XML Schemas Admitting 1-Pass Preorder Typing

Wim Martens Frank Neven Thomas Schwentick

Outline

- XML Schema Languages
- Single-Type SDTDs
- 1-Pass Preorder Typing
- Restrained Competition SDTDs
- Unique Particle Attribution vs 1PPT
- Conclusion

XML Schema Languages: DTDs

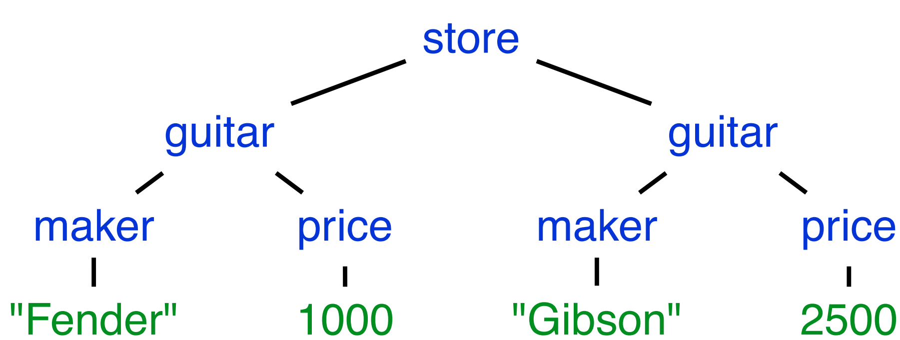
- DTDs (Document Type Definitions):

$$\begin{array}{lcl} \text{store} & \rightarrow & \text{guitar guitar}^* \\ \text{guitar} & \rightarrow & \text{maker price} \end{array}$$

XML Schema Languages: DTDs

- DTDs (Document Type Definitions):

$$
\begin{array}{lcl}
\text{store} & \rightarrow & \text{guitar guitar}^* \\
\text{guitar} & \rightarrow & \text{maker price}
\end{array}
$$



XML Schema Languages: DTDs

Regular expressions should be deterministic

- Backward compatibility with SGML:
  “It is an error if an element in the document can match more than one occurrence of an element type in the content model [without looking ahead].”
- Example:
  $bc + bd$. Where do we match b in the string bd?
  $b(c + d)$ is deterministic.

Purpose: facilitate validation!

XML Schema Languages: DTDs

Regular expressions should be deterministic

- Backward compatibility with SGML:
  "It is an error if an element in the document can match more than one occurrence of an element type in the content model [without looking ahead]."
- Example:
  $bc + bd$. Where do we match b in the string bd?
  $b(c + d)$ is deterministic.

Purpose: facilitate validation!

In which way does this constrain the schemas?

XML Schema Languages: DTDs

[Brüggemann-Klein, Wood 1998]:

- **Can you recognize deterministic regular expressions?**
- **What are the properties of deterministic regular expressions?**
- **Is it decidable whether a regexp is equivalent to a deterministic one?**

XML Schema Languages: DTDs

[Brüggemann-Klein, Wood 1998]:

- **Can you recognize deterministic regular expressions?**
  *A regular expression is deterministic (one-unambiguous) iff its Glushkov automaton is deterministic (*PTIME*).*
- **What are the properties of deterministic regular expressions?**
  - *Not every regular language can be denoted by a deterministic regular expression. E.g.,* $(a + b)^* a(a + b)$.
  - *Semantical characterization in terms of orbits*
- **Is it decidable whether a regexp is equivalent to a deterministic one?**
  Yes

XML Schema: Regular Tree Languages

Specialized DTDs (SDTDs) [Papak., Vianu, 2000]:

$\equiv$ tree automata on unranked trees

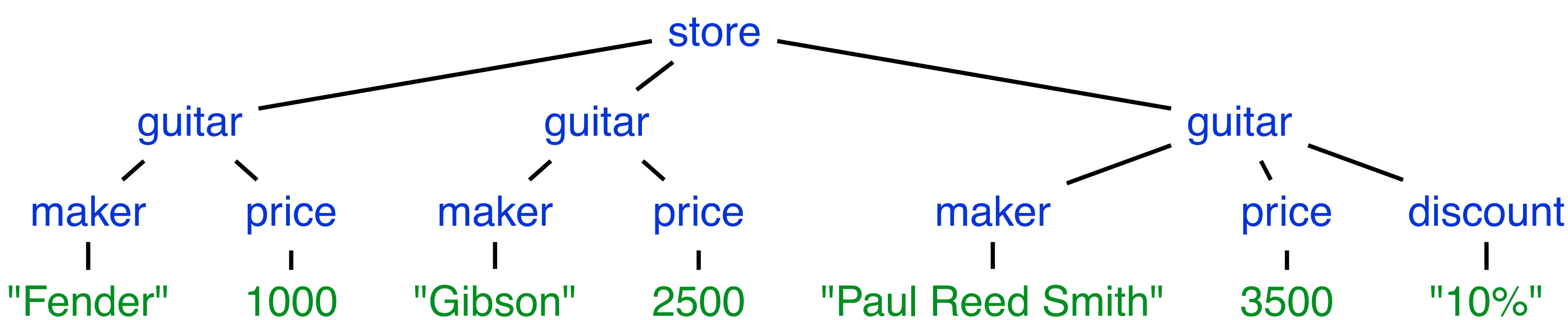
store	$\longrightarrow$	(guitar^1)* guitar^2 (guitar^2)*
guitar^1	$\longrightarrow$	maker price
guitar^2	$\longrightarrow$	maker price discount

XML Schema: Regular Tree Languages

Specialized DTDs (SDTDs) [Papak., Vianu, 2000]:

