# 3设计问题二：基于高级语言源程序格式处理工具

## 3.1 问题概述

在计算机科学中，抽象语法树（abstract syntax tree或者缩写为AST），是将[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81" \t "_blank)的语法结构的用树的形式表示，树上的每个结点都表示源程序代码中的一种语法成分。之所以说是“抽象”，是因为在抽象语法树中，忽略了源程序中语法成分的一些细节，突出了其主要语法特征。

抽象语法树(Abstract Syntax Tree ,AST)作为程序的一种中间表示形式,在[程序分析](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E5%88%86%E6%9E%90" \t "_blank)等诸多领域有广泛的应用.利用抽象语法树可以方便地实现多种[源程序](https://baike.baidu.com/item/%E6%BA%90%E7%A8%8B%E5%BA%8F" \t "_blank)处理工具,比如源程序浏览器、智能编辑器、[语言翻译器](https://baike.baidu.com/item/%E8%AF%AD%E8%A8%80%E7%BF%BB%E8%AF%91%E5%99%A8/6119005" \t "_blank)等。

在《高级语言源程序格式处理工具》这个题目中，首先需要采用形式化的方式，使用巴克斯（BNF）范式定义高级语言的词法规则（字符组成单词的规则）、语法规则（单词组成语句、程序等的规则）。再利用形式语言自动机的的原理，对源程序的文件进行词法分析，识别出所有单词；使用编译技术中的递归下降语法分析法，分析源程序的语法结构，并生成抽象语法树,最后可由抽象语法树生成格式化的源程序。

## 3.2 实现的基本原理和方法

由源程序到抽象语法树的过程，逻辑上包含2个重要的阶段，一是词法分析，识别出所有按词法规则定义的单词；二是语法分析，根据定义的语法规则，分析单词序列是否满足语法规则，同时生成抽象语法树。

词法分析的过程，就是在读取源程序的文本文件的过程中，识别出一个个的单词。在词法分析前，需要先给每一类单词定义一个类别码（用枚举常量形式），识别出一个单词后，即可得到该单词的类别码和单词自身值（对应的符号串）。实现词法分析器的相关技术是采用有穷自动机的原理，例如：用EBNF表示的标识符：

<标识符>**：：=**字母{字母|数字}

<整数>**：：=**数字 {数字}

对应的确定有穷自动机DFA如图3.1所示。其中带圆圈的数字表示状态，双箭头指向的状态0为开始状态，环形的数字表示结束状态。在DFA中，从开始状态开始进行单词的识别，一个状态识别到一个符号后转移到下一个状态，一旦到达结束状态，成功的识别出一个单词。

图3.1识别标识符和整数的DFA

根据图3.1所示的DFA，很容易得到识别标识符和整数的算法流程，如图3.2所示，识别出的单词类别码为kind，自身值为w。

图3.2识别标识符和整数的算法流程图

如何来识别特定语言的所有单词，详见《综合程序设计课程设计指导-源程序格式处理》。

语法分析的过程，建议采用的实现方法是编译技术中的递归下降子程序法，递归下降子程序法是一种非常简洁的语法结构分析算法，基本上是每个语法成分对应一个子程序，每次根据识别出的前几个单词，明确对应的语法成分，调用相应子程序进行语法结构分析。例如在分析语句的语法结构时，当识别出单词if 后，进行条件语句的处理，同时生成的子树根结点对应条件语句。处理时，首先调用表达式的子程序，得到表达式子树的根指针T1；再递归调用语句处理部分，得到if子句的子树根指针T2；再看随后的单词，如果不是else，就表示是一个if语句，条件语句子树的根结点标记为“IF语句”，有2棵子树，对应T1和T2；如果随后的单词是else，就再递归调用语句处理部分，得到else子句的子树根指针T3；最后分析出的是一个if-else语句，条件语句子树的根结点标记为“IF\_ELSE语句”，有3棵子树，对应T1、T2和T3。按此处理流程，对给定条件语句： if (a>b) m=a; else m=b；分析后生成的抽象语法树形式如图3.3所示。

