



课程 实 验 报 告

**题目： 基于 SAT 的数独游戏求解程序**

|  |  |
| --- | --- |
| **课程名称** | **程序设计综合课程设计** |
| **专业班级** | **CS2106** |
| **学 号** | **U202115514** |
| **姓 名** | **杨明欣** |
| **指导教师** | **纪俊文** |
| **助 教** | **许海涛** |
| **报告日期** | **2022 年 10 月 2 日** |

**计算机科学与技术学院**

华 中 科 技 大 学 课 程 实 验 报 告

# 目 录

[任务书](#_bookmark0) 1

1. [引言](#_bookmark1) 3
   1. [课题背景与意义](#_bookmark2) 3
      1. [SAT 问题](#_bookmark3) 3
      2. [数独游戏](#_bookmark4) 3
   2. [国内外研究现状](#_bookmark5) 3
   3. [课程设计的主要研究工作](#_bookmark6) 5
2. [系统需求分析与总体设计](#_bookmark7) 6
   1. [系统需求分析](#_bookmark8) 6
   2. [系统总体设计](#_bookmark9) 6
3. [系统详细设计](#_bookmark10) 8
   1. [有关数据结构的定义](#_bookmark11) 8
      1. [相关数据结构](#_bookmark12) 8
      2. [数据结构关联](#_bookmark13) 9
   2. [主要算法设计](#_bookmark14) 10
   3. [变量决策阶段](#_bookmark15) 11
      1. [变量选取策略](#_bookmark16) 11
      2. [冲突分析](#_bookmark17) 12
   4. [推理阶段](#_bookmark18) 12
      1. [数据结构优化](#_bookmark19) 12
      2. [传播方式优化](#_bookmark20) 13
   5. [回溯阶段](#_bookmark21) 14
      1. [智能回溯](#_bookmark22) 14
      2. [周期性重启策略](#_bookmark23) 14
   6. [数独问题的规约和求解](#_bookmark24) 15
      1. [数独终局的生成](#_bookmark25) 15
      2. [数独问题规约](#_bookmark26) 16
      3. [挖洞法生成数独终局](#_bookmark27) 16
      4. [数独问题求解](#_bookmark28) 17

[华 中 科 技 大 学 课 程 实 验 报 告](#_TOC_250000)

* + 1. [数独问题的交互和可视化](#_bookmark29) 17

1. [系统实现与测试](#_bookmark30) 19
   1. [系统实现](#_bookmark31) 19
      1. [软硬件环境](#_bookmark32) 19
      2. [各模块主要函数](#_bookmark33) 19
      3. [CNF 文件的读取和解析](#_bookmark34) 19
      4. [DPLL 算法](#_bookmark35) 20
      5. [数独问题的生成和规约](#_bookmark36) 21
      6. [程序输入输出](#_bookmark37) 21
   2. [系统测试](#_bookmark38) 22
      1. [交互页面展示](#_bookmark39) 22
      2. [CNF 文件的读取和解析以及程序输入输出测试](#_bookmark40) 24
      3. [DPLL 算法测试](#_bookmark41) 28
      4. [数独游戏求解及简易游玩模块测试](#_bookmark42) 32
2. [总结与展望](#_bookmark43) 35
   1. [全文总结](#_bookmark44) 35
   2. [工作展望](#_bookmark45) 35

[6 体会](#_bookmark46) 36

[参考文献](#_bookmark46) 37

[附录](#_bookmark46) 38

华 中 科 技 大 学 课 程 实 验 报 告

# 任务书

### 设计内容

SAT 问题即命题逻辑公式的可满足性问题（satisfiability problem），是计算机科学与人工智能基本问题，是一个典型的 NP 完全问题，可广泛应用于许多实际问题如硬件设计、安全协议验证等，具有重要理论意义与应用价值。本设计要求基于 DPLL 算法实现一个完备 SAT 求解器，对输入的CNF 范式算例文件，解析并建立其内部表示；精心设计问题中变元、文字、子句、公式等有效的物理存储结构以及一定的分支变元处理策略，使求解器具有优化的执行性能；对一定规模的算例能有效求解，输出与文件保存求解结果，统计求解时间。