$\equiv$ tree automata on unranked trees

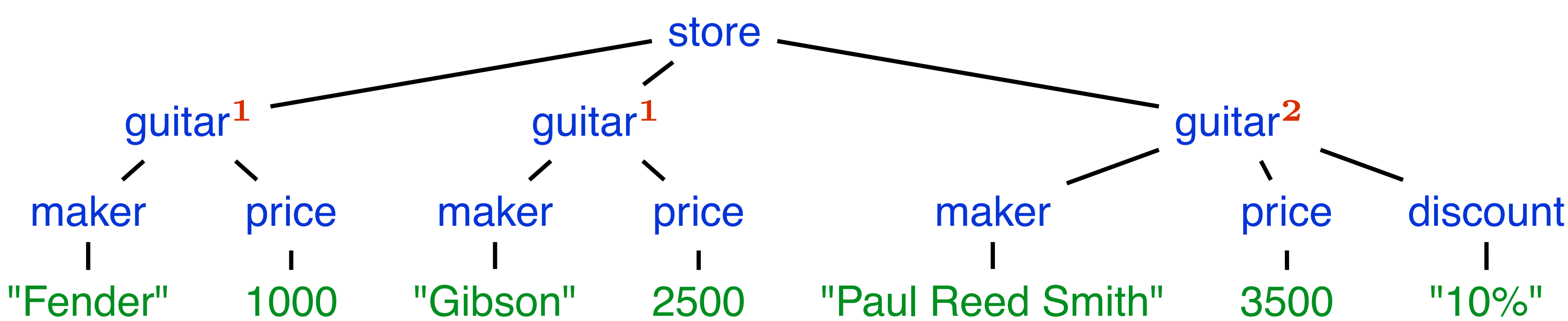
store	$\longrightarrow$	(guitar^1)* guitar^2 (guitar^2)*
guitar^1	$\longrightarrow$	maker price
guitar^2	$\longrightarrow$	maker price discount



XML Schema: Regular Tree Languages

Specialized DTDs (SDTDs) [Papak., Vianu, 2000]:
$\equiv$ tree automata on unranked trees

$$
\begin{array}{lcl}
\text{store} & \rightarrow & (\text{guitar}^1)^* \ \text{guitar}^2 \ (\text{guitar}^2)^* \\
\text{guitar}^1 & \rightarrow & \text{maker price} \\
\text{guitar}^2 & \rightarrow & \text{maker price discount}
\end{array}
$$



Typing: associating the right types to nodes

XML Schema

To facilitate validation/typing:

Element Declarations Consistent Rule (EDC):

"It is illegal to have two elements of the same name [. . .] but different types in a content model"

[XML Schema Part 0: Primer]

XML Schemas are SDTDs with a single-type restriction

[Murata,Lee,Mani 2001]

Outline

- XML Schema Languages
- Single-Type SDTDs
- 1-Pass Preorder Typing
- Restrained Competition SDTDs
- Unique Particle Attribution vs 1PPT
- Conclusion

EDC in SDTDs

Single-type SDTDs:

Different types for a label in the same rhs are not allowed!

Example:	store $\rightarrow$ (guitar$^{\mathbf{1}}$)* guitar$^{\mathbf{2}}$ (guitar^2)*	not allowed
	guitar^1 $\rightarrow$ maker^2 price^3	allowed

EDC in SDTDs

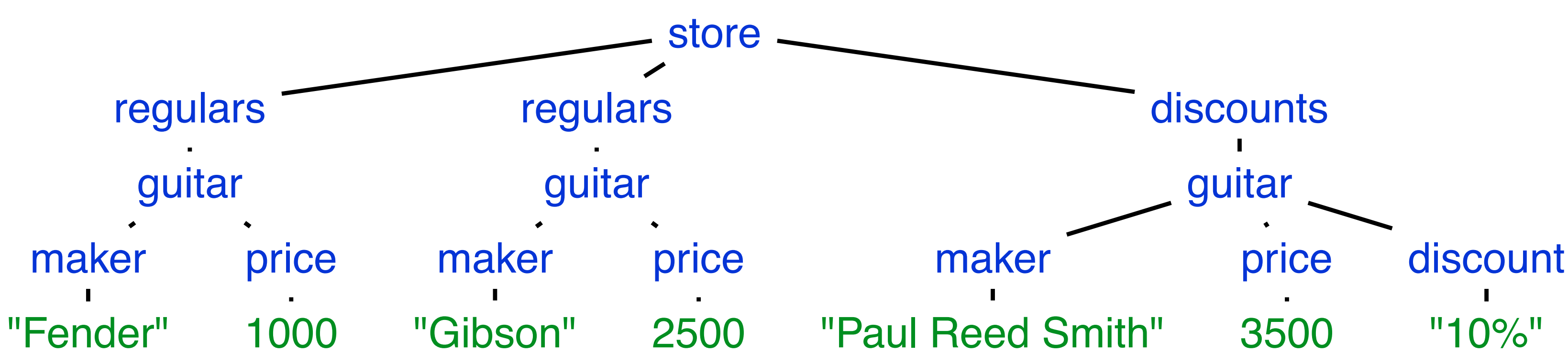
Single-type SDTDs:

$$
\begin{array}{lcl}
\text{store} & \rightarrow & \text{regulars}^* \ \text{discounts discounts}^* \\
\text{regulars} & \rightarrow & \text{guitar}^1 \\
\text{discounts} & \rightarrow & \text{guitar}^2 \\
\text{guitar}^1 & \rightarrow & \text{maker price} \\
\text{guitar}^2 & \rightarrow & \text{maker price discount}
\end{array}
$$

EDC in SDTDs

Single-type SDTDs:

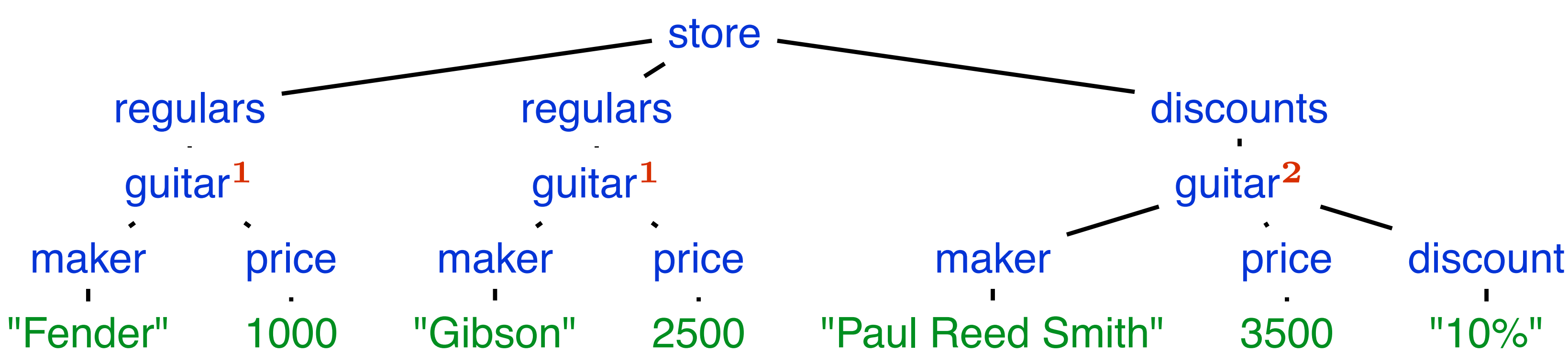
$$
\begin{array}{lcl}
\text{store} & \rightarrow & \text{regulars}^{*}\ \text{discounts discounts}^{*} \\
\text{regulars} & \rightarrow & \text{guitar}^{1} \\
\text{discounts} & \rightarrow & \text{guitar}^{2} \\
\text{guitar}^{1} & \rightarrow & \text{maker price} \\
\text{guitar}^{2} & \rightarrow & \text{maker price discount}
\end{array}
$$



EDC in SDTDs

Single-type SDTDs:

$$
\begin{array}{lcl}
\text{store} & \rightarrow & \text{regulars}^{*}\ \text{discounts discounts}^{*} \\
\text{regulars} & \rightarrow & \text{guitar}^{1} \\
\text{discounts} & \rightarrow & \text{guitar}^{2} \\
\text{guitar}^{1} & \rightarrow & \text{maker price} \\
\text{guitar}^{2} & \rightarrow & \text{maker price discount}
\end{array}
$$

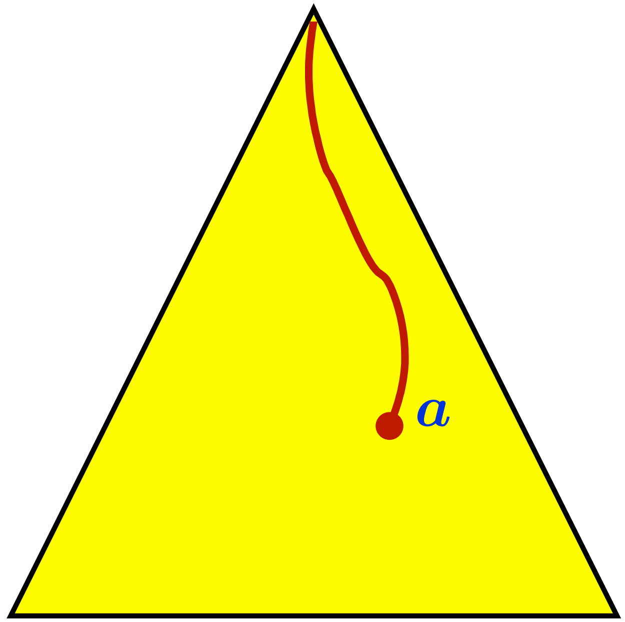


Note: DTD $\subseteq$ single-type SDTD $\subseteq$ SDTD

Questions

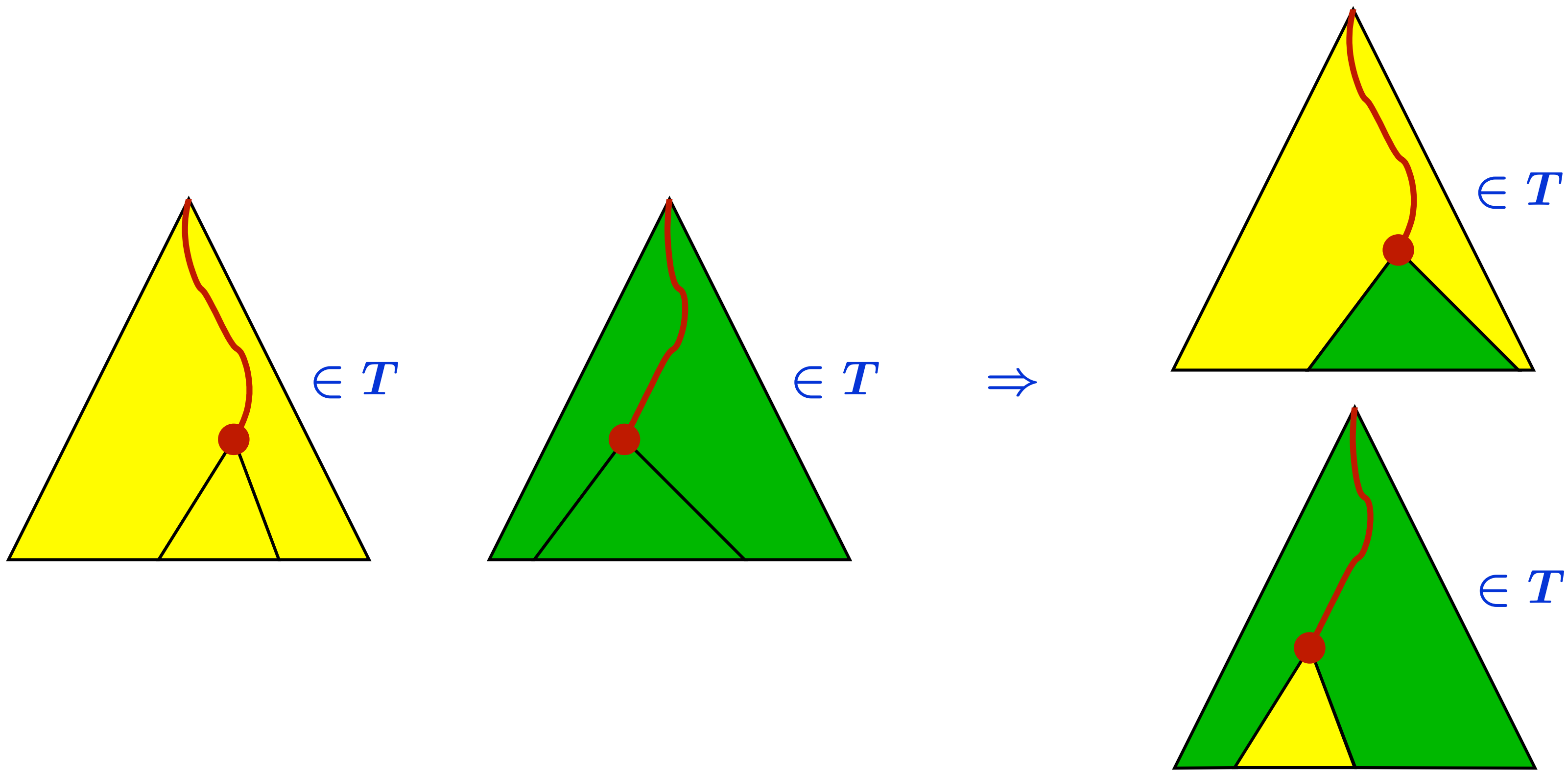
- **Can you recognize single-type SDTDs?**
  Trivial
- **What kind of languages can be defined by single-type SDTDs?**
  ???
- **Is it decidable whether an SDTD is equivalent to a single-type SDTD?**
  ???

The Ancestor-String



Ancestor-Guarded Subtree Exchange

T a regular tree language



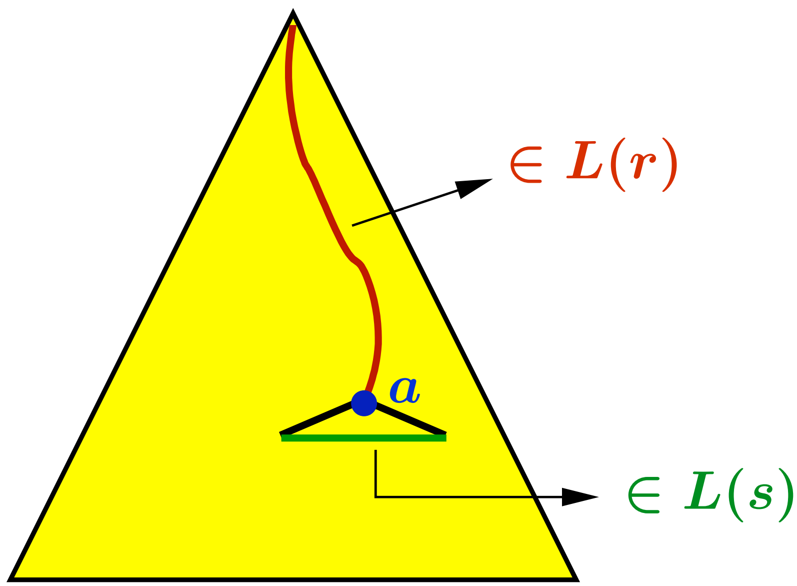
The Equivalence

Let T be a regular tree language

THEOREM: The following are equivalent:

- T is definable by a single-type SDTD
- T is closed under ancestor-guarded subtree exchange

Ancestor-Guarded DTDs



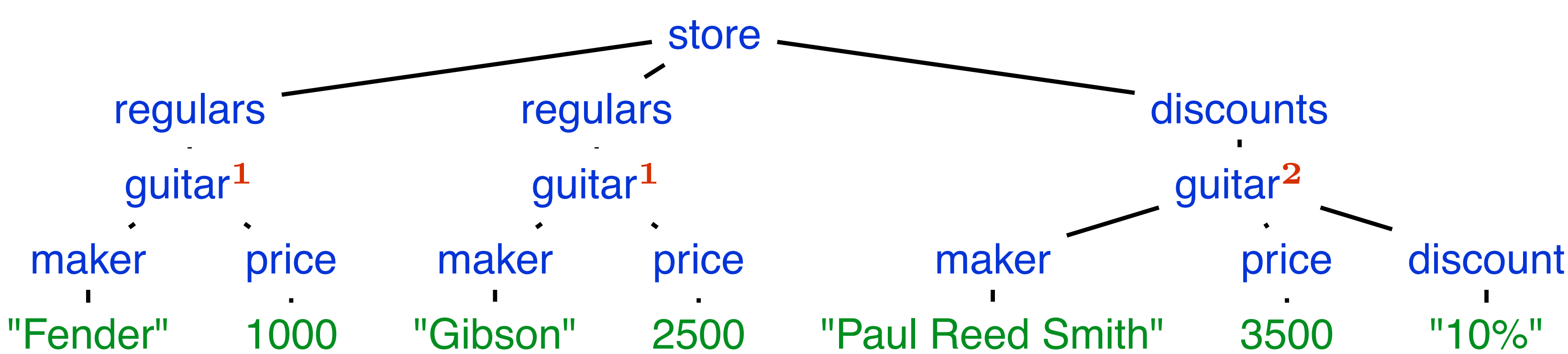
Ancestor-guarded DTD consists of triples $(r, a) \longrightarrow s$

Ancestor-Guarded DTDs

Ancestor-guarded DTD consists of triples $(r, a) \longrightarrow s$

Example:

(ε,	store)	$\rightarrow$	regulars* discounts discounts*
(store,	regulars)	$\rightarrow$	guitar
(store,	discounts)	$\rightarrow$	guitar
(store regulars,	guitar)	$\rightarrow$	maker price
(store discounts,	guitar)	$\rightarrow$	maker price discount



The Equivalence

Let T be a regular tree language

THEOREM: The following are equivalent:

- T is definable by a single-type SDTD
- T is closed under ancestor-guarded subtree exchange
- T is definable by an ancestor-guarded DTD

Questions

- **Can you recognize single-type SDTDs?**
  Trivial
- **What kind of languages can be defined by single-type SDTDs?**
  - *Semantical characterizations:*
    - *ancestor-guarded subtree exchange*
    - ...
  - *Syntactical characterizations:*
    - *ancestor-guarded DTDs*
    - ...
- **Is it decidable whether an SDTD is equivalent to a single-type SDTD?**

Questions

- **Can you recognize single-type SDTDs?**
  Trivial
- **What kind of languages can be defined by single-type SDTDs?**
  - *Semantical characterizations:*
    - *ancestor-guarded subtree exchange*
    - ...
  - *Syntactical characterizations:*
    - *ancestor-guarded DTDs*
    - ...
- **Is it decidable whether an SDTD is equivalent to a single-type SDTD?**
  Yes, EXPTIME-*complete*

