函数依赖挖掘作业

2015013200 软工51 黄 超

2015080121 软工52 李在弦（이재현）

# 实验环境

操作系统: OSX

处理器：2.9 GHz Intel Core i7

内存：16 GB 2133 MHz LPDDR3

IDE: CLion，XCode

编程语言: C++

# 算法实现

# 变量说明

为了节省在申请和释放容器空间时产生的时间消耗，我们将可能占用很大空间的容器写为全局变量。

为了便于进行集合的交并等运算，我们用整数的二进制表示法的数位来表示各个属性，比如一共有4种属性时，AB为0011，AC为0101，两者取并集即为ABC，表示为0111。

**N\_ATTR** : 保存被读入数据的属性个数值

**N\_DATA** : 读入的数据的条数

**data** : 保存从文件读入的数据

**L** : 保存当前层和上一层的属性集合值

**T** : strippedProduct函数里会用到的表

**S** : strippedProduct函数里会用到的表

**RHS** : 保存所有属性组合的右候选值

**partitions** : 保存所有属性集合的等价类划分

**computed** : 用来判断某个属性集合的等价类划分被计算的与否

**ans** : 最终的答案

# 重要函数说明

### scanData

scanData函数获取文件内的数据并把每个属性的数据保存在data中。获取数据的过程中，把属性个数保存在N\_ATTR变量中，数据总数保存在N\_DATA中。

### Init

初始化所有变量，预先计算出第0层和第1层的备选属性组合（分别为空集和所有单一属性）。

### computeDependencies

根据当前层和上一层的属性组合，计算函数依赖。

### judgeRelation

利用已经计算出的等价类划分来快速判断一个函数依赖是否成立。如果涉及的属性组合的等价类划分还没有计算就先用strippedProduct计算出来。

### prune

筛掉多余的属性组合以减少搜索树的宽度。

### generateNextLevel

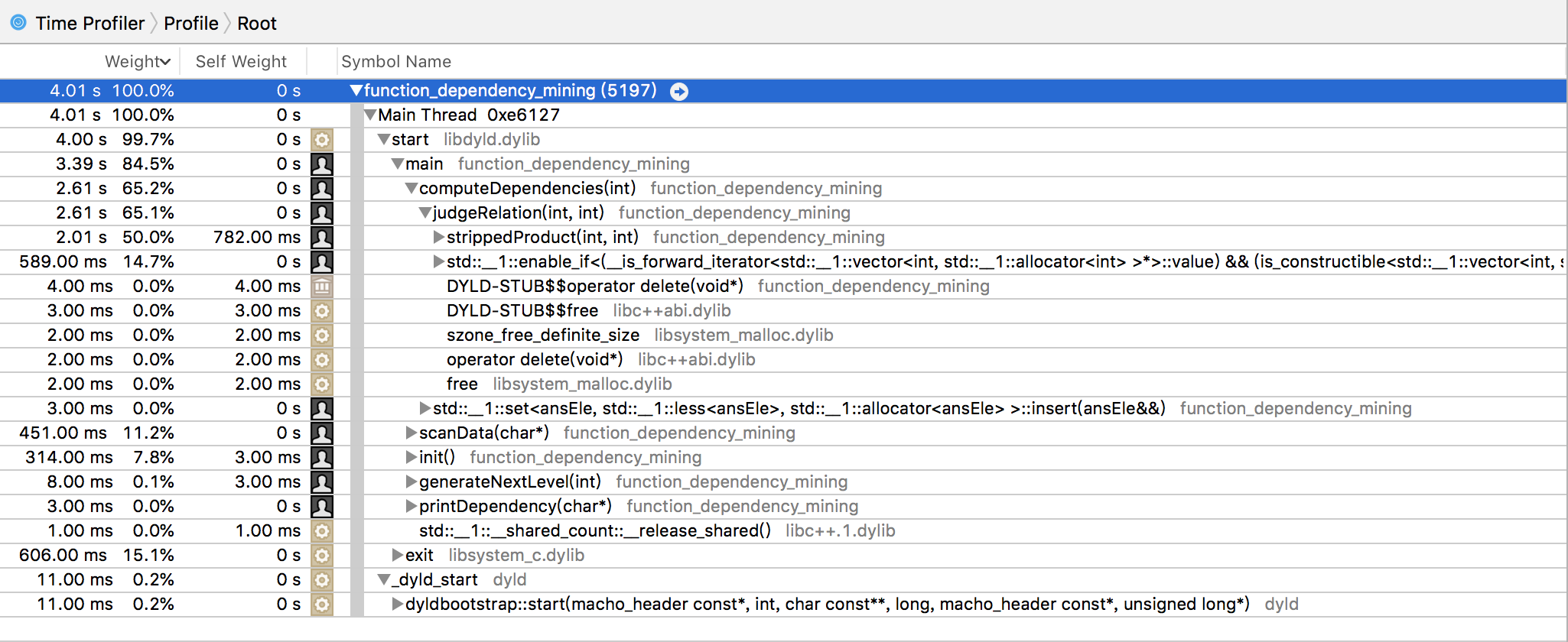
生成下一层待计算的属性组合。

### strippedProduct

用于计算等价类划分的函数。

# 实验结果

内存消耗400MB，需要4秒左右计算出data.txt中的所有函数依赖。



对程序运行进行profiling，可以看出，等价类划分的计算耗费掉了程序大部分的运行时间。

# 参考文献

TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies

< [Y Huhtala](http://xueshu.baidu.com/s?wd=author%3A%28Huhtala%2C%20Yk%C3%A4%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson)， [J Kärkkäinen](http://xueshu.baidu.com/s?wd=author%3A%28K%C3%A4rkk%C3%A4inen%2C%20Juha%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson)， [P Porkka](http://xueshu.baidu.com/s?wd=author%3A%28Porkka%2C%20Pasi%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson)， [H Toivonen](http://xueshu.baidu.com/s?wd=author%3A%28Toivonen%2C%20Hannu%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson)