Preferentially Annotated Regular Path Queries

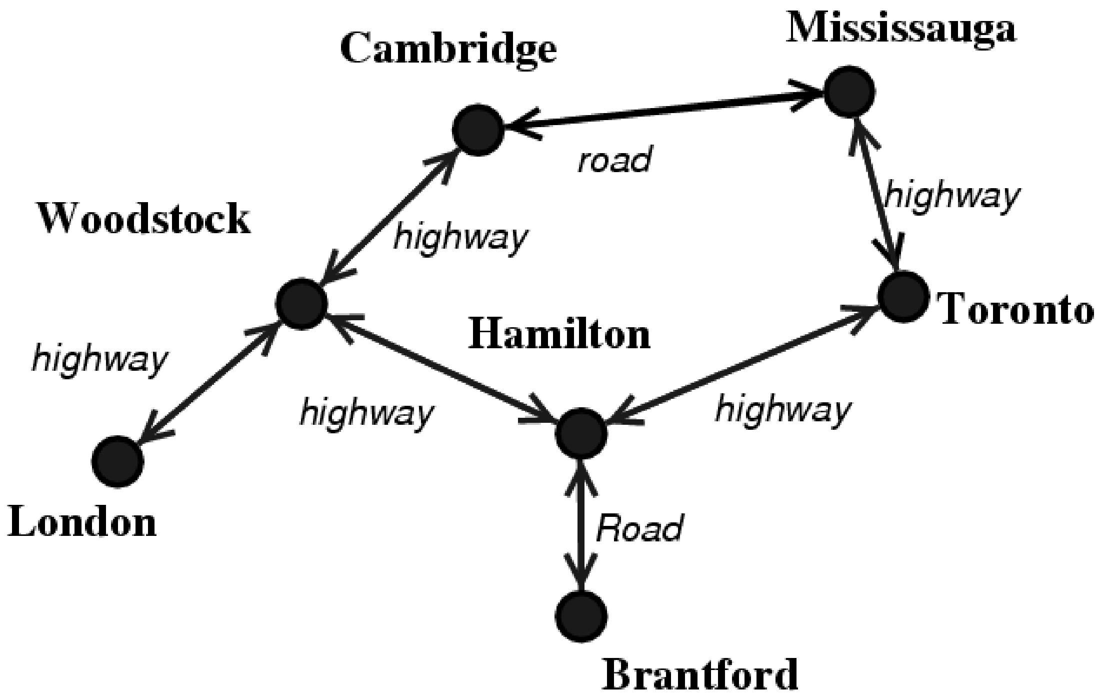
Gosta Grahne
Concordia U.
Canada

Alex Thomo
U. of Victoria
Canada

Bill Wadge
U. of Victoria
Canada

Regular Path Queries (RPQ's)

- Essentially regular expressions over database labels.



- E.g.: Q = highway*
- Meaning: Find highway routes.

RPQ's vs. Datalog

Semantically RPQ's are a fragment of Datalog.

However,

- They are easier for people to use
  - they have used reg.ex. from the early days of computers
- Important reasoning services on RPQ's are decidable
  - E.g. Containment/Equivalence is decidable for RPQ's, while for Datalog it's not.

Not any database path, but…

- Surely, I prefer highways, but can tolerate one road:

  Q = highway* . road . highway*

- Well, I prefer highways, but can tolerate up to *k* roads or city streets:

  $Q = highway^* \parallel (road + street + \varepsilon)^k$

Preferences: Boolean Way

$Q = highway^* \,||\, (road + street + \varepsilon)^k$

- Pair of objects will be produced as an answer if there exists a path between them satisfying the user query.
- There is just a "yes" or "no" qualification for the query answers.

- But, answers aren't equally good!
  - A pair of objects connected by a highway path with only 1 intervening road is a "better" answer than a pair of objects connected by a highway path with 5 intervening roads.

