Regular Object Types

Vladimir Gapeyev

Benjamin C. Pierce

University of Pennsylvania
Philadelphia, USA

XML

```
<readings>
  <paper>
    <title>Import is
           not Inheritance</title>
    <author>Szyperski</author>
    <inproc>ECOOP'92</inproc>
  </paper>
  <paper>
    <title>Classes and Mixins</title>
    <author>Flatt</author>
    <author>Krishnamurthi</author>
    <author>Felleisen</author>
    <inbook>LNCS 1523</inbook>
  </paper>
</readings>
```

XML and Regular Expression (RE) types

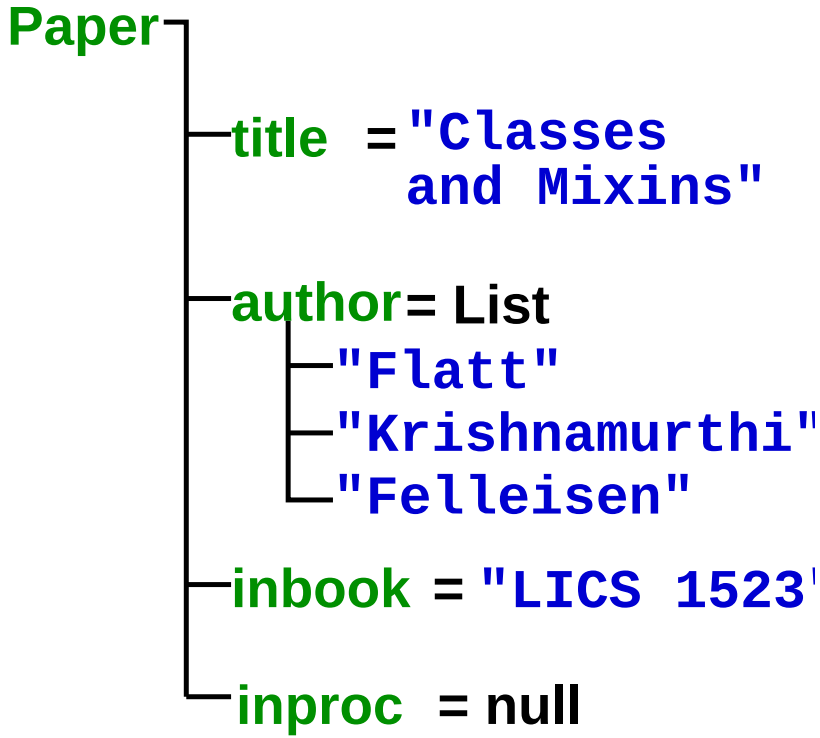
```
<readings>
  <paper>
    <title>Import is
           not Inheritance</title>
    <author>Szyperski</author>
    <inproc>ECOOP'92</inproc>
  </paper>
  <paper>
    <title>Classes and Mixins</title>
    <author>Flatt</author>
    <author>Krishnamurthi</author>
    <author>Felleisen</author>
    <inbook>LNCS 1523</inbook>
  </paper>
</readings>
```

```
Title  = <title>[pcdata]
Author = <author>[pcdata]

Source  =   <inproc>[pcdata]
          | <inbook>[pcdata]

Paper = <paper>[Title,
                Author+, Source]

Readings = <readings>[Paper*]
```

DOM: a generic AST for XML

```
Element
  name =   "paper"
  children
    Element
      name =  "title"
      children
        Text
          data =  "Classes and Mixins"
    Element
      name =  "author"
      children
        Text
          data =  "Flatt"
    Element
      name =  "author"
      children
        Text
          data =  "Krishnamurthi"
    Element
      name =  author"
      children
        Text
          data =  "Felleisen"
    Element
      name =  "inbook"
      children
        Text
          data =  "LICS 1523"
```

XML data binding: schema-specific ASTs




```
public interface Paper extends
  Element, PaperType {}

public interface PaperType {
    String getTitle();
    void setTitle(String value);

    List getAuthor();

    String getInbook();
    void setInbook(String value);

    String getInproc();
    void setInproc(String value);
}
```

XDUCE: RE types can be PL types

```
fun getConferenceTitles (ps as Paper*) : Title* =
  match ps with
  | <paper>[t as Title, Author*, <inproc>[pcdata]],
    rest as Paper* ->
      t, getConferenceTitles(rest)
  | Paper, rest as Paper* ->
      getConferenceTitles(rest)
  | () -> ()
```

XDUCE: [Hosoya,Vouillon,Pierce – ICFP'00], [Hosoya,Pierce – POPL'01]

XTATIC project: XDUCE to the masses!

