A Flexible DL-based Architecture for Deductive Information Systems

Michael Wessel and Ralf Möller

Software Systems Group (STS)
Hamburg University of Technology (TUHH)
Hamburg, Germany
{mi.wessel | r.f.moeller}@tuhh.de

Background and Motivation

- Description logics (DLs) provide widely accepted standards for decidable knowledge representation
- Benefits: Scalability of expressivity, foundations for ontology languages (Semantic Web), impressive theory, performant DL systems (DLS)
- Question: A basis for ontology-based information systems (IS)?
- Evaluation: How to use and extend a DL system for building an ontology-based query answering system for city maps?
- Contribution: Pragmatic solutions to tackle the design, representation and query answering problem in domains for which standard DLs are not well-suited

Drawbacks of Today's DLS for IS Building

- Scalability for ABoxes not easy to achieve (see LUBM with nRQL + RacerPro)
- Few have expressive and practical ABox QL
- Persistency for large ABoxes? Use a DB, but:
  - Query answering requires inference – DB access during reasoning? Pre-loading of (which portions of the) ABox into memory?
  - Thesis: as long as RPC is much slower, representation layer should include query answering engine and reasoning engine
- ABoxes not good for "data representation" (Strings, Polygons), Concrete Domains: overkill in many cases

Underlying Thesis

- Pragmatic solutions needed for todays IS designers
  - Existing DLS must be reused (cannot be reimplemented in a short period of time)
  - Pragmatic ways to extend DLS in case representational deficiencies are encountered should be identified $\longrightarrow$ this paper
  - Extensions must / should be easier to implement than DLS
  - Longtime perspective: DLS with “open architecture”? Plug in mechanism?
  - Even if this is achieved, extensibility is hard due to inherent intellectual complexity

Contributions

- "Road map for a difficult terrain"
- Three pragmatic settings for city maps IS
  - Setting 1: What can be done with RACERPRO ABox left alone
  - Setting 2: (RACERPRO external) spatial representation ("spatial DB") + ABox
  - Setting 3: (RACERPRO internal) spatial representation + ABox
- Base the IS on software abstractions
  - Abstract from remote vs. local procedure call
  - Abstract from ABox representation of the DLs
  - Provide more flexible means for data / information representation than ABoxes

Contributions (2)

- Base the IS on software abstractions
  - Allow for hybrid representations (layering of representations)
  - Use one internal data model for the IS which allows all this: “substrate data model”
  - flexible, extensible query language needed: “substrate query language framework”

$\Rightarrow$ “Semantic Middleware”

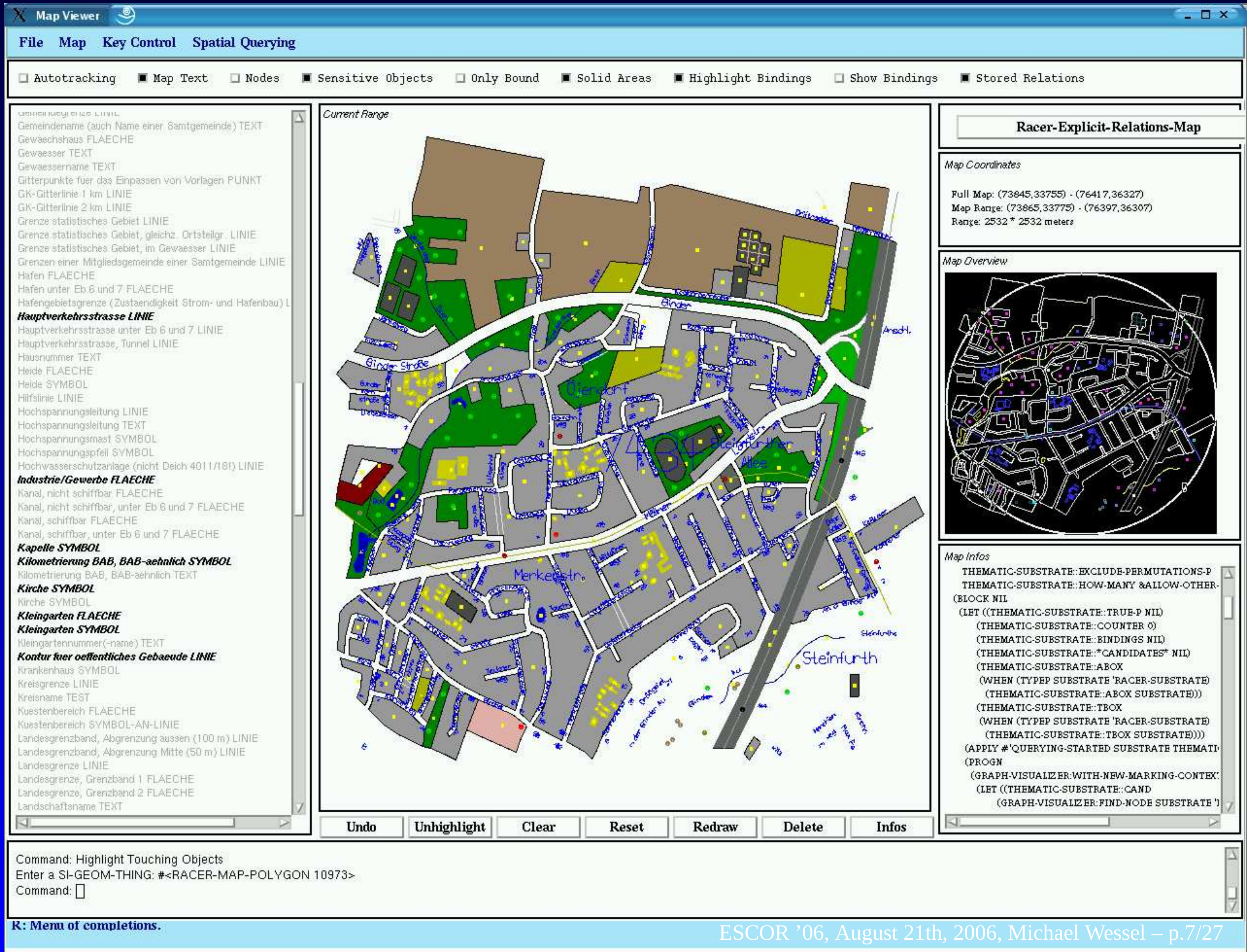
  - Thesis: flexible way to build DL-based IS
  - Example: nRQL was first external to RacerPro; since it was implemented on top of “substrate” middleware, it could be migrated easily into RacerPro

