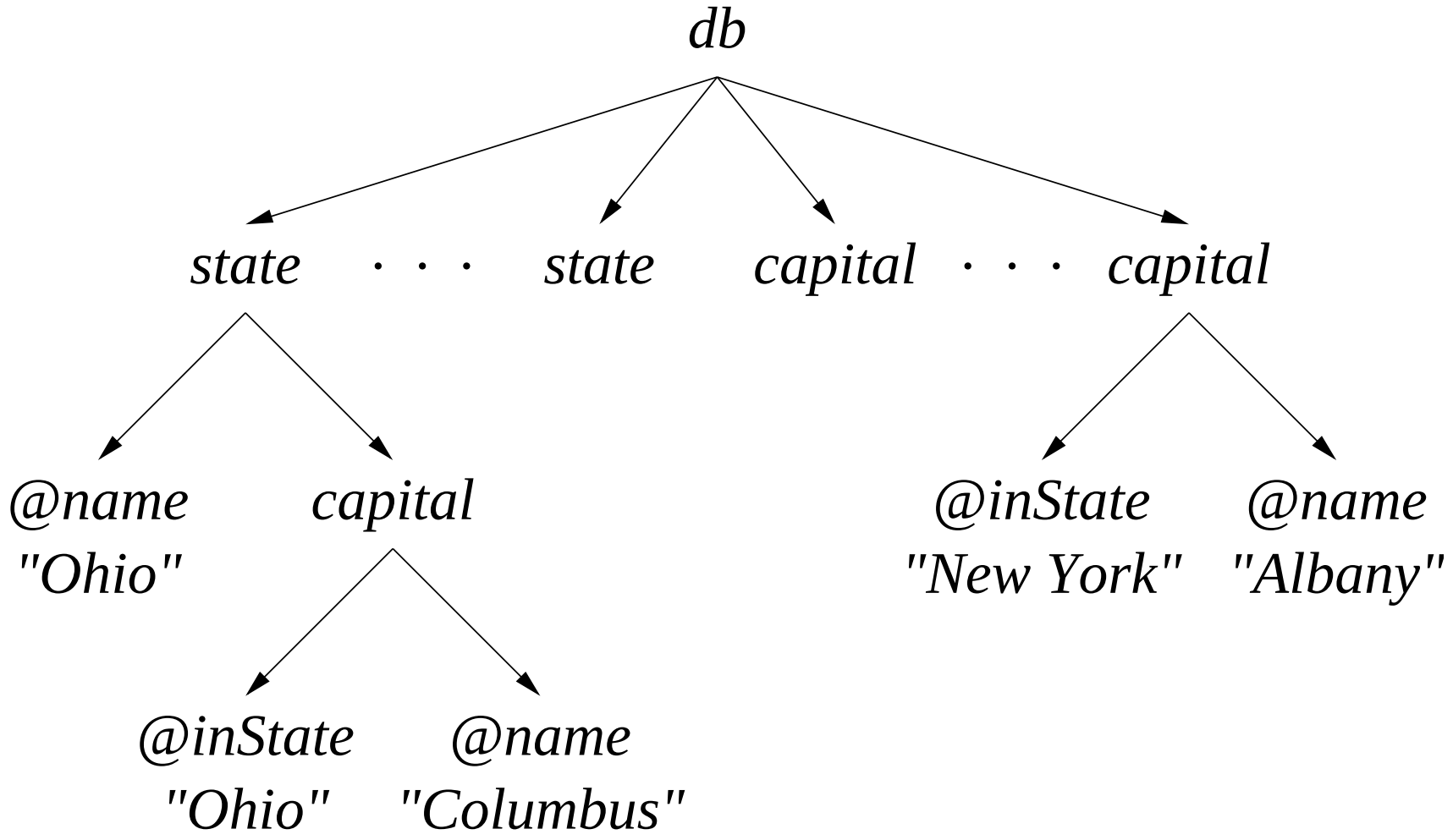
What's Hard about XML Schema Constraints?

Marcelo Arenas
U. of Toronto

Wenfei Fan
Bell Labs

Leonid Libkin
U. of Toronto

XML Documents



XML Schema

An XML Schema specification defines:

1. Structure of the documents: typing part of DTDs

```
<!ELEMENT db (state+, capital+)>
<!ELEMENT state (capital)>
<!ATTLIST state
          @name CDATA #REQUIRED>
<!ELEMENT capital EMPTY>
<!ATTLIST capital
          @inState CDATA #REQUIRED
          @name    CDATA #REQUIRED>
```

   DTD types are subsumed by XML Schema types. DTD types alone suffice to show that XML Schema constraints are hard.