Vision:

Native support
for statically typed XML programming
in a mainstream OO language.

XTATIC project: XDUCE to the masses!

Vision:

Native support
for statically typed XML programming
in a mainstream OO language.

Technical challenge:

Combine an OO type system and an RE type system.

XTATIC project: XDUCE to the masses!

Vision:

Native support
for statically typed XML programming
in a mainstream OO language.

Technical challenge:

Combine an OO type system and an RE type system.

Practical focus:

XTATIC = $C^{\#}$ + XDUCE

XTATIC project: XDUCE to the masses!

Vision:

Native support
for statically typed XML programming
in a mainstream OO language.

Technical challenge:

Combine an OO type system and an RE type system.

Practical focus:

XTATIC = C# + XDUCE

Theory:

FX = FJ + XDUCE

FJ : Igarashi,Pierce,Wadler – OOPSLA'99]

XDUCE's type system

Values: $\overline{\mathtt{t}} ::= \mathtt{t}_1, \ldots, \mathtt{t}_n \qquad \mathtt{t} ::= \langle(\mathtt{l})\rangle[\overline{\mathtt{t}}]$

XDUCE's type system

Values: $\overline{t} ::= t_1, \ldots, t_n$ $\quad t ::= \langle(l)\rangle[\overline{t}]$

$\langle(L)\rangle[T]$

T_1, T_2

$T_1 | T_2$

$[]$

$T*$

X

XDUCE's type system

Values: $\overline{\mathtt{t}} ::= \mathtt{t}_1, \ldots, \mathtt{t}_n \qquad \mathtt{t} ::= \langle(\mathtt{l})\rangle[\overline{\mathtt{t}}]$

Denotations:

$$
\begin{array}{lcl}
[\![\langle(\mathtt{L})\rangle[\mathtt{T}]]\!] & = & \{\langle(\mathtt{l})\rangle[\overline{\mathtt{t}}] \mid \mathtt{l} \in [\![\mathtt{L}]\!],\ \ \overline{\mathtt{t}} \in [\![\mathtt{T}]\!]\} \\
[\![\mathtt{T}_1,\mathtt{T}_2]\!] & = & \{\overline{\mathtt{t}}_1,\overline{\mathtt{t}}_2 \mid \overline{\mathtt{t}}_k \in [\![\mathtt{T}_k]\!]\} \\
[\![\mathtt{T}_1|\mathtt{T}_2]\!] & = & [\![\mathtt{T}_1]\!] \cup [\![\mathtt{T}_2]\!] \\
[\![[\,]]\!] & = & \{[\,]\} \\
[\![\mathtt{T}*]\!] & = & \{\overline{\mathtt{t}}_1,\ldots,\overline{\mathtt{t}}_n \mid \overline{\mathtt{t}}_k \in [\![\mathtt{T}]\!]\} \\
[\![\mathtt{X}]\!] & = & [\![\mathit{def}(\mathtt{X})]\!]
\end{array}
$$

XDUCE's type system

Values: $\overline{\mathtt{t}} ::= \mathtt{t}_1, \ldots, \mathtt{t}_n \qquad \mathtt{t} ::= \langle(\mathtt{l})\rangle[\overline{\mathtt{t}}]$