Outline

- XML Schema Languages
- Single-Type SDTDs
- 1-Pass Preorder Typing
- Restrained Competition SDTDs
- Unique Particle Attribution vs 1PPT
- Conclusion

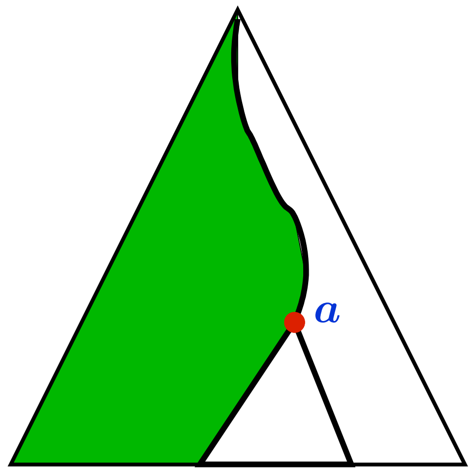
Optimize EDC?

EDC: type of a node only depends on ancestor-string!

Optimize EDC?

EDC: type of a node only depends on ancestor-string!

XML streaming: determine the type of a node when its opening tag is met

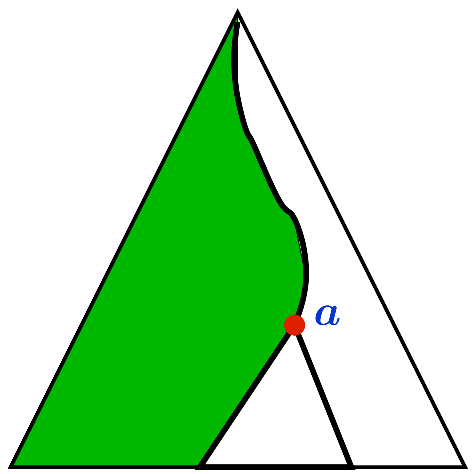


We call this 1-Pass Preorder Typing

Optimize EDC?

EDC: type of a node only depends on ancestor-string!

XML streaming: determine the type of a node when its opening tag is met



We call this 1-Pass Preorder Typing

$\neq$ 1-Pass Preorder Validation

$$\begin{array}{lll} a & \rightarrow & b^1 + b^2 \\ b^1 & \rightarrow & c \\ b^2 & \rightarrow & d \end{array}$$

defines trees $a - b^1 - c$ and $a - b^2 - d$

Questions

- **Can you recognize 1PPT SDTDs?**
  ???
- **What kind of languages can be defined by 1PPT SDTDs?**
  ???
- **Is it decidable whether an SDTD is equivalent to a 1PPT SDTD?**
  ???

Outline

- XML Schema Languages
- Single-Type SDTDs
- 1-Pass Preorder Typing
- Restrained Competition SDTDs
- Unique Particle Attribution vs 1PPT
- Conclusion

Restrained Competition SDTDs

[Murata et al., 2001]: A regular expression r over types restrains competition iff there are no strings wa^iv and wa^jv' in $L(r)$ with $i \neq j$

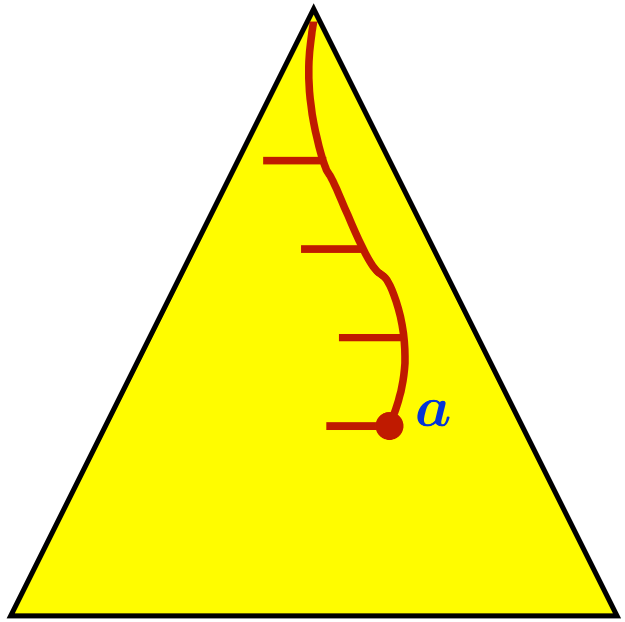
I.e. the type of a is determined by its left siblings

An SDTD is restrained competition iff every regular expression restrains competition

store	$\rightarrow$	(guitar^1)* discounts (guitar^2)$^+$
discounts	$\rightarrow$	ε
guitar^1	$\rightarrow$	maker price
guitar^2	$\rightarrow$	maker price discount

