Some Practical Issues in Building a Hybrid Deductive Geographic Information System with a DL-Component

KRDB 2003

Michael Wessel

University of Hamburg
“Cognitive Systems Groups”
Department of Computer Science
Project “DLS” (DFG Grant NE 279/8-1)
Contents

- Report about practical work in progress carried out in the context of the “DLS” project
Contents

- Report about practical work in progress carried out in the context of the “DLS” project
  - Vision of a hybrid deductive GIS
Contents

- Report about practical work in progress carried out in the context of the “DLS” project
  - Vision of a hybrid deductive GIS
  - Introduction of a first prototype
Contents

- Report about practical work in progress carried out in the context of the “DLS” project
  - Vision of a hybrid deductive GIS
  - Introduction of a first prototype
  - A software framework for similar “hybrid” KR&R tasks
Contents

- Report about *practical work in progress* carried out in the context of the “DLS” project
  - Vision of a hybrid deductive GIS
  - Introduction of a first prototype
  - A software framework for similar “hybrid” KR&R tasks
- The “vision” of a deductive GIS
Contents

• Report about practical work in progress carried out in the context of the “DLS” project
  • Vision of a hybrid deductive GIS
  • Introduction of a first prototype
  • A software framework for similar “hybrid” KR&R tasks

• The “vision” of a deductive GIS
  • The domain & domain assumptions
Contents

- Report about practical work in progress carried out in the context of the “DLS” project
  - Vision of a hybrid deductive GIS
  - Introduction of a first prototype
  - A software framework for similar “hybrid” KR&R tasks
- The “vision” of a deductive GIS
  - The domain & domain assumptions
  - Desirable reasoning tasks
Contents

• Report about practical work in progress carried out in the context of the “DLS” project
  • Vision of a hybrid deductive GIS
  • Introduction of a first prototype
  • A software framework for similar “hybrid” KR&R tasks
• The “vision” of a deductive GIS
  • The domain & domain assumptions
  • Desirable reasoning tasks
• Implementation issues
Contents

• Report about practical work in progress carried out in the context of the “DLS” project
  • Vision of a hybrid deductive GIS
  • Introduction of a first prototype
  • A software framework for similar “hybrid” KR&R tasks
• The “vision” of a deductive GIS
  • The domain & domain assumptions
  • Desirable reasoning tasks
• Implementation issues
  • Representing the data
Contents

- Report about **practical work in progress** carried out in the context of the “DLS” project
  - Vision of a hybrid deductive GIS
  - Introduction of a first prototype
  - A software framework for similar “hybrid” KR&R tasks

- The “vision” of a deductive GIS
  - The domain & domain assumptions
  - Desirable reasoning tasks

- Implementation issues
  - Representing the data
  - Value of a DL system in this scenario
Contents

- Report about practical work in progress carried out in the context of the “DLS” project
  - Vision of a hybrid deductive GIS
  - Introduction of a first prototype
  - A software framework for similar “hybrid” KR&R tasks
- The “vision” of a deductive GIS
  - The domain & domain assumptions
  - Desirable reasoning tasks
- Implementation issues
  - Representing the data
  - Value of a DL system in this scenario
  - ‘Hybrid conjunctive queries’
The Vision of a Deductive GIS

Starting point: a digital vector map
The Vision of a Deductive GIS

Concrete Geometry

Qualitative Description

Extensional Component

Thematic information in a map

public_park

lake

living_area

school

KRDB, 16.9.2003, Michael Wessel – p.3/10
The Vision of a Deductive GIS

Intensional Component

"Concept definitions" / GEO−Ontology
area, house, ...
green_area → area & ...
lake → area & (not green_area) ...
park → green_area & ...
living_area → area & (not green_area) ...

Extensional Component

Concrete Geometry

Qualitative Description

Modeling of thematic concepts
The Vision of a Deductive GIS

Intensional Component

"Concept definitions" / GEO-Ontology
area, house, ...
green_area -> area & ...
lake -> area & (not green_area) ...
park -> green_area & ...
living_area -> area & (not green_area) ...
park_wa_lake -> park & some cont. lake

Extensional Component

Concrete Geometry

Qualitative Description

Some concepts are really “spatio-thematic”
The Vision of a Deductive GIS

Simple Spatial Queries:
Retrieve all areas contained within this area

Concrete Geometry

Purely spatial queries
The Vision of a Deductive GIS

**Query Component**

Simple Spatial Queries:
- Retrieve all areas contained within this area

Thematic Queries:
- `retrieve_concept_instances(lake)`

**Intensional Component**

"Concept definitions" / GEO-Ontology
- area, house, ...
- `green_area` → area & ...
- `lake` → area & (not `green_area`) ...
- `park` → `green_area` & ...
- `living_area` → area & (not `green_area`) ...
- `park_wa_lake` → park & some cont. lake

