Evaluating an Automata Approach to Query Containment

Michael Minock
KTH Royal Institute of Technology, Stockholm, Sweden
Umeå University, Umeå, Sweden.

September 6, 2017

Introduction

- The Query Containment Problem
- A Finite State Automata Approach
  - Database state encodings
  - Building the automaton M_Q
  - Quirks
- Software Demonstration
- Future Work

Query Containment

Q_{super} *contains* Q_{sub} if $Q_{sub}(D) \subseteq Q_{super}(D)$ for all databases D

- Increasingly expressive query classes solved
- Useful in query optimization, information integration, etc.
- Useful in natural language interfaces (e.g. (Shieber, 1993), (Bos and Oka, 2002), (Minock, 2017))

Example Database and Datalog Queries

R(A,	B,	C)
h	i	j
i	k	l
j	m	n

S(D,	E)
h	r
i	b
j	b
k	r

T(F G)

Datalog Queries:

```
Q1(X) :- R(X,Y,Z), S(Y,P)
Q2(X) :- R(X,Y,Z), S(Y,'b'), S(Z,'b')
Q3(X) :- R(X,Y,Z)
```

Q3 contains Q1

Q1 contains Q2

Classical Approaches

Consider classical approaches to show that

```
Q1(X) :- R(X,Y,Z), S(Y,P)
```

contains

```
Q2(X) :- R(X,Y,Z), S(Y,'b'), S(Z,'b')
```

- Canonical databases

  Freeze body of `Q2` into *canonical database*
  `{R(1,2,3),S(2,'b'),S(3,'b')}`
  Freeze head of `Q2` as `{(1)}`
  `{(1)}` is answer to `Q1` evaluated over *canonical database*

- Theorem proving

  The following sentence is not satisfiable:
  $(\exists x)(\neg(\exists y, z, p)(R(x, y, z) \wedge S(y, p)) \wedge$
  $(\exists y', z', p1, p2)(R(x, y', z') \wedge S(y', p1) \wedge S(y', p2) \wedge p1 =$
  $'\mathrm{b}' \wedge p2 = '\mathrm{b}'))$

- containment mappings

An Automata-based Approach

To decide if Q_{super} contains Q_{sub}:

1. Based on Q_{super} and Q_{sub} determine a 'minimal' database state encoding scheme.
2. Build $M_{Q_{super}}$ which recognizes database states that generate answers to Q_{super}.
3. Build $M_{Q_{sub}}$ which recognizes database states that generate answers to Q_{sub}.
4. Q_{super} contains Q_{sub} if $L(\overline{M}_{Q_{super}}) \cap L(M_{Q_{sub}})$ is empty

Database State Encoding

The database state on slide 4 is represented as:
`hij_ikl_jmn_#hr_ib_jb_kr_##` (_ ends tuples, # ends relations)

Construct an encoding scheme for each containment problem:

1. Only relations in either Q_{super} and Q_{sub}
2. Only selected, join or simple condition attributes
3. Variables and constants of Q_{sub} are frozen and added as constants
4. (Arbitrary) Ordering of relations and attributes determined.

Example DB encoding

```
Q1(X) :- R(X,Y,Z), S(Y,P)
Q2(X) :- R(X,Y,Z), S(Y,'b'), S(Z,'b')
```

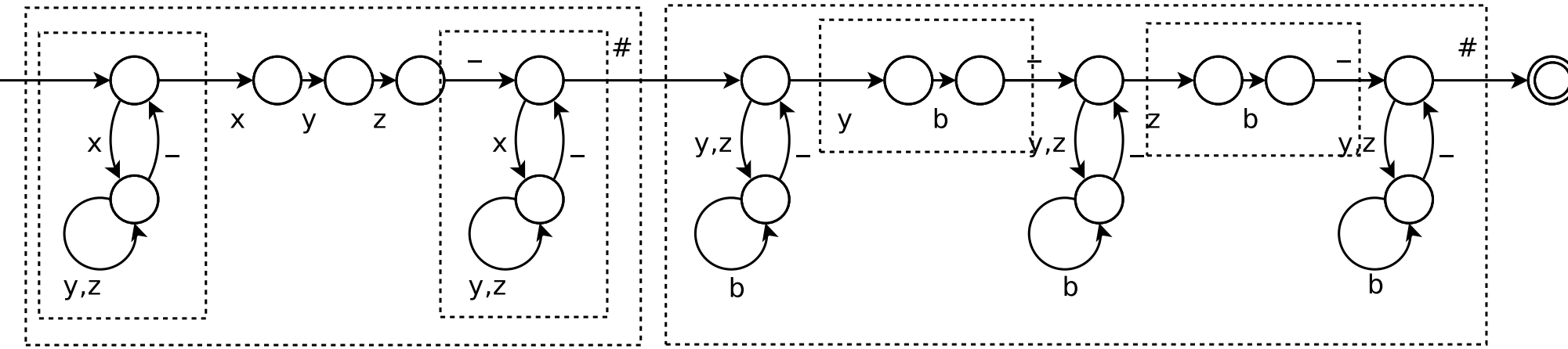
Encoding scheme for the problem if `Q1` contains `Q2`:

1. `R` and `S`.
2. All attributes of `R[A,B,C]` and `S[D,E]`
3. `'x'` under `A`, `'y'` under `B`, `'z'` under `C`, `'y'` and `'z'` under `D` and `'b'` under `E`.
4. Ordering `[R,S]` with orderings `[A,B,C]` and `[D,E]`

Construct an Automata

- String together *relation gobblers* based on encoding scheme.
- Each query predicate consists of *tuple gobblers*, followed by *witness gobblers* followed by tuple gobblers.
- Witness gobblers must be consistently sorted
- Building non-deterministic automata is easier

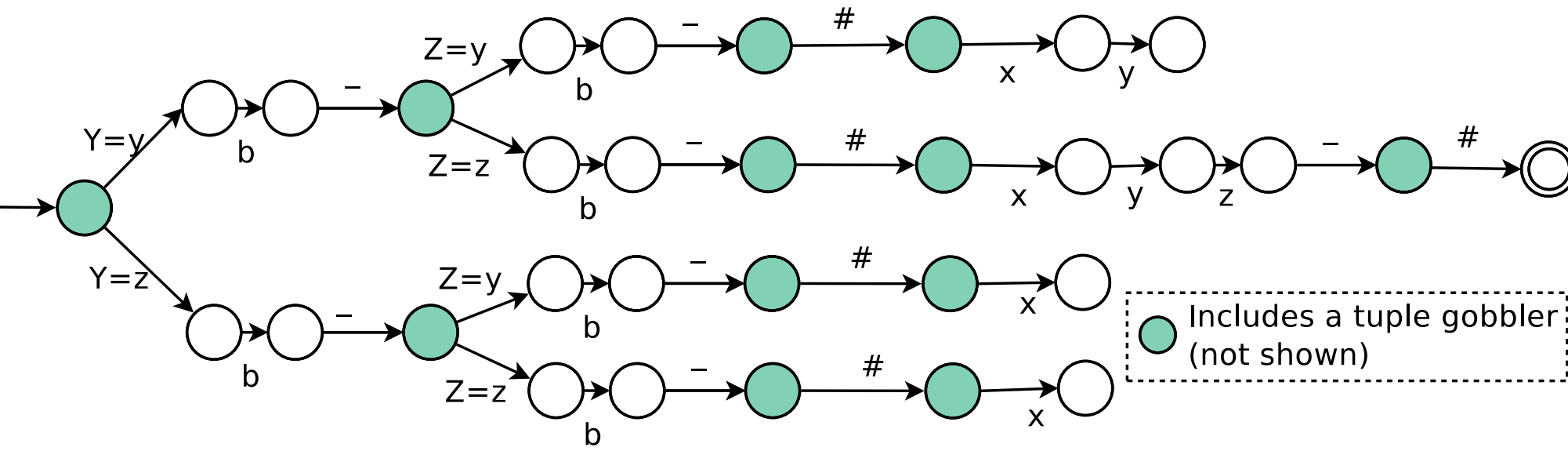
```
Q2(X) :- R(X,Y,Z), S(Y,'b'), S(Z,'b')
```



A Tree-shaped Case

Changing the encoding scheme from `[R,S]` to `[S,R]` causes the recursive construction function to build tree-shaped automaton.

```
Q2(X) :- S(Y,'b'), S(Z,'b'), R(X,Y,Z)
```



Quirks

- Consider building the automaton for `Q4(X) :- R(X,Y,Z),R(X,Y,Z)`. We must coalesce redundant witness gobblers.
- A post-processing compression routine removes such cases from automaton.
- This is absolutely required to get correct behavior (see video/photo corpus and scalability case 2).
- See paper for *ordered database state assumption* – just an optimization.
- Approach trivially extended to handle set conditions on non-join attributes.

Demonstration

- A photo/video corpus:
  27 queries over the `Picture` table over a photo/video domain – `sites.google.com/view/nli-corpora/`
- Case-1 (Merlin):
  Optimization Example 1 from [Chandra and Merlin, 1977]
- Case-2 (Colorability):
  graph colorability of graph rings of length 9 and 10 are 2-colorable (`False` for 9, `True` for 10).
- Case-3 (Big Set Conditions):
  single relation of 5 attributes, where queries have set conditions over these attributes with between 7 and 14 distinct values.

Results

Approach	Avg	Max	Min
CDB	0.3	1.5	0.1
TP	11.3	33.8	7.6
FSA	1.9	23.6	0.1

Table: Results over photo/video corpus (ms)

Approach	Case 1	Case 2	Case 3
CDB	1.0	0.6	1703
TP	62009	64.3	56235
FSA	0.9	294.9	222

Table: Results for special cases (ms)

Future Work (More expressive queries)

- point inequalities
  (e.g. *"photos of Alice not taken on June 1,2017"*)
- negated sub-goals
  (e.g. *"photos of Alice without Bob"*)
  (e.g. *"latest photo of Alice with Bob"*)
- database key constraints
  (e.g. *"no two distinct videos are stored in the same file"*)
- domain knowledge
  (e.g. *"Manhattan is in New York"*)

Future Work (Reasoning over sets)

(a) over 75% of objects are red

(b) over 75% of objects are square

(c) the majority of objects are red squares

(d) the majority of objects are red

(e) over 66% of objects are red

Example deductions:

$a \wedge b \models c$

$e \wedge b \not\models c$

$a \models d$

$c \models d$

$\alpha \vdash \beta$ when the blind k-counter machine (see Greibach,78) built to recognize $\alpha \wedge \neg\beta$ recognizes the empty language.

Conclusions

- Presented an FSA Approach for Query Containment
  - Straight forward with only a few quirks
  - Not proven correct, nor complete
  - Competitive with state of the art canonical database approach
- Software Demonstration
- Future Work