### 设计要求

要求具有如下功能：

1. 输入输出功能：包括程序执行参数的输入，SAT 算例cnf 文件的读取， 执行结果的输出与文件保存等。(15%)
2. 公式解析与验证：读取cnf 算例文件，解析文件，基于一定的物理结构， 建立公式的内部表示；并实现对解析正确性的验证功能，即遍历内部结构逐行输出与显示每个子句，与输入算例对比可人工判断解析功能的正确性。数据结构的设计可参考文献 [1-3]。(15%)
3. **DPLL** 过程：基于 DPLL 算法框架，实现 SAT 算例的求解。(35%)
4. 时间性能的测量：基于相应的时间处理函数（参考time.h），记录 DPLL

过程执行时间（以毫秒为单位），并作为输出信息的一部分。(5%)

1. 程序优化：对基本 DPLL 的实现进行存储结构、分支变元选取策略 [1-3] 等某一方面进行优化设计与实现，提供较明确的性能优化率结果。优化率的计 算公式为：[(t-to)/t]\*100%, 其中t 为未对 DPLL 优化时求解基准算例的执行时间， to 则为优化 DPLL 实现时求解同一算例的执行时间。(15%)
2. **SAT** 应用：将数双独游戏 [5] 问题转化为 SAT 问题 [6-8]，并集成到上面的求解器进行数独游戏求解，游戏可玩，具有一定的/简单的交互性。应用问题归约为 SAT 问题的具体方法可参考文献 [3] 与 [6-8]。(15%)

## 华 中 科 技 大 学 课 程 实 验 报 告

### 参考文献

* 1. 张健著. 逻辑公式的可满足性判定—方法、工具及应用. 科学出版社，2000
  2. Tanbir Ahmed. An Implementation of the DPLL Algorithm. Master thesis,

Concordia University,Canada,2009

* 1. 陈稳. 基于 DPLL 的 SAT 算法的研究与应用. 硕士学位论文，电子科技大学，2011
  2. Carsten Sinz. Visualizing SAT Instances and Runs of the DPLL Algorithm. J Autom Reasoning (2007) 39:219–243
  3. 360 百科：数独游戏 https://baike.so.com/doc/3390505-3569059.html

Twodoku：https://en.grandgames.net/multisudoku/twodoku

* 1. Tjark Weber. A sat-based sudoku solver. In 12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2005, pages 11– 15, 2005.
  2. Ins Lynce and Jol Ouaknine. Sudoku as a sat problem. In Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics, AIMATH 2006, Fort Lauderdale. Springer, 2006.
  3. Uwe Pfeiffer, Tomas Karnagel and Guido Scheffler. A Sudoku-Solver for Large Puzzles using SAT. LPAR-17-short (EPiC Series, vol. 13), 52–57
  4. Sudoku Puzzles Generating: from Easy to Evil.

<http://zhangroup.aporc.org/images/files/Paper_3485.pdf>

* 1. 薛源海，蒋彪彬，李永卓. 基于“挖洞”思想的数独游戏生成算法. 数学的实践与认识,2009,39(21):1-7
  2. 黄祖贤. 数独游戏的问题生成及求解算法优化. 安徽工业大学学报 (自然科学版), 2015,32(2):187-191

华 中 科 技 大 学 课 程 实 验 报 告

# 1 引言

### 课题背景与意义

#### SAT 问题

SAT 问题即布尔可满足性问题，是确定是否存在满足给定布尔公式解释的问题。如果对于给定的布尔公式变量可以一致地用 TRUE 或者 FALSE 替换，那么该布尔公式可满足，相反则不满足。

可满足性问题（Satisfiability Problem,SAT）是人工智能领域中的一个重要研究方向，也是自动推理中的核心问题之一。SAT 问题广泛应用于工程技术、软件验证、数学定理证明、组合优化以及逻辑电路等价性验证等多个领域，因此研究SAT 问题有助于拓展知识面以及今后的实际应用。

