FAKULTÄT FÜR INFORMATIK
Faculty of Informatics

Lightweight Spatial Conjunctive Query Answering using Keywords

Thomas Eiter, Thomas Krennwallner, Patrik Schneider

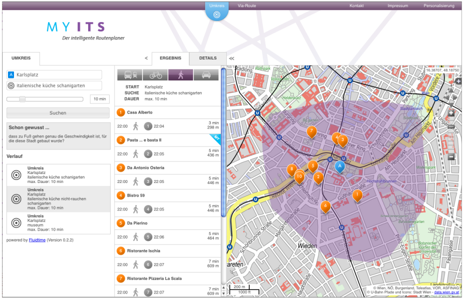
TU WIEN

May 28, 2013

Overview

Motivation

- Semantically enriched route planing
- Expressing intentions by keywords for searching POIs by
  - Categories (e.g., restaurants, supermarkts)
  - Attributes (e.g., having a guest garden and WLAN)
  - Spatial relations (e.g., Next and Within) between POIs



- Example 1: Italian Cuisine, Non-smoking, Next to, Fountain, In, Park

Aims and Challenges

- Aims
  - Support keyword-based input without prior knowledge
  - Ontology-based integration of multiple data sources. E.g., OpenStreetMap, Open Government Data, restaurant guides
  - Scalable spatial query answering (QA) $\rightarrow$ DL-Lite$_R$ (Calvanese, 2007)

- Challenges
  - Create “meaningful” queries from keywords
  - Capture semantics and algorithms for QA with spatial relations
  - Extending DL-Lite$_R$ but keeping its properties (FO-rewritability)
  - Evaluate the queries on a RDBMS $\rightarrow$ complex SQL queries

Spatial Relations I

- Spatial relations are pure set theoretic operations (Güting, 1988) $\rightarrow$ no separation between interior and boundary (as in the DE-9IM)

- Spatial relations are based on
  - Spatial objects: Γ_S
  - Geometries: $p = (p_1, \ldots, p_n)$, where $\{p_1, \ldots, p_n\} \in P_F$
  - Points Sets: $P_F \subseteq P_E \subseteq \mathbb{R}^2$, P_E is the spatial extent of the database
  - A function g: maps Γ_S to P_F

- The full point set of sp. objects is given by the function $points(g(s))$

- E.g., a line segment $s_1 = (p_1, p_2)$ is defined by the linear equation

$$points(g(s_1)) = \{\alpha p_1 + (1 - \alpha) p_2 | \alpha \in \mathbb{R}, 0 \leq \alpha \leq 1\}$$

Spatial Relations II

- We define the spatial relations by the function $points(g(x))$
  - $Equals(x, y)$: $points(g(x)) = points(g(y))$
  - $NotEquals(x, y)$: $points(g(x)) \neq points(g(y))$
  - $Inside(x, y)$: $points(g(x)) \subseteq points(g(y))$
  - $Outside(x, y)$: $(points(g(x)) \cap points(g(y))) = \emptyset$
  - $Intersect(x, y)$: $(points(g(x)) \cap points(g(y))) \neq \emptyset$
- Given a spatial relation $S(s_1, s_2)$ and a spatial database $\mathcal{D} = (P_F, g)$ over Γ_S:
  $\mathcal{D} \models S(s_1, s_2)$, if $S(s_1, s_2)$ evaluates to *true* relative to $points()$
- Captured by a first-order formula over $(\mathbb{R}^2, \leq)$

DL-Lite$_R(S)$ Syntax

- Introduce the localization of concepts and binding $\mathcal{B}$ between ABox $\mathcal{A}$ and spatial database $\mathcal{D}$ (as in Kutz, 2001)
- We have a spatio-thematic KB $\mathcal{L}_\mathcal{S} = \langle \mathcal{T}, \mathcal{A}, \mathcal{D}, \mathcal{B} \rangle$
- We (mildly) extend DL-Lite$_R$ with new complex concepts:
  $C ::= B \mid \neg B \mid (loc\ A) \mid (loc_s\ A), \quad s \in \Gamma_S$
  - $(loc\ A)$: the individuals in A can have a spatial extension
  - $(loc_s\ A)$: the individuals in A have the extension s

- Example 2:
  $Park$, $CityParkCafe$ (concepts);
  $poly$ (the spatial object for "City Park");
  $odeon$, cp (the individuals Odeon and City Park);
  $Park \sqsubseteq (loc\ Park)$, $CityParkCafe \sqsubseteq (loc_{poly} Park)$ (TBox asr.);
  $CityParkCafe(odeon)$, $Park(cp)$ (ABox assertions);
  $(cp, poly)$ (Binding)

DL-Lite$_R(\mathcal{S})$ Syntax

- Introduce the localization of concepts and binding $\mathcal{B}$ between ABox $\mathcal{A}$ and spatial database $\mathcal{D}$ (as in Kutz, 2001)
- We have a spatio-thematic KB $\mathcal{L}_\mathcal{S} = \langle \mathcal{T}, \mathcal{A}, \mathcal{D}, \mathcal{B} \rangle$
- We (mildly) extend DL-Lite$_R$ with new complex concepts:
  $C ::= B \mid \neg B \mid (loc\ A) \mid (loc_s\ A), \quad s \in \Gamma_\mathcal{S}$
  - $(loc\ A)$: the individuals in A can have a spatial extension
  - $(loc_s\ A)$: the individuals in A have the extension s

- Example 2:
  $Park$, $CityParkCafe$ (concepts);
  $poly$ (the spatial object for "City Park");
  $odeon$, cp (the individuals Odeon and City Park);
  $Park \sqsubseteq (loc\ Park)$, $CityParkCafe \sqsubseteq (loc_{poly} Park)$ (TBox asr.);
  $CityParkCafe(odeon)$, $Park(cp)$ (ABox assertions);
  $(cp, poly)$ (Binding)

DL-Lite$_R(\mathcal{S})$ Semantics I

- Transform $\mathcal{L}_\mathcal{S}$ into a DL-Lite$_R$ KB $\mathcal{K}_\mathcal{S}$ and show that the models correspond
- An interpretation of $\mathcal{L}_\mathcal{S}$ is $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I}, b^\mathcal{I} \rangle$, where $b^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Gamma_\mathcal{S}$ is a partial function that assigns a location to some individuals
- We define the interpretation function for $\mathcal{L}_\mathcal{S}$ (extend the semantics of DL-Lite$_R$)
  $(loc\ A)^\mathcal{I} \supseteq \{e \in \Delta^\mathcal{I} \mid e \in A^\mathcal{I} \wedge \exists s \in \Gamma_\mathcal{S} : (e, s) \in b^\mathcal{I}\}$ and
  $(loc_s\ A)^\mathcal{I} = \{e \in \Delta^\mathcal{I} \mid e \in A^\mathcal{I} \wedge (e, s) \in b^\mathcal{I}\}$
- Transformation from $\mathcal{L}_\mathcal{S}$ to $\mathcal{K}_\mathcal{S}$:
  - Add $C_\mathcal{T}$ as a spatial top concept
  - Add C_s for every $s \in \Gamma_\mathcal{S}$
  - Replace $(loc\ A)$ with $C_\mathcal{T} \sqcap A$
  - Replace $(loc_s\ A)$ with $C_s \sqcap A$
  - Add the axioms $C_s \sqsubseteq C_\mathcal{T}$ and $C_s \sqsubseteq \neg C_{s'}$ for all $s \neq s' \in \Gamma_\mathcal{S}$
  - Add $C_s(a)$ for every $(a, s) \in \mathcal{B}$, but not $\neg C_s(a)$ for $(a, s) \notin \mathcal{B}$

DL-Lite$_R(S)$ Semantics II

- Transformation of Example 2:

	$\mathit{TopFeature}$ is added as $C_{\mathcal{T}}$
$\mathit{Park} \sqsubseteq (\mathit{loc}\ \mathit{Park})$	$\mathit{Park} \sqsubseteq (\mathit{TopFeature} \sqcap \mathit{Park})$
$\mathit{CityParkCafe} \sqsubseteq (\mathit{loc}_{\mathit{poly}} \mathit{Park})$	$\mathit{CityParkCafe} \sqsubseteq (C_{\mathit{poly}} \sqcap \mathit{Park})$,
	$C_{\mathit{poly}} \sqsubseteq \mathit{TopFeature}$
$(\mathit{cp}, \mathit{polycp})$	$C_{\mathit{poly}}(\mathit{cp})$

- The models of $\mathcal{L}_S$ and $\mathcal{K}_S$ correspond with the same domain, concepts, and roles:
  (i) if $\mathcal{I} \models \mathcal{L}_S$, then $\mathcal{I}' \models \mathcal{K}_S$ where $C_s^{\mathcal{I}'} = \{e \in \Delta^{\mathcal{I}} \mid (e, s) \in b^{\mathcal{I}}\}$ and $C_{\mathcal{T}}^{\mathcal{I}'} = \bigcup_{s \in \Gamma_S} C^{\mathcal{I}'}$ $(= dom(b^{\mathcal{I}}))$
  (ii) if $\mathcal{I}' \models \mathcal{K}_S$, then $\mathcal{I} \models \mathcal{L}_S$ where $b^{\mathcal{I}} = \{(e, s) \mid e \in C_s^{\mathcal{I}'}\}$ and $(\mathit{loc}\ A)^{\mathcal{I}} = C_{\mathcal{T}}^{\mathcal{I}'} \cap A^{\mathcal{I}'}$

- Proposition 1:
  Satisfiability checking and conjunctive query (CQ) answering for ontologies in DL-Lite$_R(S)$ is FO-rewritable.

DL-Lite$_R(S)$ Semantics II

- Transformation of Example 2:

	$\mathit{TopFeature}$ is added as $C_{\mathcal{T}}$
$\mathit{Park} \sqsubseteq (\mathit{loc}\ \mathit{Park})$	$\mathit{Park} \sqsubseteq (\mathit{TopFeature} \sqcap \mathit{Park})$
$\mathit{CityParkCafe} \sqsubseteq (\mathit{loc}_{\mathit{poly}} \mathit{Park})$	$\mathit{CityParkCafe} \sqsubseteq (C_{\mathit{poly}} \sqcap \mathit{Park})$,
	$C_{\mathit{poly}} \sqsubseteq \mathit{TopFeature}$
$(\mathit{cp}, \mathit{polycp})$	$C_{\mathit{poly}}(\mathit{cp})$

- The models of $\mathcal{L}_{\mathcal{S}}$ and $\mathcal{K}_{\mathcal{S}}$ correspond with the same domain, concepts, and roles:
  (i) if $\mathcal{I} \models \mathcal{L}_{\mathcal{S}}$, then $\mathcal{I}' \models \mathcal{K}_{\mathcal{S}}$ where $C_s^{\mathcal{I}'} = \{e \in \Delta^{\mathcal{I}} \mid (e, s) \in b^{\mathcal{I}}\}$ and $C_{\mathcal{T}}^{\mathcal{I}'} = \bigcup_{s \in \Gamma_S} C^{\mathcal{I}'}\ (= \mathit{dom}(b^{\mathcal{I}}))$
  (ii) if $\mathcal{I}' \models \mathcal{K}_{\mathcal{S}}$, then $\mathcal{I} \models \mathcal{L}_{\mathcal{S}}$ where $b^{\mathcal{I}} = \{(e, s) \mid e \in C_s^{\mathcal{I}'}\}$ and $(\mathit{loc}\ A)^{\mathcal{I}} = C_{\mathcal{T}}^{\mathcal{I}'} \cap A^{\mathcal{I}'}$

- Proposition 1:
  Satisfiability checking and conjunctive query (CQ) answering for ontologies in DL-Lite$_R(S)$ is FO-rewritable.

Spatial Conjunctive Queries

- Spatial Conjunctive Queries (SCQ) $\rightarrow$ extend CQ with spatial atoms for spatial relations
- A SCQ $q(\mathbf{x})$ over $\mathcal{L}_{\mathcal{S}}$ is as:
  $O_1(\mathbf{x}, \mathbf{y}) \wedge \cdots \wedge O_n(\mathbf{x}, \mathbf{y}) \wedge S_1(\mathbf{x}, \mathbf{y}) \wedge \cdots \wedge S_m(\mathbf{x}, \mathbf{y})$
  - $\mathbf{x}$ are distinguished variables
  - $\mathbf{y}$ are non distinguished variables or individuals
  - $O_i(\mathbf{x}, \mathbf{y})$ is a concept or role from $\mathcal{T}$
  - $S_i(\mathbf{x}, \mathbf{y})$ is a spatial relation
- Example 3:
  $q(x) : Restaurant(x) \wedge NextTo(x, y) \wedge Fountain(y) \wedge Within(x, z) \wedge Park(z)$

Spatial Conjunctive Queries

- Spatial Conjunctive Queries (SCQ) $\rightarrow$ extend CQ with spatial atoms for spatial relations

- A SCQ $q(\mathbf{x})$ over $\mathcal{L}_{\mathcal{S}}$ is as:
  $O_1(\mathbf{x}, \mathbf{y}) \wedge \cdots \wedge O_n(\mathbf{x}, \mathbf{y}) \wedge S_1(\mathbf{x}, \mathbf{y}) \wedge \cdots \wedge S_m(\mathbf{x}, \mathbf{y})$
  - $\mathbf{x}$ are distinguished variables
  - $\mathbf{y}$ are non distinguished variables or individuals
  - $O_i(\mathbf{x}, \mathbf{y})$ is a concept or role from $\mathcal{T}$
  - $S_i(\mathbf{x}, \mathbf{y})$ is a spatial relation

- Example 3:
  $q(x) : \mathit{Restaurant}(x) \wedge \mathit{NextTo}(x, y) \wedge \mathit{Fountain}(y) \wedge \mathit{Within}(x, z) \wedge \mathit{Park}(z)$

SCQ are FO-Rewritable

- Show that query $q(x)$ over $\mathcal{L}_\mathcal{S}$ is transformed into a union of CQ $uq(x)$ over $\mathcal{L}'_\mathcal{S} = \langle \mathcal{T}', \mathcal{A}', \mathcal{D}, \mathcal{B} \rangle$ by replacing each $S(z, z')$ with:

$$\bigvee_{s,s' \in \Gamma_S} (C_s(z) \wedge C_{s'}(z') \wedge S(s, s'))$$

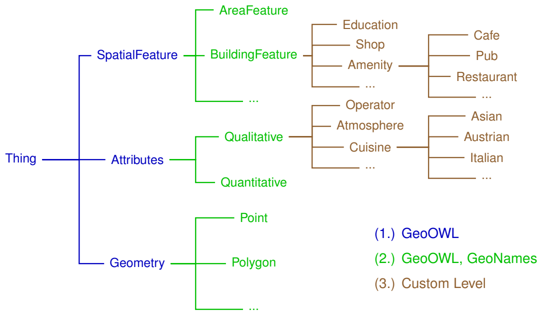
  where the C_s' are fresh spatial concepts for spatial objects
- Answering SCQ in DL-Lite$_R(S)$ is FO-rewritable (Proposition 2), by
  - The semantic correspondance of $\mathcal{L}_\mathcal{S}$ and $\mathcal{K}_\mathcal{S}$
  - The transformation of $q(x)$ into $uq(x)$ by replacing the spatial atoms
- We can simplify the transformation:
  - For a fixed $\mathcal{D}$, we can eliminiate $S(s, s')$
  - Replace $S(s, s')$ with a fresh concept $S_{s,s'}$ and extend $\mathcal{L}'_\mathcal{S}$ with $C_s \sqsubseteq S_{s,s'}$, if $\mathcal{D} \models S(s, s')$

Query Evaluation Algorithm

- Exponential blow up of query size regarding spatial atoms
- For the computation on a RDBMS we restrict SCQ
  - Only distinguished variables in spatial atoms
  - Acyclic queries
- Now, we can separate evaluation (by a join tree) into an ontology and a spatial query part:
  - (1) Evaluate the ontological part, by applying the standard DL-Lite$_R$ query rewriting with *PerfectRef*
  - (2) Filter the result of (1) according to the spatial atoms and bindings
- For step (2), two strategies are possible:
  - Database (O_D): using the spatial join function of a spatial-relational DBMS (single evaluation)
  - Internal (O_I): calculate the join internally by keeping intermediate results in-memory (multiple evaluation)