XML Schema (cont'd)

2. Types of element and attribute values.

   The values of attributes *@name* and *@inState* must be strings.

3. Constraints on the values of elements and attributes: Keys and Foreign Keys

   - Every state must be uniquely identified by its name:

     $$(db/state, \{@name\})$$

   - Every state can have at most one capital:

     $$(db//capital, \{@inState\})$$

   - Every capital must be a city in some state:

     $$(db//capital, \{@inState\}) \subseteq_{FK} (db/state, \{@name\})$$

XML Consistency

- We are interested on the interaction between structural constraints, keys and foreign keys.

- Relational databases: given any schema and keys, foreign keys, one can always find a nonempty instance of the schema satisfying the constraints.

- An XML Schema specification - DTD and constraints - can be inconsistent.

An Inconsistent XML Schema Specification

No XML document conforms to the DTD and satisfies the set of constraints of the geographical database:

- The number of *capital* elements is greater than the number of *state* elements:

```
<!ELEMENT db (state+, capital+)>
<!ELEMENT state (capital)>
```

- The number of *capital* elements is at most the number of *state* elements:

$$(db//capital, \{@inState\})$$
$$(db/state, \{@name\})$$
$$(db//capital, \{@inState\}) \subseteq_{FK} (db/state, \{@name\})$$

The XML Schema Consistency Problem

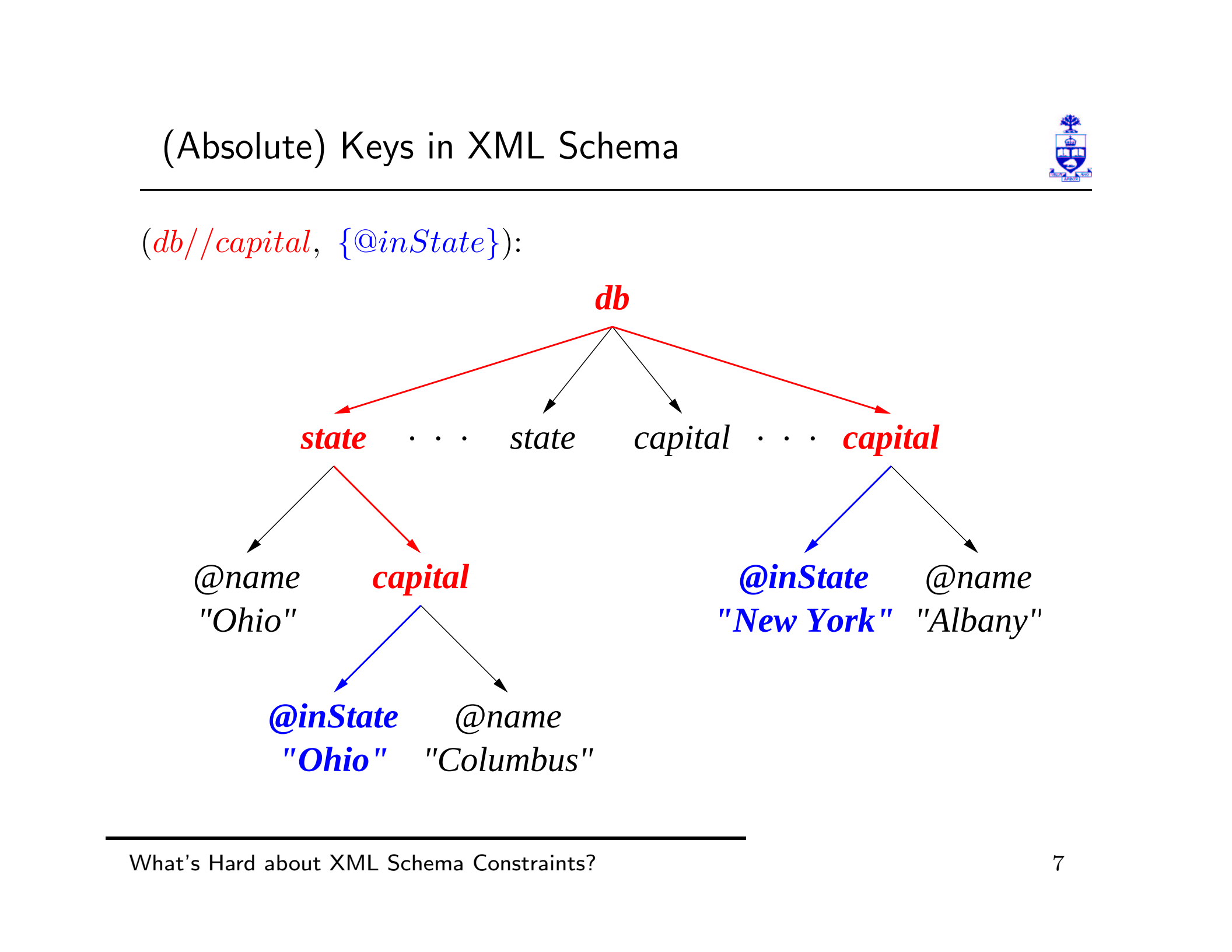
INPUT: A DTD D and a set of constraints Σ.

