Atitit.编程语言and 自然语言的比较and 编程语言未来的发展

[1. 单词的间隔靠空格,编程的单词的间隔靠分界符.. 1](#_Toc24360)

[2. 语句分界符:自然语言使用逗号,编程语言使用分号 1](#_Toc2743)

[3. 换行 1](#_Toc2487)

[4. 段落and fun method 2](#_Toc25690)

[5. 上下文相关 2](#_Toc30588)

[6. 操作泛型化 2](#_Toc5925)

[7. 动词和名词之间的互换性 2](#_Toc9837)

[8. 词类的判断 3](#_Toc385)

[8.1.1. 词类划分的形式标志 3](#_Toc6766)

[8.1.2. 自然语言中的指代词（你、我、他、它等等）好比编程语言中的“指针(point)”。 4](#_Toc16984)

[8.2. 这类疑问句似乎在编程语言 是包括函数在内的“表达式”。 4](#_Toc24313)

[8.3. 语句分析的“终极”在哪里？ 4](#_Toc18088)

[9. 词法检查 4](#_Toc9938)

[10. 句法检查 5](#_Toc4322)

[11. 一个类型系统。 自然语言中常用’token'，编程语言常用‘instance’。 5](#_Toc634)

[12. ref 6](#_Toc28679)

# 单词的间隔靠空格,编程的单词的间隔靠分界符..

括号 逗号 stop 句号

# 语句分界符:自然语言使用逗号,编程语言使用分号

# 换行

自然语言很少的中间做..编程语言很多中个..

作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

# 段落and fun method

编程靠fun method来定义段落..

# 上下文相关

编程语言跟个自然语言都是上下文相关的..业务角度...

技术上的ast绝大部分还是上下文无相关的.

# 操作泛型化

自然语言中存在一些“ 泛义动词 ” ，例如汉语中的“打” ， “搞” ， “干”„„， 英语中的“ get ” ， “ take ”等。它们的进一步的具体含义要由这些动词所涉及 的对象来确定。例如： “打毛衣”中的“打”应理解为“编织” ，   “打篮球” 中的“打”应理解为“玩” ，   “打开水”中的“打”应理解为“取得” ，等 等。



在面向对

在面向对象的编程语言中，也有类似的“动态绑定（Dynamic Binding）”机 制：一个函数名或者子程序名字的具体含义要在“运行时”依据所涉及对象在当时的指的类型来决定

数据：常数、变量名。它们对应于自然语言中的名词。

 其他：起连接作用、用来表述token 之间的关系的符号，例如if, then等等。 它们对应于自然语言中的连接词、感叹词等等

# 动词和名词之间的互换性

在自然语言中，有一个引人瞩目的现象，那就是“名词和动词之间的呼唤性”。这是因为，“词”或者token是用来表达“概念”的，而一个概念往往有多种侧面。所以自然语言中常常出现用同一个“符号”来表达不同的侧面的现象。举例来说： “钉”本来是名词，但是可以转化，作为动词来使用：我把画钉在墙上

函数十编程走十中个.不个fun trans as data..

# 词类的判断

从语法分析的角度看，对词或 token 进行分类的依据是它的语法功能。从语言使用者的角度看，则是他们的使用方式。因此如何对词或 token所属类别进行判断乃是一件至关重要的事情。

不管在自然语言中，还是在编程语言中，判断一个词或者 token 的类别的办法基本上是两种：词典 和 词本身所携带的形态标志。

词典中提供的信息

在编程语言中，为判断一个 token 的类别而提供的“词典”有两种：

1. 一种是语言中“先验地”规定好了的外部词典。例如关键字和一些保留字。

2. 另外一种是编程人员（用户）临时定义的内部词典，这就是程序中的 declaration.

### 词类划分的形式标志

如果能够从词或 token的外形（形式上的）特征就能判断出他是属于哪一类，那么无论从“书写者撰写”的角度，还是从“阅读者理解”的角度，都能够大大减少出错的机会。

在某些编程语言（例如Visual Basic）中，如果一个变量没有在任何地方加以声明,也可以从变量名字的外形上看出它的类型。例如：名字以%结尾的变量是?整数?， 名字以&结尾的变量是?字符串?，名字以&结尾的变量是?浮点数?等等。

英语中，在某种程度上也有类似的机制，例如：以tion, ing 结尾的基本上是名词。以-lize 结尾的基本上是名词。以-ful 结尾的基本上是形容词。以-ly 结尾的大概是副词。

汉语使用方块字，没有办法添加尾缀，所以没有这样的形态标志。因此大家认为，这一现象给汉语的计算机处理增加了困难。不过话不能说得太绝对。在某些情况下，汉语还是有“形态标志”的。例如，在名词的前面加“很”“还”之类一般用来修饰形容词的副词，就是在“形态”上指出：后面的这个名词已经转化为“形容词”了。例子如：“同学们说我穿这条裙子很青春。”，“我们排演的这套节目还是很生活的。”，

### 自然语言中的指代词（你、我、他、它等等）好比编程语言中的“指针(point)”。

但是自然语言中从不明显地交待：从现在起，“他”表示“张三”，直到遇见新的声明为止。每一个具体的代词指向何方要根据句子域句子之间的前后语义来联系来判断

## 这类疑问句似乎在编程语言 是包括函数在内的“表达式”。

## 语句分析的“终极”在哪里？

计算机语言的编译器在分析计算机程序中的语句时，最终总是要分析到语法规则中所规定的“终极符（terminator）”为止。

# 词法检查

从词法的角度，语素和语素合成单词是有一定规则的。例如中文的‘碧’字，作为一个语素，和其它语素结合成单词时有一个限制条件，必须是单音语素。如：碧空，碧蓝，碧绿；但是，碧天空，碧宝蓝等就不合法了。程序设计语言的单词（称作identifier，标识词）也有着类似的严格规定。例如，任何identifier必须以字母或下划线开头，像 bilan、 \_blian等就是合法的identifier，而23bilan、@bilan就属于非法的。任何一个identifier中间不能有空格，否则非法

# 句法检查

从句法的角度，有类似自然语言的条从句：if x < y then print x +y;  
意思是，如果两个identifier所代表的数不相等且x小于y，那么，将x所代表的数和y所代表的数相加，然后将结果打印出来。  
  
从语义的角度，任何一个identifier都可以定义为不同的类型。例如 name可以代表‘张三’，age可以代表25  
String name = ‘张三’；

程序设计语言中关于语法和语义的定义是非常严格、明确的，不能模棱两可。例如，单复数的定义，  
在汉语中，根本就是残缺的，只有在指人类实体是可以‘可选地’用‘们’，而且用法非常复杂。英语简单一些，用加s的方式表达。但有些单词单复数一样，如equipment，有些单词只有复数没有单数，如alms。无法只从形态上判断是单复数。  
而程序设计语言就没有这样的问题：  
任何定义了品词的identifier都是单数名词。而复数名词则需要一个特定的词类，通常叫做数组，或者叫做’表‘。例如

# 一个类型系统。 自然语言中常用’token'，编程语言常用‘instance’。

这些只是编程语言和自然语言的表面的异同。更本质的共通点，我觉得是二者之间都有一个类型系统。  
谈到类型，就要引入另一个术语，自然语言中常用’token'，编程语言常用‘instance’。我们经常重复已经说过的话

当然，我们有时需要区分类型和实例。自然语言的办法是加限定词。你的朋友刚想拿一块糖，你说，不是叫你吃这块糖，而是那块（当然在这种情景下必须有身体语言辅助：你得指着某一特定的糖块）。‘这’和‘那’还是很模糊，一定要明确，恐怕就要加更多的限定、形容

# ref

自然语言与机器语言的比较 南京廖华.htm

自然语言和程序设计语言的某些共通点.htm