Inferring XML Schema Definitions from XML Data

Geert Jan Bex, Frank Neven and
Stijn Vansummeren

Hasselt University and transnational
University of Limburg, Belgium

Overview

- Introduction
- Complete algorithm *i*LOCAL
- Heuristic *i*XSD
- Experiments
- Conclusions

Motivation for schemas

- Why schemas?
  - automation & optimization of search
  - integration of XML data sources
  - translation & processing of XML data
  - used by software tools, e.g., JAXB, Castor
  - schema matching & model management
- Why infer schemas?
  - 50 % of XML document on the web have none [Barbosa et al., 2005]
  - 33 % of schemas are not valid [Bex et al., 2004, 2005]

real world XML & XSDs

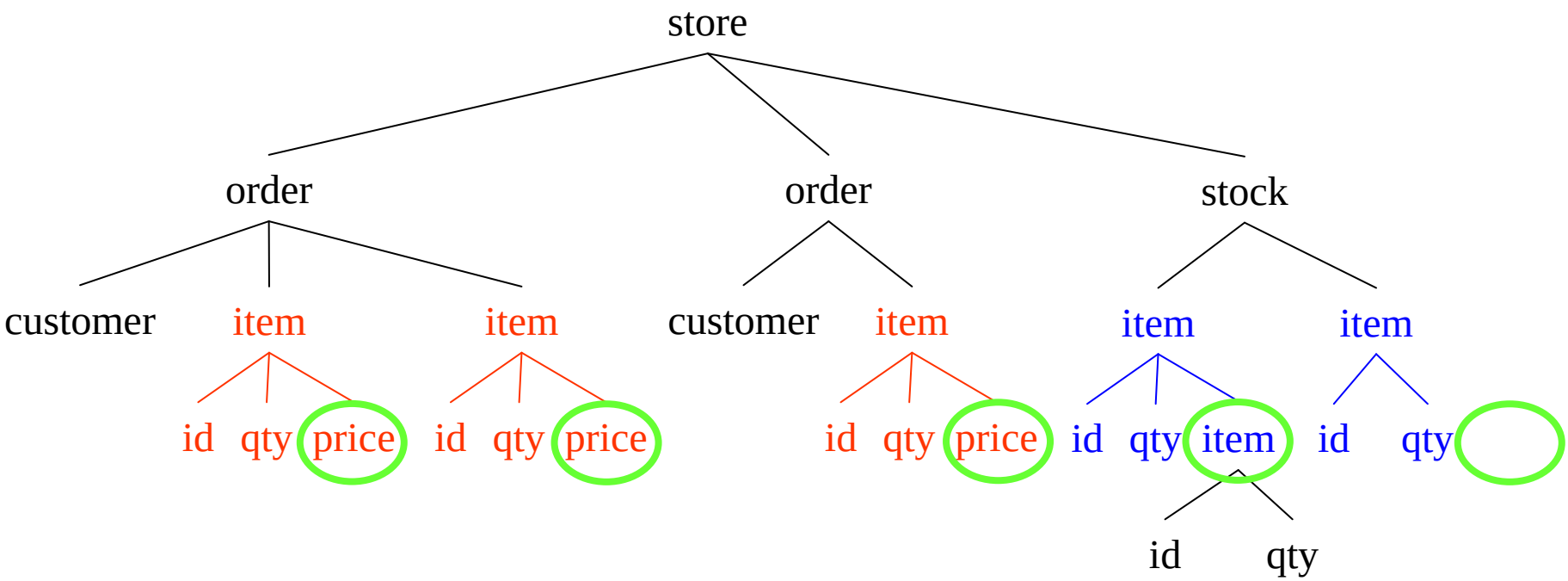
Motivation for XSD inference

- DTD inference
  - XTract [Garofalakis et al., 2003]
  - trang [Clark]
  - *i*DTD [Bex et al., 2006]
- XSD inference
  - trang
  - XStruct
  - JAXB, .Net

} output XSD syntax, but equivalent to DTD

expressive power limited to that of DTDs!