Digitale Stadtkarte (DISK)

- ©"Amt für Geoinformation und Vermessung"
- Two digital vector maps in SQD format
- Objects are "classified" according to a fixed list of categories ("Objektschlüssel-Katalog"):
  - 5164 ⇒ [lake, navigable], 4128 ⇒ [meadow], 2224 ⇒ [park], 2119 ⇒ [living area,] ...
  - Taxonomic relationships ("is-a") implicitly present, but not explicitly modeled ⇒ needs remodeling
  - Very specific categories, no generalizing categories: *cemetry_for_non_christians*, but *cemetry* is missing

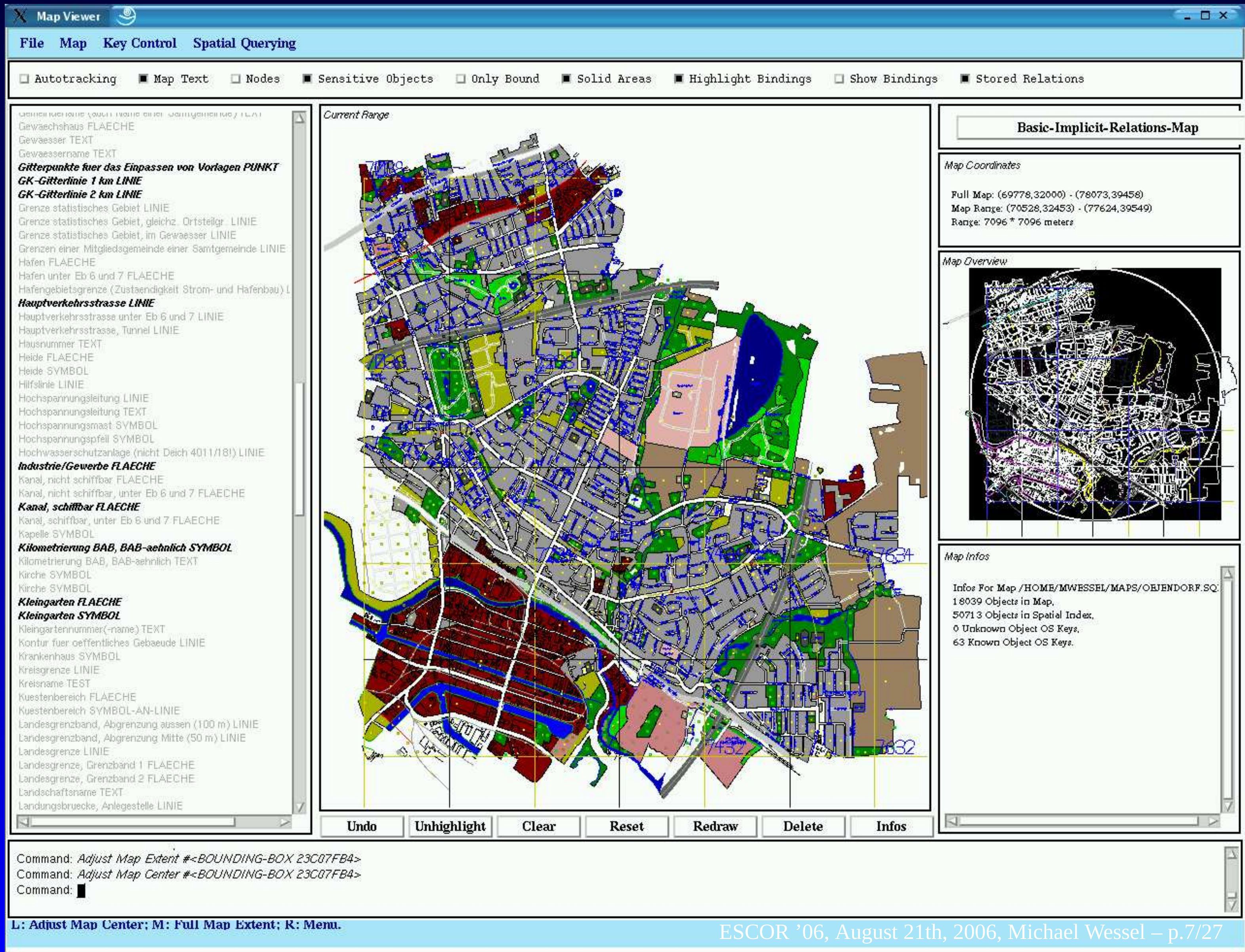
Digitale Stadtkarte (DISK)

- Map 1: 2694 geo objects, 361 primary



Digitale Stadtkarte (DISK)

- Map 2: 18.039 geo objects, 5.418 primary



Digitale Stadtkarte (DISK)

- ©"Amt für Geoinformation und Vermessung"
- Two digital vector maps in SQD format
- Objects are "classified" according to a fixed list of categories ("Objektschlüssel-Katalog"):
  - 5164 ⇒ [lake, navigable], 4128 ⇒ [meadow], 2224 ⇒ [park], 2119 ⇒ [living area,] ...
  - Taxonomic relationships ("is-a") implicitly present, but not explicitly modeled ⇒ needs remodeling
  - Very specific categories, no generalizing categories: *cemetry_for_non_christians*, but *cemetry* is missing

Aspects of Geographic Data

- Two groups of aspects
  - Thematic: Geographic Category (Semantics)
  - Spatial: Area, Shape, Relationships, . . .
- Space has specific properties
- Instances of spatial data types (polygons etc.) <u>automatically preserve</u> many important spatial aspects of the represented geo object (a polygon represents both shape and area)
- A map intrinsically represents spatial relationships ⇒ rich, “analogical” representation
- In an ABox, everything must be represented symbolically ⇒ “symbolic bottleneck”

Spatio-Thematic Querying

- DISK ontology in a TBox
- Remodelling of thematic DISK categories:
  $cemetry\_for\_non\_christians \dot{\sqsubseteq} cemetry$
- Additional spatio-thematic vocabulary:
  $park\_with\_lake \doteq park \sqcap \exists contains.lake$
- Use spatio-thematic vocabulary in queries:

$$ans(?lake, ?park, ?creek, ?industrial\_area, ?chemical\_plant) \leftarrow$$
$$lake(?lake), chemically\_contaminated(?lake), park(?park),$$
$$contains(?park, ?lake), creek(?creek),$$
$$flows\_in(?creek, ?lake), crosses(?creek, ?industrial\_area),$$
$$contains(?industrial\_area, ?chemical\_plant),$$
$$unreliable(?chemical\_plant).$$

Representing DISK: Setting 1

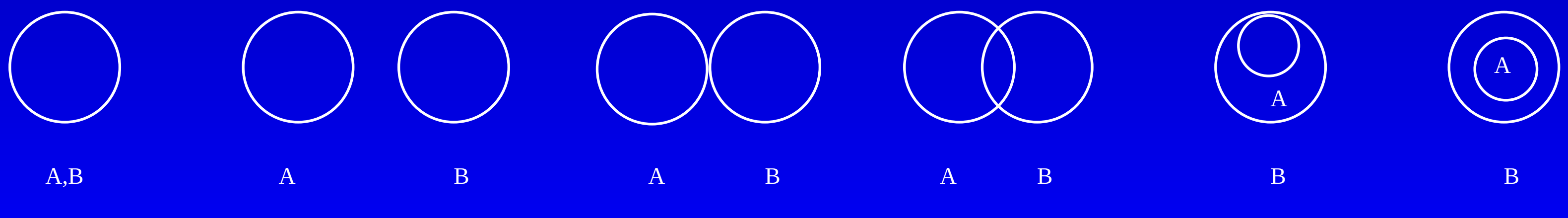
- What can be done with RacerPro left alone?
- Remodeling of thematic categories in a TBox
  - concept_for_key$(2224) =_{def} park$

Representing DISK: Setting 1

- What can be done with RacerPro left alone?
- Remodeling of thematic categories in a TBox
  - concept_for_key$(2224) =_{def} park$
- Representation of the map as an ABox $\mathcal{A}$
  - For each map object i with key n, add $i :$ concept_for_key(n) to $\mathcal{A}$
  - Represent dedicated metric aspects in the CD $(i : \exists(area). =_{123456})$

Representing DISK: Setting 1

- What can be done with RacerPro left alone?
- Remodeling of thematic categories in a TBox
  - concept_for_key$(2224) =_{def} park$
- Representation of the map as an ABox $\mathcal{A}$
  - For each map object i with key n, add i : concept_for_key(n) to $\mathcal{A}$
  - Represent dedicated metric aspects in the CD $(i : \exists(area). =_{123456})$
  - Compute qualitative RCC8 relationships: (i, j) : EC, (i, k) : TPPI, ...