A simple syntactic addition

- User can annotate the symbols in the regular expressions with "markers" (typically natural numbers), which "strengthen" or "weaken" his (pattern) preferences.

$$Q = (\text{highway}:0)^* \mid\mid (\text{road}:1 + \text{street}:2 + \varepsilon)^k$$

- Meaning:
  - User ideally prefers highways,
  - then roads, which he prefers less,
  - and finally he can tolerate streets, but with an even lesser preference.

Semantics (Quantitative)

$Q = (highway:0)^* \parallel (road:1 + street:2 + \varepsilon)^k$

- The system should produce:
  - first the pairs of objects connected by highways,
  - then the pairs of objects connected by highways intervened by 1 road,
  - and so on.

- The "so on" raises some important semantical questions.
- Is a pair of objects connected by a

  highway path intervened by **two** roads

  equally good as another pair of objects connected by a

  highway path intervened by **one** street only?

- Indeed, in this example, it might make sense to consider them equally good, and "concatenate" weights by summing them up.

Qualitative Semantics

$Q = (viarail{:}0)^* \parallel (greyhound{:}1 + aircanada{:}2 + \varepsilon)^k$.

- Is now a pair of objects connected by

  a path with two greyhound segments

  equally preferable as a pair of objects connected

  with one aircanada segment?

- If the user is afraid of flying, she might want to "concatenate" edge-weights by choosing the maximum of the weights.
- Then an itinerary with no matter how many greyhound segments is preferable to an itinerary containing only one flight segment.

Hybrid Semantics

$Q = (\text{viarail}:0)^* \parallel (\text{greyhound}:1 + \text{aircanada}:2 + \varepsilon)^k$.

- Following a purely qualitative approach,

  greyhound itineraries

  are always preferable to

  itineraries containing aircanada segments,

  while these itineraries are equally preferable, no matter how many lags the flight has.

- Sometimes we need to distinguish among itineraries on the same "level of discomfort."
  - Namely, we should be able to (quantitatively) say for example that

    a direct aircanada route

    is preferable to

    an aircanada route with a stop-over,

    which again is preferable to

    an aircanada route with three lags.

Semirings

- In total, from all the above, we have four kind of preference semantics:
  - Boolean,
  - quantitative,
  - qualitative,
  - hybrid.
- In all these semantics, we:
  - aggregate ("concatenate") preference weights along edges of the paths, and then
  - aggregate path preferences when there are multiple paths connecting a pair of objects.
- We regard the preference annotations as elements of a semiring, with two operations:
  - "plus"
  - "times"
- The "times" aggregates the preferences along edges of a path, while the "plus" aggregates preferences among paths.

Semirings

$\mathcal{R} = (R, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ such that

$(R, \oplus, \mathbf{0})$ is a commutative monoid with $\mathbf{0}$ as the identity element for $\oplus$.

$(R, \otimes, \mathbf{1})$ is a monoid with $\mathbf{1}$ as the identity element for $\otimes$.

$\otimes$ distributes over $\oplus$: for all x, y, z $\in$ R,

$(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$

$z \otimes (x \oplus y) = (z \otimes x) \oplus (z \otimes y).$

Natural order $\leq$ on R: $x \leq y$ iff $x \oplus y = x$

Annotated Queries

$\mathfrak{R} = (R, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ semiring.

An $\mathfrak{R}$-annotated query Q over Δ is a function

$$Q : \Delta^* \to R.$$

We write $(w, x) \in Q$ instead of $Q(w) = x$.

- When annotated queries are given by "annotated regular expressions," we have annotated regular path queries (ARPQ's).

