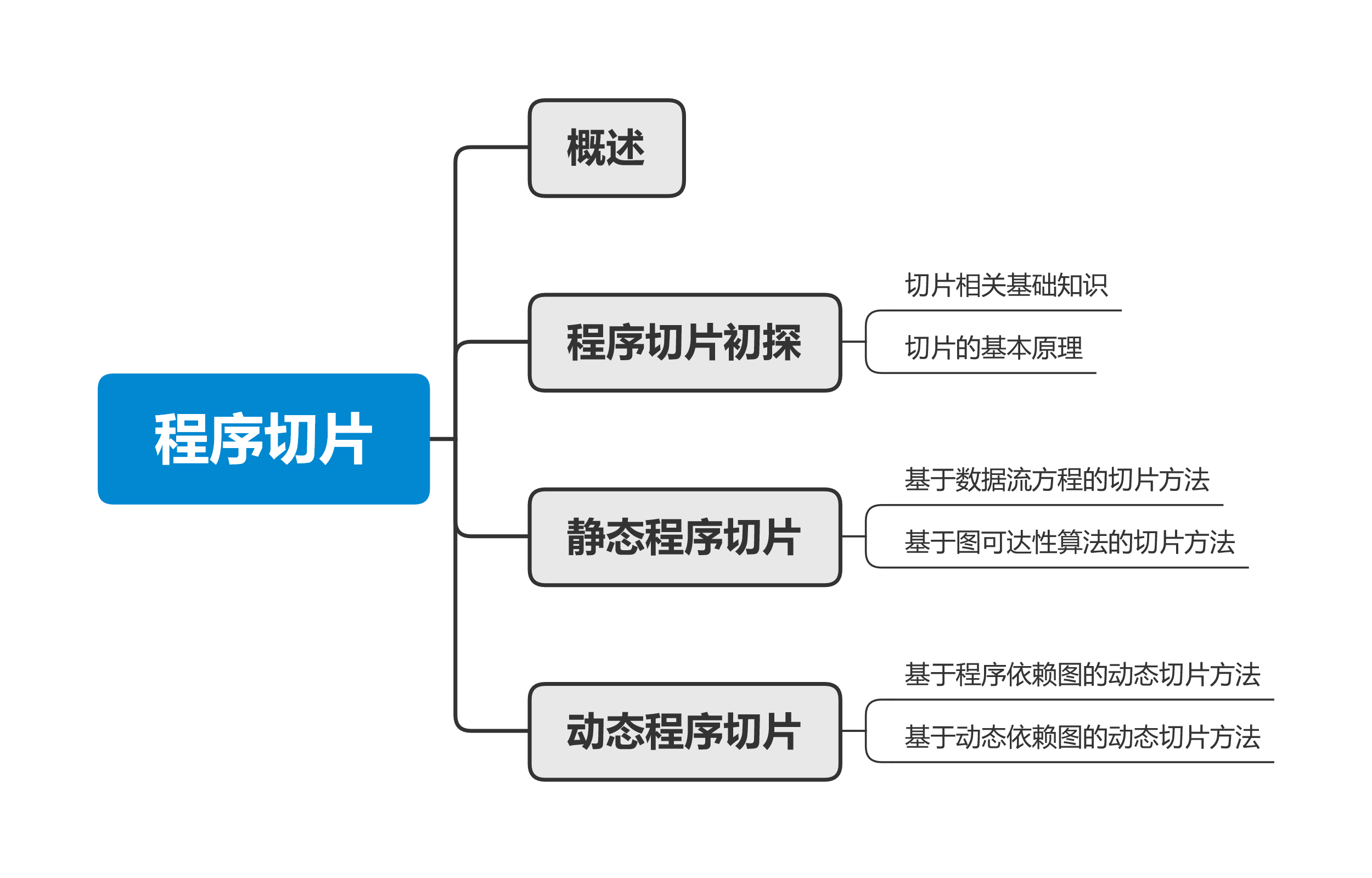
第四章 程序切片

# 一．章节主要内容



# 二．详细内容

## 4.1 概述

程序切片旨在从程序中提取满足一定约束条件的代码片段(对指定变量施加影响的代码、指令，或者指定变量所影响的代码片段)，主要在程序**调试、优化、排错、逆向、安全分析**等阶段采用，是在程序开发到**一定阶段**或者**已完成情况**下直接分析程序代码，通过寻找程序内部的相关性来分解程序，然后针对分解得到的程序切片进行分析来实现对程序部分功能特性的理解

程序切片经过了四个阶段：

* 基于数据流方程求解的程序切片计算
* 基于依赖图的程序切片计算
* 面向对象的程序切片计算
* 面向不同应用的各类切片计算

程序切片可应用于**软件逆向工程**、**软件调试**、**软件维护**以及**软件可靠性及安全性分析**

## 4.2 程序切片初探

### 4.2.1 切片相关基础知识

程序切片的过程是对程序代码进行分析，并依据分析人员的特定需求提取部分感兴趣的的代码的过程

基于程序代码，可以直接获取：①**控制流**信息 ②**数据流**信息

控制流：程序中一系列指令(语句、函数调用)执行的顺序

数据流：数据在程序中一系列执行指令间产生、传递、复制和消失的过程

* 控制流分析：
* 三种基本形成结构：顺序结构、条件结构、循环结构
* 基本块：

1. 程序执行时只能从该基本块的第一条指令进入
2. 程序执行时离开该基本块前的最后一条指令必须是该基本块的最后一条指令

* 基本块计算方法:



* 程序划分成基本块之后，将基本块视为基本单元节点，基本块之间在程序执行流程上互为前驱和后继是为两个基本块之间存在一条边，则程序形成一个有向图，即控制流图(Control Flow Graph, CFG)
* **CFG计算算法**：



* 在CFG中，从任意节点开始进行节点遍历会形成一条路径，路径上的基本块串联之后可以形成程序的一条执行路径，该路径成为CFG的一个可执行路径
* 数据流分析
  + 可到达定义
  + 目的：获取某程序语句所引用变量的来源，即引用变量是如何产生、复制和传递的
  + 路径定义：路径由程序中连续执行的代码语句位置的序列组成，位置序列长度为n，定义为P1，P2，…，Pn。 Pn是某条语句执行之前的代码位置，Pn+1是该语句执行之后的代码位置
  + 可到达定义：针对变量x的一个定义(赋值定义)语句，该语句的定义到达程序的某个代码位置P，并且该路径上没有变量x的其他定义，同时称语句s是代码位置P的一个可到达定义
* 基本概念：

|  |  |  |
| --- | --- | --- |
|  | 基本概念 | 定义 |
| 变量x | 定义集Def(x) | 定义x的所有语句的集合 |
| 引用集Use(x) | 任何使用x的语句的集合 |
| 语句s | 产生集Gen(s ) | 所有由s给出的变量定义所在的语句构成的集合 |
| 消灭集Kill(s) | 若重新定义变量x，而x此前由语句s’定义，则称s消灭定义s’ |
| 入集In(s) | 所有在语句s之前仍然有效的定义语句的集合 |
| 出集Out(s) | 所有离开语句s的定义语句的集合，添加s产生的语句，同时去掉语句s所消灭的定义语句 |