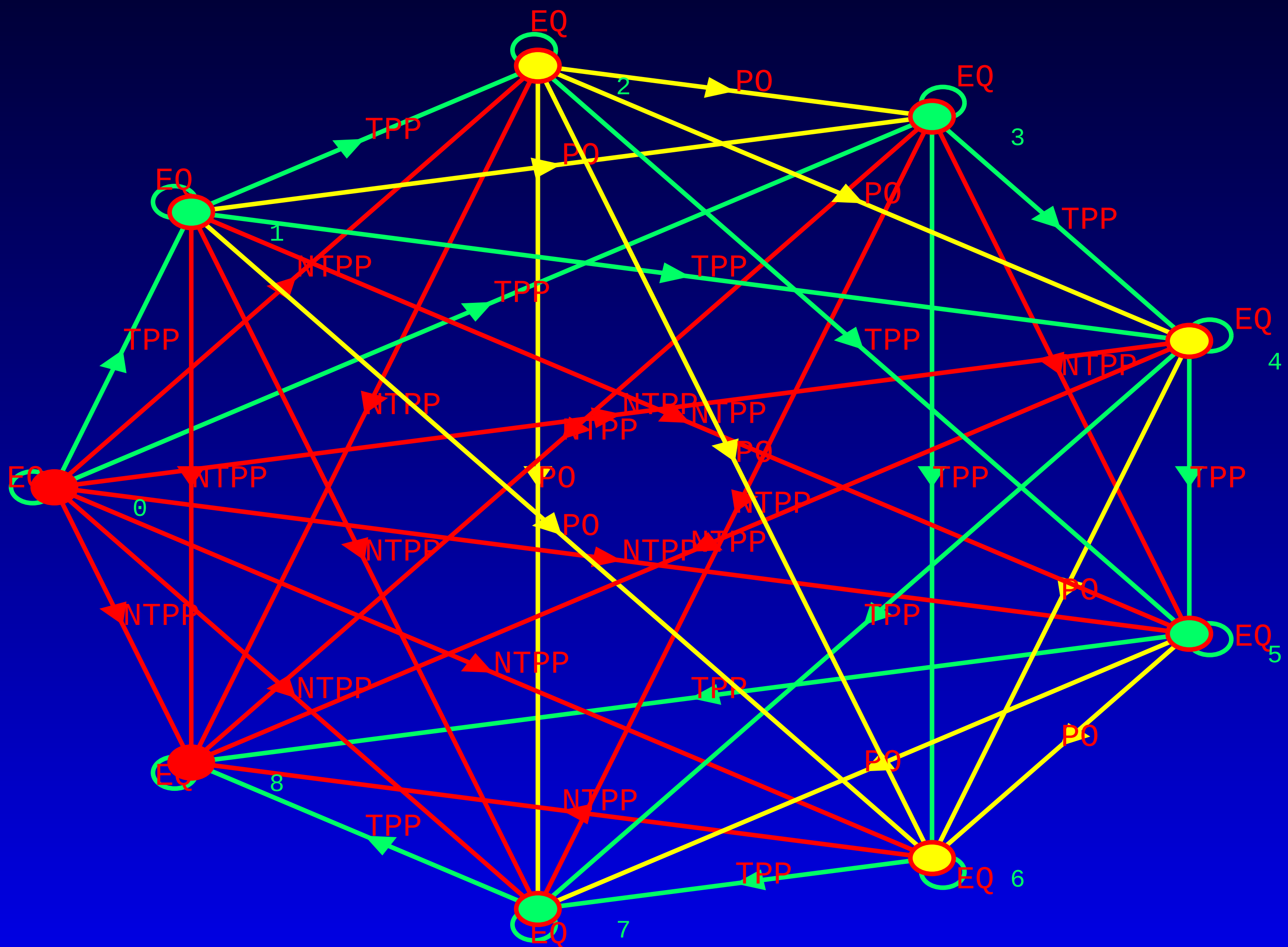


Representing DISK: Setting 1

- What can be done with RacerPro left alone?
- Remodeling of thematic categories in a TBox
  - concept_for_key$(2224) =_{def} park$
- Representation of the map as an ABox $\mathcal{A}$
  - For each map object i with key n, add i : concept_for_key(n) to $\mathcal{A}$
  - Represent dedicated metric aspects in the CD $(i : \exists(area). =_{123456})$
  - Compute qualitative RCC8 relationships: (i, j) : EC, (i, k) : TPPI, …

$\Rightarrow$ RCC8 network in the ABox (network is always consistent)

Illustration: DISK **ABox**



Querying the DISK (Setting 1)

- Simple spatio-thematic QL: instance retrieval queries
- concept_instances$(park_with_lake) = \{i, \ldots\}$
- $i \in$ concept_instances$(park \sqcap (\exists NTPPI.lake \sqcup \exists TPPI.lake))$
- If $\{\ i : park, k : lake, j : meadow,$
  $(i, j) : TPPI, (j, k) : NTPPI\} \ \subseteq \mathcal{A},$
  then also $(i, k) : NTPPI \in \mathcal{A}$ (due to map)
- RCC roles in ABox must be closed: for each i, add $i : (\leq\ n\ R) \sqcap (\geq\ n\ R)$
- $n =_{def} |\{\, j \mid (i, j) : R \in \mathcal{A}\,\}|$

Querying the DISK (Setting 1) (2)

- concept_instances$(bird_sanctuary) = \{i, \ldots\}$
- $bird\_sanctuary \doteq park \sqcap \forall NTPPI.\neg building \sqcap \forall TPPI.\neg building$

$$\mathcal{A} \cup \{(i, k) : NTPPI\} \cup$$

$$\{i : (\leq_1 \; TPPI) \sqcap (\geq_1 \; TPPI), i : (\leq_1 \; NTPPI) \sqcap (\geq_1 \; NTPPI), \ldots\} \cup$$

$$\{i : (\neg park \sqcup ((\exists TPPI.building) \sqcap (\exists NTPPI.building)))\}$$

- is unsatisfiable, if $\{building, park, meadow\}$ are mutually disjoint
- Problems in the TBox: incomplete subsumption relationships (not a problem for query answering)
- Moreover, nRQL can be used

Querying the DISK with nRQL

- nRQL offers classical negation and NAF
- For which living areas can it be proven that there are no adjacent freeways?

$$ans(?living\_area) \leftarrow living\_area(?living\_area), \forall adjacent.\neg freeway(?living\_area)$$

- Requires RCC closed ABox and disjointness axioms, as just discussed (not so good . . .)
- Living areas with no known adjacent freeways?

$$ans(?living\_area) \leftarrow living\_area(?living\_area), \backslash(\pi(?living\_area)\ adjacent(?living\_area, ?freeway), freeway(?freeway))$$

nRQL **Concrete Syntax**

```
Q1:
(retrieve (?x)
  (and (?x living-area)
       (?x (all adjacent
                (not freeway))))))

Q2:
(retrieve (?x)
  (and (?x living-area)
       (not (project-to (?x)
               (and (?x ?y adjacent)
                    (?y freeway)))))))
```

Extentible Substrate QL Framework

- nRQL is a specialized "Substrate QL" for substrates S of type ABox
- Two kinds of atoms: unary $C(x)$, binary $R(x,y)$
- x, y: Individuals, variables (with act. dom. sem.)
- Extensions of atoms on substrate S:
$C(x)^{\mathcal{E}} =_{def} \{ \, i \mid S \models C(i) \, \}$,
$R(x,y)^{\mathcal{E}} =_{def} \{ \, (i,j) \mid S \models R(i,j) \, \}$.
- Due to NAF: $\models_{NAF}$ instead of $\models$
- Complex queries: Relational operators `AND` ("$\times$"), `UNION` ($\cup$), `NEG` ($\backslash$), `PROJECT-TO` (π)
- New atoms can be added: $\models$ must be defined for $S \times atom$, $\Rightarrow$ very flexible!