Annotated Automata

- Computationally, ARPQ's are represented by "annotated automata" $(P, \Delta, \mathcal{R}, \tau, p_0, F)$
- The language defined by an annotated automaton A is:

$$[A] = \{(w, x) \in \Delta^* \times R :$$

$$w = r_1 r_2 \ldots r_n,$$

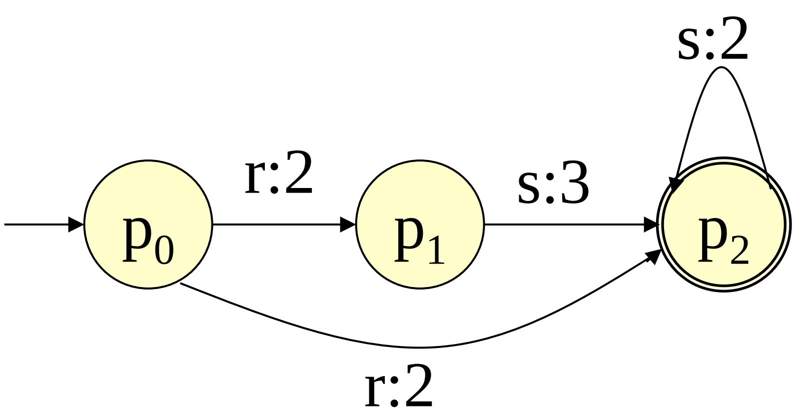
$$x = \oplus \{x_1 \otimes \ldots \otimes x_n :$$

$$(p_0, r_1, x_1, p_1) \in \tau,$$

$$\ldots$$

$$(p_{n-1}, r_n, x_n, p_1) \in \tau,$$

$$p_n \in F\}\}.$$



$(rs,4) \in [A]$

Query Answers

Given a database DB, and

an annotated Q over semiring $\mathcal{R} = (R, \oplus, \otimes, \mathbf{0}, \mathbf{1})$

Ans(Q, DB, $\mathcal{R}$) = {(a, b, x) :

$x = \oplus\{y : (w, y) \in Q$ and

w labels some path from a to b in DB}.

We have (a, b, $\mathbf{0}$) as an answer to Q, if there is no path in DB spelling some word in Q.

Preference Semirings

Boolean preferences: $\mathcal{B} = (\{T, F\}, \vee, \wedge, F, T)$

Quantitative preferences: $\mathcal{N} = (N \cup \{\infty\}, \min, +, \infty, 0)$

Qualitative preferences: $\mathcal{F} = (N \cup \{\infty\}, \min, \max, \infty, 0)$

Preference Semirings

Hybrid preferences: $\mathcal{H} = (R, \oplus, \otimes, \mathbf{0}, \mathbf{1})$

- Interface again is N.
- However, R is bigger to allow for a finer ranking

$R = \{0, 1, 1^{(2)}, \ldots, 2, 2^{(2)}, \ldots\} \cup \{\infty\}$

1, 2, . . . are shorthand for $1^{(1)}, 2^{(1)}, \ldots$

- $n^{(i)}$: n -- level of discomfort,

  i -- how many times we are "forced to endure" that level of discomfort.

$$n^{(i)} \oplus m^{(j)} = \begin{cases} n^{(i)} & \text{if } n < m \\ m^{(j)} & \text{if } n > m \\ n^{(min\{i,j\})} & \text{if } n = m, \end{cases} \qquad n^{(i)} \otimes m^{(j)} = \begin{cases} n^{(i)} & \text{if } n > m \\ m^{(j)} & \text{if } n < m \\ n^{(i+j)} & \text{if } n = m \end{cases}$$