**Extensional Component**

- `public_park`
- `lake`
- `living_area`
- `school`

**Concrete Geometry**

**Qualitative Description**

Purely thematic queries
The Vision of a Deductive GIS

Simple Spatial Queries:
Retrieve all areas contained within this area

Thematic Queries:
retrieve_concept_instances(lake)

Spatio-thematic Queries:
Retrieve all parks that contain a lake

"Concept definitions" / GEO-Ontology
area, house, ...
green_area -> area & ...
lake -> area & (not green_area) ...
park -> green_area & ...
living_area -> area & (not green_area) ...
park_wa_lake -> park & some cont. lake

Concrete Geometry

Qualitative Description

“Spatio-thematic” queries

KRDB, 16.9.2003, Michael Wessel – p.3/10
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $\mathcal{E}$
The Vision of a Deductive GIS

• System metaphor: we want a GIS similar to a DL system

• Extensional component $\mathcal{E}$
  • Representation of certain selected spatio-thematic aspects of a concrete map (“geographic world”)
  • Which spatial and thematic aspects?
  • Data vs. information / knowledge?
  • Unified or hybrid representation of spatial and/or thematic aspects (different “sources”)?

KRDB, 16.9.2003, Michael Wessel – p.3/10
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$

Query component $\mathcal{Q}$

Reasoning tasks

Multi-dimensional space of design-decisions

How can RACER be of value in this setting (RACER offers ALCQHI $\mathcal{R} + (\mathcal{D})$, but is not a "spatio-thematic" DL)

Development of a flexible software OO framework allowing for experiments
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
  - Modeling of ontologies with “concepts” in a description language (not necessary DL)
    - Which spatial and thematic aspects?
    - Thematic, spatial, spatio-thematic concepts?
    - Combined or separated description languages for different aspects?
    - Spatio-thematic interaction?
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $Q$
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
  - Retrieval of interesting objects / constellations; “map analysis / reasoning”
  - Kind of queries
  - With spatial and thematic aspects are addressable?
  - Usage of concepts from the ontologies within queries
  - Evaluation of queries (“specialists” for sources)?
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $E$
- Intensional component $I$
- Query component $Q$
- Reasoning tasks
  - $E, I$: consistency checking
  - $I, Q$: satisfiability and entailment of queries / concepts
  - $I, Q$: computation of query / concept subsumption hierarchies ("taxonomies")
  - $E \times I$: instance "realization"
  - $Q \times E \times I$: query answering using vocabulary from $I$
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks

$\Rightarrow$ Multi-dimensional space of design-decisions
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks

$\Rightarrow$ Multi-dimensional space of design-decisions

? How can RACER be of value in this setting (RACER offers $\mathcal{ALCQHI}_R^+ (\mathcal{D}^-)$, but is not a “spatio-thematic” DL)
The Vision of a Deductive GIS

- System metaphor: we want a GIS similar to a DL system
- Extensional component $\mathcal{E}$
- Intensional component $\mathcal{I}$
- Query component $\mathcal{Q}$
- Reasoning tasks

$\Rightarrow$ Multi-dimensional space of design-decisions

? How can RACER be of value in this setting (RACER offers $\mathcal{ALCQHI}_{R^+}(D^-)$, but is not a “spatio-thematic” DL)

- Development of a flexible software OO framework allowing for experiments
The Data

- Data from the “Amt für Vermessung und Geo-Information Hamburg”
The Data

- Data from the “Amt für Vermessung und Geo-Information Hamburg”
- Two digital vector maps in the proprietary SQD format
The Data

- Data from the “Amt für Vermessung und Geo-Information Hamburg”
- Two digital vector maps in the proprietary SQD format
  - Map 1: 2694 objects, 361 primary objects
The Data

Data from the "Amt für Vermessung und Geo-Information Hamburg"

Two digital vector maps in the proprietary SQD format

Objects are "classified" according to object key

5164 (lake, navigable)
4128 (meadow)
2224 (park)
2119 (living area)

Subsumption implicitly present, but not explicitly modeled

Some concepts really have a spatio-thematic flavor, e.g. park with (containing) a lake
The Data

- Data from the “Amt für Vermessung und Geo-Information Hamburg”
- Two digital vector maps in the proprietary SQD format
  - Map 2: 18,039 geometric objects, 5,418 primary object
The Data

Data from the "Amt für Vermessung und Geo-Information Hamburg"
Two digital vector maps in the proprietary SQD format
Objects are "classified" according to object key
dictionary:
- 5164 (lake, navigable)
- 4128 (meadow)
- 2224 (park)
- 2119 (living area, ...)