How do DTDs and XSDs differ?



in DTDs, either:

item → id, qty, (price + item*)

or

order_item → id, qty, price

stock_item → id, qty, stock_item*

can be done in XSDs

XSD: abstract syntax

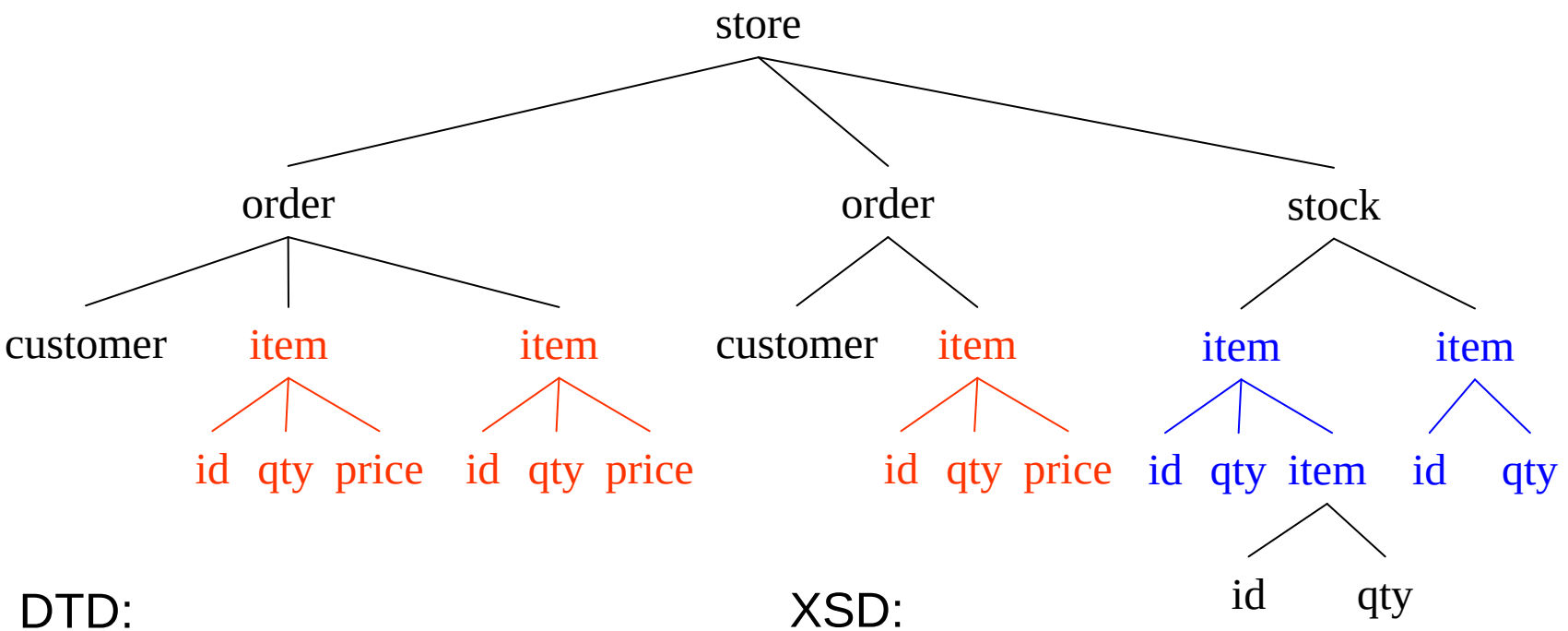
```
<xsd:element name="store" type="store"/>

<xsd:complexType name="store">
  <xsd:sequence>
    <xsd:element name="order" type="order" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="stock" type="stock"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="order">
  <xsd:sequence>
    <xsd:element name="customer" type="customer"/>
    <xsd:element name="item" type="item1" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

root → **store**[*store*]
store → **order**[*order*]*, **stock**[*stock*]
order → **customer**[*customer*], **item**[$item_1$]$^+$

Motivating example for XSD



DTD:

root → **store**
store → **order***, **stock**
order → **customer**, **item**$^{+}$
stock → **item**$^{+}$
item → **id**, **qty**, (**price** + **item***)

XSD:

root → **store**[*store*]
store → **order**[*order*]*, **stock**[*stock*]
order → **customer**[*customer*], **item**[$item_1$]$^{+}$
stock → **item**[$item_2$]$^{+}$
$item_1$ → **id**[*id*], **qty**[*qty*], **price**[*price*]
$item_2$ → **id**[*id*], **qty**[*qty*], **item**[$item_2$]*

Inference of XSDs

- Problem: infer XSD from XML corpus
- Requirement: concise, i.e., humans can interpret/validate