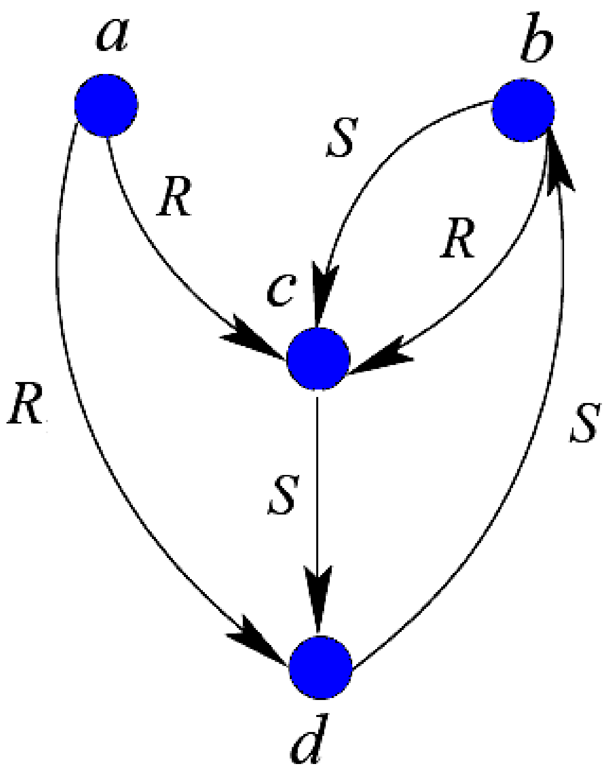
Hybrid Preferences

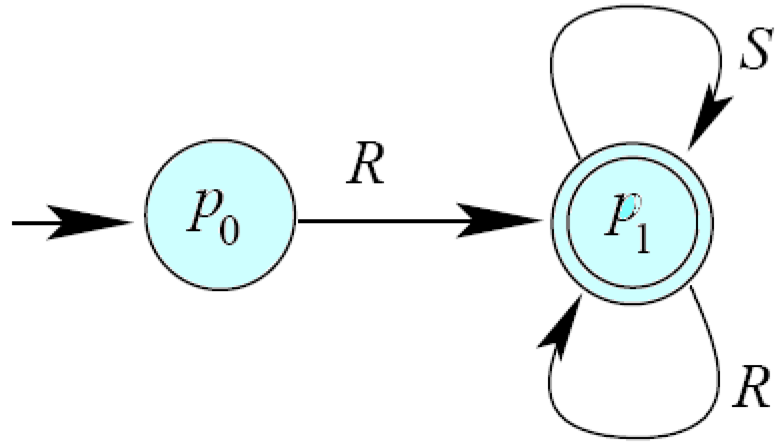
- The user, annotates query symbols with natural numbers representing his preferences.

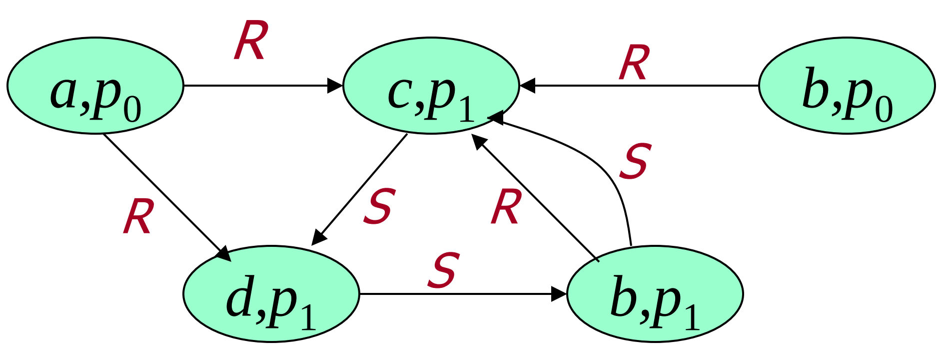
- Similarly with qualitative semantics, only database edges matched by transitions annotated with the "worst" level of discomfort will really count.

- Similarly with quantitative semantics, paths with same "worst-level of discomfort" are comparable.
  - Namely, the best path will be the one with the fewest "worst-level of discomfort" edges.

Answering of RPQ's (Classical)