#### 数独游戏

“数独 sudoku”是十八世纪瑞士数学家欧拉发明的。2004 年 11 月 12 日,“数独”游戏登上了《泰晤士报》的版面, 很快, 作为该报每日内容的“数独”游戏就风靡了英美! 规则很简单: 有 9×9 共 81 个方格 ( 即依次有 9 个九宫格, 详细图可参见第 4 节内容) , 在其中填入 1 到 9 的数字, 让每个数字在每一行、每一列及每一个九宫格里都只出现一次。谜题中会预先填入若干数字, 其它方格为空白, 玩家得依谜题中的数字分布状况, 逻辑推敲出剩下的空格里是什么数字。由于规则简单, 在推敲之中完全不必用到数学计算, 只需运用逻辑推理能力, 所以无论男女老幼, 人人都可以玩, 而且容易上手、容易入迷。世界各地有很多数独俱乐部, 还有的国家如法国等专门举行过数独比赛, 其风靡程度可见一斑。

国内外许多学者已对数独的求解算法做了深入研究，例如递归法、回溯候选数法、枚举算法等。

### 国内外研究现状

可满足性问题是计算机科学领域和人工智能等领域中的重要研究对象。但是其求解算法的时间开销和空间开销却异常惊人。对于一个含有 *n* 个命题逻辑变量的合取范式来说，如果使用穷举法来罗列所有真值指派进行求解，虽然在理

## 华 中 科 技 大 学 课 程 实 验 报 告

论上是可行的，但算法的时间复杂度却是指数级规模，为 *O*(2*n*)，计算机如果采用这种方式进行求解将负担不起如此大的开销。搜索空间如此庞大，使得计算机在可等待的时间里不能计算出结果，进而产生组合爆炸问题，所花费的计算时间是人们不能容忍的。S.A.Cook 于 1971 年首次证明了布尔表达式的可满足性问题属于 *NP* 完全问题。*NP* 完全问题排在七大数学难题之首，在计算复杂性理论中具有非常重要的地位，一方面因为它有着极大的理论价值并且非常难解，另一方面是一旦被破解以后，在诸多的工程领域里还可以得到广泛的应用。但是在

*P* ≠*NP* 的假设条件下，*SAT* 问题不可能有多项式时间的求解算法。*SAT* 问题已经成为了一类问题的难度标准，所以称其为 *NP* 完全问题的种子。

20 世纪 60 年代开始，*SAT* 问题倍受世人关注，不少学者在这方面做了大量工作，取得了很大突破。

1960 年，Martin Davis 和 Hilary Putnam 提出了首个求解 *SAT* 问题的完备算法，称为 DP 算法。

1962 年，George Logemann 和 Donald W. Loveland 等人在 *DP* 算法的基础上提出了 *DPLL* 算法。

1971 年，Stephen Arthur. Cook 证明了 *SAT* 问题是 *NP* 完全问题。

1999 年，João Marques Silva 等人在 *DPLL* 算法的基础上提出了 *GRASP* 算法，首次引入了冲突学习回溯策略。

2003 年，Bart Selman 和 Henry Kautz 在人工智能国际合作会议上指出当前

SAT 问题面临的十大挑战性问题。

2005 年，Niklas Eén 等人提出的 SatEliteE 首次实现了预处理简化问题的规模和复杂性。

2007 年，Bart Selman 和 Henry Kautz 对当前 *SAT* 问题的现状做了全面的整理和叙述。

每年此领域都会组织一次可满足性理论和应用方面的国际会议并组织 *SAT* 竞赛，以寻找高效的 *SAT* 求解器，并且详细展示出这些求解器各方面的性能， 旨在进一步促进 *SAT* 问题算法的研究。

近年来，由于在计算机理论研究和实际应用中的重要作用，*SAT* 问题成为国内外研究的热点问题，许多研究人员在这方面做了大量研究，并在算法研究和技术实现上取得了较大的突破，这也间接推动了形式化验证和自动推理等领域的发展。

## 华 中 科 技 大 学 课 程 实 验 报 告

### 课程设计的主要研究工作

本设计基于 *DPLL* 的算法与程序框架，实现一个完备 *SAT* 求解器，同时程序的改进，对输入的 *CNF* 范式算例文件，解析并建立其内部表示；精心设计问题中变元、文字、子句、公式等有效的物理存储结构，设计分支变元处理策略， 使求解器具有优化的执行性能；对一定规模的算例能有效求解，输出与文件保存求解结果，统计求解时间。