Problems with Setting 1

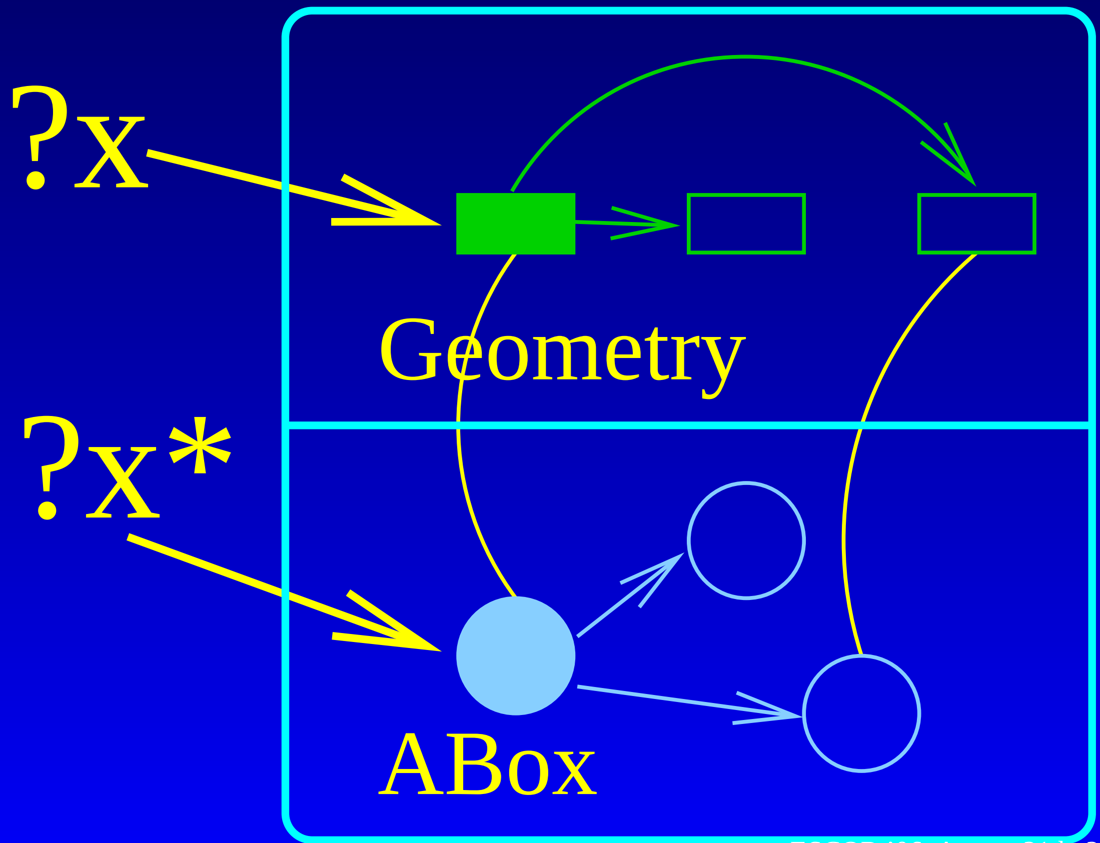
- n^2 size of generated ABoxes (29 million role assertions with DC, 19.880 without DC)
- Missing practically relevant query atoms (e.g., distance queries)
- Qualitative representation of “spatial data” in an ABox
- “Closed domain reasoning” requiered $\Rightarrow$ missuse of the DL system (open domain reasoning)
- Geometry needed anyway, at least for presentation purposes
- $\Rightarrow$ Motivates hybrid representation and query language

Representing the DISK: Setting 2

- Map Substrate: $(ABox, SBox, *)$
  - Substrate 1: ABox - thematic aspects
  - Substrate 2: SBox - map geometry
  - $*$: part. inject. mapping from nodes in S_1 to S_2

Representing the DISK: Setting 2

- Map Substrate: $(ABox, SBox, *)$
  - Substrate 1: ABox - thematic aspects
  - Substrate 2: SBox - map geometry
  - $*$: part. inject. mapping from nodes in S_1 to S_2



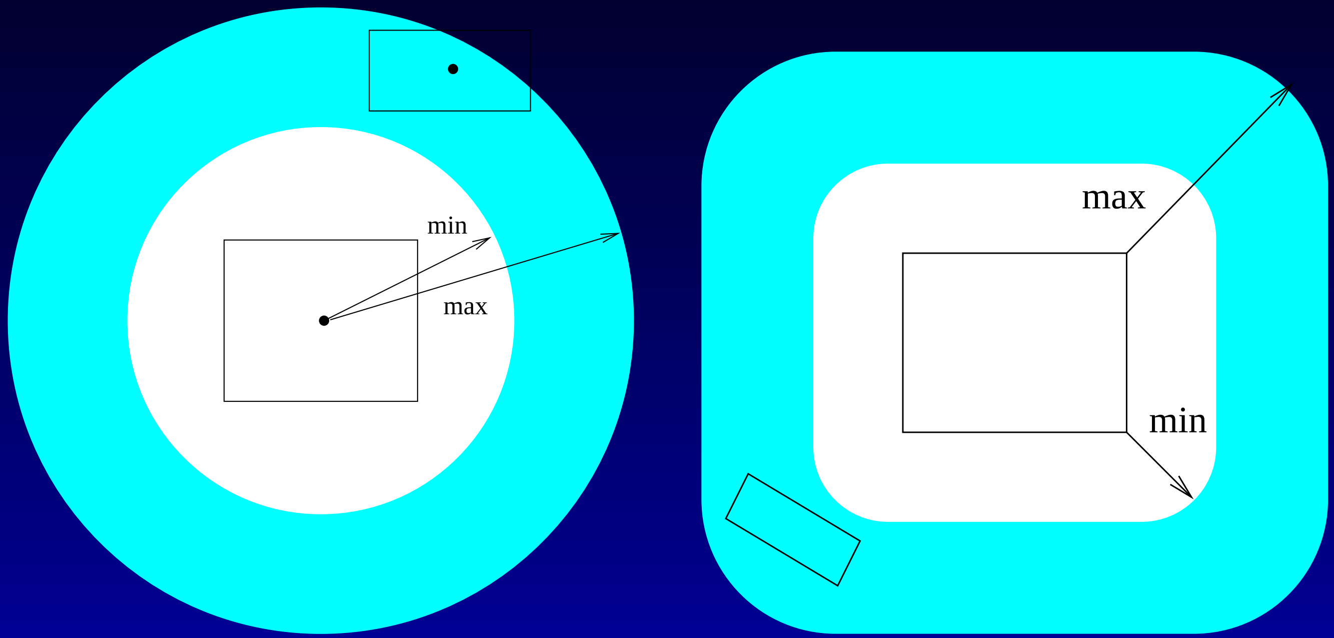
Representing the DISK: Setting 2

- Map Substrate: $(ABox, SBox, *)$
  - Substrate 1: ABox - thematic aspects
  - Substrate 2: SBox - map geometry
  - $*$: part. inject. mapping from nodes in S_1 to S_2
- ⊕ Non-symbolic spatial binary query atoms
- ⊕ On-demand computation and inspection of spatial aspects
- ⊕ Dedicated index structures
- ⊕ Closed world reasoning in $SBox$
- ⊕ Simple model checking
- ⊖ No more “reasoning” on spatial aspects

Spatial nRQL- SnRQL

- Two sorts of atoms in SnRQL
- ABox atoms:
  - Variables range over ABox individuals
  - Atoms as in nRQL (concept, role, constraint query atoms)
- Spatial atoms:
  - Variables range over SBox individuals
  - RCC atoms
  - Geometric attributes: area, length, ...
  - Metric relationships: range queries, epsilon queries, ...
- Variables are bound in parallel, bindings reflect the “*” mapping

Spatial nRQL- SnRQL



Range Query

Epsilon Query

SNRQL Concrete Syntax

```
(retrieve (?*x ?*y)
     (and (?*x
            (and living-area
                (all living-quality
                     first-class-area)))
          (?y ?x (:inside-distance 750))
          (?*y subway-station)
          (?x ?y :adjacent)
          (?*y golf-club)
          (?y (:area 10000000 nil))))
```

Hybridness can be made transparent

- Add end-user syntax for DLMAPS system: user must not be aware of the details of the map representation
- $ans(?x, ?y) \leftarrow$

  $park(?x), contains(?x, ?y), lake\_or\_pond(?y)$
- $lake\_or\_pond(?y) \rightarrow (lake \sqcup pond)(? * y)$
- $contains(?x, ?y) \rightarrow NTPPI(?x, ?y) \vee TPPI(?x, ?y)$

Problems with the Approach

- Theoretical problems: No spatial reasoning
- Practical problems (perspective: IS designer)
  - ABox / SBox separated, communication overhead (caches required)
  - Hybrid QL required
  - IS designers probably do not want to implement a query answering engine, thus:
    - Split hybrid query into subqueries, send to different sources, combine sub-results
    - Probably bad performance (no overall query optimization, communication overhead, combination of results, . . .)

$\Rightarrow$ Shows a way, but too hard to realize

Hybrid Substrates in RACERPRO

- Thus, in order to avoid these problems, this functionality should be put into the DL system

$\Rightarrow$ Hybrid substrates for RACERPRO

- Makes functionality available for other IS designers
- Compensate for representational deficiencies of the ABox
  - Data substrate: stores told value data (from CD of ABox or OWL documents, enabled retrieval facilities)
  - RCC substrate: associate an RCC network with an ABox (next) $\Rightarrow$ limited form of spatial reasoning

The Data Substrate

```
(retrieve (?x ?*name ?*age)
   (and (?x (and |http://...#person|
                 (an |http://...#age|)))
        (?*x ?*name |http://...#name|)
        (?*name ( (:predicate (search "wessel"))
                  ((:predicate (search "michael"))
                   (:predicate (search "achim")))))
        (?*x ?*age |http://...#age|)
        (?*age ((:predicate (< 40))))))
```

- New sort of variables: `*?x (*$?x)`, ranging over data nodes
- Data nodes can also be data values in OWL documents
- Data nodes/edges have descriptive labels: kind, role, property, . . .
- Notion of entailment for labels of nodes/edges
- Data query atoms are in pos. CNF & contain literals and predicates.

The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in <u>all</u> models (“certain answer”)

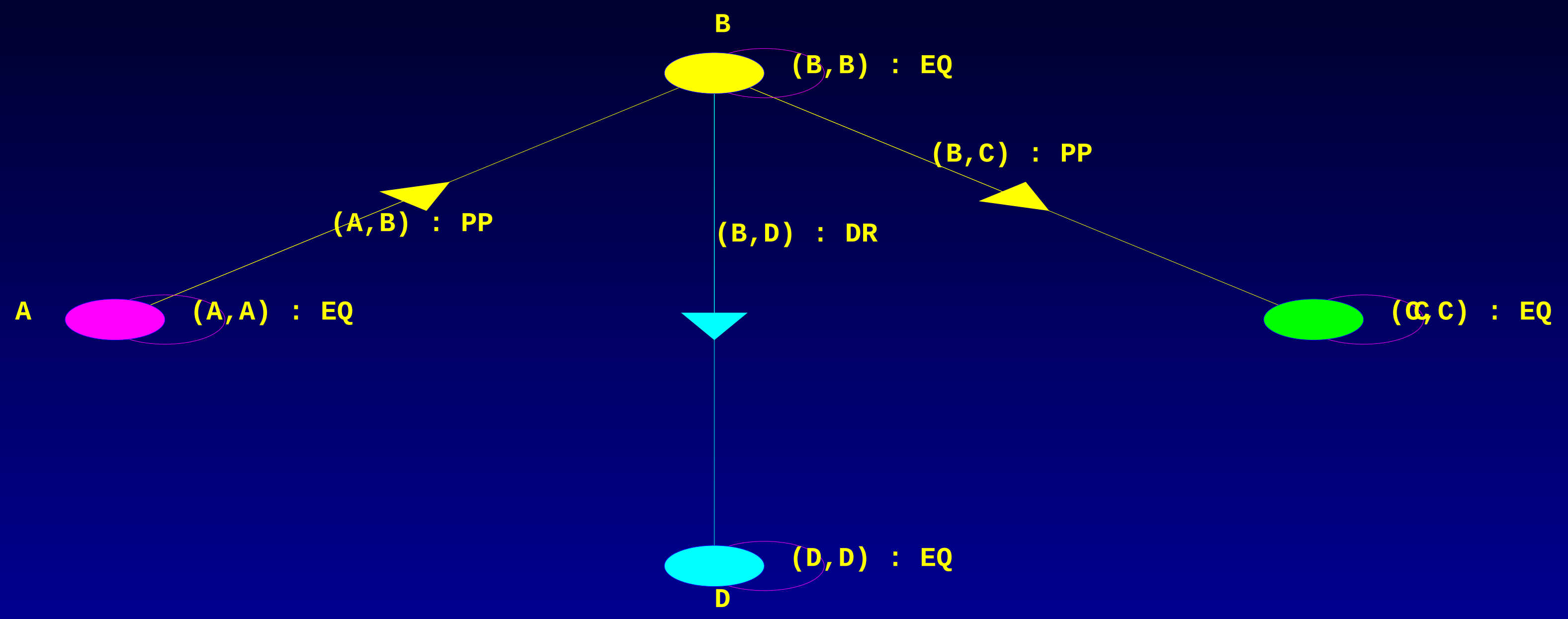
The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?