Subsumption implicitly present, but not explicitly modeled
Some concepts really have a spatio-thematic flavor, e.g. park with (containing) a lake
The Data

- Data from the “Amt für Vermessung und Geo-Information Hamburg”
- Two digital vector maps in the proprietary SQD format
- Objects are “classified” according to object key dictionary:
The Data

- Data from the “Amt für Vermessung und Geo-Information Hamburg”
- Two digital vector maps in the proprietary SQD format
- Objects are “classified” according to object key dictionary:
  - 5164 ⇒ lake, navigable
  - 4128 ⇒ meadow
  - 2224 ⇒ park
  - 2119 ⇒ living area, ...

Subsumption implicitly present, but not explicitly modeled
The Data

- Data from the “Amt für Vermessung und Geo-Information Hamburg”
- Two digital vector maps in the proprietary SQD format
- Objects are “classified” according to object key dictionary:
  - 5164 $\Rightarrow$ lake, navigable
  - 4128 $\Rightarrow$ meadow
  - 2224 $\Rightarrow$ park
  - 2119 $\Rightarrow$ living area
- Subsumption implicitly present, but not explicitly modeled $\Rightarrow$ needs remodeling
The Data

- Data from the “Amt für Vermessung und Geo-Information Hamburg”
- Two digital vector maps in the proprietary SQD format
- Objects are “classified” according to object key dictionary:
  - 5164 ⇒ lake, navigable
  - 4128 ⇒ meadow
  - 2224 ⇒ park
  - 2119 ⇒ living area
- Subsumption implicitly present, but not explicitly modeled ⇒ needs remodeling
- Some concepts really have a spatio-thematic flavor, e.g. park with (containing) a lake
Design Decisions

- Various representation possibilities for the map
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
  - Closed Domain Assumption (there are no other spatial objects than the present ones)
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
  - Closed Domain Assumption (there are no other spatial objects than the present ones)
  - Closed World Assumption (complete theory of a fixed single structure)
Design Decisions

• Various representation possibilities for the map
• Concrete spatial “data”: use a spatially indexed geometric representation
• Qualitative (spatio-)thematic “information”:
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- Qualitative (spatio-)thematic “information”:
  - Setting 1: Modeled as a RACER ABox with Concept Membership Assertions like
    
    $\text{area}_{123} : \text{lake} \sqcap \text{meadow} \sqcap \ldots$

KRDB, 16.9.2003, Michael Wessel – p.5/10
**Design Decisions**

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- Qualitative (spatio-)thematic “information”:
  - Setting 2: Like Setting 1, but additionally with Role Membership Assertions like $(area_{123}, area_{456}) : contains$, mirroring qualitative spatial relationships as found in the map (e.g., using RCC8 relationships)
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- Qualitative (spatio-)thematic “information”:
  - Setting 3: Do not use a RACER ABox, but simply annotate map objects with RACER concept expressions (or expressions of an other reasoning engine)
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- Qualitative (spatio-)thematic “information”:
  - Setting 4: Do not use RACER at all, but implement your own “truly” spatio-thematic DL
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- Qualitative (spatio-)thematic “information”:
  - Setting 5: Don’t even use a DL
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- Qualitative (spatio-)thematic “information”:
  ⇒ In order to allow for flexible experiments, description languages are not hard-wired into the software framework; e.g. we can use the same framework if we change the spatio-thematic description vocabulary, e.g. switching from RCC8 relationships to qualitative distance relationships
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- (Qualitative) (spatio-)thematic “information”: use your favorite description language
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- (Qualitative) (spatio-)thematic “information”: use your favorite description language
- Notion of a “reasoning substrate”:
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- (Qualitative) (spatio-)thematic “information”: use your favorite description language
- Notion of a “reasoning substrate”:
  - General-purpose “labeled graph”-like notion with exchangeable node and edge labeling languages
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- (Qualitative) (spatio-)thematic “information”: use your favorite description language
- Notion of a “reasoning substrate”:
  - General-purpose “labeled graph”-like notion with exchangeable node and edge labeling languages
  - Use inheritance to get specialized substrate classes, languages and reasoners
Design Decisions

Subclasses of class substrate
Design Decisions

Subclasses of class semantic entity
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- (Qualitative) (spatio-)thematic “information”: use your favorite description language
- Notion of a “reasoning substrate”:
  - General-purpose “labeled graph”-like notion with exchangeable node and edge labeling languages
  - Use inheritance to get specialized substrate classes, languages and reasoners
  - Special-purpose index structures
Design Decisions

- Various representation possibilities for the map
- Concrete spatial “data”: use a spatially indexed geometric representation
- (Qualitative) (spatio-)thematic “information”: use your favorite description language
- Notion of a “reasoning substrate”:
  - General-purpose “labeled graph”-like notion with exchangeable node and edge labeling languages
  - Use inheritance to get specialized substrate classes, languages and reasoners
  - Special-purpose index structures
  - DL-system inspired protocols (interfaces)
An Experiment with RACER