数独游戏可转化为 *SAT* 问题，用本系统实现的 *SAT* 求解器可以快捷地对数独问题转化的 *CNF* 文件进行求解，再以变元真值数据转化的数独盘格式输出求解答案。本系统具有一定的交互功能，用户可以利用本系统进行数独游戏，系统将自动判断解的正确性，并输出正确答案。

具体研究工程步骤如下：

1. 对于 *SAT* 求解问题的相关背景、基本原理以及传统的 *DPLL* 框架进行深入了解，根据相关资料对于项目确定整体方向，设计项目整体流程。
2. 根据 *SAT* 求解器设计相应的数据结构和算法，进而实现基于 *DPLL* 框架的 *SAT* 求解器，并使用相应的算例进行测试:
   1. 数据结构主要使用十字链表的结构
   2. *CNF* 文件的读取、解析和初始化
   3. 基于 *DPLL* 框架的 *SAT* 求解器（核心模块）
   4. 双数独问题的生成和归约，转化成 *SAT* 问题并求解
3. 通过查阅文献和不断尝试，优化数据结构和算法，实现求解效率的提高。主要优化方向有：
   1. 数据结构优化
   2. 变元分支选取策略优化
   3. 算法框架优化
4. 设计问题转化策略将数独问题归约为 *SAT* 问题并求解。
5. 在程序实现的过程中将程序各个结构模块化

华 中 科 技 大 学 课 程 实 验 报 告

# 2 系统需求分析与总体设计

### 系统需求分析

本设计基于 *DPLL* 的算法与程序框架，实现一个完备 *SAT* 求解器，通过一定的策略在 *DPLL* 框架的基础上实现优化。然后将数独问题规约为 *SAT* 问题， 并基于完备 *SAT* 求解器完成数独游戏求解的设计。

### 系统总体设计

系统总体设计分为两个大的模块：基于 *DPLL* 算法的 *SAT* 求解器和简易数独游戏，各自模块下面还有一些小的功能，大致介绍如下：

* + 1. 基于 *DPLL* 算法的 *SAT* 求解器，需要依次实现：
       1. *CNF* 的读取解析，遍历输出，保存建立其对应的数据存储结构；
       2. *DPLL* 求解，计算求解时间并显示，将结果保存到同名 *.res* 文件里；
       3. 实现对解析正确性的验证功能，即遍历内部结构逐行输出与显示每个子句，与输入算例对比可人工判断解析功能的正确性。
    2. 数独游戏，需要依次实现：
       1. 随机数独游戏的创建，用户可选择挖洞数量进而自行调整难度；
       2. 通过挖洞法生成数独具有唯一解，转化为 *CNF* 文件 *DPLL* 进行求解， 再可视化地将结果打印到屏幕上；
       3. 实现数独界面，具有一定的交互性和可玩性，实现一定程度的可视化交互。