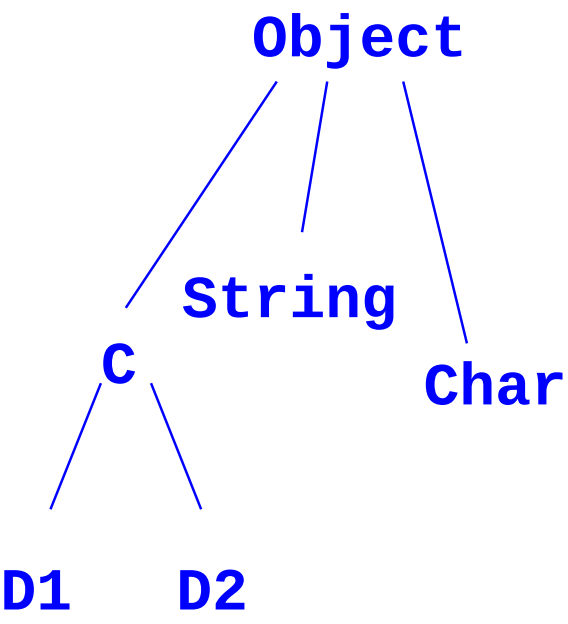
Denotations:

$$
\begin{array}{lcl}
[\![\langle(\mathtt{L})\rangle[\mathtt{T}]]\!] & = & \{\langle(\mathtt{l})\rangle[\overline{\mathtt{t}}] \mid \mathtt{l} \in [\![\mathtt{L}]\!],\ \ \overline{\mathtt{t}} \in [\![\mathtt{T}]\!]\} \\
[\![\mathtt{T}_1,\mathtt{T}_2]\!] & = & \{\overline{\mathtt{t}}_1,\overline{\mathtt{t}}_2 \mid \overline{\mathtt{t}}_k \in [\![\mathtt{T}_k]\!]\} \\
[\![\mathtt{T}_1|\mathtt{T}_2]\!] & = & [\![\mathtt{T}_1]\!] \cup [\![\mathtt{T}_2]\!] \\
[\![[\,]]\!] & = & \{[\,]\} \\
[\![\mathtt{T}*]\!] & = & \{\overline{\mathtt{t}}_1,\ldots,\overline{\mathtt{t}}_n \mid \overline{\mathtt{t}}_k \in [\![\mathtt{T}]\!]\} \\
[\![\mathtt{X}]\!] & = & [\![\mathit{def}(\mathtt{X})]\!]
\end{array}
$$

Semantic subtyping:

$$\mathtt{T}_1 <: \mathtt{T}_2 \qquad \Longleftrightarrow \qquad [\![\mathtt{T}_1]\!] \subseteq [\![\mathtt{T}_2]\!]$$

