

# Learning XML : VPAs and Discrimination Trees

Cinzia Di Giusto, Davide Fissore, Etienne Lozes

Université Nice Côte d'Azur, CNRS, I3S, Sophia Antipolis, France

## Why VPA ?

For  $\forall$  Non-Deterministic VPA  $V_1$ ,  
there  $\exists$  a Deterministic VPA  $V_2$   
such that  $L(V_1) = L(V_2)$   
 $\rightarrow$  Every binary operation  
between 2 VPA is decidable !

Note :

Push symbols  $\Leftrightarrow$  Open tags  
Pop symbols  $\Leftrightarrow$  Close tags

## VPAs

VPA := Visibly pushdown automata.  
They can recognize context free  
languages.  
The alphabet is :

$$\hat{\Sigma} = \Sigma_{call} \uplus \Sigma_{ret} \uplus \Sigma_{int}$$

Acceptance for XML :  
Empty stack + final states

## XML

XML (eXtensible Markup  
Language) is a standard  
format for data exchange.  
XML representable w/VPA!

## And Communication ?



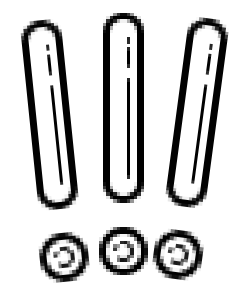
Arthur : Does  $w \in U$  ?  
Merlin : Yes/No

Arthur creates a conjecture  $C$ .

Arthur : Does  $C = U$  ?

Merlin : if  $C = U \rightarrow$  Yes

else  $\rightarrow$  a counter-example



## What is Learning ?



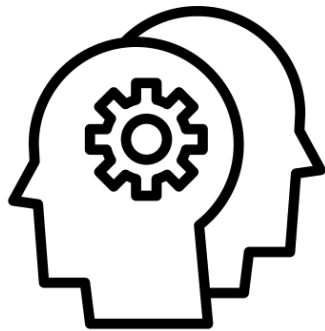
Dana Angluin's framework :

The Learner wants to learn a  
language  $U$   
The Teacher knows  $U$



## «Canonical» VPA

Regular automata have a  
unique minimal representant,  
this is not true for VPA



## k-SEVPA

Single entry VPA are VPAs  
where states are partitioned  
into  $k$  modules.  
Each module has only one  
entry for call transitions

## An XML grammar to LEARN

$G :=$

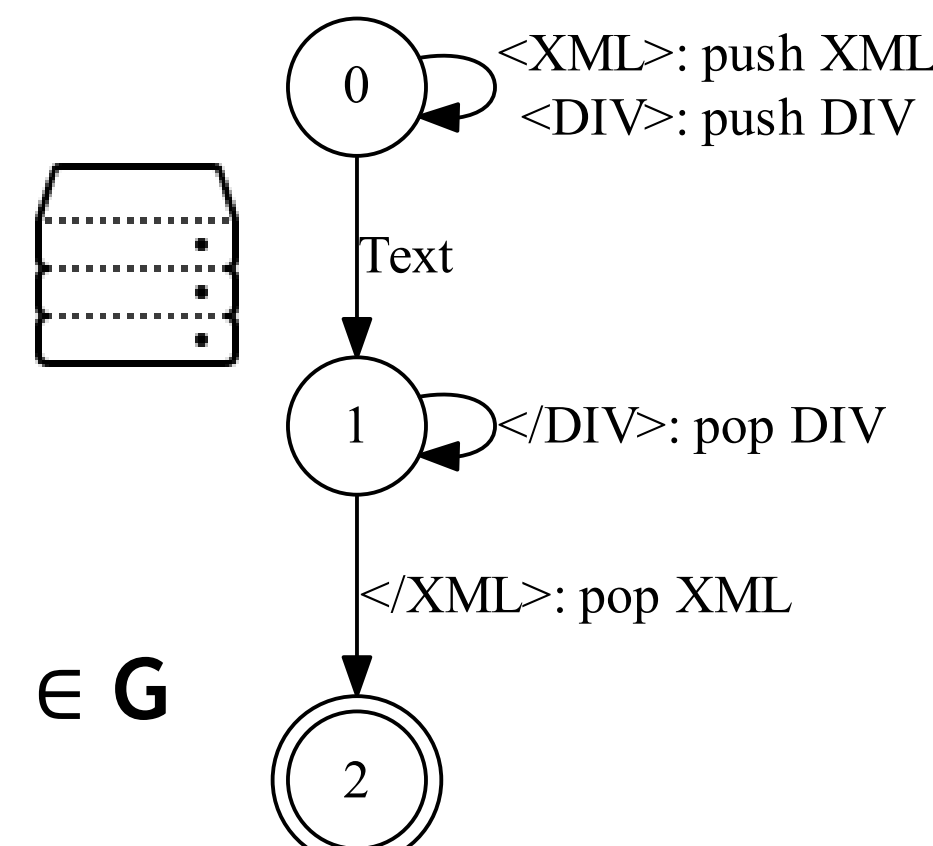
$d(\text{XML}) = \text{Text} + \text{DIV}$

$d(\text{DIV}) = \text{Text} + \text{DIV}$

$d : X \rightarrow \langle X \rangle \text{ RULE } \langle /X \rangle$

Example:

$\langle \text{XML} \rangle \langle \text{DIV} \rangle \text{Text} \langle / \text{DIV} \rangle \langle / \text{XML} \rangle \in G$



## The learning phase



In Visibly Pushdown Language  
(VPL), we can adapt the Myhill-  
Nerode congruence :  
two words  $(\omega_1, \omega_2) \in \hat{\Sigma}^2$  are  
equivalent if

$$\forall (u_1, u_2) \in \text{WM}(\hat{\Sigma})$$

$$u_1 \cdot \omega_1 \cdot u_2 \in L \Leftrightarrow u_1 \cdot \omega_2 \cdot u_2 \in L$$

$\text{WM}(\hat{\Sigma})$

It is a couple of words  
 $u_1, u_2$  such that in  
 $u = u_1 \circ u_2$  each call symbol  
has a corresponding  
ret symbol

## Discrimination Tree

From WM, we can build a particular  
binary tree called Discrimination  
tree.