具体流程图如下：

## 华 中 科 技 大 学 课 程 实 验 报 告

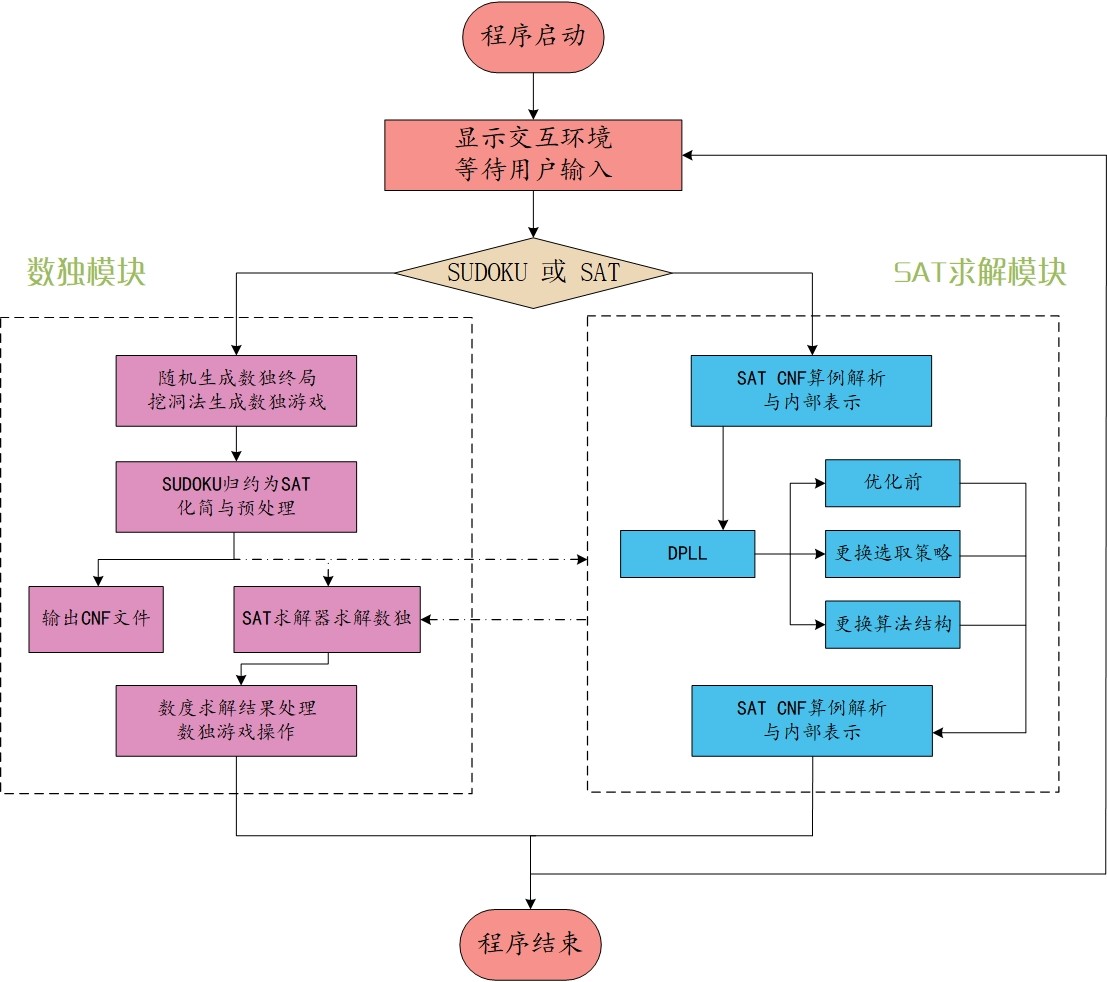


图 2-1 系统整体设计图

华 中 科 技 大 学 课 程 实 验 报 告

# 3 系统详细设计

### 有关数据结构的定义

#### 相关数据结构

本次实验对于 *CNF* 文件的解析结果整体采用十字链表的数据结构存储。首先对于文字建立一维链表，然后对于子句建立二维链表存储信息，建立 *CNF* 结构存储子句和相关信息，实现对于 *CNF* 文件信息的存储。具体数据结构及包含的数据项如表 3-1：

表 3-1 **数据结构相关定义**

需要处理的数据 所包含的数据项 数据类型

变元数 (literal\_num) 整型变量

CNF 文件

子句

文字

子句数 (clause\_num) 整型变量

首子句 (first\_clause) 结构体指针类型变量文字数 (number) 整型变量

首文字 (first\_literal) 结构体类型变量被监视节点 (watch\_literal[2]) 整型数组

下一子句 (next\_clause) 结构体指针类型变量文字 (literal) 整型变量

下一子句文字 (next\_literal) 结构体指针类型变量下一相同文字 (next\_same\_literal) 结构体指针类型变量对应子句句首 (head) 结构体指针类型变量

对于变元建立结构体数组，存储处理变元所需要的相关信息，包括真值情况、存储处理时对应栈的深度、变量决策时对应的得分、文字总数、正文字总数和负文字总数等，具体的数据结构如表 3-2：

## 华 中 科 技 大 学 课 程 实 验 报 告

表 3-2 **数据结构相关定义**

需要处理的数据 所包含的数据项 数据类型

真值情况 (is\_value) 整型变量栈的深度 (dep) 整型变量

变量得分 (score) 整型变量

文字变元

文字总数 (num) 整型变量

正文字数目 (pos) 整型变量

负文字数目 (nev) 整型变量

首子句 (first) 结构体类型变量

#### 数据结构关联

建立好的文字、子句和 *CNF* 结构等数据结构，具有一定的关联性，具体阐述如下：由文字结点链接成子句链表，同时子句结点链接又形成 *CNF* 链表，同时由相同文字建立好文字链表，形成一定的网状结构，便于搜索和修改。具体图示如图：