OO subtyping

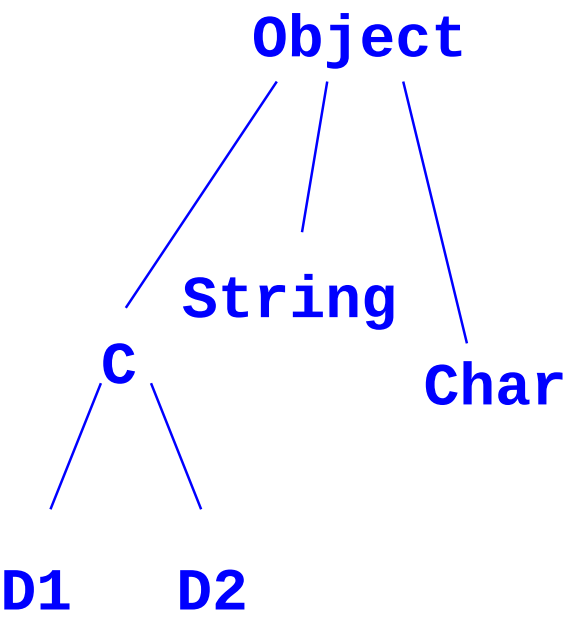


Subclassing: `D` $\sqsubset$: `C`

— reflexive and transitive closure of

$$\texttt{class C}_1\ \texttt{extends C}_2\ \texttt{\{...\}}$$

OO subtyping — semantically



Subclassing: $\mathtt{D} \sqsubset: \mathtt{C}$

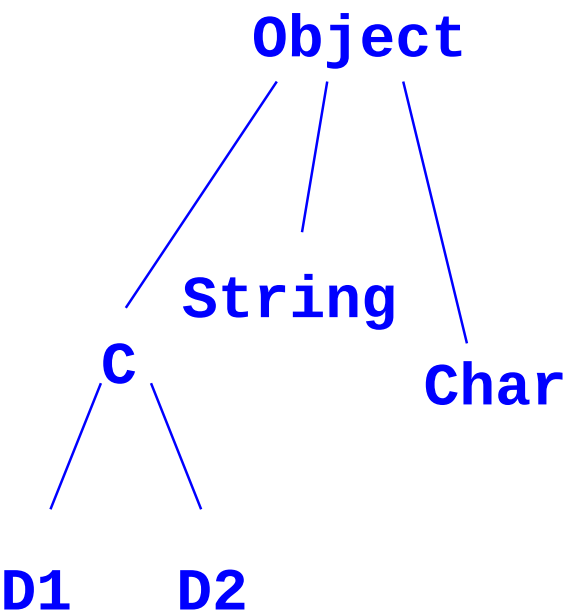
— reflexive and transitive closure of

$$\mathtt{class\ C_1\ extends\ C_2\ \{...\}}$$

Define:

$$[\![\mathtt{C}]\!] \;=\; \{\mathtt{new\ D}(\overline{\mathtt{o}}) \mid \mathtt{D} \sqsubset: \mathtt{C},\ \mathit{fields}(\mathtt{D}) = \overline{\mathtt{F}}\ \overline{\mathtt{f}},\ \mathtt{o_i} \in [\![\mathtt{F_i}]\!]\}$$

OO subtyping — semantically



Subclassing: $\mathtt{D} \sqsubset: \mathtt{C}$

— reflexive and transitive closure of

$$\mathtt{class\ C_1\ extends\ C_2\ \{...\}}$$

Define:

$$[\![\mathtt{C}]\!] \;=\; \{\mathtt{new\ D}(\overline{\mathtt{o}}) \mid \mathtt{D} \sqsubset: \mathtt{C},\ \mathit{fields}(\mathtt{D}) = \overline{\mathtt{F}}\ \overline{\mathtt{f}},\ \mathtt{o_i} \in [\![\mathtt{F_i}]\!]\}$$

Then:

$$\mathtt{C_1} \sqsubset: \mathtt{C_2} \iff [\![\mathtt{C_1}]\!] \subseteq [\![\mathtt{C_2}]\!]$$

XTATIC = C# + XDUCE

Values:

$$t ::= \langle(a)\rangle[\overline{t}]$$

Types:

$$T ::= \langle(A)\rangle[T] \quad T_1,T_2 \quad T_1|T_2 \quad [] \quad T* \quad X$$

XTATIC = C# + XDUCE

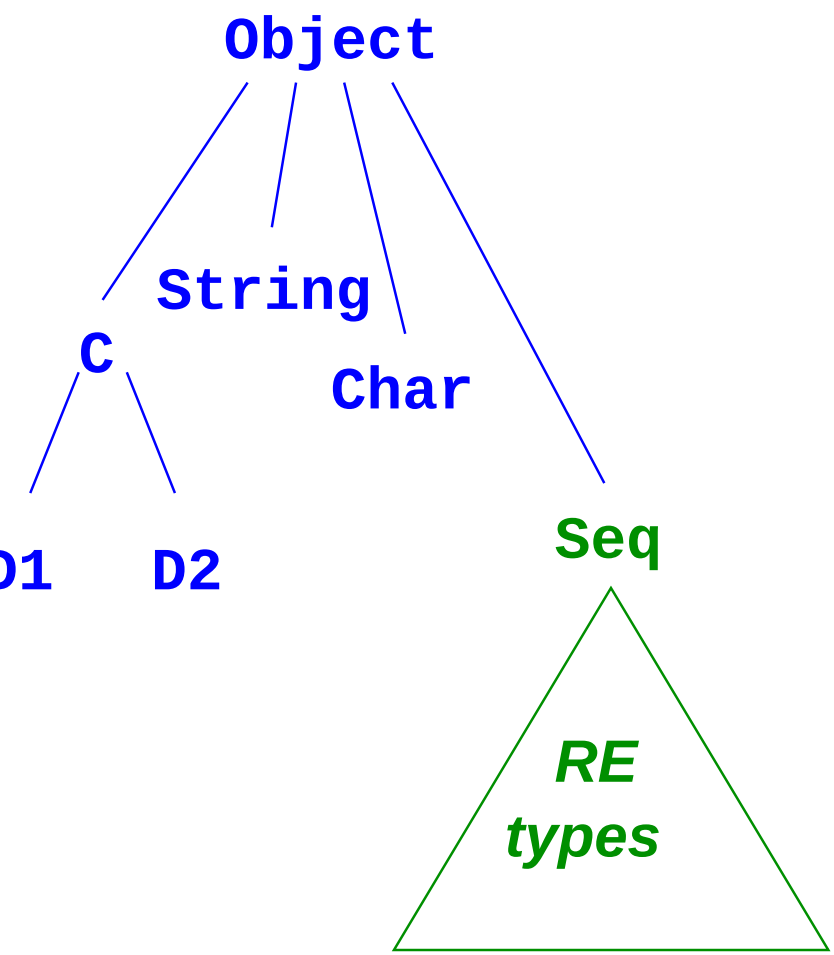
Values:

a	::=	
	new C($\overline{a}$)	
	[$\overline{t}$]	
t	::=	
	<(a)>[$\overline{t}$]	

Types:

A	::=	
	C	
	[T]	
T	::=	
	<(A)>[T]	
	T_1,T_2	
	T_1|T_2	
	[]	
	T*	
	X	

XTATIC = C# + XDUCE



Values:

a	::=	
	new C($\overline{a}$)	
	[$\overline{t}$]	
t	::=	
	<(a)>[$\overline{t}$]	

Types:

A	::=	
	C	
	[T]	
T	::=	
	<(A)>[T]	
	T_1,T_2	
	T_1|T_2	
	[]	
	T*	
	X	

PIZZA: [Odersky,Wadler – POPL'97]

Type denotations and subtyping

$$[\![[\langle(\mathtt{A})\rangle[\mathtt{T}]]]\!] \quad = \quad \{\langle(\mathtt{a})\rangle[\overline{\mathtt{t}}] \mid \mathtt{a} \in [\![\mathtt{A}]\!], \quad [\overline{\mathtt{t}}] \in [\![[\mathtt{T}]]\!]\}$$

$$\ldots$$

Type denotations and subtyping

$$[\![[\langle(\mathtt{A})\rangle[\mathtt{T}]]]\!] \quad = \quad \{ \langle(\mathtt{a})\rangle[\overline{\mathtt{t}}] \mid \mathtt{a} \in [\![\mathtt{A}]\!], \quad [\overline{\mathtt{t}}] \in [\![[\mathtt{T}]]\!] \}$$

$$\ldots$$

$$[\![\mathtt{Seq}]\!] \quad = \quad \{ [\overline{\mathtt{t}}] \mid \overline{\mathtt{t}} \in [\![[\mathtt{T}]]\!] \text{ for some } \mathtt{T} \}$$

Type denotations and subtyping

$$[\![[\langle(\mathtt{A})\rangle[\mathtt{T}]]]\!] \;=\; \{\, \langle(\mathtt{a})\rangle[\overline{\mathtt{t}}] \mid \mathtt{a} \in [\![\mathtt{A}]\!],\quad [\overline{\mathtt{t}}] \in [\![[\mathtt{T}]]\!] \,\}$$

$$\ldots$$

$$[\![\mathtt{Seq}]\!] \;=\; \{\, [\overline{\mathtt{t}}] \mid \overline{\mathtt{t}} \in [\![[\mathtt{T}]]\!] \text{ for some } \mathtt{T} \,\}$$

$$[\![\mathtt{C}]\!] \;=\; \{\, \mathtt{new}\ \mathtt{D}(\overline{\mathtt{a}}) \mid \mathtt{D} \sqsubset\!: \mathtt{C},\ \mathit{fields}(\mathtt{D}) = \overline{\mathtt{F}}\ \overline{\mathtt{f}},\ \mathtt{a}_i \in [\![\mathtt{F}_i]\!] \,\}$$