- $\mathcal{I} = TBox$: modeling of purely thematic concepts; “object keys” are remodeled as (quite simple) RACER TBox
An Experiment with RACER

- $I = TBox$ : modeling of purely thematic concepts; “object keys” are remodeled as (quite simple) RACER TBox
  
  $\Rightarrow$ even a simple ontology is of great value; the query $\text{retrieve\_concept\_instances}(\text{green\_area})$ would not return instances of the (intuitive) sub-concepts $\text{meadow}$ and $\text{park}$ otherwise
An Experiment with RACER

- $\mathcal{I} = TBox$ : modeling of purely thematic concepts; “object keys” are remodeled as (quite simple) RACER TBox

- $\mathcal{E} = (ABox, ‘Map Substrate’) = ‘Racer Map Substrate’$: exhaustively add RCC8 role membership assertions, computed from the geometry, and thematic descriptions to the ABox
An Experiment with RACER

- $\mathcal{I} = TBox$: modeling of purely thematic concepts; “object keys” are remodeled as (quite simple) RACER TBox
- $\mathcal{E} = (ABox, ‘Map Substrate’) = ‘Racer Map Substrate’: exhaustively add RCC8 role membership assertions, computed from the geometry, and thematic descriptions to the ABox
- Map 1: 130.321 RMAs if we represent the disconnected relationship $DC$, 1804 without $DC$
An Experiment with RACER

- $\mathcal{I} = TBox$: modeling of purely thematic concepts; “object keys” are remodeled as (quite simple) RACER TBox
- $\mathcal{E} = (\text{ABox, ‘Map Substrate’}) = ‘\text{Racer Map Substrate}’$: exhaustively add RCC8 role membership assertions, computed from the geometry, and thematic descriptions to the ABox
  - Map 2: 29.354.724 with $DC$, 19.988 without $DC$
An Experiment with RACER

Illustration of a typical ABox
An Experiment with RACER

- $\mathcal{T} = TBox$: modeling of purely thematic concepts; “object keys” are remodeled as (quite simple) RACER TBox
- $\mathcal{E} = (ABox, \text{‘Map Substrate’}) = \text{‘Racer Map Substrate’}$: exhaustively add RCC8 role membership assertions, computed from the geometry, and thematic descriptions to the ABox
- Pose simple instance retrieval queries to RACER
An Experiment with RACER

<table>
<thead>
<tr>
<th>Spatial Querying</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green Area \ E\text{contains} \ Lake</td>
</tr>
</tbody>
</table>

**RACER 1.7.7 performs much better than RACER 1.7.6, but only until we add DC and close the roles.**
An Experiment with RACER

I = T Box:
modeling of purely thematic
concepts; "objectkeys" are remodeled as (quite
simple) RACER TBox

E = (ABox, "MapSubstrate") = "RacerMapSubstrate":
exhaustively add RCC8 role
membership assertions, computed from the
geometry, and thematic descriptions to the ABox

Pose simple instance retrieval queries to RACER

Closing of spatial roles (R) required to realize
"spatial closed domain assumption" in order to
answer \( R:C \) queries correctly

RACER 1.7.7 performs much better than RACER
1.7.6, but only until we add DC and close the
roles

KRDB, 16.9.2003, Michael Wessel – p.6/10
An Experiment with RACER

- $\mathcal{I} = TBox$: modeling of purely thematic concepts; “object keys” are remodeled as (quite simple) RACER TBox
- $\mathcal{E} = (ABox,‘Map Substrate’) = ‘Racer Map Substrate’: exhaustively add RCC8 role membership assertions, computed from the geometry, and thematic descriptions to the ABox
- Pose simple instance retrieval queries to RACER
- Closing of spatial roles ($R$) required to realize “spatial closed domain assumption” in order to answer $\forall R.C$ queries correctly
An Experiment with RACER

- $\mathcal{I} = TBox$: modeling of purely thematic concepts; “object keys” are remodeled as (quite simple) RACER TBox

- $\mathcal{E} = (ABox, ‘Map Substrate’) = ‘Racer Map Substrate’: exhaustively add RCC8 role membership assertions, computed from the geometry, and thematic descriptions to the ABox

- Pose simple instance retrieval queries to RACER

- Closing of spatial roles ($R$) required to realize “spatial closed domain assumption” in order to answer $\forall R.C$ queries correctly

\[ \Rightarrow \text{add } i : (\leq n R) \cap (\geq n R) \text{ to individual } i, \]

\[ \text{where } n = \text{def } | \{ j \mid (i, j) : R \in \mathcal{A} \} | \]
An Experiment with RACER

- $T = TBox$: modeling of purely thematic concepts; “object keys” are remodeled as (quite simple) RACER TBox

