems with caching as cached models can affect parent
- **Qualified number restrictions** can also cause problems
  - Even relatively small numbers can mean significant non-determinism
- Reasoning with **very large KBs/ontologies**
  - Web ontologies can be expected to grow very large
- Reasoning with **individuals** (Abox)
  - Deployment of web ontologies will mean reasoning with (possibly very large numbers of) individuals
  - Unlikely that standard Abox techniques will be able to cope

Performance Solutions (Maybe)

- Excessive **memory usage**
  - Problem exacerbated by over-cautious double blocking condition (e.g., root node can never block)
  - Promising results from more precise blocking condition [Sattler & Horrocks]
- **Qualified number restrictions**
  - Problem exacerbated by naive expansion rules
  - Promising results from optimised expansion using Algebraic Methods [Haarslev & Möller]
- **Caching** and merging
  - Can still work in some situations (work in progress)
- Reasoning with **very large KBs**
  - DL systems shown to work with $\approx$100k concept KB [Haarslev & Möller]
  - But KB only exploited small part of DL language

Tools and Infrastructure

Tools and infrastructure required in order support use of DAML+OIL

- Ontology **design and maintenance**
  - Several **editors** available, e.g, OilEd (Manchester), OntoEdit (Karlsruhe), Protégé (Stanford)
  - Need integrated **environments** including modularity, versioning, visualisation, explanation, high-level languages, . . .
- Ontology **Integration**
  - Some tools available, e.g., Chimera (Stanford)
  - Need integrated **environments** . . .
  - Can learn from DB integration work [Lenzerini, Calvanese et al]
- **Reasoning** engines
  - Several DL systems available
  - Need for improved usability/connectivity
- . . .

New Reasoning Tasks

- **Querying**
  - Retrieval (instances of a concept) and realisation (most specific class of instance) wont be sufficient
  - Minimum requirement will be conjunctive query style language [Tessaris & Horrocks]
  - May also need to answer “what can I say about x?” style of query [Bechhofer & Horrocks]
- **Explanation** (e.g., to support ontology design) [McGuinness, Borgida et al]
- **Least common subsumer** and/or matching (e.g., to support ontology integration and “bottom up” design) [Baader, Küsters & Molitor]
- ...

Summary

- **Ontologies** will play key role in **Semantic Web**
- **DAML+OIL** is web ontology language based on **Description Logic**
- Ontology design, integration and deployment **supported by reasoning**
- DLs are **logic based KR formalisms** with emphasis on reasoning
- DL systems provide **efficient reasoning services**
  - Careful choice of logic/algorithm
  - Highly optimised implementation
- Still many **challenges** for DL and Semantic Web research
  - Expressive power
  - Performance
  - Tools and infrastructure
  - New reasoning tasks

Resources

Slides from this talk

`www.cs.man.ac.uk/~horrocks/Slides/hp-labs.pdf`

FaCT system

`www.cs.man.ac.uk/fact`

OIL

`www.ontoknowledge.org/oil/`

DAML+OIL

`www.daml.org/language/`

OilEd

`img.cs.man.ac.uk/oil`

I.COM

`www.cs.man.ac.uk/~franconi/icom/`

Select Bibliography

F. Baader, E. Franconi, B. Hollunder, B. Nebel, and H.-J. Profitlich. An empirical analysis of optimization techniques for terminological representation systems or: Making KRIS get a move on. In B. Nebel, C. Rich, and W. Swartout, editors, *Proc. of KR'92*, pages 270–281. Morgan Kaufmann, 1992.

F. Giunchiglia and R. Sebastiani. A SAT-based decision procedure for $\mathcal{ALC}$. In *Proc. of KR'96*, pages 304–314. Morgan Kaufmann, 1996.

V. Haarslev and R. Möller. High performance reasoning with very large knowledge bases: A practical case study. In *Proc. of IJCAI 2001* (to appear).

B. Hollunder and W. Nutt. Subsumption algorithms for concept languages. In *Proc. of ECAI'90*, pages 348–353. John Wiley & Sons Ltd., 1990.

Select Bibliography

I. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. PhD thesis, University of Manchester, 1997.

I. Horrocks and P. F. Patel-Schneider. Comparing subsumption optimizations. In *Proc. of DL'98*, pages 90–94. CEUR, 1998.

I. Horrocks and P. F. Patel-Schneider. Optimising description logic subsumption. *Journal of Logic and Computation*, 9(3):267–293, 1999.

I. Horrocks and S. Tobies. Reasoning with axioms: Theory and practice. In *Proc. of KR'00* pages 285–296. Morgan Kaufmann, 2000.

E. Franconi and G. Ng. The i.com tool for intelligent conceptual modelling. In *Proc. of (KRDB'00)*, August 2000.

D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. F. Patel-Schneider. OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.