Type denotations and subtyping

$$
\begin{array}{lcl}
[\![[\langle(\mathtt{A})\rangle[\mathtt{T}]]]\!] & = & \{\, \langle(\mathtt{a})\rangle[\overline{\mathtt{t}}] \mid \mathtt{a} \in [\![\mathtt{A}]\!],\quad [\overline{\mathtt{t}}] \in [\![[\mathtt{T}]]\!] \,\} \\
 & \ldots & \\
[\![\mathtt{Seq}]\!] & = & \{\, [\overline{\mathtt{t}}] \mid \overline{\mathtt{t}} \in [\![[\mathtt{T}]]\!] \text{ for some } \mathtt{T} \,\} \\
[\![\mathtt{C}]\!] & = & \{\, \mathtt{new}\ \mathtt{D}(\overline{\mathtt{a}}) \mid \mathtt{D} \sqsubset\!: \mathtt{C},\ \mathit{fields}(\mathtt{D}) = \overline{\mathtt{F}}\ \overline{\mathtt{f}},\ \mathtt{a}_i \in [\![\mathtt{F}_i]\!] \,\} \\
 & & (\cup [\![\mathtt{Seq}]\!],\ \text{if } \mathtt{C} = \mathtt{Object})
\end{array}
$$

Type denotations and subtyping

$$[\![[\langle(\mathtt{A})\rangle[\mathtt{T}]]]\!] \quad = \quad \{\, \langle(\mathtt{a})\rangle[\overline{\mathtt{t}}] \mid \mathtt{a} \in [\![\mathtt{A}]\!],\quad [\overline{\mathtt{t}}] \in [\![[\mathtt{T}]]\!] \,\}$$

$$\ldots$$

$$[\![\mathtt{Seq}]\!] \quad = \quad \{\, [\overline{\mathtt{t}}] \mid \overline{\mathtt{t}} \in [\![[\mathtt{T}]]\!] \text{ for some } \mathtt{T} \,\}$$

$$[\![\mathtt{C}]\!] \quad = \quad \{\, \mathtt{new}\ \mathtt{D}(\overline{\mathtt{a}}) \mid \mathtt{D} \sqsubset\!: \mathtt{C},\ \mathit{fields}(\mathtt{D}) = \overline{\mathtt{F}}\ \overline{\mathtt{f}},\ \mathtt{a}_i \in [\![\mathtt{F}_i]\!] \,\}$$

$$(\cup [\![\mathtt{Seq}]\!],\ \text{if } \mathtt{C} = \mathtt{Object})$$

Define XTATIC subtyping $<\!:$ as

$$\mathtt{A}_1 <\!: \mathtt{A}_2 \quad \Longleftrightarrow \quad [\![\mathtt{A}_1]\!] \subseteq [\![\mathtt{A}_2]\!]$$

Corollary:

$$\mathtt{C}_1 <\!: \mathtt{C}_2 \quad \Longleftrightarrow \quad [\![\mathtt{C}_1]\!] \subseteq [\![\mathtt{C}_2]\!] \quad \Longleftrightarrow \quad \mathtt{C}_1 \sqsubset\!: \mathtt{C}_2$$

Featherweight XTATIC expressions

e	::=	
	new C($\overline{e}$)	object creation
	‹(e)›[$\overline{e}$]	tree
	[$\overline{e}$]	sequence
	e.f	field access
	e.m($\overline{e}$)	method call
	match(e){case [$\overline{P}$]: $\overline{e}$}	pattern match
	x	value variable

Standard properties (for FX)

Theorem [Preservation]: Suppose Σ conforms to Γ.

If $\Gamma \vdash \mathtt{e} \in \mathtt{A}$

and $\Sigma \vdash \mathtt{e} \Downarrow \mathtt{a}$

then $\mathtt{a} \in [\![\mathtt{A}]\!]$.