- $E = (\text{ABox, ‘Map Substrate’}) = \text{‘Racer Map Substrate’}$: exhaustively add RCC8 role membership assertions, computed from the geometry, and thematic descriptions to the ABox

- Pose simple instance retrieval queries to RACER

- Closing of spatial roles ($R$) required to realize “spatial closed domain assumption” in order to answer $\forall R.C$ queries correctly

- RACER 1.7.7 performs much better than RACER 1.7.6, but only until we add $DC$ and close the roles
Problems with the Approach

- Incompleteness of reasoning in $I$, $Q$

Inherently unsatisfiable queries will not be recognized as such. TBox (resp. $I$, $Q$): might become inconsistent without being noticed, missing subsumption relationships etc.

"Query subsumption" incomplete, but okay for optimization purposes (caching/reusing of answer sets).

Since the ABox is "correctly closed", query answering is complete (assuming an unfoldable TBox).

RACER performs "spatial closed domain reasoning".

RACER has problems with the specific structure of the ABoxes (probably "worst case" for a tableaux-based system!)

Using RACER in this way seems to be inappropriate.

Implementation of special-purpose reasoners. We can still use RACER for "sub-reasoning" tasks.
Problems with the Approach

- Incompleteness of reasoning in $\mathcal{I}$, $\mathcal{Q}$
  - Inherently unsatisfiable queries will not be recognized as such
Problems with the Approach

- Incompleteness of reasoning in $\mathcal{I}$, $\mathcal{Q}$
  - Inherently unsatisfiable queries will not be recognized as such
- $\text{TBox (resp. } \mathcal{I})$: might become inconsistent without being noticed, missing subsumption relationships etc. $\Rightarrow$ model only purely thematic concepts

KRDB, 16.9.2003, Michael Wessel – p.7/10
Problems with the Approach

- Incompleteness of reasoning in $\mathcal{I}$, $\mathcal{Q}$
  - Inherently unsatisfiable queries will not be recognized as such
  - TBox (resp. $\mathcal{I}$): might become inconsistent without being noticed, missing subsumption relationships etc. $\Rightarrow$ model only purely thematic concepts
  - “Query subsumption” incomplete, but okay for optimization purposes (caching/reusing of answer sets)

KRDB, 16.9.2003, Michael Wessel – p.7/10
Problems with the Approach

- Incompleteness of reasoning in $I$, $Q$
- Since the ABox is “correctly closed”, query answering is complete (assuming an unfoldable TBox) $\implies$ RACER performs “spatial closed domain reasoning”
Problems with the Approach

- Incompleteness of reasoning in $\mathcal{I}$, $\mathcal{Q}$
- Since the ABox is “correctly closed”, query answering is complete (assuming an unfoldable TBox) $\Rightarrow$ RACER performs “spatial closed domain reasoning”
- RACER has problems with the specific structure of the ABoxes (probably “worst case” for a tableaux-based system!)
Problems with the Approach

- Incompleteness of reasoning in $I$, $Q$

- Since the ABox is “correctly closed”, query answering is complete (assuming an unfoldable TBox) $\Rightarrow$ RACER performs “spatial closed domain reasoning”

- RACER has problems with the specific structure of the ABoxes (probably “worst case” for a tableaux-based system!)
  - RACER cannot handle even small ABoxes with DC and closed roles
Problems with the Approach

- Incompleteness of reasoning in $\mathcal{I}, \mathcal{Q}$
- Since the ABox is “correctly closed”, query answering is complete (assuming an unfoldable TBox) $\Rightarrow$ RACER performs “spatial closed domain reasoning”
- RACER has problems with the specific structure of the ABoxes (probably “worst case” for a tableaux-based system!)
  - RACER cannot handle even small ABoxes with DC and closed roles
  - Explicit representation of 29 million role membership assertions is not a good idea
Problems with the Approach

• Incompleteness of reasoning in $\mathcal{I}, \mathcal{Q}$

• Since the ABox is “correctly closed”, query answering is complete (assuming an unfoldable TBox) $\Rightarrow$ RACER performs “spatial closed domain reasoning”

• RACER has problems with the specific structure of the ABoxes (probably “worst case” for a tableaux-based system!)
  - RACER cannot handle even small ABoxes with DC and closed roles
  - Explicit representation of 29 million role membership assertions is not a good idea
  - “Specialized reasoners” will perform much better
Problems with the Approach

• Incompleteness of reasoning in $\mathcal{I}$, $\mathcal{Q}$

• Since the ABox is “correctly closed”, query answering is complete (assuming an unfoldable TBox) $\Rightarrow$ RACER performs “spatial closed domain reasoning”

• RACER has problems with the specific structure of the ABoxes (probably “worst case” for a tableaux-based system!)