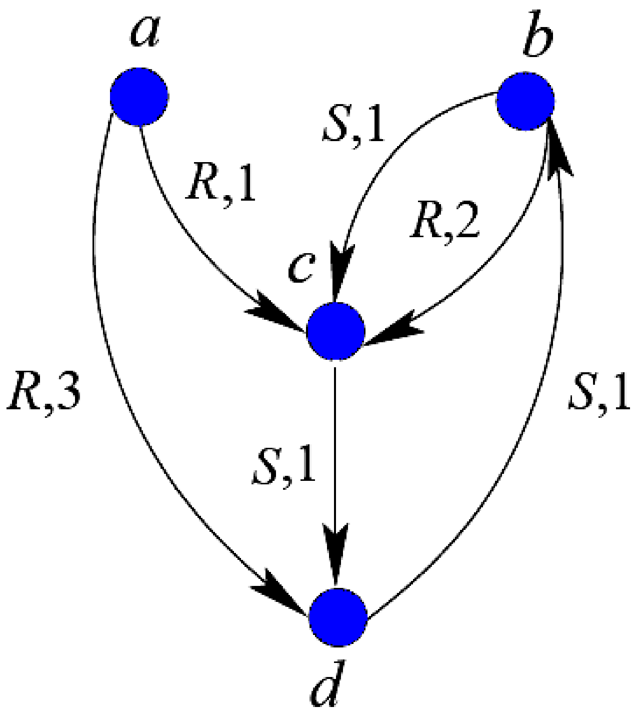


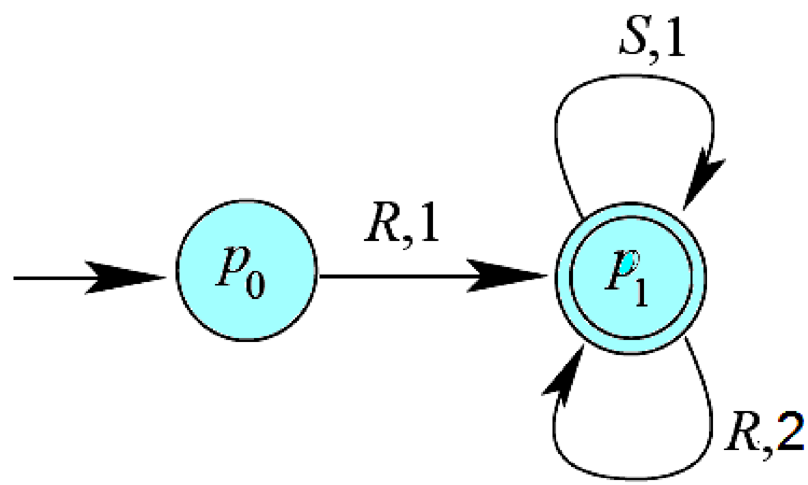


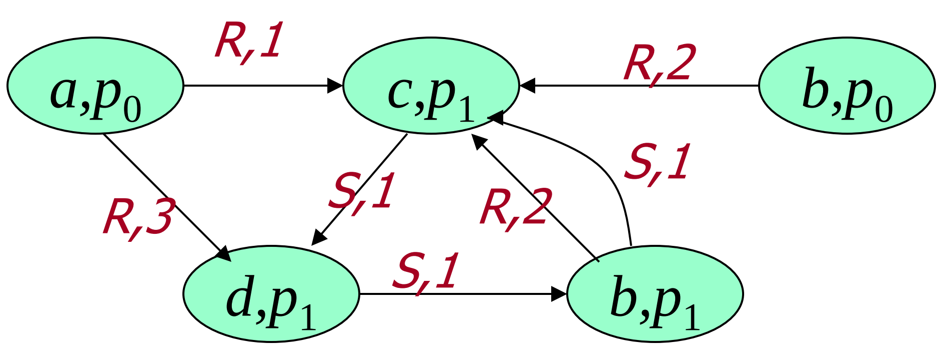


Then, do reachability in the green graph.

Answering of ARPQ's







Then, compute generalized shortest paths in the green graph.

LAV Data Integration

- No database in the classical sense.
- We have "data-sources," characterized by a definition over a "global schema": $\Delta = \{R, \ldots\}$

- Each data-source also has a name, and the set of these names constitutes the "local schema": $\Omega = \{s_1, \ldots, s_n\}$
- Mapping: $def(s_i) = S_i$

- LAV system also has a set of tuples over the local schema.

- Queries are formulated on the global schema.

