

笔记模板2

1. 文章解决的问题

利用Genetic improvement来选择以及应用插入、删除、替换等操作

实验使用的是PyGGI这个GI遗传改进工具。

2. 解决思路

1. 工具大致思路：如果一个ast上有n条语句，则有n种删除的可能， n^2 的插入和替换的可能，以及5m个比较运算符的编辑。选择这三种的操作（变异算子）概率有偏差，见截图。

2. 实验结果：

2.0采用统一策略：

Program	Patches		Iterations			
	all	unique	avg	min	max	stdev
DFS	19	5	83	10	222	51
KHEAP-SORT	0	0	-	-	-	-
KNAP-SACK	9	2	159	1	397	157.3
LIS	17	6	171	10	451	154.3
MERGE-SORT	3	3	393	330	474	73.7
PASCAL	1	1	430	430	430	-
QUICK-SORT	13	2	251	17	499	173
SIEVE	6	6	312	161	467	113.1
Avg	8.50	3.13	257.0	-	-	-
Stdev	7.25	2.3	128.3	-	-	-

27个bug程序中8个程序找到了68个补丁。其中25个补丁在语法上是唯一的。在后面用这个25补丁做研究

1.0采用的初始策略：有85个补丁， 33个语法唯一的补丁

Program	Patches		Iterations			
	all	unique	avg	min	max	stdev
DFS	14	6	192	22	393	126.8
KHEAP-SORT	1	1	63	63	63	-
KNAP-SACK	17	5	102	4	495	128.6
LIS	13	9	179	18	442	128.3
MERGE-SORT	1	1	95	95	95	-
PASCAL	15	4	153	2	490	148.4
QUICK-SORT	20	4	25	1	64	18.7
SIEVE	3	3	256	65	389	169.8
Avg	10.50	4.13	133.13	-	-	-
Stdev	7.63	2.64	75.68	-	-	-

下表为多少个独特的补丁（只有这个策略出现的补丁）：

Program	Standard strategy	Uniform strategy	# other APR tools [8]
DFS	✓ (6)	✓ (5)	6
KHEAPSORT	✓ (1)	(0)	0
KNAPSACK	✓ (5)	✓ (2)	1
LIS	✓ (9)	✓ (6)	5
MERGESORT	✓ (1)	✓ (3)	1
PASCAL	✓ (4)	✓ (1)	0
QUICKSORT	✓ (4)	✓ (2)	6
SIEVE	✓ (3)	✓ (6)	0

下表为各种编辑操作在补丁中的占比：patch下的百分比指的是操作出现在补丁中的占比，edits指的是操作在所有操作的占比。

Operator	Initial strategy		Uniform strategy	
	patches	edits	patches	edits
All	33	73	25	49
Deletion	18.2%	12.3%	0%	0%
Insertion	45.4%	21.9%	40%	24.5%
Replacement	45.4%	24.7%	68%	57.1%
Comparison op mod	63.6%	41.1%	32%	18.4%

反正就是偏向插入与替换，而不是删除

下表为各操作有效性的表。<表示为生成的补丁或变体比原来的适应度更好。>表示比原来的差，=表示与原来的一致。

TABLE V
MUTATION OPERATORS EFFICACY

Operator	Patches	Success	Fitness			
			<%	>%	=%	==%
Deletion	7189	47.8%	3.9	20.2	23.7	9.5
Insertion	81110	28.7%	0.7	5.0	23.0	10.5
Replacement	74303	22.8%	1.5	8.4	12.9	6.3
Comp op mod	5329	89.2%	13.6	36.2	39.4	17.9

3. 核心知识点或名词定义

- 插入操作：只能在语句前插入
- 如何表示ast：用SrcML来表示基于XML的ast，然后PYGGI用这样子的ast表示形式来修改程序
- **tion of the program's AST.** The types of tags we consider are break, continue, decl_stmt, do, expr_stmt, for, goto, if, return, and switch, for statements; and operator_comp for operators.
- 插入、删除、替换（语句级别）以及6个比较操作符（变异）的修改操作

- 什么是统一策略：每一种编辑方法都有相等的概率被选中，包括操作符的空间
- PyGGI2.0采用局部爬山搜索算法（寻找当前最好的相邻状态）
- 文中说的两步选择策略就是先选择用哪个操作，然后操作的语句或比较符。
-

设一个程序有 n 条语句， m 个比较操作符。有 n 种删除操作， n^2 种插入操作， n^2 种替换操作。 $5m$ 种比较符修改操作

$$T = n + n^2 + n^2 + 5m$$

每种操作被选择的概率： n/T , n^2/T , n^2/T , $5m/T$

4.程序功能说明

5. 存在的问题

6. 改进的思路

7. 想法来源