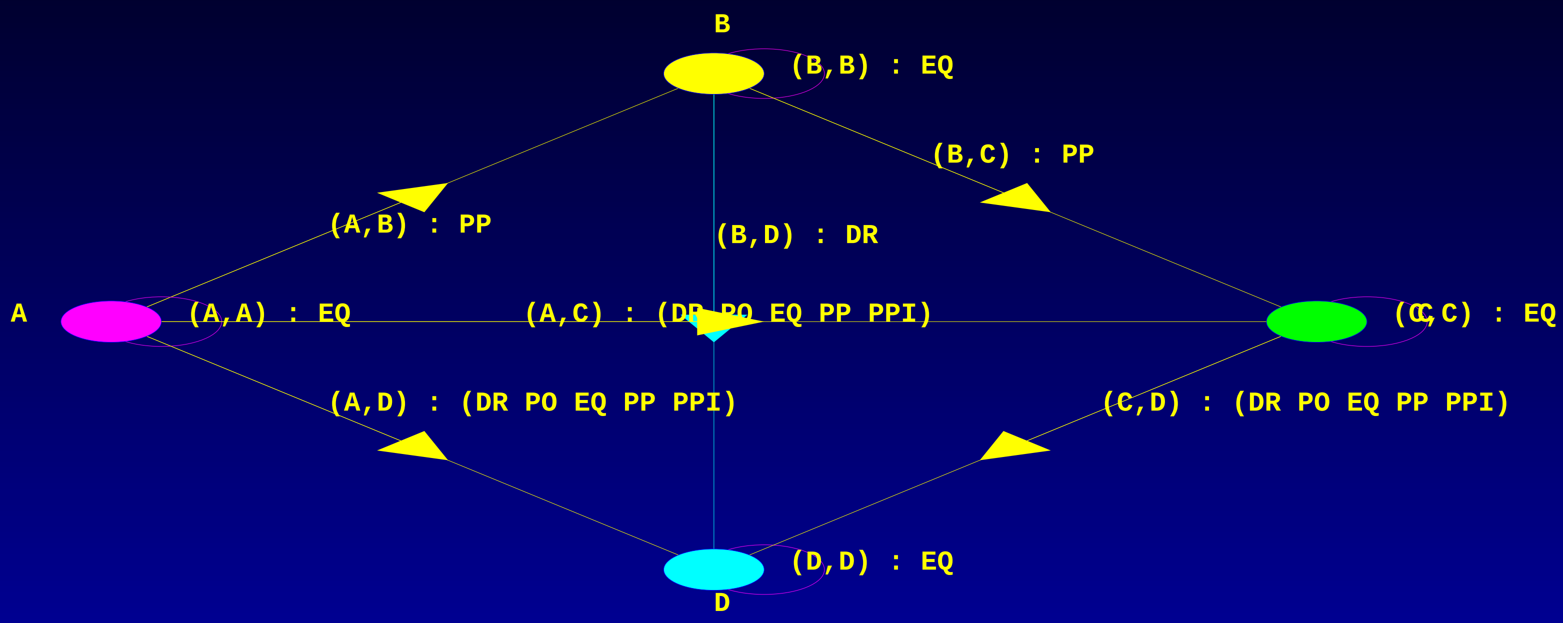
The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations

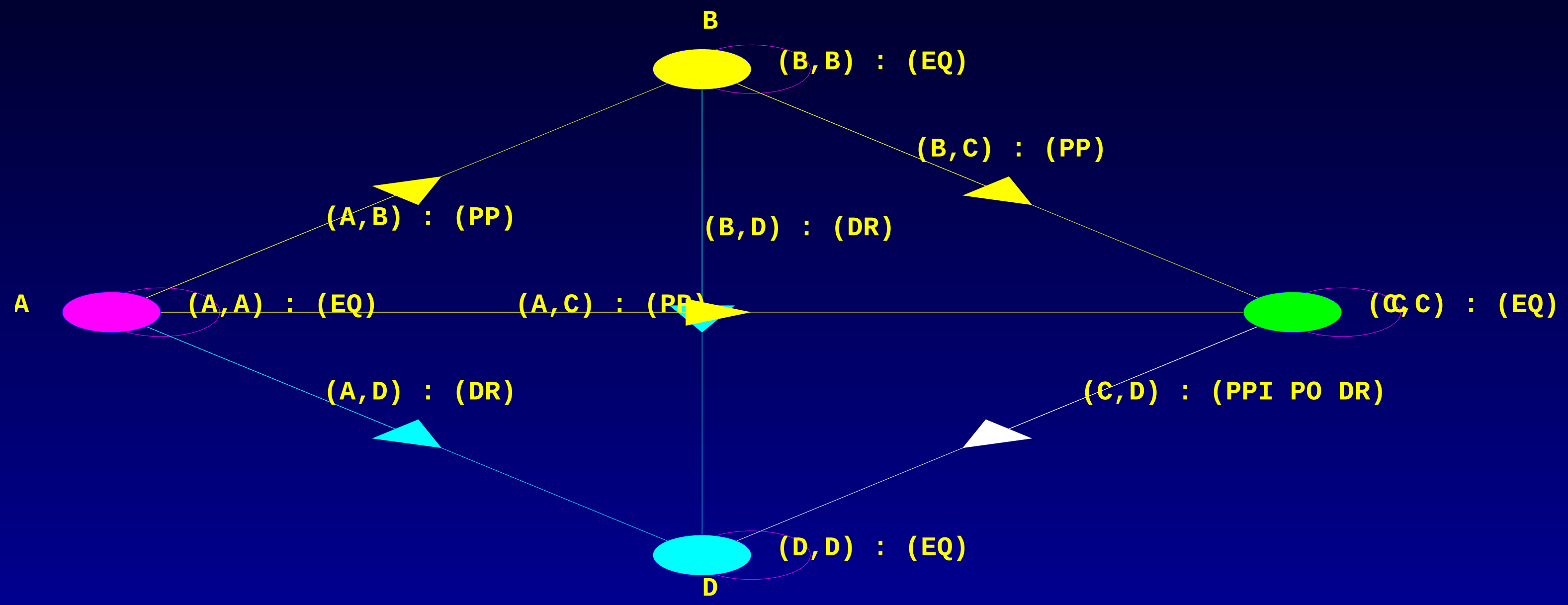
The RCC Substrate



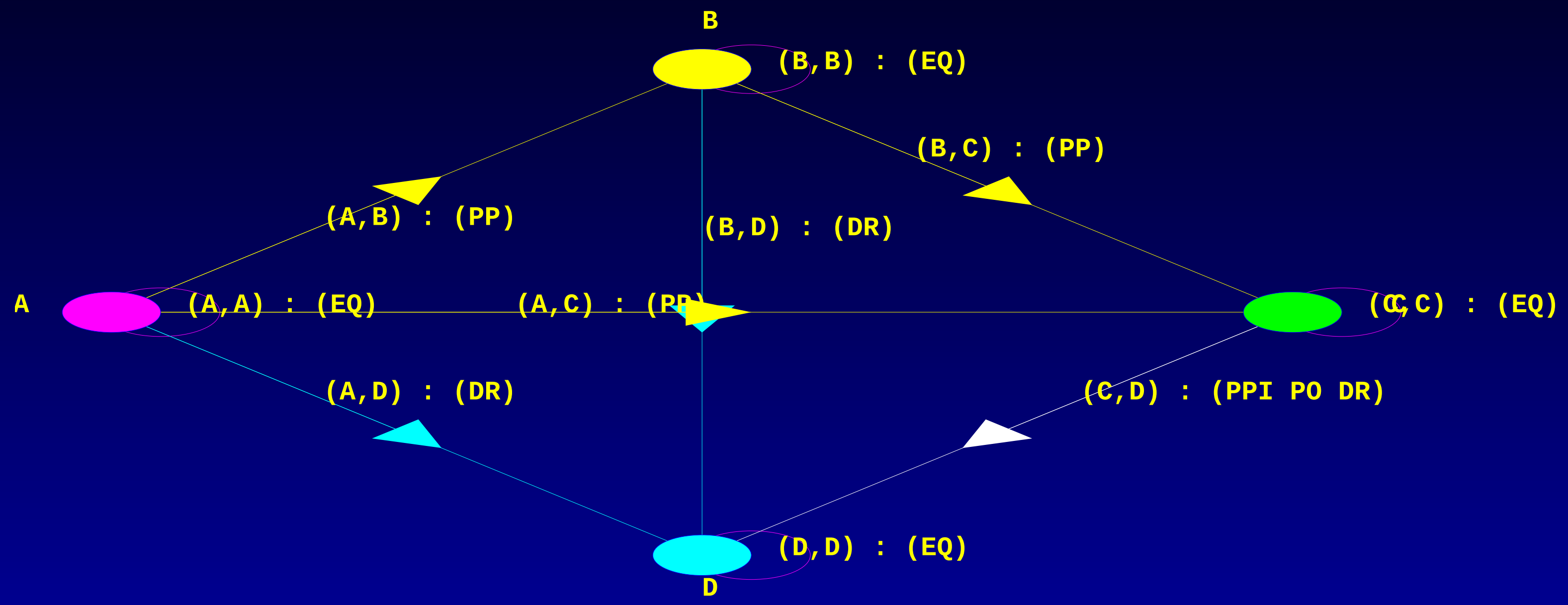
The RCC Substrate



The RCC Substrate



The RCC Substrate



$$\models DR(A, D)$$
$$\models PP(A, C)$$
$$\models \{PPI, PO, DR\}(C, D)$$
$$\models \ldots$$

The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable

The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable
- Setting 3 for DLMAPS

The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable
- Setting 3 for DLMAPS
- (Proof of concept implementation)

The RCC Substrate

- Substrate QL based on notion of logical consequence: a binding to a variable is only established if this binding holds in all models ("certain answer")
- Question: Holds $R(x, y)$ in all models of the RCC network $\mathcal{R}$?
- Not so easy, since $\mathcal{R}$ (or R) can contain non-base relations
- Reduction to satisfiability: $R \models R(x, y)$ iff $\mathcal{R} \cup (\mathcal{RCC} \setminus R)(x, y)$ unsatisfiable
- Setting 3 for DLMAPS
- (Proof of concept implementation)

Conclusion

- Spatial parts from spatio-thematic concepts removed from the TBox
- Instead, spatially aware query answering instead of TBox / concept reasoning
- Simple layering of representations can be of great value in practice
- (Possibly) Hybrid substrate QL framework allows for extensibility
- Base the IS on abstractions so that the representation can be changed easily
- (Theoretically) simple techniques can be successful in practice

Thanks

for your

attention!