$\Rightarrow$ Using RACER in this way seems to be inappropriate
Problems with the Approach

- Incompleteness of reasoning in $\mathcal{I}$, $\mathcal{Q}$
- Since the ABox is “correctly closed”, query answering is complete (assuming an unfoldable TBox) $\Rightarrow$ RACER performs “spatial closed domain reasoning”
- RACER has problems with the specific structure of the ABoxes (probably “worst case” for a tableaux-based system!)

$\Rightarrow$ Using RACER in this way seems to be inappropriate

$\Rightarrow$ Implementation of special-purpose reasoners
Problems with the Approach

- Incompleteness of reasoning in $\mathcal{I}, \mathcal{Q}$
- Since the ABox is "correctly closed", query answering is complete (assuming an unfoldable TBox) $\Rightarrow$ RACER performs "spatial closed domain reasoning"
- RACER has problems with the specific structure of the ABoxes (probably "worst case" for a tableau-based system!)

$\Rightarrow$ Using RACER in this way seems to be inappropriate

$\Rightarrow$ Implementation of special-purpose reasoners

$\Rightarrow$ We can still use RACER for "sub-reasoning" tasks
More Expressive Queries

- Simple non-recursive conjunctive queries
More Expressive Queries

- Simple non-recursive conjunctive queries
  - ‘Find a living area, a green area and a parking lot which are pairwise adjacent’
  - \[ \text{query}(?x, ?y, ?z) \leftarrow \]
  - \[ \text{living\_area}(?x), \text{green\_area}(?y), \text{parking\_lot}(?z), \]
  - \[ \text{adjacent}(?x, ?y), \text{adjacent}(?x, ?z), \text{adjacent}(?y, ?z) \]
  - \[ \Rightarrow \text{not expressible with standard DL concepts} \]
More Expressive Queries

Queries make use of "hybrid" spatio/thematic vocabulary from the ontologies. Vision: according to where and how the data/information is represented ("sources"), queries will be "rewritten". Result of the reformulation process: a "hybrid" spatio-thematic query which we can already process.

"Active domain" semantics for variables (note: conjuncts like ?x : R:C can be used).

Query processing: parsing ! plan generation ! plan optimization ! compilation ! execution.
More Expressive Queries

- Simple non-recursive conjunctive queries
  - ‘Find a contaminated lake in a park in which a creek flows which borders an industrial area containing a chemical plant’
  - \(\text{query}(\?x, \?y, \?z, \?f) \leftarrow\)
    
    \(\text{industrial\_area}(\?x), \text{creek}(\?y), \text{lake\_or\_pond}(\?z),\)
    
    \(\text{contaminated}(\?z), \text{chemical\_plant}(\?f), \text{park}(\?u),\)
    
    \(\text{borders}(\?y, \?x), \text{flows\_in}(\?y, \?z), \text{contains}(\?u, \?z),\)
    
    \(\text{contains}(\?x, \?f)\)
More Expressive Queries

Queries make use of "hybrid" spatio/thematic vocabulary from the ontologies. Vision: according to where and how the data/information is represented ("sources"), queries will be "rewritten". Result of the reformulation process: a "hybrid" spatio-thematic query which we can already process.

"Active domain" semantics for variables (note: conjuncts like $x:R:C$ can be used).

Query processing: parsing → plan generation → plan optimization → compilation → execution.
More Expressive Queries

- Simple non-recursive conjunctive queries
- Queries make use of ‘hybrid’ spatio/thematic vocabulary from the ontologies
  → ‘Plug in’ definitions of terms
More Expressive Queries

- Simple non-recursive conjunctive queries
- Queries make use of ‘hybrid’ spatio/thematic vocabulary from the ontologies
  → ‘Plug in’ definitions of terms
- \( \text{lake}_\text{or}_\text{pond}(\?z) \rightarrow \) 
- \( \?z^* : (\text{lake} \sqcap \text{pond}) \)
More Expressive Queries

- Simple non-recursive conjunctive queries
- Queries make use of ‘hybrid’ spatio/thematic vocabulary from the ontologies
  → ‘Plug in’ definitions of terms
- $borders(?y, ?x), flows\_in(?y, ?x) \rightarrow EC(?y, ?x)$
More Expressive Queries

- Simple non-recursive conjunctive queries
- Queries make use of ‘hybrid’ spatio/thematic vocabulary from the ontologies
  - ‘Plug in’ definitions of terms
  - $\text{contaminated}(?z) \rightarrow \ ?z^* : \exists \text{water\_quality}.\text{poisoned}$
More Expressive Queries

- Simple non-recursive conjunctive queries
- Queries make use of ‘hybrid” spatio/thematic vocabulary from the ontologies
- Vision: according to where and how the data/information is represented (‘sources”), queries will be ‘rewritten”
More Expressive Queries