- Data exists in the local schema, so, a translation from Δ to Ω has to be performed in order to be able to compute query answers.

Source Collections and Possible DB's

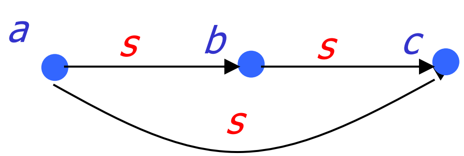
- Let $\Omega = \{s_1, . . . s_n\}$ be the local schema.
- Then, a source collection $\mathbb{S}$ is a graph database on Ω.

- *poss*($\mathbb{S}$) : Set of all databases from which the given source collection $\mathbb{S}$ might have been generated.

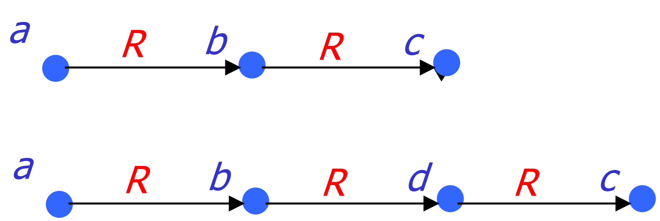
- E.g., consider $S=R^*$ and

Under sound source assumption

$\mathbb{S}$



Possible DB's



…

Certain Answer

$CAns(Q, \mathbb{S}) = \bigcap_{DB \in poss(\mathbb{S})} Ans(Q,DB)$

How to express this using the Boolean Semiring?

$CAns(Q, \mathbb{S}, \mathfrak{B}) = \bigwedge_{DB \in poss(\mathbb{S})} Ans(Q,DB,\mathfrak{B})$

where

$Ans(Q,DB_1,\mathfrak{B}) \wedge Ans(Q,DB_2,\mathfrak{B}) =$

$\{(a, b, x \wedge y) :$

$(a, b, x) \in Ans(Q,DB_1,\mathfrak{B})$ and

$(a, b, y) \in Ans(Q,DB_2,\mathfrak{B})\}$

Dual Operator and Certain Answer

- We aggregated the answers on possible DB's by using $\wedge$, which is the dual of $\vee$, which is the $\oplus$ of $\mathfrak{B}$ semiring.

- Generalizing, we define

$$x \odot y = \begin{cases} x & \text{if } x \oplus y = y \\ y & \text{if } x \oplus y = x \end{cases}$$

$$\text{CAns}(Q, \mathbb{S}, \mathfrak{R}) = \bigodot_{DB \in poss(\mathbb{S})} \text{Ans}(Q, DB, \mathfrak{R})$$

Differently said…

- A tuple $(a, b, x) \in \mathrm{CAns}(Q, \mathfrak{S}, \mathfrak{R})$, with $x \neq \mathbf{0}$,

    iff

  for each $DB \in \mathrm{poss}(\mathfrak{S})$ there exists $y \leq x$ s.t.
  $(a, b, y) \in \mathrm{Ans}(Q, DB, \mathfrak{R})$.

- Definition reflects:

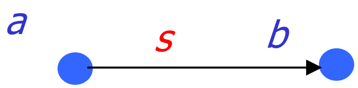
    certainty that objects *a* and *b* are always connected with paths, which are preferentially weighted not more than *x*.

Practically

Query: $Q = (\text{highway} : 0)^* \parallel (\text{road} : 1 + \varepsilon)^*$

Source Collection:

$\mathfrak{S}$



Source Definition: $S = \text{highway}^* \parallel (\text{road} + \varepsilon)^5$

Possible Databases:

All those, which have at least a path (between *a* and *b*) labeled by **highways** intervened by at most 5 **roads**.

Quantitative Semiring

- $\odot$ is *max*, and we have (*a*, *b*, 5) as a certain answer.

- Weight of 5 states our certainty that in any possible database, there is a path from *a* to *b*, whose preferential weight w.r.t. the given query is not more than 5.

- Also, there exists a possible database in which the best path between a and b is exactly 5.

