Walk Logic as a framework for path query languages on graph databases

Jelle Hellings, Bart Kuijpers
Jan Van den Bussche, and Xiaowang Zhang
Hasselt University and transnational University of Limburg

19 March 2013

Overview

Motivation

- Expressing graph-queries
- Properties of paths, walks, ...

Route planning

We want to travel from *our office* to a *cafetaria* and from this *cafetaria* get back to the *office* using a *different route*

General logics

- First-order logic: limited to local reasoning
- Monadic second-order logic:
  - Focus on sets: *bipartite graph*

$$\exists S \exists T\, (\forall x\, (x \in S \iff x \notin T) \wedge \forall y\, edge(x,y) \implies ((x \in S \wedge y \in T) \vee (y \in T \wedge x \in S)))$$

  - Paths non-straightforward: *y is reachable from x*

$$\forall S\, [(x \in S) \wedge \forall u \forall v\, (u \in S \wedge edge(u,v) \implies v \in S) \implies y \in S]$$

  - *Nodes* versus *nodes and edges*

Specific logics

- Family of Conjunctive Regular Path Queries (CRPQs)
  - Focus on labelling of paths *('regular expression')*

$$Q(a, b) := a\pi b, (\alpha\beta + \gamma\delta)^*(\pi)$$

  - Limited reasoning between paths *('equal length')*

$$Q(\pi_1, \pi_2) := a\pi_1 b \wedge a\pi_2 b, \left[\begin{smallmatrix}\alpha \\ \beta\end{smallmatrix}\right]^* \left(\begin{smallmatrix}\pi_1 \\ \pi_2\end{smallmatrix}\right)$$

- Family of verification logics (CTL* and hybrid extensions)
  - Focus on behaviour single/independent paths

$$\mathrm{AF}(\textit{produce}\ \mathrm{U}\ \textit{break} \vee \textit{no-resources})$$

Idea: extend first-order logic

- Add *walks*
- Add *positions on walks*
- Necessary operators to compare positions

Route planning

We want to travel from *our office* to a *cafetaria* (W) and from this *cafetaria* get back to the *office* using a *different route* (W')

$$\exists W \exists W' \exists t_1^W \exists t_2^W \exists {u_1}^{W'} \exists {u_2}^{W'} \exists {u_3}^{W'}$$
$$(\mathrm{office}(t_1) \wedge t_1 < t_2 \wedge \mathrm{cafeteria}(t_2) \wedge u_1 < u_3 < u_2$$
$$\wedge\, u_1 \sim t_2 \wedge u_2 \sim t_1 \wedge \forall t_3^W (t_1 < t_3 < t_2 \implies t_3 \not\sim u_3))$$

Overview

Definitions

Definition (Directed node-labeled graph)

A directed node-labeled graph is a triple $G = (N, E, l)$:

- N is a finite set of *nodes*
- $E \subseteq N \times N$ is the set of *edges*
- $l : N \to 2^{\mathcal{AP}}$ is a node-label function

Definition (Walk)

A *walk* in G is a finite nonempty sequence $v_1 \dots v_n$ of nodes such that $(v_i, v_{i+1}) \in E$ for each $1 \leq i < n$

Definition (Path)

A *path* in G is a walk without node repetition

Walk Logic

- Quantification over *walks* and *positions on walks*
- Atomic formulae: properties on positions

$a(t)$	Node referred to by position variable t has labelling a
$t_1 \sim t_2$	Position variables t_1, t_2 refer to the same node
$t_1 < t_2$	Position variable t_1 comes before t_2 in walk W
	Position variables t_1 and t_2 *must* be of the same sort

- Logical connectives
- Optionally: syntactic sugar (quantification over nodes, $=$, $\ldots$)

Path logic: Walk Logic with *path*-semantics

- Paths are useful themselves *(Hamiltonian path)*:

$$\exists P \forall Q \forall t^Q \exists u^P \, (t \sim u)$$

- Walk logic can express walk P is a path:

$$\mathit{isPath}(P) \equiv \forall t^P \forall u^P \, (t^P \sim u^P) \implies (t^P = u^P)$$

- Set of edges can describe a path
  MSO over nodes and edges subsumes Path Logic
- Can we also express Walk Logic in Path Logic or MSO?

Overview

Walk-based Graph Properties - 1

Strongly Connected

$$\forall P \forall Q \forall t^P \forall u^Q \exists R \exists v^R \exists w^R \, (v < w \wedge t \sim v \wedge u \sim w)$$

Hamiltonian Path *(in Path Logic)*

$$\exists P \forall Q \forall t^Q \exists u^P \, (t \sim u)$$

Eulerian Trail

$$\exists W \, (W \textit{ is a trail} \wedge \textit{every edge is part of } W)$$

Walk-based Graph Properties - 2

Theorem

Weakly Connected is not expressible on directed graphs

Proof.

$$n_1 \longleftarrow n_2 \longrightarrow n_3 \longleftarrow n_4 \longrightarrow n_5 \longleftarrow n_6$$

All walks contain at most 2 nodes: *reduce to first-order logic* □

- Direction matters!
- On undirected graphs:
  Weakly Connected same way as strongly connected
  Planar Graph using Kuratowski's Theorem

Set-based Graph Properties

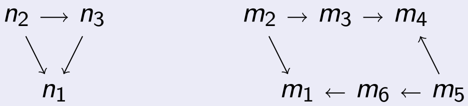
Theorem

Bipartite graph is not expressible on directed graphs

Lemma (Dénes Kőnig)

A graph is bipartite iff it does not contain an odd cycle

Proof.