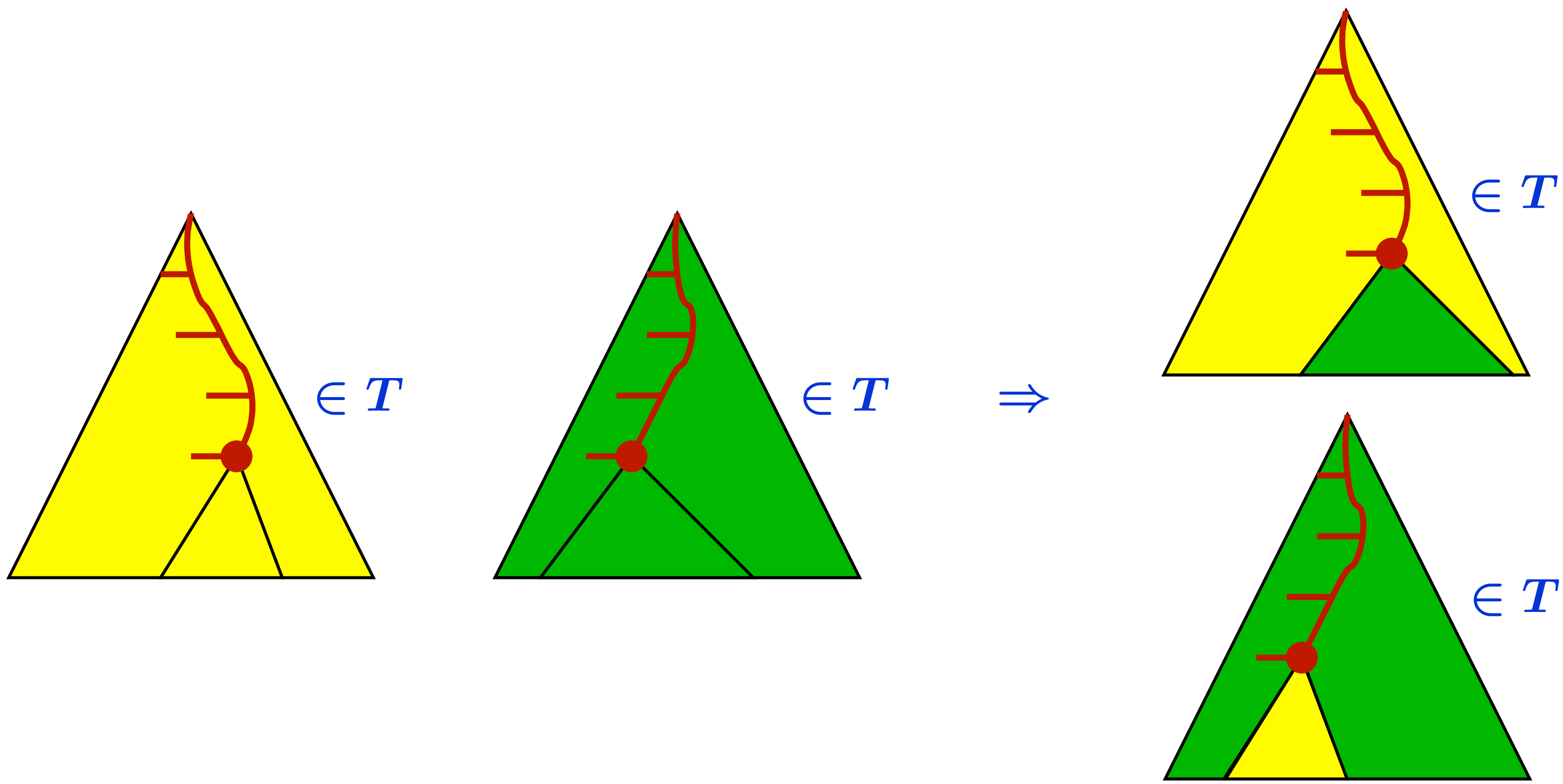
Note: DTD $\subseteq$ stSDTD $\subseteq$ rcSDTD $\subseteq$ SDTD

The Ancestor-Sibling-String

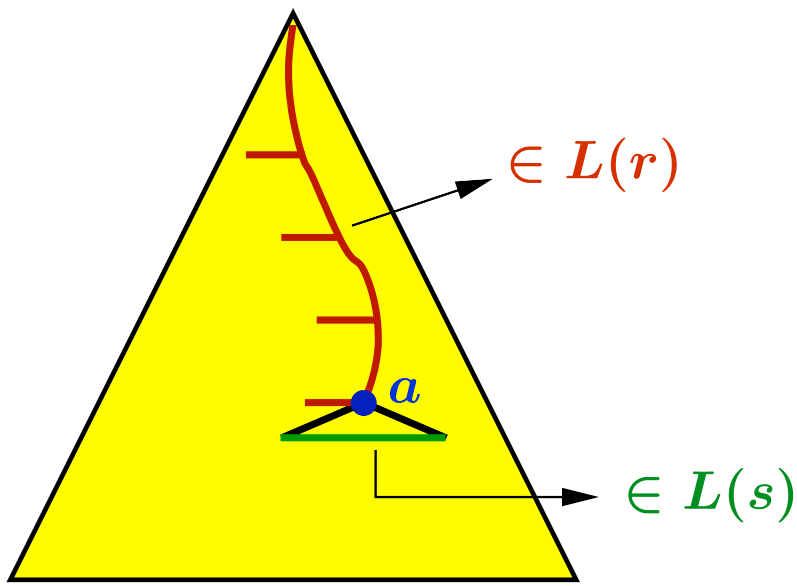


Anc-Sib-Guarded Subtree Exchange

$\boldsymbol{T}$ a regular tree language

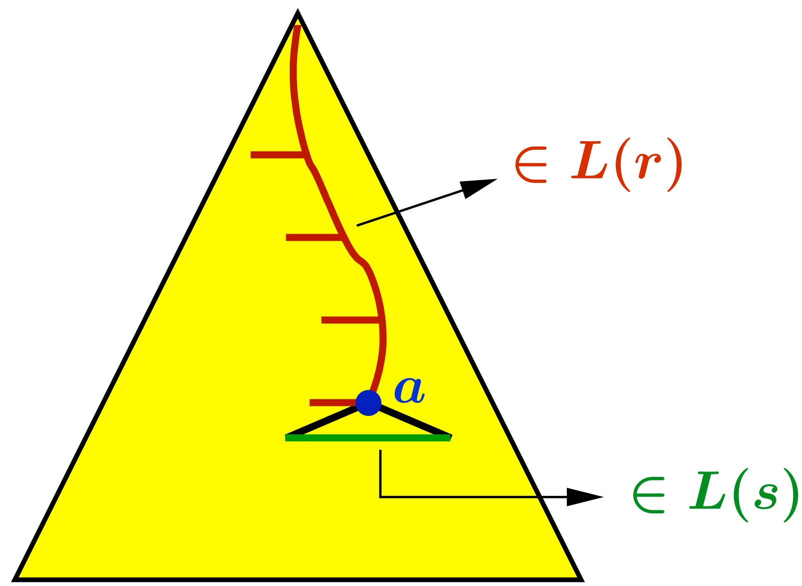


Ancestor-Sibling-Guarded DTDs



Anc-Sib guarded DTD consists of triples $(r, a) \rightarrow s$

Ancestor-Sibling-Guarded DTDs



Anc-Sib guarded DTD consists of triples $(r, a) \longrightarrow s$

Example:

$$
\begin{array}{rrcl}
(\varepsilon, & \text{store}) & \longrightarrow & \text{guitar}^* \text{ discounts guitar}^+ \\
(\text{store} \# \text{guitar}^*, & \text{discounts}) & \longrightarrow & \varepsilon \\
(\text{store} \# \text{guitar}^*, & \text{guitar}) & \longrightarrow & \text{maker price} \\
(\text{store} \# \text{guitar}^* \text{ discounts guitar}^*, & \text{guitar}) & \longrightarrow & \text{maker price discount}
\end{array}
$$

The Equivalence

Let T be a regular tree language.

THEOREM: The following are equivalent:

- T is definable by a restrained competition SDTD
- T is closed under ancestor-sibling-guarded subtree exchange
- T is definable by an ancestor-sibling-guarded DTD

The Equivalence

Let T be a regular tree language.

THEOREM: The following are equivalent:

- T is definable by a restrained competition SDTD
- T is closed under ancestor-sibling-guarded subtree exchange
- T is definable by an ancestor-sibling-guarded DTD
- T allows 1-Pass Preorder Typing

1-Pass Preorder Typeable SDTDs are exactly the rcSDTDs!

The Equivalence: 1PPT vs rcSDTD

Intuition: not rcSDTD implies not 1PPT

Suppose we have $x \rightarrow r$

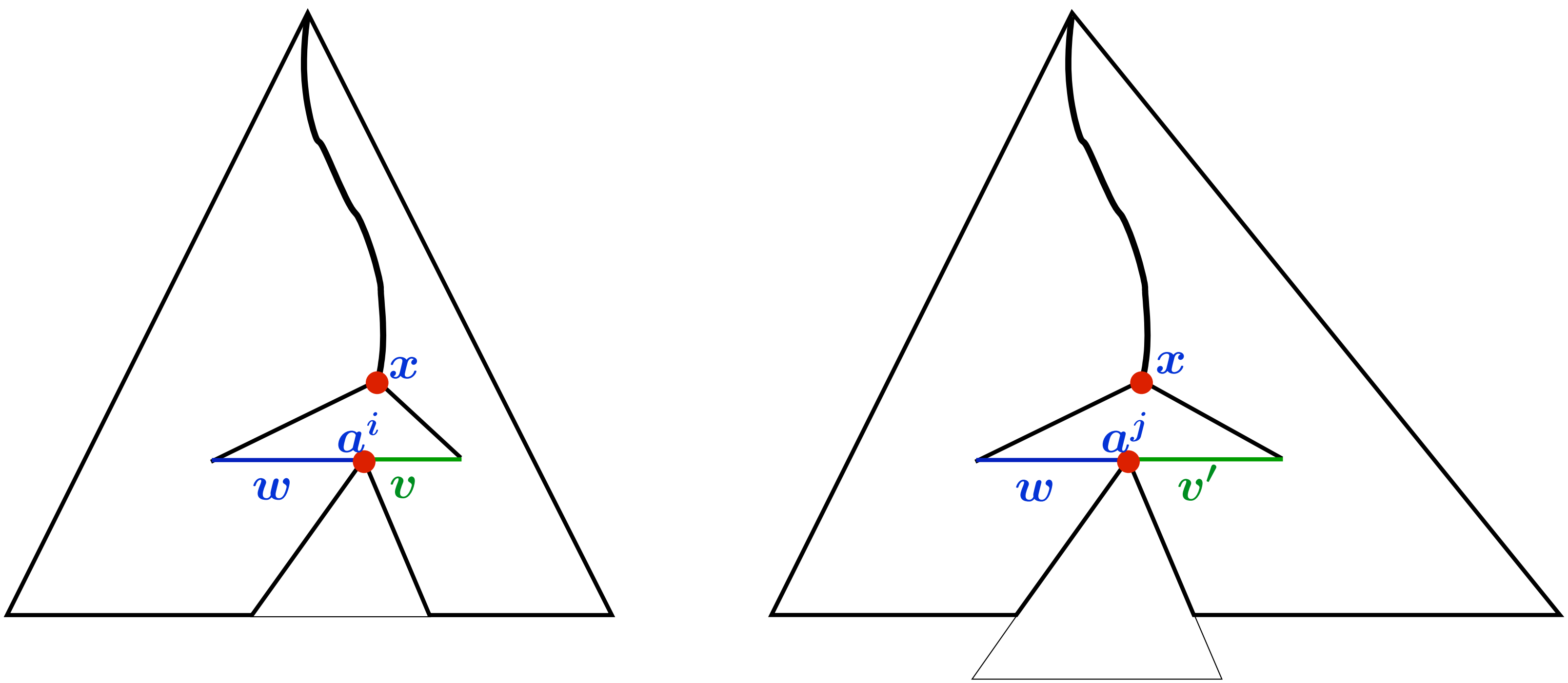
where wa^iv and wa^jv' in $L(r)$ and $i \neq j$

The Equivalence: 1PPT vs rcSDTD

Intuition: not rcSDTD implies not 1PPT
Suppose we have $x \longrightarrow r$

where wa^iv and wa^jv' in $L(r)$ and $i \neq j$

Then we can make trees



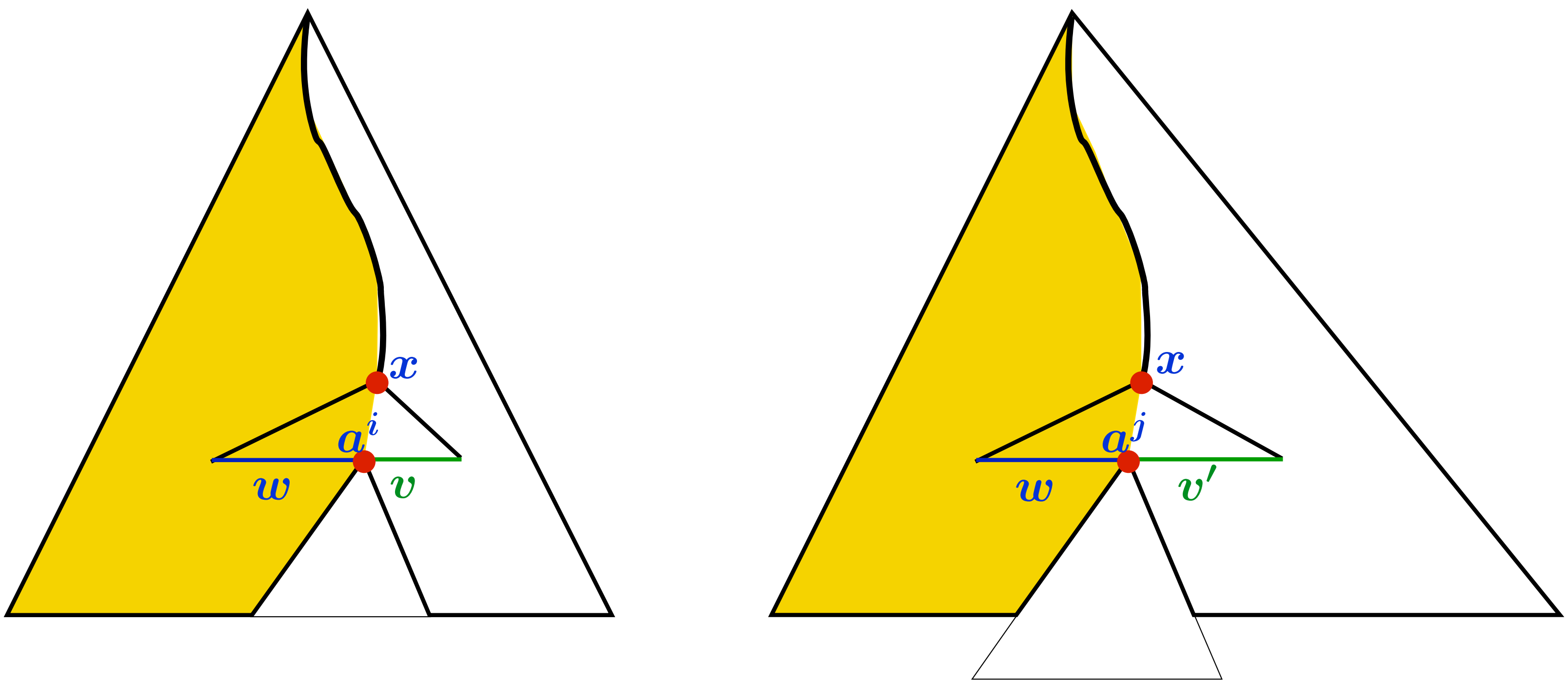
The Equivalence: 1PPT vs rcSDTD

Intuition: not rcSDTD implies not 1PPT

Suppose we have $x \longrightarrow r$

where wa^iv and wa^jv' in $L(r)$ and $i \neq j$

Then we can make trees



Questions

- **Can you recognize 1PPT SDTDs?**
- **What kind of languages can be defined by 1PPT SDTDs?**
  - *Semantical characterizations:*
    - *restrained competition SDTDs*
    - *ancestor-sibling guarded subtree-exchange*
    - *...*
  - *Syntactical characterizations:*
    - *ancestor-sibling guarded DTD*
    - ...
- **Is it decidable whether an SDTD is equivalent to a 1PPT SDTD?**

Questions

- **Can you recognize 1PPT SDTDs?**
  Yes, in NLOGSPACE
- **What kind of languages can be defined by 1PPT SDTDs?**
  - *Semantical characterizations:*
    - *restrained competition SDTDs*
    - *ancestor-sibling guarded subtree-exchange*
    - *...*
  - *Syntactical characterizations:*
    - *ancestor-sibling guarded DTD*
    - ...
- **Is it decidable whether an SDTD is equivalent to a 1PPT SDTD?**
  Yes, EXPTIME*-complete*

Outline

- XML Schema Languages
- Single-Type SDTDs
- 1-Pass Preorder Typing
- Restrained Competition SDTDs
- Unique Particle Attribution vs 1PPT
- Conclusion

Unique Particle Attribution vs 1PPT

Unique Particle Attribution

New name for one-unambiguous or determinism constraint

- Unique Particle Attribution $\Rightarrow$ 1PPT
  Intuition:

$$
\begin{array}{ccccccc}
a^1? & b^1 & (b^2 & + & c^1)^* & a^2 & c^1 \\
a_1? & b_2 & (b_3 & + & c_4)^* & a_5 & c_6
\end{array}
$$

rc:	$wa^iv \in L(r), wa^jv' \in L(r) \Rightarrow i = j$
deterministic:	$wa_iv \in L(r), wa_jv' \in L(r) \Rightarrow i = j$

Unique Particle Attribution vs 1PPT

Unique Particle Attribution

New name for one-unambiguous or determinism constraint

- Unique Particle Attribution $\Rightarrow$ 1PPT
  Intuition:

$$
\begin{array}{ccccccc}
a^1? & b^1 & (b^2 & + & c^1)^* & a^2 & c^1 \\
a_1? & b_2 & (b_3 & + & c_4)^* & a_5 & c_6
\end{array}
$$

- 1PPT $\not\Rightarrow$ Unique Particle Attribution
  Example: $(a^1 + b^1)^* a^1 (a^1 + b^1)$

Outline

- XML Schema Languages
- Single-Type SDTDs
- 1-Pass Preorder Typing
- Restrained Competition SDTDs
- Unique Particle Attribution vs 1PPT
- Conclusion

Conclusions

- EDC, 1PPT have elegant semantical characterizations
- Characterizations provide a toolbox for proofs

Conclusions

- 1PPT is strictly larger than EDC
- 1PPT is a robust notion
- 1PPT has a syntactical counterpart
- Validation and typing against 1PPT essentially not harder than against EDC
- When content models are deterministic:
  - inclusion/equivalence of 1PPT SDTDs in **PTIME**
  - minimizing is in **PTIME**, unique minimal 1PPT SDTD

These are the same complexities as for EDC SDTDs!

Conclusions

- 1PPT is strictly larger than EDC
- 1PPT is a robust notion
- 1PPT has a syntactical counterpart
- Validation and typing against 1PPT essentially not harder than against EDC
- When content models are deterministic:
  - inclusion/equivalence of 1PPT SDTDs in **PTIME**
  - minimizing is in **PTIME**, unique minimal 1PPT SDTD

  These are the same complexities as for EDC SDTDs!
- EDC is currently in XML Schema specification!