QUESTION: Is there an XML document T that both conforms to D and satisfies Σ?

One wants to know whether an XML Schema specification makes sense!

Why do we call this problem "XML Schema Consistency" instead of "DTD Consistency"? We consider constraints with the semantics proposed by XML Schema.

(Absolute) Keys in XML Schema



$(db//capital,\ \{@inState\})$:

Keys in XML Schema (cont'd)

XML Schema keys are slightly different from those studied in the integrity constraint literature.

Key: $(P, \{Q_1, \ldots, Q_n\})$

- P is called the selector of the key. It is a regular expression conforming to the BNF grammar:

$$
\begin{array}{lcl}
\mathit{selector} & ::= & \mathit{path} \mid \mathit{path} \cup \mathit{selector} \\
\mathit{path} & ::= & r//\mathit{sequence} \mid \mathit{sequence} \\
\mathit{sequence} & ::= & \tau \mid \_ \mid \mathit{sequence}/\mathit{sequence}
\end{array}
$$

Keys in XML Schema (cont'd)

- Expressions $Q_1, \ldots, Q_n$ are called the fields of the key. They are regular expressions conforming to the BNF grammar:

$$
\begin{array}{lcl}
\mathit{field} & ::= & \mathit{path} \mid \mathit{path} \cup \mathit{field} \\
\mathit{path} & ::= & \mathit{//sequence/last} \mid \mathit{/sequence/last} \\
\mathit{sequence} & ::= & \epsilon \mid \tau \mid \_ \mid \mathit{sequence/sequence} \\
\mathit{last} & ::= & \mathtt{S} \mid @l \mid @\_
\end{array}
$$

This grammar differs from the "selectors grammar" in restricting the final step to match a text node or an attribute.

Keys in XML Schema (cont'd)

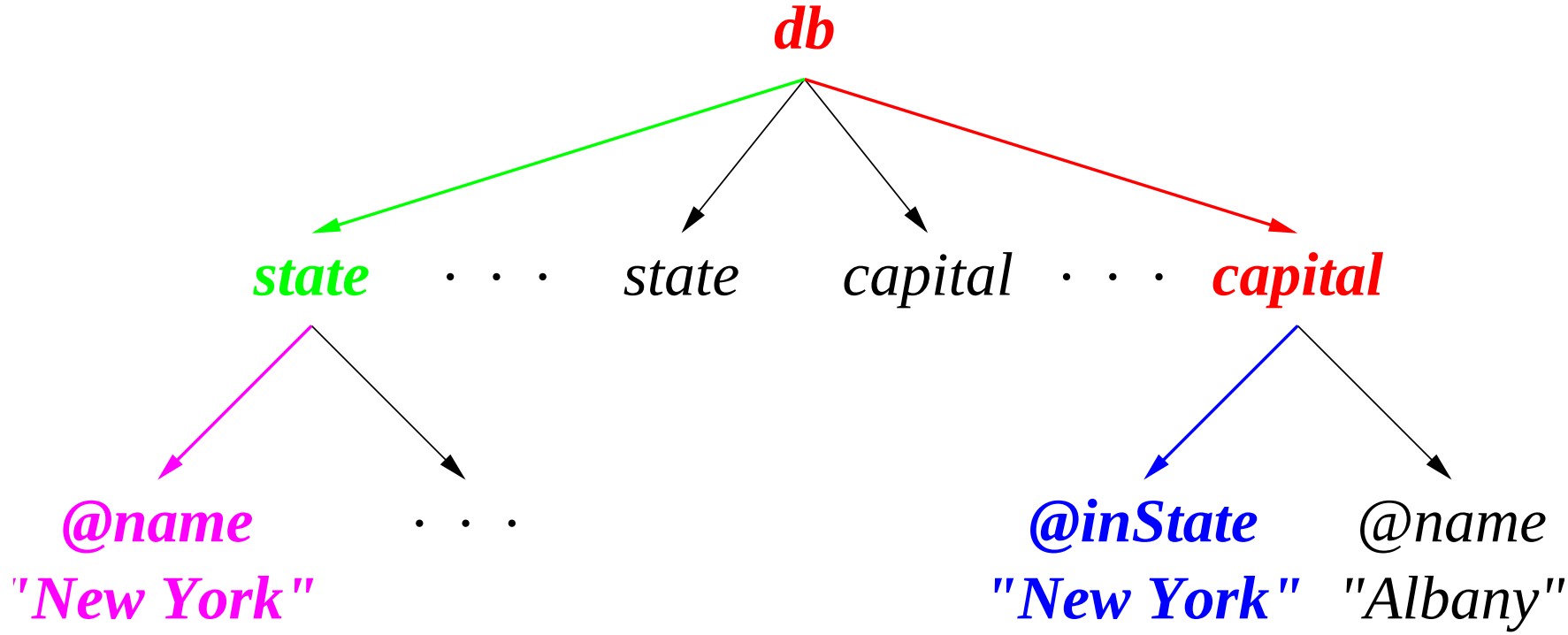
$(P, \{Q_1, \ldots, Q_m\})$ is satisfied by a document if for every node x reachable from the root by path P,