Qualitative Semiring

- $\odot$ is again *max*, but we have (*a*, *b*, 1) as a certain answer.

- Weight of 1 states our certainty that in any possible database, there is a path from *a* to *b*, and the level of discomfort (w.r.t. the query) for traversing that path is not more than 1.

Hybrid Semiring

$$n^{(i)} \odot m^{(j)} = \begin{cases} m^{(j)} & \text{if } n < m \\ n^{(i)} & \text{if } n > m \\ n^{(max\{i,j\})} & \text{if } n = m \end{cases}$$

- We have (a, b, $1^{(5)}$) as a certain answer.
- Because although the level of discomfort of the best path connecting a with b in any possible database is 1, in the worst case (of such best paths), we need to endure up to 5 times such discomfort (w.r.t. the query).
- Of course $1^{(5)}$ is infinitely better than 2.

Certain Answers via Query Spheres

- Given Q, the y-sphere of Q is

  $Q^y = \{(w, x) \in \Delta^* \times R : (w, x) \in Q \text{ and } x \leq y\}$

- Call them "spheres" because: $Q^x \subseteq Q^y \subseteq Q$ for $x \leq y$
- Discrete Semirings:

  $\forall x \quad \exists$ "the next element" y

  i.e. $x<y$ and there isn't z, s.t $x<z<y$

- Theorem.

  $(a, b, y) \in CAns(Q, \mathfrak{S}, \mathfrak{R})$ iff

  $(a, b, T) \in CAns(Q^y, \mathfrak{S}, \mathfrak{B})$ and

  $(a, b, T) \notin CAns(Q^x, \mathfrak{S}, \mathfrak{B})$

Certain Answers via Query Spheres

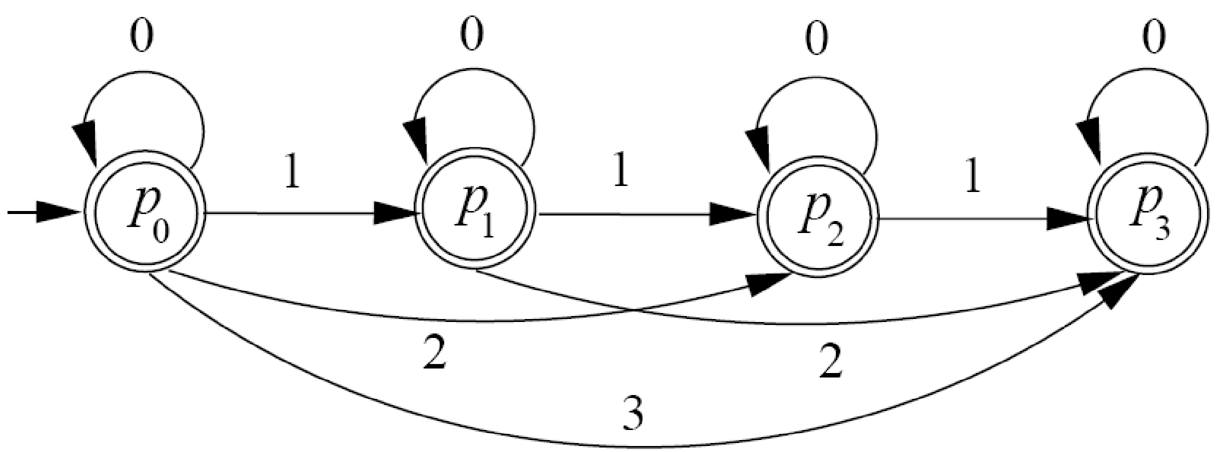
- We know how to compute the certain answer in the Boolean (classical) case: Calvanese, Di Giacomo, Lenzerini, Vardi, ICDE'00.
- And, we present next how to compute query spheres.
- But, is there an upper limit in the index of the spheres?
- Answer:
  - For the qualitative semiring there is always such bound.
  - For the quantitative and hybrid semirings, we reduce the problem to the Limitedness Problem in distance automata introduced and solved by Hashiguchi.
    - If there is such limit, then all the certain answers can be ranked.
    - Otherwise, the certain answers can be computed, but eventually ranked.
    - In practice, the user can provide a bound for the quality of certain answers he is interested in.

Computing Query Spheres

Computing Q^k

- Qualitative: Keep only transitions weighted $\leq k$.

- Quantitative: Intersect with mask automaton (e.g.):



$\mathcal{M}_3$

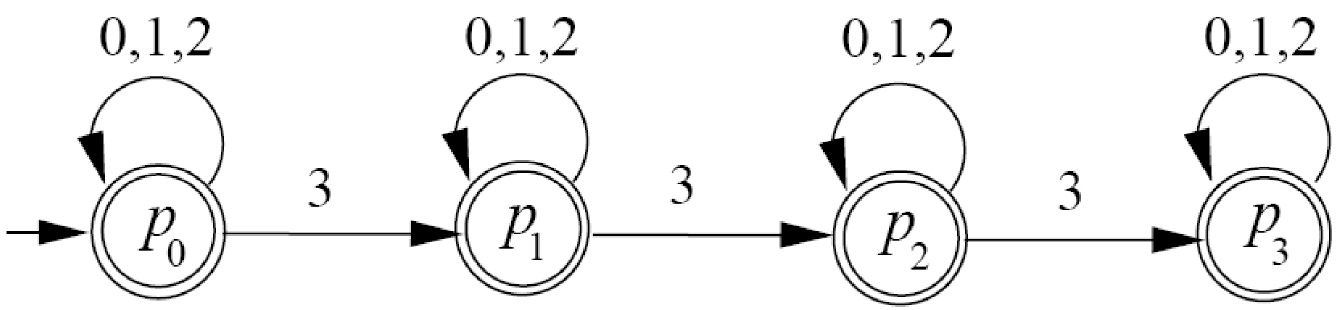
Computing Query Spheres: Hybrid

- Computing Q^y where $y = n^{(k)}$
- Intersect with a mask automaton, which extracts from the query automaton all the paths with

  (a) any number of transitions weighted strictly < n, and

  (b) not more than k transitions weighted exactly n.

E.g.



$\mathcal{M}_{3,2}$

Containment and Equivalence (Full Paper)

- We also study in detail the query containment for various semirings.

- We show that the containment is decidable for deterministic queries.

- Allauzen, Mohri, TCS 328, 2004.

  Show that large classes of weighted NFA's can successfully be determinized.

Conclusions

- Introduced preferential regular path queries
  - whose symbols are annotated with preference weights for “scaling” up or down the intrinsic importance of matching a symbol against a database edge label.
- Different specializations for the same syntactic annotations.
- Various semantics in a unifying semiring framework.
- Studied three important aspects:

  (1) query answering

  (2) (certain) query answering in LAV data-integration systems

  (3) query containment and equivalence.

- In all these, obtained important positive results, which encourage the use of our preference framework for enhanced querying of semistructured databases.

References

- Gösta Grahne, Alex Thomo, William W. Wadge: Preferentially Annotated Regular Path Queries. ICDT 2007: 314-328
- Gösta Grahne, Alex Thomo. Boundedness of Regular Path Queries in Data Integration Systems. IDEAS 2007: 85-92
- Gösta Grahne, Alex Thomo: Regular path queries under approximate semantics. Ann. Math. Artif. Intell. 46(1-2): 165-190 (2006)
- Dan C. Stefanescu, Alex Thomo. Enhanced Regular Path Queries on Semistructured Databases. EDBT Workshops 2006: 700-711
- Dan C. Stefanescu, Alex Thomo, Lida Thomo. Distributed evaluation of generalized path queries. SAC 2005: 610-616
- Gösta Grahne, Alex Thomo. Query Answering and Containment for Regular Path Queries under Distortions. FoIKS 2004: 98-115
- Gösta Grahne, Alex Thomo. Approximate Reasoning in Semistructured Data. KRDB 2001