- Simple non-recursive conjunctive queries
- Queries make use of ‘hybrid’ spatio/thematic vocabulary from the ontologies
- Vision: according to where and how the data/information is represented (“sources”), queries will be ‘rewritten’
- Result of the reformulation process: a ‘hybrid’ spatio-thematic query which we can already process
More Expressive Queries

\(?x^* / ?x\) are bound in parallel
More Expressive Queries

- Simple non-recursive conjunctive queries
- Queries make use of ‘hybrid’ spatio/thematic vocabulary from the ontologies
- Vision: according to where and how the data/information is represented (“sources”), queries will be ‘rewritten’
- Result of the reformulation process: a ‘hybrid’ spatio-thematic query which we can already process
- ‘Active domain’ semantics for variables (note: conjuncts like $?x^* : \exists R.C$ can be used)
More Expressive Queries

- Simple non-recursive conjunctive queries
- Queries make use of ‘hybrid’ spatio/thematic vocabulary from the ontologies
- Vision: according to where and how the data/information is represented (‘sources’), queries will be ‘rewritten’
- Result of the reformulation process: a ‘hybrid’ spatio-thematic query which we can already process
- ‘Active domain’ semantics for variables (note: conjuncts like \(?x^* : \exists R.C\) can be used)
- Query processing: parsing → plan generation → plan optimization → compilation → execution
Reasoning about Queries

- Example: query consistency
Reasoning about Queries

- Example: query consistency
- Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts
Reasoning about Queries

- Example: query consistency
  - Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts
  - Check satisfiability separately

Reduction to appropriate ABox / RCC consistency checks

Example: hybrid query containment

By reduction to query consistency

Customizable: notion of consistency has to be provided by the framework user (implementation of specialized methods)

Vision: since queries can also be seen as “concept definitions” it might be reasonable to base the ontology on them (instead of a truly spatio-thematic description logic)
Reasoning about Queries

- Example: query consistency
  - Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts
  - Check satisfiability separately
  - ABox assertions: construct an ABox from the conjuncts, replacing variables with individuals (RACER offers unique name assumption), check for ABox satisfiability
Reasoning about Queries

- Example: query consistency
  - Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts
  - Check satisfiability separately
  - ABox assertions: construct an ABox from the conjuncts, replacing variables with individuals (RACER offers unique name assumption), check for ABox satisfiability
  - RCC conjuncts: construct an RCC network and check for its consistency

Conjecture: somehow “weak” since no interaction, but quite useful in this scenario, and complete (unlike using RACER concepts)

Reduction to appropriate ABox / RCC consistency checks

Example: hybrid query containment

By reduction to query consistency

Customizable: notion of consistency has to be provided by the framework user (implementation of specialized methods)

Vision: since queries can also be seen as “concept definitions” it might be reasonable to base the ontology on them (instead of a truly spatio-thematic description logic)
Reasoning about Queries

- Example: query consistency
  - Two kinds of conjuncts: “RCC” and “ABox assertion” conjuncts
  - Check satisfiability separately
  - ABox assertions: construct an ABox from the conjuncts, replacing variables with individuals (RACER offers unique name assumption), check for ABox satisfiability
  - RCC conjuncts: construct an RCC network and check for its consistency
  - Conjecture: somehow “weak” since no interaction, but quite useful in this scenario, and complete (unlike using RACER concepts)
Reasoning about Queries

- Example: query consistency
  \[ \Rightarrow \text{Reduction to appropriate ABox / RCC consistency checks} \]
Reasoning about Queries

• Example: query consistency
  ⇒ Reduction to appropriate ABox / RCC consistency checks

• Example: hybrid query containment
Reasoning about Queries

- Example: query consistency
  \[\Rightarrow \text{Reduction to appropriate ABox / RCC consistency checks}\]

- Example: hybrid query containment

\[
\text{query}(\text{?germany}, \text{?city}, \text{?sea}) \leftarrow \\
\text{germany}(\text{?germany}^*), \text{federal\_division}(\text{?division}^*), \\
\text{german\_city}(\text{?city}^*), (\text{baltic\_see} \sqcap \text{north\_sea})(\text{?sea}^*), \\
\text{PPI}(\text{?germany}, \text{?division}), \text{PPI}(\text{?division}, \text{?city}), \\
\text{DR}(\text{?division}, \text{?sea}) \]

\[
\models \\
\text{query}(\text{?country}, \text{?city}, \text{?ocean}) \leftarrow \\
\text{country}(\text{?country}^*), \text{city}(\text{?city}^*), \text{ocean}(\text{?ocean}^*), \\
\text{DR}(\text{?ocean}, \text{?city}), \text{PPI}(\text{?country}, \text{?city})
\]
Reasoning about Queries