> Reachability: For each Q_i, there is exactly one node reachable from x by path Q_i, and
>
> Uniqueness: The values of Q_is uniquely determine x.

Usually, in the integrity constraint literature only uniqueness is considered.

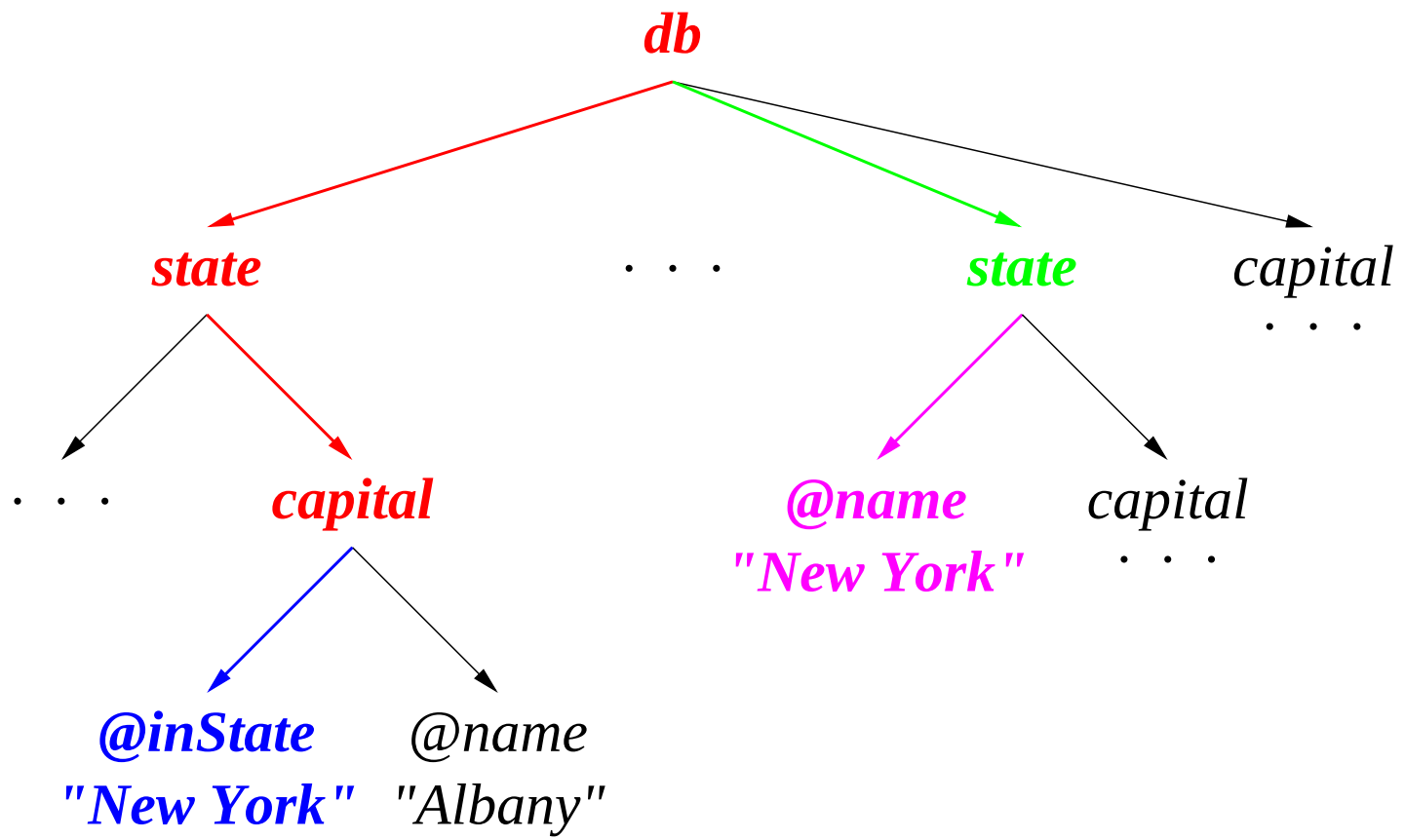
(Absolute) Foreign Keys in XML Schema

$(db//capital, \{@inState\}) \subseteq_{FK} (db/state, \{@name\})$:



Foreign Keys in XML Schema (cont'd)

$(db//capital, \{@inState\}) \subseteq_{FK} (db/state, \{@name\})$:



Foreign Keys in XML Schema (cont'd)

Foreign Key: $(P, \{Q_1, \ldots, Q_n\}) \subseteq_{FK} (U, \{S_1, \ldots, S_n\})$

P and U are selectors, Q_1, ..., Q_n, S_1, ..., S_n are fields.

$(P, \{Q_1, \ldots, Q_n\}) \subseteq_{FK} (U, \{S_1, \ldots, S_n\})$ is satisfied if

1. $(U, \{S_1, \ldots, S_n\})$ is satisfied.
2. For every node x reachable from the root by path P, there is a node x' reachable from the root by path U such that the Q_1, ..., Q_n-values of x are equal to the S_1, ..., S_n-values of x'.

What is Known about XML Consistency?

Run-time check: attempts to validate documents with (D, Σ). Are repeated failures due to a bad specification or problems with the documents? Static analysis is a better approach!

Only uniqueness condition was considered.

Fan & Libkin, PODS'01:

- The consistency problem for DTDs, keys and foreign keys of the form:

$$(r//\tau, \{@l_1, \ldots, @l_n\})$$
$$(r//\tau, \{@l_1, \ldots, @l_n\}) \subseteq_{FK} (r//\tau', \{@l'_1, \ldots, @l'_n\})$$

  is undecidable.

What is Known about XML Consistency? (cont'd)

- The consistency problem for DTDs and keys of the form:

$$(r//\tau, \{@l_1, \ldots, @l_n\})$$

  is solvable in linear time.

- When restricted to unary constraints:

$$(r//\tau, \{@l\})$$
$$(r//\tau, \{@l\}) \subseteq_{FK} (r//\tau', \{@l'\})$$