- But… theorem [Gold, 1967]:

impossible to learn from positive data only

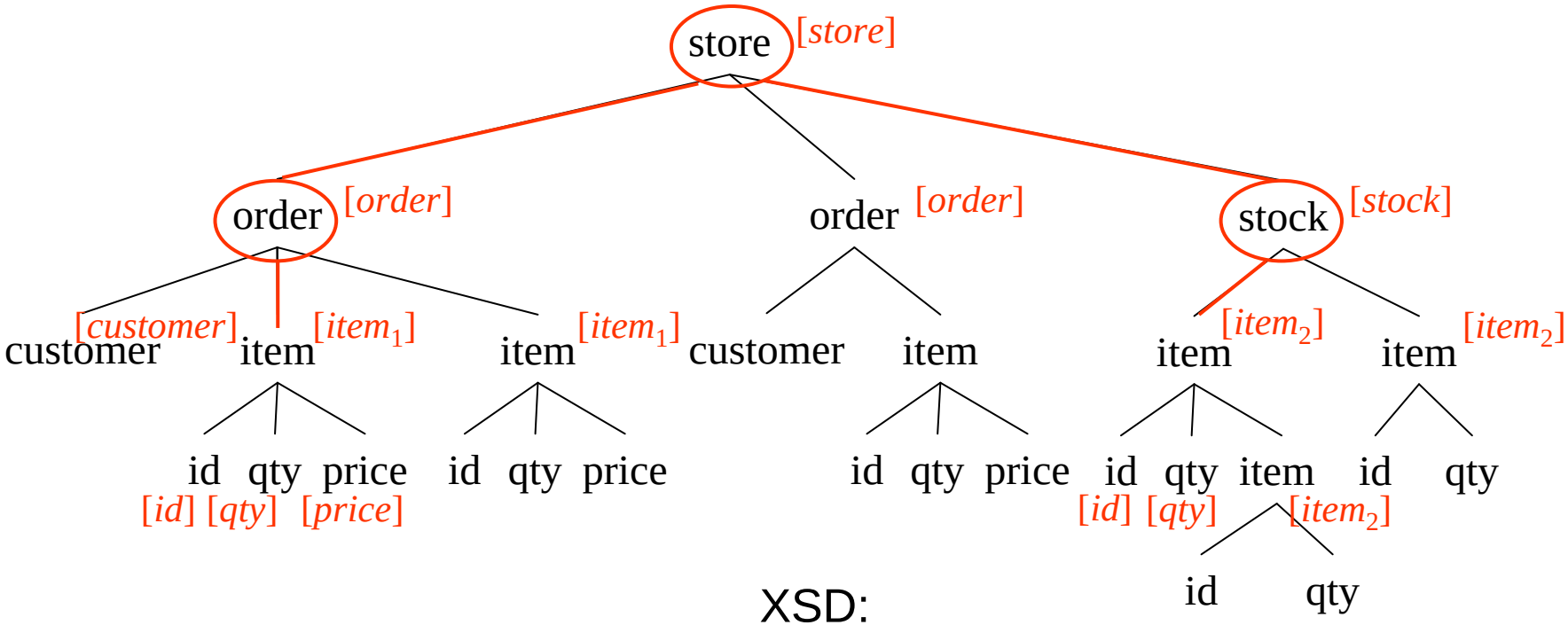
XSD property

W3C specs: Element Declarations Consistent (EDC):

no elements with distinct type in same content model

content model of an element depends on its context

XML validation for XSD



XSD:

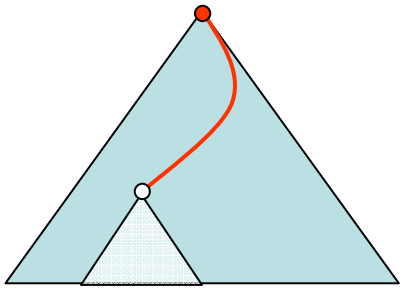
root → **store**[*store*]
store → **order**[*order*]*, **stock**[*stock*]
order → **customer**[*customer*], **item**[$item_1$]$^+$
stock → **item**[$item_2$]$^+$
$item_1$ → **id**[*id*], **qty**[*qty*], **price**[*price*]
$item_2$ → **id**[*id*], **qty**[*qty*], **item**[$item_2$]*

if XML is valid:
type assignment is determined by path from element to root

XML validation for XSD

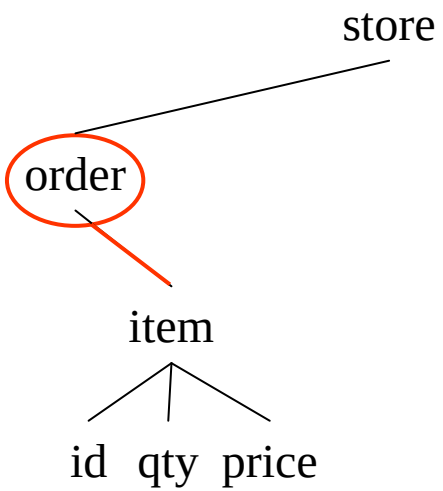
Theorem [Martens et al., 2006]

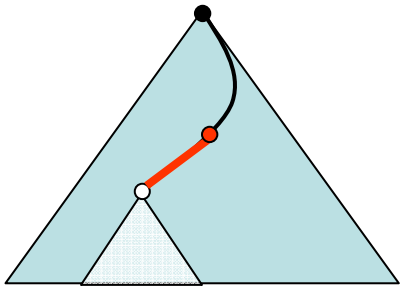
> Content model of an element is uniquely determined by the path from the root to that element



XSD observations: local context

- Large, diverse corpus of real world XSDs [Bex et al., 2004, Martens et al., 2006]
  - 98 % of XSDs only local context: relevant ancestor path has length of at most 3, i.e., "greatgrandfather"





XSD observations: SOREs

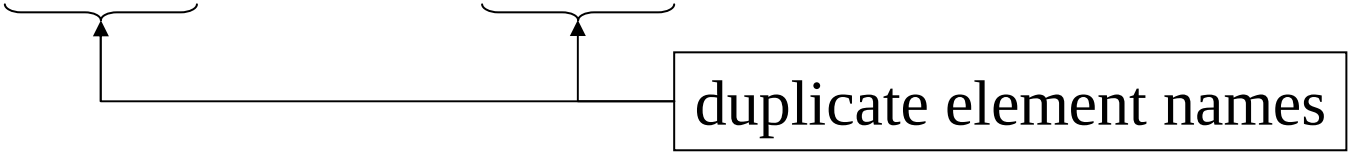
- Large, diverse corpus of real world XSDs [Bex et al., 2004, Martens et al., 2006]
  - 99 % of regular expressions is single occurrence