Theorem [Soundness]:

If $\bullet \vdash \mathtt{e} \in \mathtt{A}$

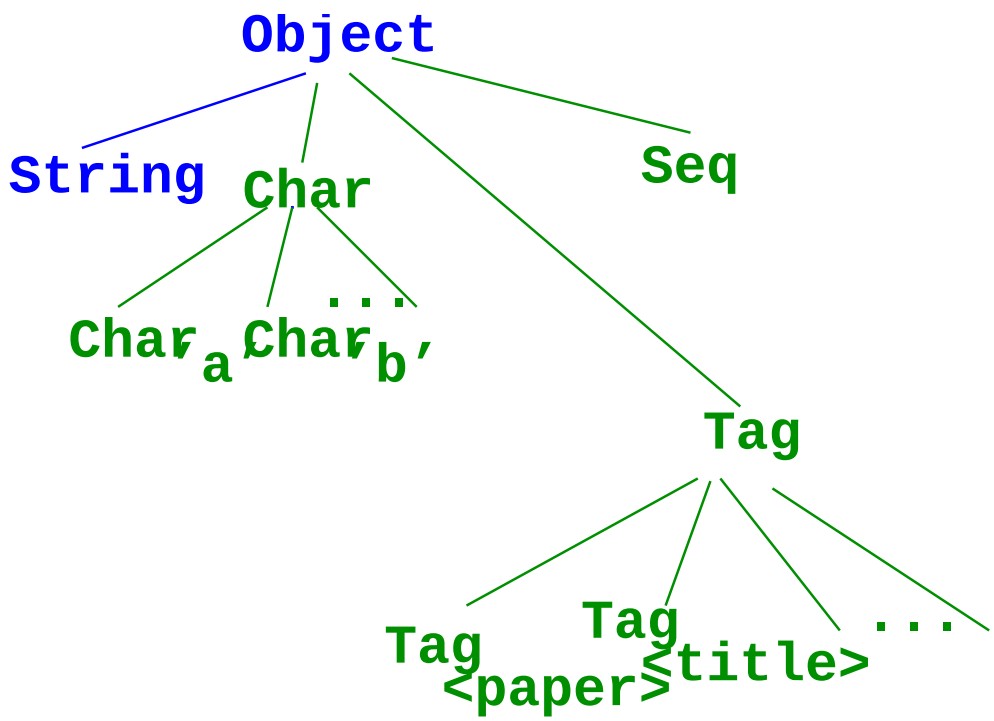
then evaluation of $\mathtt{e}$ does not go wrong.

XML representation: types

$\mathtt{a}$	$::=$	$\mathtt{t}$	$::=$
	$\mathtt{new}\ \mathtt{C}(\overline{\mathtt{a}})$		$\langle(\mathtt{a})\rangle[\overline{\mathtt{t}}]$
	$[\overline{\mathtt{t}}]$		
$\mathtt{A}$	$::=$	$\mathtt{T}$	$::=$
	$\mathtt{C}$		$\langle(\mathtt{A})\rangle[\mathtt{T}]$
	$[\mathtt{T}]$		$\mathtt{T}_1,\mathtt{T}_2$
			$\ldots$

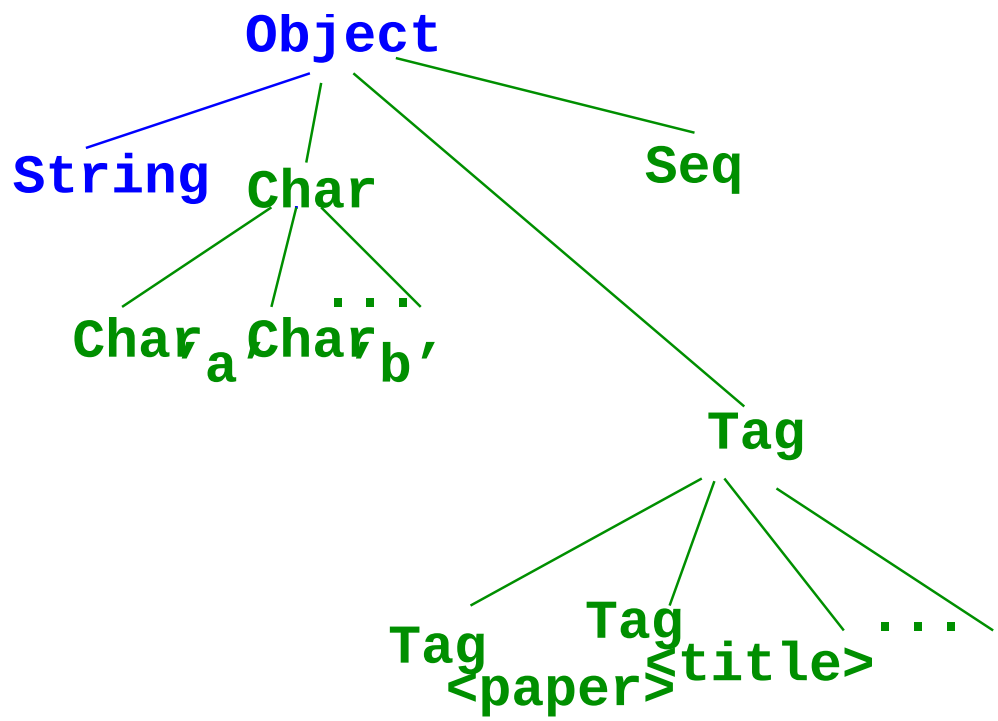
XML representation: types

$$
\begin{array}{lll}
a & ::= & \\
 & \texttt{new}\ C(\overline{a}) & \\
 & [\overline{t}] & \\
A & ::= & \\
 & C & \\
 & [T] &
\end{array}
\qquad
\begin{array}{lll}
t & ::= & \\
 & \langle(a)\rangle[\overline{t}] & \\
 & & \\
T & ::= & \\
 & \langle(A)\rangle[T] & \\
 & T_1, T_2 & \\
 & \ldots &
\end{array}
$$



XML representation: types

$$
\begin{array}{lll}
a & ::= & \\
 & \texttt{new}\ C(\overline{a}) & \\
 & [\overline{t}] & \\
A & ::= & \\
 & C & \\
 & [T] &
\end{array}
\qquad
\begin{array}{lll}
t & ::= & \\
 & \langle(a)\rangle[\overline{t}] & \\
 & & \\
T & ::= & \\
 & \langle(A)\rangle[T] & \\
 & T_1, T_2 & \\
 & \ldots &
\end{array}
$$




```
pcdata = [ pcchar* ]
pcchar = [ <(Char)>[] ]
xml = [ [pcchar | element]* ]
element = [ <(Tag)>[xml] ]
```

XML representation: values

XML:

```
<paper>
   <title>Mixins</title>
   <author>Flatt</author>
</paper>
```

XML representation: values

XML:

```
<paper>
   <title>Mixins</title>
   <author>Flatt</author>
</paper>
```

Abstract syntax:

```
[ ‹(new Tag<paper>)› [
    ‹(new Tag<title>)› [
      ‹(new Char'M')›[], ‹(new Char'i')›[], ...]
    ‹(new Tag<autor>)› [
      ‹(new Char'F')›[], ‹(new Char'l')›[], ...]
  ] ]
```

XML representation: values

XML:

```
<paper>
   <title>Mixins</title>
   <author>Flatt</author>
</paper>
```

Abstract syntax:

```
[ ‹(new Tag<paper>)› [
    ‹(new Tag<title>)› [
      ‹(new Char'M')›[], ‹(new Char'i')›[], ...]
    ‹(new Tag<autor>)› [
      ‹(new Char'F')›[], ‹(new Char'l')›[], ...]
  ] ]
```

Concrete syntax:

```
[ <paper>[
    <title>[‘Mixins‘]
    <author>[‘Flatt‘] ] ]
```

An example in XTATIC

```
[Title*] getEcoopTitles ([Paper*] ps) {
  [Title*] result = [];
  while (!ps.Equals([])) {
    match (ps) {
      case [ <paper>[Title t, Author*,
                     <inproc>[‘ECOOP‘, pcdata] ],
             Paper* rest  ]:
        result = [result, t];
        ps = rest;
      case [ Paper, Paper* rest ]:
        ps = rest;
      case []:  ;
    }
  return result;
  }
```

Summary

XTATIC extends a class-based OO type system with RE types:

- a pure OO program is a valid XTATIC program
- XDUCE-style pattern matching
- objects and trees can “interweave”
- XTATIC subtyping combines both OO and RE subtyping

Related work

XML data binding:

- JAXB, Castor, Relaxer
- HaXML, XMλ

OO type system extensions:

- Pizza, GJ
- ODMG

Semistructural data:

- OEM, Ozone

Static guarantees for XML processing:

- XDUCE, CDUCE
- XQUERY
- JWIG