图3.3if-else语句的抽象语法树

在《综合程序设计课程设计指导-源程序格式处理》中，首先使用巴克斯（BNF）范式定义定义了一个简单的语言，给出了各个语法成分的处理流程框架作为实验的参考。

## 3.3 功能要求

1. 语言定义

选定C语言的一个子集，要求包含：

（1）基本数据类型的变量、常量，以及数组。不包含指针、结构，枚举等。

（2) 双目算术运算符（+-\*/%），关系运算符、逻辑与（&&）、逻辑或（||）、赋值运算符。不包含逗号运算符、位运算符、各种单目运算符等等。

（3）函数定义、声明与调用。

（4）表达式语句、复合语句、if语句的2种形式、while语句、for语句，return语句、break语句、continue语句、外部变量说明语句、局部变量说明语句。

（5）编译预处理（宏定义，文件包含）

（6）注释（块注释与行注释）

2. 单词识别

设计DFA的状态转换图（参见实验指导），实验时给出DFA，并解释如何在状态迁移中完成单词识别（每个单词都有一个种类编号和单词的字符串这2个特征值），最终生成单词识别（词法分析）子程序。

**注：含后缀常量，以类型不同作为划分标准种类编码值，例如123类型为int，123L类型为long，单词识别时，种类编码应该不同；但0x123和123类型都是int，种类编码应该相同。**

3. 语法结构分析

（1）外部变量的声明；

（2）函数声明与定义；

（3）局部变量的声明；

（4）语句及表达式；

（5）生成（1)-(4)（包含编译预处理和注释）的抽象语法树并显示。

4. 按缩进编排生成源程序文件。

## 3.4实现与评测说明

实验部分按阶段进行检查评分，检查时，要求同学们自行准备好测试用例，可以不必考虑测试的源程序文件功能意义，以测试用例能覆盖全部任务要求为准。即:自行设计的测试用例务必反映系统的功能(含异常情况处理)，用例未能反映的功能，视同未实现情况评定。

（1）识别语言的全部单词。（50%）

要求测试用例包含所有种类的单词，测试用例中没有出现的单词种类视作没有完成该类单词的识别。由于每类单词有一个种类编码（参见实验指导书用枚举常量定义），可以将识别出来的单词按种类编码进行排序显示，这样既能方便自己的调试，也能方便检查。注意相同种类编码的多种形式，都应该包含在测试用例中，例如类型为int的常量，有三种形式0123、123、0x123。

报错功能，指出不符合单词定义的符号位置。测试文件中不必包含错误符号，检查时由老师随机修改测试文件，设置错误，检查报错功能是否实现。

（2）语法结构分析与生成抽象语法树。（40%）

要求测试用例包含函数声明，定义、表达式（各种运算符均在某个表达式中出现）、所有的语句，以及if语句的嵌套，循环语句的嵌套。测试用例中没有出现的语句和嵌套结构，视作没有完成该种语法结构的分析。

常见的语句包括：

1.表达式语句； 2.if语句； 3.if else语句； 4.while语句;

5.for语句； 6. return语句； 7.break语句； 8.continue语句；

9.复合语句； 10.函数定义； 11.函数声明； 12.函数调用；

13. if语句嵌套； 14.循环语句嵌套； 15.外部变量说明语句;