Inner Nodes contain a couple  $(u_1,$   
 $u_2)$  and leaves are labelled with a  
string.

## VPA from Disc. Tree ?

From this discriminator tree, we can  
build, through membership queries  
the same VPA for the grammar  $G$ .

Where:

state 0 :=  $\epsilon$

state 1 := Text

state 2 :=  $\langle \text{XML} \rangle \text{Text} \langle / \text{XML} \rangle$

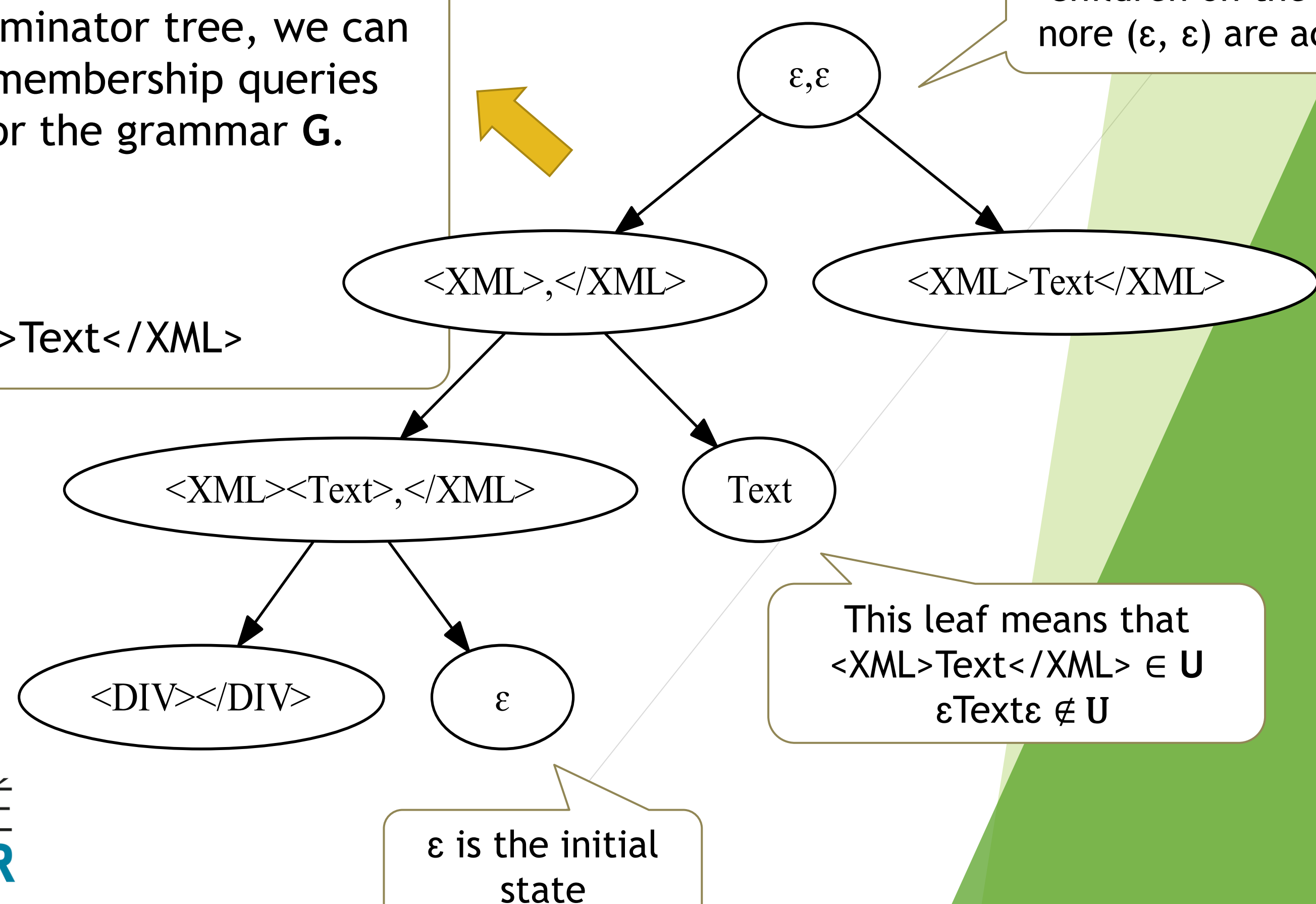
## Leaves meaning

Leaves represent the VPA  
states and through  
Membership questions, we  
build the corresponding VPA

## LCA

The LCA  $L$  (Lowest Common  
Ancestor) of two leaves  $l_1, l_2$  is the  
unique inner node such that  $l_1$  is on  
the right of  $L \Leftrightarrow l_2$  is on the left of  $L$

Children on the right of  
nore  $(\epsilon, \epsilon)$  are accepting



## References

## Demo