Meta-Model for SCQs

- Sequence of keywords: Italian Cuisine, Non-smoking, In, Park
- We take an ontology O_U as a meta-model for the generation of SCQs
- Need for a meta-model, e.g. a query $q(x) \leftarrow ItalianCuisine(x)$ would return pizza, pasta, etc.
- Three levels: GeoOWL, GeoNames, Custom level



Generation of SCQs

- Rewriting of a sequence of keywords K to a SCQ applying a set of completion rules and a nesting function
- Example 4:
  - $K_1 =$ (Italian Cuisine, Non-smoking, In, Park)
  - $K_2 = (ItalianCuisine, NonSmoking, Within, Park)$
  - $K_3 = (((SpatialFeature\ hasValue\ \ ItalianCuisine)$ $hasValue\ NonSmoking)\ Within\ Park)$
  - $K_4 = SpatialFeature(x_1) \wedge hasValue(x_1, y_1) \wedge$ $ItalianCuisine(y_1) \wedge hasValue(x_1, y_2) \wedge$ $NonSmoking(y_2) \wedge Within(x_1, x_2) \wedge\ Park(x_2)$

Refinement of SCQs

- Query rewriting with DL-Lite$_R$ can lead to exponentially larger UCQ than the original; we have very general SCQs $\rightarrow$ large UCQs
- Syntactic Connectivity by capturing the inclusion assertions in O_U
- Example of the refinement algorithm:
  - $SpatialFeature(x_1) \wedge hasValue(x_1, y_1) \wedge ItalianCuisine(y_1) \wedge hasValue(x_1, y_2) \wedge NonSmoking(y_2)$
  - $Restaurant \rightarrow \exists hasCuisine \rightarrow ItalianCuisine$ and $Restaurant \rightarrow \exists provides \rightarrow NonSmoking$ are shorter paths
  - $Restaurant$ is a subconcept of $SpatialFeature$
  - Which leads to $Restaurant(x_1) \wedge hasCusine(x_1, y_1) \wedge ItalianCuisine(y_1) \wedge provides(x_1, y_2) \wedge NonSmoking(y_2)$
- We might lose completeness with respect to the original SCQ

Scenario and Implementation

- Part of the MyITS system (http://myits.at/) with
  - Neighborhood routing
  - Via routing
- We have implemented a prototype based on
  - Java 1.6 and PostGIS 1.5.1
  - Owlgres 0.1 for the DL-Lite$_R$ rewriting
  - PostGIS functions and/or JTS Topology Suite for spatial atoms
- DL-Lite$_R$ Ontology
  - 324 concepts with 327 inclusion assertions
  - 30 roles with 19 inclusion assertions
  - 23 (resp. 25) domains (resp. ranges) of roles
- Instances (spatial objects)
  - $\approx$ 70k OSM instances
  - $\approx$ 7200 OGD instances
  - $\approx$ 3700 other instances

Experimental Setup

- Mac OS X 10.6.8 system; Intel Core i7 2.66GHz; 4 GB of RAM
- Average of five runs for query rewriting and evaluation time
- Benchmark 1 for evaluating the refinement:
  - $Q_1 : (Spar)$
  - $Q_2 : (Guest\ Garden)$
  - $Q_3 : (Italian\ Cuisine, Guest\ Garden)$
  - $Q_4 : (Italian\ Cuisine, Guest\ Garden, Wlan)$
  - $Q_5 : (Italian\ Cuisine, Guest\ Garden, Wlan, Child\ Friendly)$
- Benchmark 2 for comparing database and internal evaluation of spatial atoms:
  - $Q_6 : (Playground, Within, Park)$
  - $Q_7 : (Supermarket, Next\ To, Pharmacy)$
  - $Q_8 : (Italian\ Cuisine, Guest\ Garden, Next\ To, ATM, Next\ To, Metro\ Station)$
  - $Q_9 : (Playground, Disjoint, Park)$

Results

- Benchmark 1 with unrefined in parentheses (time in secs):

	Instances	Query Size	Time
Q_1	106 (109)	438 (2256)	1.66 (4.96)
Q_2	1623 (1623)	51 (2256)	1.23 (5.59)
Q_3	204 (—^{s})	28 (71712)	1.14 (—^{s})
Q_4	32 (—^{m})	56 (—^{m})	1.48 (—^{m})
Q_5	3 (—^{m})	112 (—^{m})	4.11 (—^{m})

- Benchmark 2 for O_I (internal) and O_D (database) (time in secs):

	Instances	Query Size	Time	
			O_I	O_D
Q_6	93	2	1.54	19.3
Q_7	378	4	2.22	—^{t}
Q_8	26	30	3.37	—^{t}
Q_9	151	2	2.02	—^{t}

- Observations
  - Refinement is essential for feasibility of our approach
  - In case of Q_1, we lose completeness
  - Large performance difference between O_I and O_D

Conclusion and Outlook

- Conclusion
  - DL-Lite$_R$ with spatial objects using point-set relations
  - Evaluate (restricted) CQs with spatial atoms over an RDBMS
  - Provide a technique for generation of spatial SCQs from keywords
  - Implemented a prototype and performed experiments to evaluate it in a real-world scenario
- Future Research
  - Extend the ontology and query language, e.g., $\mathcal{EL}$ or Datalog$^{\pm}$
  - Extend the point set model to the DE-9IM
  - Address restriction on query evaluation algorithm
  - Investigate further query generation and refinement
  - Compare to similar approaches as Geo-SPARQL engines

Conclusion and Outlook

Thanks for your attention!

Generation of SCQs Ext.

- Rewriting of a sequence of keywords K in three steps to a SCQ
- (1) Replace every keyword in K with concepts of O_U or a predefined spatial predicate $\rightarrow K'$
- (2) Apply of completion rules on $K' \rightarrow K''$, some examples:
  - If $C_1 \sqsubseteq \mathit{QualAttribute}$ and $C_2 \sqsubseteq \mathit{QualAttribute}$ rewrite to $((\mathit{SpatialFeature}\ \ \mathit{hasValue}\ \ C_1)\ \ \mathit{hasValue}\ \ C_2)$
  - If $E_1 \sqsubseteq \mathit{SpatialFeature}$ or E_1 is a subquery, $E_2 \sqsubseteq \mathit{SpatialFeature}$ or E_2 is subquery, and S is a spatial predicate, rewrite to $((E_1)\ \ S\ \ E_2)$;
- (3) Generate a SCQ from K'' by the nesting function $f(K'') = (\cdots((C_1(x_1) \wedge E_{1,1}(x_1, y_1) \wedge E_{1,2}(y_1)) \wedge \chi_2) \wedge \cdots) \wedge \chi_n$ where $\chi_i = E_{i,1}(\vartheta(E_{i-1,1}), y_i) \wedge E_{i,2}(y_i)$ and $E_{i,1}$ (resp. $E_{i,2}$) is either empty, a role, or a spatial (resp. either empty or a concept) atom

Syntactic Connectivity Ext.

- Determine the connectivity of concepts for two purposes:
  - Auto completion and combination
  - Refinement of SCQ (later)
- Based on the DL-Lite$_R$ O_U and captures the inclusion assertions:
  - Concept inclusion $M_C : C_1 \sqsubseteq C_2$; role hierarchies $M_H : R_1 \sqsubseteq R_2$
  - Role membership which covers the range (resp. domain) of a role as $M_R : \exists R^- \sqsubseteq C$ (resp. $M_D : \exists R \sqsubseteq C$)
  - Mandatory participation $M_P : C \sqsubseteq \exists R$
  - But not disjoint concepts: $C_1 \sqsubseteq \neg C_2$
- We have two types of connections:
  - $Supermarket \rightarrow_{M_C} Shop \rightarrow_{M_P} \exists hasOperator \rightarrow_{M_R} Operator$ (direct connection)
  - $GuestGarden \rightarrow_{M_C} QualVal \rightarrow_{M_P} \exists hasValue \rightarrow_{M_R} SpatialFeature \leftarrow_{M_R} \exists hasValue \leftarrow_{M_P} QualVal \leftarrow_{M_C} Wlan.$ (indirect connection)