16. 局部变量说明语句。

报错功能，指出不符合语法规则的错误位置。测试文件中不必包含错误语句等，检查时由老师随机修改测试文件，设置错误，检查报错功能是否实现。

显示抽象语法树，要求能由抽象语法树说明源程序的语法结构，这也是检查时验证语法结构分析正确性的依据。

（3）缩进编排重新生成源程序文件（10%）。对（2）的测试用例生成的抽象语法树进行先根遍历，按缩进编排的方式写到.c文件中，查看文件验证是否满足任务要求。

实现语法结构分析时，不局限使用递归下降子程序法，但不能使用工具自动生成单词识别和语法结构分析的程序。

## 3.5参考文献

[1] 王生原，董渊，张素琴，吕映芝等. 编译原理（第3版）. 北京：清华大学出版社. 前4章

[2] 严蔚敏等.数据结构(C语言版).北京：清华大学出版社

# 4程序设计综合课程设计总体要求

## 4.1 坚守学术诚信

鼓励创新，进行有一定特色的设计。严禁对程序与报告的抄袭行为（包括对网络资源及其他同学的设计），一经发现，课程设计成绩计0分，以考试抄袭舞弊行为处理。

## 4.2 程序规范

程序遵从一般性规范：

1. 源码依据模块组织到不同.h与.c文件中，不要将全部程序放到一个源文件中。
2. 变量尽量基于描述性命名，看其名知其意。
3. 函数头有统一注释，说明功能，输入输出与条件等。

⑷函数内部关键处理步骤处加上注释予以说明。

## 4.3 报告规范，内容完善

按照计算机学院课程设计报告的要求及本课程设计报告的格式规范与内容要求撰写设计报告，避免出现错别字及形式的不规范现象。报告主要内容应至少涵盖如下方面(以下非报告目录)。

一、问题描述

二、需求与技术现状分析

三、程序总体设计(含模块结构图)

四、数据结构和算法详细设计

五、程序实现

(C语言程序实现的简要说明，如开发环境、支持包、函数原型与功能及调用关系；全部源程序以电子版提供，报告中只能作为附录内容之一)

六、程序测试及结果分析

七、复杂度分析

八、总结、特色与不足

主要参考文献

附录一：源程序

附录二：程序使用说明

## 4.4课堂与考勤要求

要求按时到实验室完成综合程序设计，根据完成与验收情况由指导老师批准方可在其它场所查阅资料，撰写报告。设计课坚持记录考勤。

## 4.5检查与验收

在设计课内，全体同学需给指导老师或助教演示程序，解释程序，回答老师提问，验收或报告完成情况。

## /\*\*

## \* 函数名称:

## \* 函数功能：

## \* 函数输入:

## \*函数输出：

## \*/

# 指导参考书目录

[1] 曹计昌，卢萍，李开. C语言与程序设计. 电子工业出版社，2013

[2]严蔚敏等.数据结构（C语言版）.清华大学出版社，

[3] [Larry Nyhoff](http://www.calvin.edu/~nyhl/index.html). [ADTs, Data Structures, and Problem Solving with C++.](http://vig.prenhall.com/catalog/academic/product/0,1144,0131409093,00.html)Second Edition,[Calvin College](http://cs.calvin.edu/),2005

[4] 殷立峰. Qt C++跨平台图形界面程序设计基础. 清华大学出版社,2014:192～197

[5] 严蔚敏等.数据结构题集（C语言版）.清华大学出版社

# 附录A程序设计综合课程设计评价指标

|  |  |  |
| --- | --- | --- |
| **评价指标** | **满分** | **评价标准** |
| 程序功能（40%） | 100 | 题目一成绩=（1）至（6）项功能得分之和，其中，处理满足算例记0.8，不满足算例记0.2。 |
| 题目二成绩=（2）至（4）项功能得分之和。其中（2）（3），处理正常用例记0.9，报错用例记0.1。 |
| 程序规范（10%） | 100 | 程序规范：80，注释：80+，模块化且注释好：90+，不规范：80-。 |
| 设计特色（10%） | 100 | 特色不明显：70，有一定特色：80+，特色突出或有创意：90+ |
| 报告内容（30%） | 100 | 问题描述与分析：20，程序总体设计、数据结构、算法设计和理论分析：60，测试计划及测试分析：20。 |
| 报告规范（10%） | 100 | 基本规范：80，规范：80+，不规范：80-。 |
| 逾期扣分 | 10 | 逾期提交：2/天。超过5天者本次实验记0。 |
| **综合成绩＝设计成绩×92%＋实验考勤×8%**  设计成绩＝(∑程序功能×40%＋设计特色×10%＋程序规范×10%＋报告内容×30%＋报告规范×10%－逾期扣分) | | |

**注：实验考勤原则上仅记录签到情况，不考虑任何请假情形。**

# 3设计问题二：基于高级语言源程序格式处理工具

## 3.1 问题概述

在计算机科学中，抽象语法树（abstract syntax tree或者缩写为AST），是将[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81" \t "_blank)的语法结构的用树的形式表示，树上的每个结点都表示源程序代码中的一种语法成分。之所以说是“抽象”，是因为在抽象语法树中，忽略了源程序中语法成分的一些细节，突出了其主要语法特征。

抽象语法树(Abstract Syntax Tree ,AST)作为程序的一种中间表示形式,在[程序分析](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E5%88%86%E6%9E%90" \t "_blank)等诸多领域有广泛的应用.利用抽象语法树可以方便地实现多种[源程序](https://baike.baidu.com/item/%E6%BA%90%E7%A8%8B%E5%BA%8F" \t "_blank)处理工具,比如源程序浏览器、智能编辑器、[语言翻译器](https://baike.baidu.com/item/%E8%AF%AD%E8%A8%80%E7%BF%BB%E8%AF%91%E5%99%A8/6119005" \t "_blank)等。

在《高级语言源程序格式处理工具》这个题目中，首先需要采用形式化的方式，使用巴克斯（BNF）范式定义高级语言的词法规则（字符组成单词的规则）、语法规则（单词组成语句、程序等的规则）。再利用形式语言自动机的的原理，对源程序的文件进行词法分析，识别出所有单词；使用编译技术中的递归下降语法分析法，分析源程序的语法结构，并生成抽象语法树,最后可由抽象语法树生成格式化的源程序。

## 3.2 实现的基本原理和方法

由源程序到抽象语法树的过程，逻辑上包含2个重要的阶段，一是词法分析，识别出所有按词法规则定义的单词；二是语法分析，根据定义的语法规则，分析单词序列是否满足语法规则，同时生成抽象语法树。

词法分析的过程，就是在读取源程序的文本文件的过程中，识别出一个个的单词。在词法分析前，需要先给每一类单词定义一个类别码（用枚举常量形式），识别出一个单词后，即可得到该单词的类别码和单词自身值（对应的符号串）。实现词法分析器的相关技术是采用有穷自动机的原理，例如：用EBNF表示的标识符：

<标识符>**：：=**字母{字母|数字}

<整数>**：：=**数字 {数字}

对应的确定有穷自动机DFA如图3.1所示。其中带圆圈的数字表示状态，双箭头指向的状态0为开始状态，环形的数字表示结束状态。在DFA中，从开始状态开始进行单词的识别，一个状态识别到一个符号后转移到下一个状态，一旦到达结束状态，成功的识别出一个单词。

图3.1识别标识符和整数的DFA

根据图3.1所示的DFA，很容易得到识别标识符和整数的算法流程，如图3.2所示，识别出的单词类别码为kind，自身值为w。

图3.2识别标识符和整数的算法流程图

如何来识别特定语言的所有单词，详见《综合程序设计课程设计指导-源程序格式处理》。

语法分析的过程，建议采用的实现方法是编译技术中的递归下降子程序法，递归下降子程序法是一种非常简洁的语法结构分析算法，基本上是每个语法成分对应一个子程序，每次根据识别出的前几个单词，明确对应的语法成分，调用相应子程序进行语法结构分析。例如在分析语句的语法结构时，当识别出单词if 后，进行条件语句的处理，同时生成的子树根结点对应条件语句。处理时，首先调用表达式的子程序，得到表达式子树的根指针T1；再递归调用语句处理部分，得到if子句的子树根指针T2；再看随后的单词，如果不是else，就表示是一个if语句，条件语句子树的根结点标记为“IF语句”，有2棵子树，对应T1和T2；如果随后的单词是else，就再递归调用语句处理部分，得到else子句的子树根指针T3；最后分析出的是一个if-else语句，条件语句子树的根结点标记为“IF\_ELSE语句”，有3棵子树，对应T1、T2和T3。按此处理流程，对给定条件语句： if (a>b) m=a; else m=b；分析后生成的抽象语法树形式如图3.3所示。