- What's a Single Occurrence RegExp

  header, protein, organism, reference*, comment*, genetics*, complex*, function*, classification?, keywords?, feature*, summary, sequence

  authors, citation, volume?, month?, year, pages?, (title + descr)?, xrefs?

  title, (author, affiliation?)$^{+}$, abstract
- … and what's not

  title, ((author, affiliation)$^{+}$ + (editor, affiliation)$^{+}$), abstract



Overview

- Introduction
- Complete algorithm *i*LOCAL
- Heuristic *i*XSD
- Experiments
- Conclusions

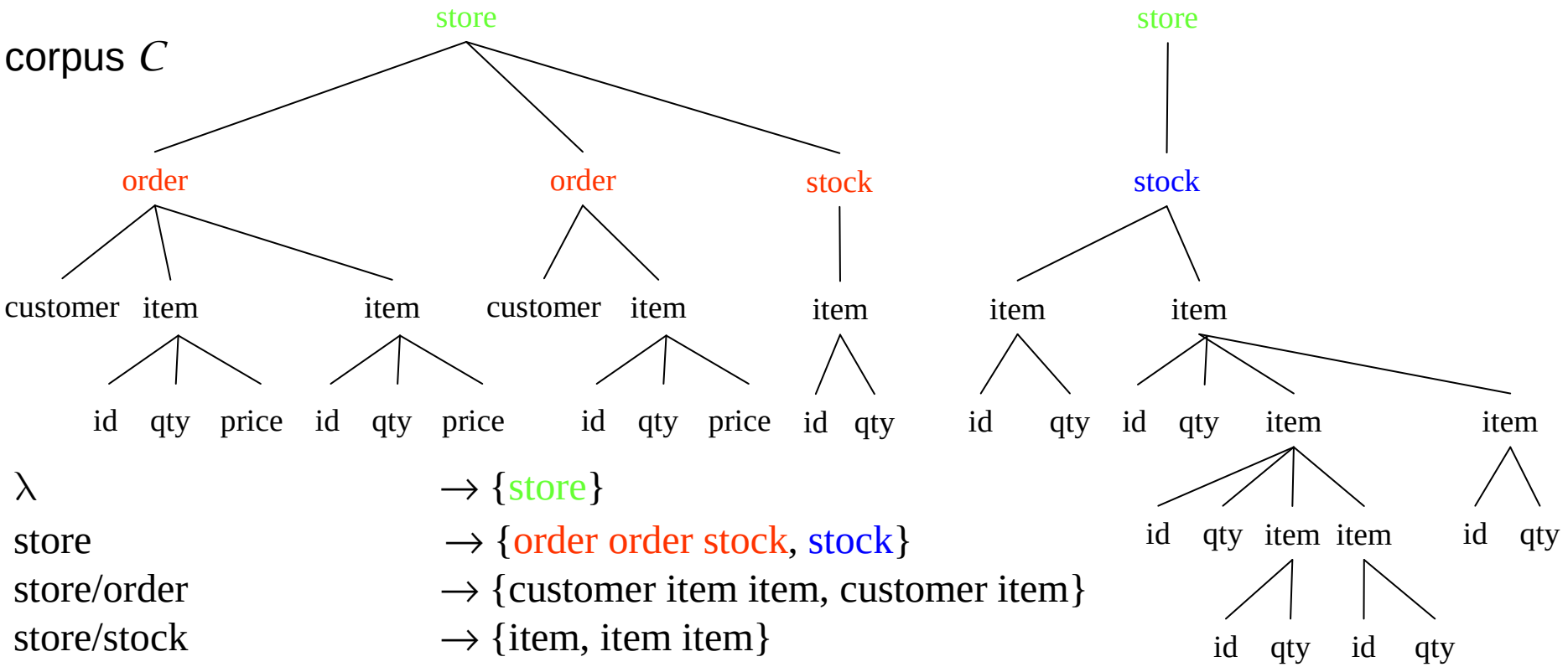
Main result

Theorem:

> XSDs with local context and SORE content models are learnable from positive examples only

Algorithm *i*LOCAL

corpus C



λ	$\rightarrow$ {store}
store	$\rightarrow$ {order order stock, stock}
store/order	$\rightarrow$ {customer item item, customer item}
store/stock	$\rightarrow$ {item, item item}
store/order/item	$\rightarrow$ {id qty price}
store/stock/item	$\rightarrow$ {id qty, id qty item item}
store/stock/item/item	$\rightarrow$ {id qty item item, id qty}
store/stock/item/item/item	$\rightarrow$ {id qty}

paths are types [Martens et al., 2006]

Algorithm *i*LOCAL

λ	→ {store}
store	→ {order order stock, stock}
store/order	→ {customer item item, customer item}
store/stock	→ {item, item item}
store/order/item	→ {id qty price}
store/stock/item	→ {id qty, id qty item item}