  the consistency problem for DTDs, keys and foreign keys is NP-complete.

What is Known about XML Consistency? (cont'd)

Regular expressions were also considered.

Arenas & Fan & Libkin, PODS'02:

- The consistency problem for DTDs, keys and foreign keys of the form:

$$(P, \{@l\})$$
$$(P, \{@l\}) \subseteq_{FK} (P', \{@l'\})$$

  where P, P' are regular expressions, is PSPACE-hard and is in NEXPTIME.

- The consistency problem for DTDs and keys of the form:

$$(P, \{@l_1, \ldots, @l_n\})$$

  where P is a regular expressions, is solvable in linear time.

What do we do here?

- All the previous results are applicable to the XML Schema consistency problem.
- We obtain lower bounds as corollaries of these results.
- We cannot obtain upper bounds as corollaries because of the reachability condition.

New Results

We saw before that without foreign keys, consistency is solvable in linear time.

Reachability condition makes the problem hard.

Theorem XML Schema consistency problem is NP-hard, even if:

- No foreign keys are considered.
- DTDs do not include recursion and Kleene star.
- Keys are unary.

New Results (cont'd)

The language for expressing selectors and fields makes the problem hard.

Theorem XML Schema consistency problem is NP-hard, even if:

- Keys and foreign keys are of the form:

$$(P, \{@l\}),$$
$$(P, \{@l\}) \subseteq_{FK} (P', \{@l'\}).$$

  P, P' are selector.

  $@l$ ($@l'$) is an attribute of all the element types that are the last symbol of some string in P (P').

- The number of element types and attributes is fixed (greater than 10).

Conclusion

	DTD	XML Schema
Keys and foreign keys	undecidable	undecidable
Unary keys and foreign keys	NP-complete	PSPACE-hard
Keys only	linear time	NP-hard

Future work: Exact complexity of many problems remains unknown.