图3.3if-else语句的抽象语法树

在《综合程序设计课程设计指导-源程序格式处理》中，首先使用巴克斯（BNF）范式定义定义了一个简单的语言，给出了各个语法成分的处理流程框架作为实验的参考。

## 3.3 功能要求

1. 语言定义

选定C语言的一个子集，要求包含：

（1）基本数据类型的变量、常量，以及数组。不包含指针、结构，枚举等。

（2) 双目算术运算符（+-\*/%），关系运算符、逻辑与（&&）、逻辑或（||）、赋值运算符。不包含逗号运算符、位运算符、各种单目运算符等等。

（3）函数定义、声明与调用。

（4）表达式语句、复合语句、if语句的2种形式、while语句、for语句，return语句、break语句、continue语句、外部变量说明语句、局部变量说明语句。

（5）编译预处理（宏定义，文件包含）

（6）注释（块注释与行注释）

2. 单词识别

设计DFA的状态转换图（参见实验指导），实验时给出DFA，并解释如何在状态迁移中完成单词识别（每个单词都有一个种类编号和单词的字符串这2个特征值），最终生成单词识别（词法分析）子程序。

**注：含后缀常量，以类型不同作为划分标准种类编码值，例如123类型为int，123L类型为long，单词识别时，种类编码应该不同；但0x123和123类型都是int，种类编码应该相同。**

3. 语法结构分析

（1）外部变量的声明；

（2）函数声明与定义；

（3）局部变量的声明；

（4）语句及表达式；

（5）生成（1)-(4)（包含编译预处理和注释）的抽象语法树并显示。

4. 按缩进编排生成源程序文件。

## 3.4实现与评测说明

实验部分按阶段进行检查评分，检查时，要求同学们自行准备好测试用例，可以不必考虑测试的源程序文件功能意义，以测试用例能覆盖全部任务要求为准。即:自行设计的测试用例务必反映系统的功能(含异常情况处理)，用例未能反映的功能，视同未实现情况评定。

（1）识别语言的全部单词。（50%）

要求测试用例包含所有种类的单词，测试用例中没有出现的单词种类视作没有完成该类单词的识别。由于每类单词有一个种类编码（参见实验指导书用枚举常量定义），可以将识别出来的单词按种类编码进行排序显示，这样既能方便自己的调试，也能方便检查。注意相同种类编码的多种形式，都应该包含在测试用例中，例如类型为int的常量，有三种形式0123、123、0x123。

报错功能，指出不符合单词定义的符号位置。测试文件中不必包含错误符号，检查时由老师随机修改测试文件，设置错误，检查报错功能是否实现。

（2）语法结构分析与生成抽象语法树。（40%）

要求测试用例包含函数声明，定义、表达式（各种运算符均在某个表达式中出现）、所有的语句，以及if语句的嵌套，循环语句的嵌套。测试用例中没有出现的语句和嵌套结构，视作没有完成该种语法结构的分析。

常见的语句包括：

1.表达式语句； 2.if语句； 3.if else语句； 4.while语句;

5.for语句； 6. return语句； 7.break语句； 8.continue语句；

9.复合语句； 10.函数定义； 11.函数声明； 12.函数调用；

13. if语句嵌套； 14.循环语句嵌套； 15.外部变量说明语句;

16. 局部变量说明语句。

报错功能，指出不符合语法规则的错误位置。测试文件中不必包含错误语句等，检查时由老师随机修改测试文件，设置错误，检查报错功能是否实现。

显示抽象语法树，要求能由抽象语法树说明源程序的语法结构，这也是检查时验证语法结构分析正确性的依据。

（3）缩进编排重新生成源程序文件（10%）。对（2）的测试用例生成的抽象语法树进行先根遍历，按缩进编排的方式写到.c文件中，查看文件验证是否满足任务要求。

实现语法结构分析时，不局限使用递归下降子程序法，但不能使用工具自动生成单词识别和语法结构分析的程序。

## 3.5参考文献

[1] 王生原，董渊，张素琴，吕映芝等. 编译原理（第3版）. 北京：清华大学出版社. 前4章

[2] 严蔚敏等.数据结构(C语言版).北京：清华大学出版社

# 4程序设计综合课程设计总体要求

## 4.1 坚守学术诚信

鼓励创新，进行有一定特色的设计。严禁对程序与报告的抄袭行为（包括对网络资源及其他同学的设计），一经发现，课程设计成绩计0分，以考试抄袭舞弊行为处理。

## 4.2 程序规范

程序遵从一般性规范：

1. 源码依据模块组织到不同.h与.c文件中，不要将全部程序放到一个源文件中。
2. 变量尽量基于描述性命名，看其名知其意。
3. 函数头有统一注释，说明功能，输入输出与条件等。

⑷函数内部关键处理步骤处加上注释予以说明。

## 4.3 报告规范，内容完善

按照计算机学院课程设计报告的要求及本课程设计报告的格式规范与内容要求撰写设计报告，避免出现错别字及形式的不规范现象。报告主要内容应至少涵盖如下方面(以下非报告目录)。

一、问题描述

二、需求与技术现状分析

三、程序总体设计(含模块结构图)

四、数据结构和算法详细设计

五、程序实现

(C语言程序实现的简要说明，如开发环境、支持包、函数原型与功能及调用关系；全部源程序以电子版提供，报告中只能作为附录内容之一)

六、程序测试及结果分析

七、复杂度分析

八、总结、特色与不足

主要参考文献

附录一：源程序

附录二：程序使用说明

## 4.4课堂与考勤要求

要求按时到实验室完成综合程序设计，根据完成与验收情况由指导老师批准方可在其它场所查阅资料，撰写报告。设计课坚持记录考勤。

## 4.5检查与验收

在设计课内，全体同学需给指导老师或助教演示程序，解释程序，回答老师提问，验收或报告完成情况。

# 指导参考书目录

[1] 曹计昌，卢萍，李开. C语言与程序设计. 电子工业出版社，2013

[2]严蔚敏等.数据结构（C语言版）.清华大学出版社，

[3] [Larry Nyhoff](http://www.calvin.edu/~nyhl/index.html). [ADTs, Data Structures, and Problem Solving with C++.](http://vig.prenhall.com/catalog/academic/product/0,1144,0131409093,00.html)Second Edition,[Calvin College](http://cs.calvin.edu/),2005

[4] 殷立峰. Qt C++跨平台图形界面程序设计基础. 清华大学出版社,2014:192～197

[5] 严蔚敏等.数据结构题集（C语言版）.清华大学出版社

# 附录A程序设计综合课程设计评价指标

|  |  |  |
| --- | --- | --- |
| **评价指标** | **满分** | **评价标准** |
| 程序功能（40%） | 100 | 题目一成绩=（1）至（6）项功能得分之和，其中，处理满足算例记0.8，不满足算例记0.2。 |
| 题目二成绩=（2）至（4）项功能得分之和。其中（2）（3），处理正常用例记0.9，报错用例记0.1。 |
| 程序规范（10%） | 100 | 程序规范：80，注释：80+，模块化且注释好：90+，不规范：80-。 |
| 设计特色（10%） | 100 | 特色不明显：70，有一定特色：80+，特色突出或有创意：90+ |
| 报告内容（30%） | 100 | 问题描述与分析：20，程序总体设计、数据结构、算法设计和理论分析：60，测试计划及测试分析：20。 |
| 报告规范（10%） | 100 | 基本规范：80，规范：80+，不规范：80-。 |
| 逾期扣分 | 10 | 逾期提交：2/天。超过5天者本次实验记0。 |
| **综合成绩＝设计成绩×92%＋实验考勤×8%**  设计成绩＝(∑程序功能×40%＋设计特色×10%＋程序规范×10%＋报告内容×30%＋报告规范×10%－逾期扣分) | | |

**注：实验考勤原则上仅记录签到情况，不考虑任何请假情形。**