store/stock/item/item	→ {id qty item item, id qty}
store/stock/item/item/item	→ {id qty}

locality: k = 2 ↓

λ	→ {store}
store	→ {order order stock, stock}
store/order	→ {customer item item, customer item}
store/stock	→ {item, item item}
order/item	→ {id qty price}
stock/item	→ {id qty, id qty item item}
item/item	→ {id qty item item, id qty}

Algorithm *i*Local

λ	$\rightarrow$ {store}
store	$\rightarrow$ {order order stock, stock}
store/order	$\rightarrow$ {customer item item, customer item}
store/stock	$\rightarrow$ {item, item item}
order/item	$\rightarrow$ {id qty price}
stock/item	$\rightarrow$ {id qty, id qty item item}
item/item	$\rightarrow$ {id qty item item, id qty}

*i*SOA, ToSORE [Bex et al., 2006]

XSD

root	$\rightarrow$ **store**[*store*]
store	$\rightarrow$ **order**[*store/order*]*, **stock**[*store/stock*]
store/order	$\rightarrow$ **customer**[*order/customer*], **item**[*order/item*]$^+$
store/stock	$\rightarrow$ **item**[*stock/item*]$^+$
order/item	$\rightarrow$ **id**[*item/id*], **qty**[*item/qty*], **price**[*item/price*]
stock/item	$\rightarrow$ **id**[*item/id*], **qty**[*item/qty*], **item**[*item/item*]*
item/item	$\rightarrow$ **id**[*item/id*], **qty**[*item/qty*], **item**[*item/item*]*

Algorithm *i*LOCAL

- Theorem: *i*LOCAL is sound

> corpus C is valid with respect to inferred XSD

- Theorem: *i*LOCAL is *k*-complete

> if corpus C is "sufficiently large" then
> target XSD is equivalent with inferred XSD

Algorithm MINIMIZE

$root$	→ **store**[*store*]
store	→ **order**[*store/order*]*, **stock**[*store/stock*]
store/order	→ **customer**[*order/customer*], **item**[*order/item*]$^+$
store/stock	→ **item**[*stock/item*]$^+$
order/item	→ **id**[*item/id*], **qty**[*item/qty*], **price**[*item/price*]
stock/item	→ **id**[*item/id*], **qty**[*item/qty*], **item**[*item/item*]*
item/item	→ **id**[*item/id*], **qty**[*item/qty*], **item**[*item/item*]*

} duplicate types

MINIMIZE ↓

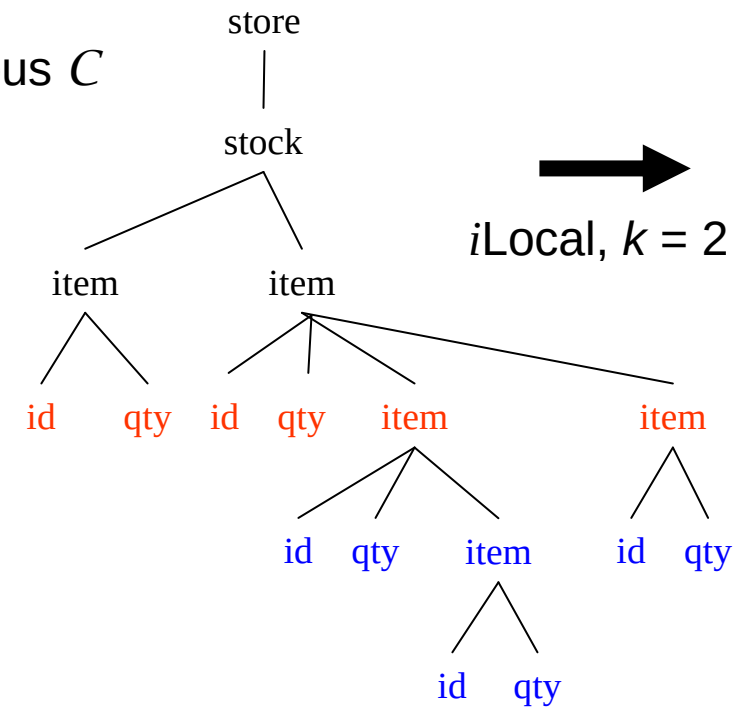
$root$	→ **store**[*store*]
store	→ **order**[*store/order*]*, **stock**[*store/stock*]
store/order	→ **customer**[*order/customer*], **item**[*order/item*]$^+$
store/stock	→ **item**[$item_2$]$^+$
order/item	→ **id**[*item/id*], **qty**[*item/qty*], **price**[*item/price*]
$item_2$	→ **id**[*item/id*], **qty**[*item/qty*], **item**[$item_2$]*

Overview

- Introduction
- Complete algorithm *i*LOCAL
- Heuristic *i*XSD
- Experiments
- Conclusions

In practice: incomplete data

corpus C



*i*Local, $k = 2$

stock/item → {id qty, id qty item item}
item/item → {id qty item, id qty}

*i*SOA, ToSORE

stock/item → **id**[*item/id*], **qty**[*item/qty*], **item**[*item/item*]*
item/item → **id**[*item/id*], **qty**[*item/qty*], **item**[*item/item*]?

MINIMIZE can't minimize!

incomplete data ⇒ *i*Local derives too many types!

Practical heuristics

- Define "distance" between types
  - details: see paper
- For types s, t: if $\mathrm{distance}(s,t) < \varepsilon$, unify s and t

  = REDUCE
- Our practical algorithm *i*XSD:

$$\boxed{i\mathrm{XSD}(k,\ C) = \mathrm{REDUCE}(i\mathrm{LOCAL}(k,\ C))}$$

Overview

- Introduction
- Complete algorithm *i*LOCAL
- Heuristic *i*XSD
- Experiments
- Conclusions

Experiments

- Corpora:
  - 697 real world XSD documents: C_{XSD}
    - XSD schema is local with $k = 2$
    - **attributeGroup**, **group**, **extension**: 2 contexts
    - **restriction**: 3 contexts
  - 8 corpora for synthetic XSDs, 200 XML documents each: $C_1,\ldots,C_8$
    - XSD schemas define documents of unbounded depth, width
    - local with $k = 2, 3$
    - 12 to 23 types
    - one schema associates multiple types with six element names
    - XML generated with ToXgene

real world corpora are hard to find

Precision

types of *i*XSD imprecisions:

1. content model for target and inferred type can differ
2. type in target XSD can corresponds to multiple types in inferred XSD: false positives
3. type in inferred XSD can corresponds to multiple types in target XSD: false negatives
4. type in target XSD is not derived

incomplete corpus, can't be avoided

1. Content models

- adapt content model of target XSD to information present in corpus = baseline
- compare derived content model with baseline
- C_{XSD}, $k = 2$:
  - 38/47 as good
  - 9/47 better than baseline

ToSORE generalization + REDUCE smoothing

2/3. False positives/negatives

- C_{XSD}, $k = 2$
  - iXSD: no false positives/negatives
  - iLOCAL, no REDUCE: 29 false positives

$$i\text{XSD}(k,\ C) = \text{REDUCE}(i\text{LOCAL}(k,\ C))$$

illustrates need for and power of REDUCE

- C_1,…, C_8: no false positives/negatives

Sensitivity to parameters k *and* ε

- context size $k \nearrow \Rightarrow$ false positives $\nearrow$
  $\Rightarrow$ false negatives $\searrow$

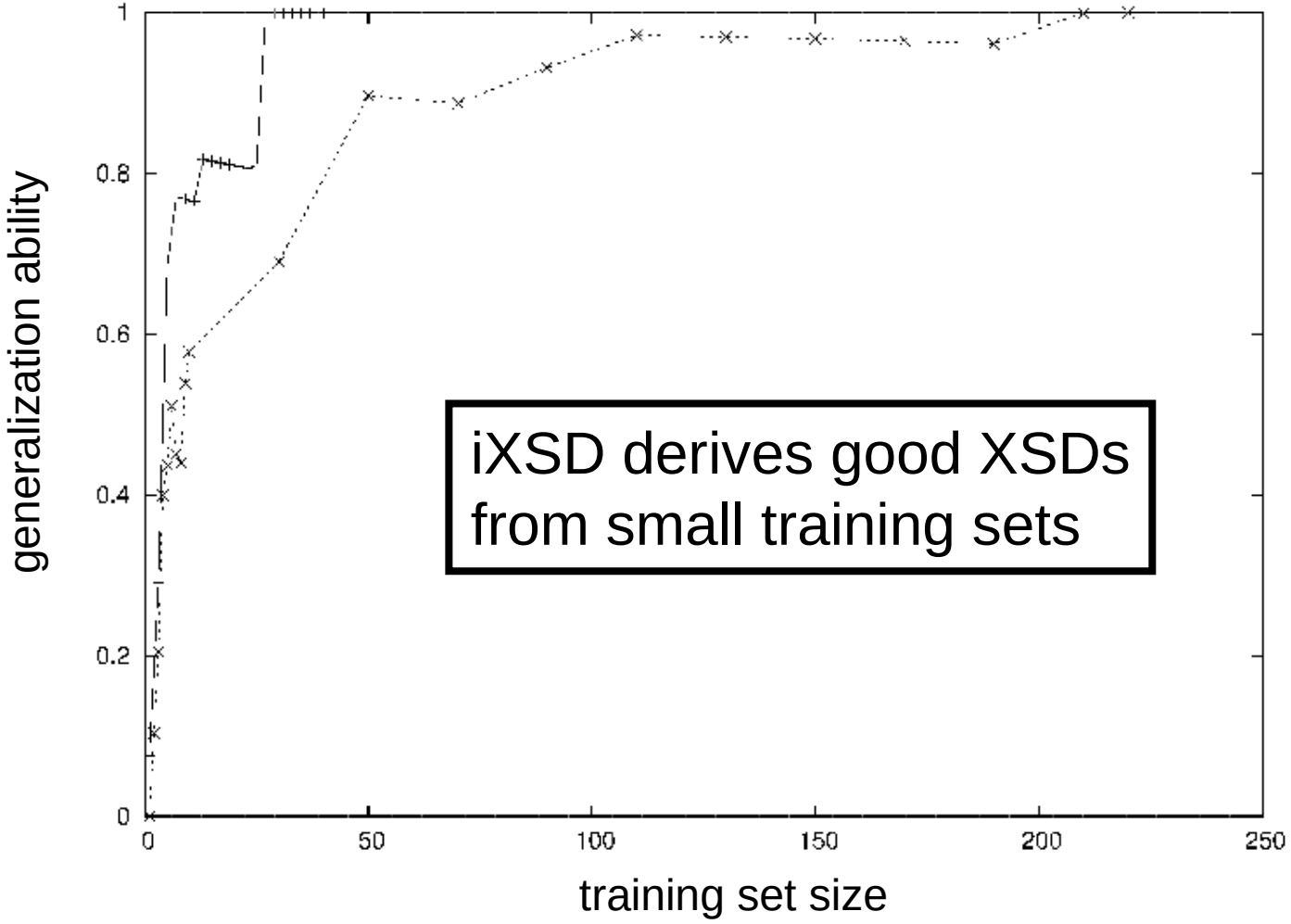
rule of thumb:
increase k until types are derived with too few examples

- $\varepsilon \nearrow \Rightarrow$ false positives $\searrow$
  $\Rightarrow$ false negatives $\nearrow$

safe range: $0.05 \lesssim \varepsilon \lesssim 0.15$

Generalization

*i*XSD on training set
generalization ability = fraction of valid XML docs in test set



Overview

- Introduction
- Complete algorithm *i*LOCAL
- Heuristic *i*XSD
- Experiments
- Conclusions

Conclusions

- Two algorithms
  - *i*LOCAL: sound & k-complete
  - *i*XSD: extends *i*LOCAL to deal with poor data
    - good performance on real world & synthetic data
    - good runtime performance
    - rule of thumb to determine context size k
- Future work

  principled approach to determine best locality k

Thank you