All walks contain at most 3 nodes: *reduce to first-order logic* □

- MSO *can* express bipartite graph
- Is Walk Logic strictly subsumed by MSO?

Open questions

- Can we express Walk Logic in Path Logic?
- Can we express Walk Logic in MSO?
- Is Walk Logic strictly subsumed by MSO?

Eulerian Trail

Theorem

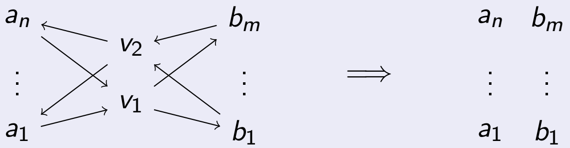
MSO(nodes, edges) and Path Logic cannot express Eulerian Trail

Lemma (well known result)

MSO cannot distinguish sets with i from sets with j elements

Proof.

For MSO: existence of Eulerian Trail in the graph



Reduces to sets A and B having the equal number of elements □

Relations with FO and MSO

Lemma (Courcelle and Engelfriet)

MSO(nodes) cannot express Hamiltonian Path

- FO and Path Logic are strictly subsumed by Walk Logic
- MSO(nodes) incomparable with Path Logic and Walk Logic
- MSO(nodes, edges) strictly subsumes Path Logic
- MSO(nodes, edges) incomparable with Walk Logic

Overview

Regular walk logic

- Conjunctive Regular Path Queries (CRPQs)
  Regular expressions over single walk
- Extended Conjunctive Regular Path Queries (ECRPQs)
  Regular expressions over n-tuples of walks
- (Extended) Regular Walk Logic ((E)RWL)[1]:
  Generalize (E)CRPQs by adding Boolean connectives

$$\exists \pi_1 \exists \pi_2 \exists v_1 \exists v_2 \left(v_1 \pi_1 v_2 \wedge v_1 \pi_2 v_2 \wedge \left[\begin{smallmatrix} \alpha \\ \beta \end{smallmatrix} \right]^* \left(\begin{smallmatrix} \pi_1 \\ \pi_2 \end{smallmatrix} \right) \right)$$

- Purpose: study open problems for (E)CRPQs

[1] In the literature this variant is also called $\text{ECRPQ}^{\neg}$

ECRPQs with *path*-semantics

- Standard (E)CRPQs work with *walk* semantics
- Efficient query evaluations
- Under *path* semantics:
  No efficient query evaluation algorithm is known

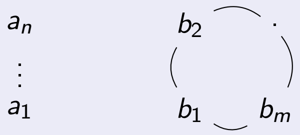
- SPARQL 1.1: property paths had path-based semantic
- Regular Path Logic (RPL) is RWL with path-based semantic

Hamiltonian path - 1

Theorem

ERWL cannot express Hamiltonian Path

Definition ($\overline{\mathbf{K}}_n \times \mathbf{C}_m$-graphs)



- n point-nodes, m nodes on an undirected cycle
- Undirected edges between every point-node and cycle-node

Lemma

$\forall$ length $l > 2$ and nodes v_1, v_2: there is a walk $v_1 \pi v_2$ of length l

Hamiltonian path - 2

Theorem (*repeated*)

ERWL cannot express Hamiltonian Path

Lemma (*repeated*)

$\forall$ *length* $l > 2$ *and nodes* v_1, v_2*: there is a walk* $v_1 \pi v_2$ *of length* l

Corollary

Using a unary alphabet for the labelling:

- *Regular expressions reduce to reachability in* $\overline{\mathbf{K}}_n \times \mathbf{C}_m$*-graphs*
- *ERWL on* $\overline{\mathbf{K}}_n \times \mathbf{C}_m$*-graphs reduces to FO-logic*

Proof (de Rougemont).

FO logic on $\overline{\mathbf{K}}_n \times \mathbf{C}_m$ graphs cannot express Hamiltonian Path. □

RWL and RPL

Theorem

ERPL is not subsumed by ERWL

Proof.

- ERWL cannot distinguish $\overline{\mathbf{K}}_n \times \mathbf{C}_m$- from $\overline{\mathbf{K}}_{n'} \times \mathbf{C}_{m'}$-graphs
- ERPL can express *'Longest path has even length'*

$$\exists \pi_1((\alpha\alpha)^* \pi_1 \land \neg \exists \pi_2 \left[\begin{smallmatrix}\alpha\\ \alpha\end{smallmatrix}\right]^* \left[\begin{smallmatrix}\bot\\ \alpha\end{smallmatrix}\right]^+ (\pi_1, \pi_2))$$

□

Additional results

- Eulerian Path not expressible in RWL or RPL
- CRPQ and star-free ECRPQ are incomparable with WL
- Path-based CRPQ is not subsumed by ECRPQ

Overview

Open Problems

- Relations with verification logic:
  - Infinite walks are the standard in verification logics
  - Can we express the verification logics in Walk Logic?
  - Walk Logic with infinite walks?
- Complexity bounds on model checking for WL:
  - WL model checking is decidable
  - Current approach has horrible complexity

Conclusion

- General walk-based reasoning on graphs
- Relates to practical graph languages
- Framework for studying expressivity