

$$\int_0^1 \int_0^{1-x} \int_0^{10(x+3y)} x^2 dz dy =$$

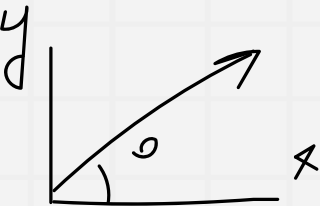
$$V: z=10(x+3y), x+y+z=10, x=0, y=0, z=0$$

$$\frac{3, x=5}{2+4y^2} \frac{2+4y^2}{2+4y^2}$$

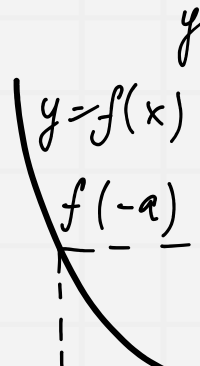
03

Generator Usage

How do we use the generator?



$$= \int_0^1$$



Usage 1: Parse and Unparse PGML codes

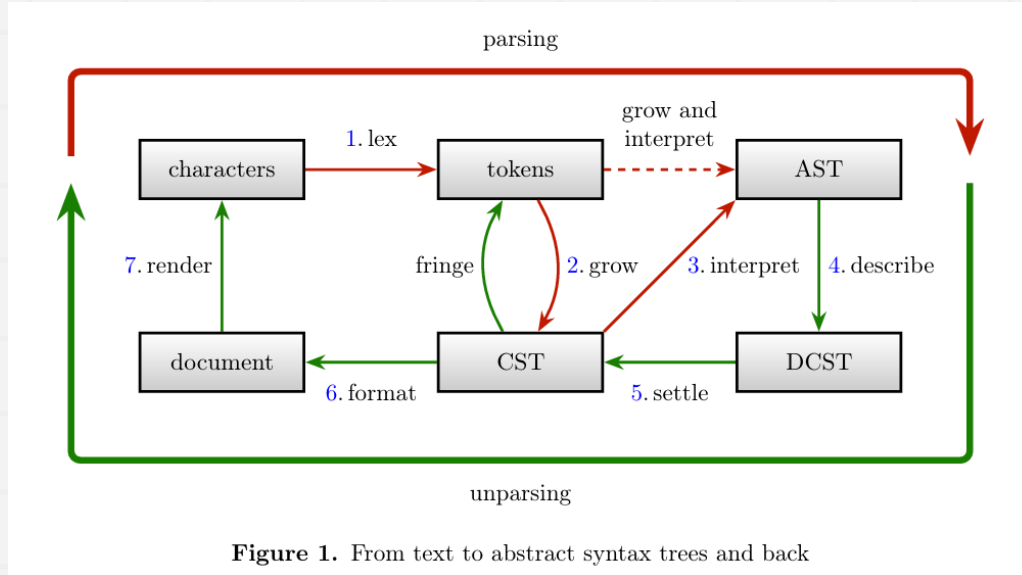


Figure 1. From text to abstract syntax trees and back

Enter this command in terminal:

```
printf <PGML_Code> | dune exec _build/default/unparser.exe
```



Usage 1: Parse and Unparse PGML codes

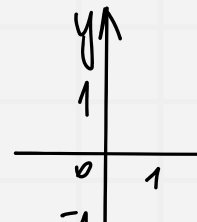
Example:

```
printf 'DOCUMENT();  
loadMacros("PGstandard.pl", "MathObjects.pl", "PGML.pl");  
Context("Numeric");  
BEGIN_PGML  
What is the value of [5]? [____]{5}  
END_PGML  
ENDDOCUMENT();' | dune exec _build/default/unparser.exe  
Done: 100% (58/58, 0 left) (jobs: 1)
```

Semantic: OK

AST:

```
Doc_Minimal(LoadMacros(None), Context_Only("Numeric"),  
Problem([Q_Item(Q_Value(IC_Math(ME_Term(T_Fac(F_Int(5))))),  
A_Var(Var_Value(VInt(5))),  
None]))  
..... (* More on next page *)
```





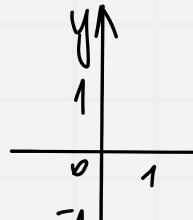
Usage 1: Parse and Unparse PGML codes

Example(continue):

```
→ describe_document  
..... (* Describe steps *)  
→ describe_header
```

After unparsing:

```
DOCUMENT();  
loadMacros(  
  "PGstandard.pl",  
  "MathObjects.pl",  
  "PGML.pl"  
);  
Context("Numeric");  
BEGIN_PGML  
What is the value of [5]?[____]{5}  
  
END_PGML  
ENDDOCUMENT();
```





Usage 2: Generate PGML codes according to Student ID

Enter this command in terminal:

```
dune exec _build/default/unparser.exe 2230026100
```

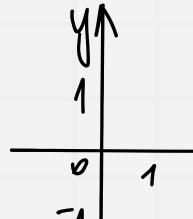
* **2230026100** is the student id.

Example:

```
dune exec _build/default/unparser.exe 2230026100
Done: 100% (58/58, 0 left) (jobs: 1)
Generated AST:

Doc_Minimal(LoadMacros(None), Context_With_Config("Numeric",
Cfg_Last(Operators, CM_Undefine([Add_Op; Sub_Op; Mul_Op; Div_Op]))),
Problem([Q_Item(Q_IsEqual(IC_Math(ME_Add(ME_Term(T_Fac(F_Int(0))),
T_Mul(T_Fac(F_Int(19)), F_Int(19)))))),
A_Var(Var_Method(M_Real(361.000000))),
None]))

..... (* More on next page *)
```





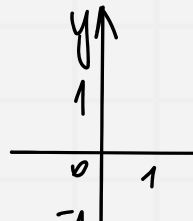
Usage 2: Generate PGML codes according to Student ID

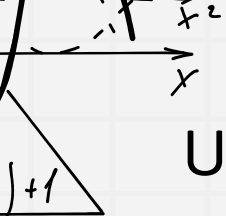
Example(continue):

```
→ describe_document  
..... (* Describe steps *)
```

Generated PGML code:

```
DOCUMENT();  
loadMacros(  
  "PGstandard.pl",  
  "MathObjects.pl",  
  "PGML.pl"  
);  
Context("Numeric");  
Context()->operators->undefine('+','-','*','/');  
BEGIN_PGML  
[ `0+19*19` ] is equal to [____]{Real(361)}  
  
END_PGML  
ENDDOCUMENT();
```





Usage 2: Generate PGML codes according to Student ID

What if I want the generated questions to be harder?

➤ Just change these **TWO parameters** in **generate.ml**:

1. **max_expr_depth** determines the **max depth** of the generated **math expressions**;
2. **question_num** determines the **max quantity** of the generated **questions** between BEGIN_PGML and END_PGML.

Code snippet in **generate.ml**:

```
(* ..... *)
(* ===== These could be difficulty levels: ===== *)
(* ..... *)
(* The absolute max depth of an expression: *)
let max_expr_depth = 2

(* The maximum amount of questions: *)
let question_num = 5
(* ===== *)
(* ..... *)
```

