

小型代码分析系统的实现方式

指针分析[Pointer analysis]是静态程序分析的基本组成部分之一，它的目的是找出在程序执行过程中通过特定指针变量访问哪些对象。现在我们希望您对测试数据执行上下文无关指针分析。一个程序包含26个用小写字母表示的对象，每个对象也有26个用小写字母表示的成员变量（又称字段，可能指向某些对象的指针）。同时，程序中有26个用大写字母指定的全局指针。程序中有四种语句。我们使用[Variable]表示指针的名称，[Field]表示成员变量的名称，[Object]表示对象。

A = x分配：指针A可以指向对象x，即可以通过A访问x。 A = B转让：指针A可以指向通过B访问的每个对象。 A.f = B贮存：对于通过A访问的每个对象o，o的成员变量f可以指向通过B访问的每个对象。 A = B.f装载：对于通过B访问的每个对象o，A可以指向通过o的成员变量f访问的每个对象。

上下文无关指针分析假设程序的语句将以任何顺序执行足够的次数。例如，在下面两个程序中，A和B都可以指向对象x和对象o，原因是在现实世界中，语句的确切执行顺序和执行时间很难预测。

现在，您需要对由N个语句组成的给定程序执行上下文无关指针分析，对于每个指针，输出它可以指向的对象。

输入的第一行包含一个整数N，1 ≤ N ≤ 200，表示程序中的语句数。等号“=”前后只有一个空格。以下N行中的每一行都包含一个语句。

输出应该包含26行。在第i行中，输出第i个指针的名称（第i个大写字母），后跟冒号“:”和空格，然后按字母顺序列出可通过该指针访问的对象。

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: <https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E5%B0%8F%E5%9E%8B%E4%BB%A3%E7%A0%81%E5%88%86%E6%9E%90%E7%B3%BB%E7%BB%9F%E7%9A%84%E5%AE%9E%E7%8E%B0%E6%96%B9%E5%BC%8F&rev=1596685938>

Last update: 2020/08/06 11:52