Two queries - does Green entail Blue?
Reasoning about Queries

Adding entailed constraints for Green
Reasoning about Queries

Example: query consistency
Reduction to appropriate ABox / RCC consistency checks

Example: hybrid query containment
By reduction to query consistency
Customizable: notion of consistency has to be provided by the framework user (implementation of specialized methods)
Vision: since queries can also be seen as "concept definitions" it might be reasonable to base the ontology on them (instead of a truly spatio-thematic description logic)
Reasoning about Queries

Example: query consistency

Reduction to appropriate ABox / RCC consistency checks

Example: hybrid query containment

By reduction to query consistency

Customizable: notion of consistency has to be provided by the framework user (implementation of specialized methods)

Vision: since queries can also be seen as “concept definitions” it might be reasonable to base the ontology on them (instead of a truly spatio-thematic description logic)
Reasoning about Queries

Example: query consistency
Reduction to appropriate ABox / RCC consistency checks

Example: hybrid query containment
By reduction to query consistency
Customizable: notion of consistency has to be provided by the framework user (implementation of specialized methods)
Vision: since queries can also be seen as "concept definitions" it might be reasonable to base the ontology on them (instead of a truly spatio-thematic description logic)
Reasoning about Queries

Example: query consistency
Reduction to appropriate ABox / RCC consistency checks

Example: hybrid query containment
By reduction to query consistency
Customizable: notion of consistency has to be provided by the framework user (implementation of specialized methods)
Vision: since queries can also be seen as "concept definitions" it might be reasonable to base the ontology on them (instead of a truly spatio-thematic description logic)
Reasoning about Queries

Example: query consistency

Reduction to appropriate ABox / RCC consistency checks

Example: hybrid query containment

By reduction to query consistency

Customizable: notion of consistency has to be provided by the framework user (implementation of specialized methods)

Vision: since queries can also be seen as "concept definitions" it might be reasonable to base the ontology on them (instead of a truly spatio-thematic description logic)
Reasoning about Queries

Match - Green is more specific than Blue
Reasoning about Queries

- Example: query consistency
  ⇒ Reduction to appropriate ABox / RCC consistency checks

- Example: hybrid query containment
  ⇒ By reduction to query consistency
Reasoning about Queries

- Example: query consistency
  ⇒ Reduction to appropriate ABox / RCC consistency checks

- Example: hybrid query containment
  ⇒ By reduction to query consistency

- Customizable: notion of consistency has to be provided by the framework user (implementation of specialized methods)
Reasoning about Queries

- Example: query consistency
  \[\Rightarrow\] Reduction to appropriate ABox / RCC consistency checks

- Example: hybrid query containment
  \[\Rightarrow\] By reduction to query consistency

- Customizable: notion of consistency has to be provided by the framework user (implementation of specialized methods)

- Vision: since queries can also be seen as “concept definitions” it might be reasonable to base the ontology \( I \) on them (instead of a truly spatio-thematic description logic)
Summary

- Usage of (one of) the fastest DL systems does not guarantee good overall performance in arbitrary application contexts.

Application-specific reasoners and/or reasoning services are still needed.

Application-specific index structures and optimizations are needed.

It would be nice if DL systems were more open and "customizable" using inheritance (where is the DL system with arbitrary user-definable concrete domains?)

An object-oriented DL-system architecture can have advantages.
Summary

- Usage of (one of) the fastest DL systems does not guarantee good overall performance in arbitrary application contexts
- Application-specific reasoners and/or reasoning services are still needed
Summary

- Usage of (one of) the fastest DL systems does not guarantee good overall performance in arbitrary application contexts
- Application-specific reasoners and/or reasoning services are still needed
- Application-specific index structures and optimizations are needed
Summary

- Usage of (one of) the fastest DL systems does not guarantee good overall performance in arbitrary application contexts
- Application-specific reasoners and/or reasoning services are still needed
- Application-specific index structures and optimizations are needed

⇒ It would be nice if DL systems were more open and “customizable” using inheritance (where is the DL system with arbitrary user-definable concrete domains?)
Summary

- Usage of (one of) the fastest DL systems does not guarantee good overall performance in arbitrary application contexts
- Application-specific reasoners and/or reasoning services are still needed
- Application-specific index structures and optimizations are needed

⇒ It would be nice if DL systems were more open and “customizable” using inheritance (where is the DL system with arbitrary user-definable concrete domains?)

⇒ An object-oriented DL-system architecture can have advantages
Thanks for your attention!

Usage of (one of) the fastest DL systems does not guarantee good overall performance in arbitrary application contexts.

Application-specific reasoners and/or reasoning services are still needed.

Application-specific index structures and optimizations are needed.

It would be nice if DL systems were more open and "customizable" using inheritance (where is the DL system with arbitrary user-definable concrete domains?)

An object-oriented DL-system architecture can have advantages.