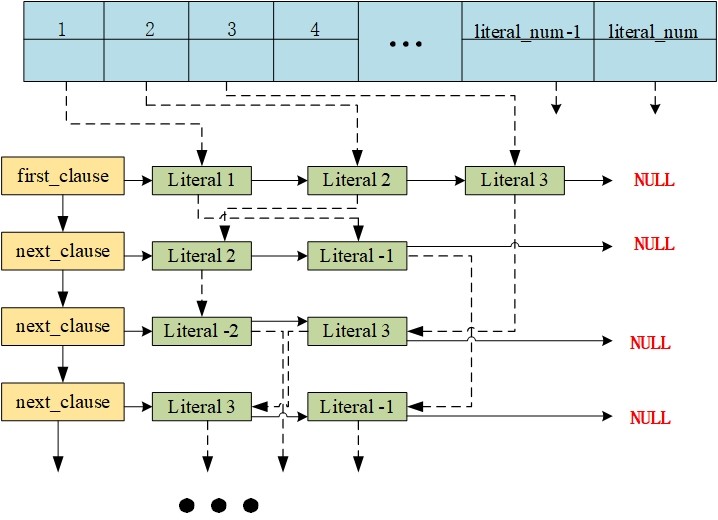


图 3-1 数据结构关联图

## 华 中 科 技 大 学 课 程 实 验 报 告

### 主要算法设计

对于 CNF 公式中为使所有的子句满足条件，需要对于所有文字进行合理的真值赋值，通过真值赋值所得出的搜索空间可以用一颗二叉树来表示。树的结点代表对应的文字变元，树的分支代表不同的赋值情况，从二叉树到叶子结点的一条包含所有变元的路径表示 CNF 公式对应的一组合理的真值赋值。基于 DPLL 的算法是通过对于这棵二叉树从根节点开始进行深度优先搜索寻找合适的通路， 以得到问题的可满足解。

**Algorithm 1** DPLL 框架

输入**:** S /\*S 代表子句集 \*/

输出**:** *True or False*

1: **function** *DPLL*(*S*)

2: **while** *S* 中存在单子句 **do**

3: 在 *S* 中选一个单子句 *L*;

4: 依据单子句规则，利用 *L* 化简 *S*;

5: **if** *S* = Φ **then return** True

6: **else if** *S* 中有空子句 **then return** False

7: **end if**

8: **end while**

9: 基于某种变元选取策略选取变元 *v*; 10: **if** *DPLL*(*S ∪ v*) **then return** True 11: **end if**

12: **if** *DPLL*(*S ∪ ¬v*) **then return** True

13: **end if**

14: **end function**

在搜索二叉树的过程中如果无法进行单子句传播，需要通过一定的策略进行变量决策；如果发生错误，则变量需要回溯到某一决策层。因此，DPLL 算法框架对应求解过程主要分为以下三个阶段：

1. 变量决策阶段：在搜索过程的每一分支阶段，选择未赋值的一个变量为其赋值为 0 或 1，该变量称为决策变量，决策变量在赋值时所处的二叉树中的高度称为它的决策层。

## 华 中 科 技 大 学 课 程 实 验 报 告

1. 推理阶段：每一次已选择变量赋值之后，识别该赋值所导致的必要的赋值或者根据已有的变量赋值对子句进行化简，即进行布尔约束传播过程。
2. 回溯阶段：推理过程中发生冲突时，实现算法的回溯，使搜索过程从较深的变量决策层返回至较浅的决策层。冲突即指 BCP 过程中，至少出现了一个子句不可满足的情况。

### 变量决策阶段

变量决策策略主要用在回溯算法的搜索过程中，对于未进行赋值的变量进行合理的选择，进而实现分裂搜索，以实现进一步传播。变量决策策略的选择可以影响决策层的数目和决策深度，对计算效率起到决定性作用。变元选择的基本准则是子句越短，子句所包含的变元越难满足，因此应该先进行选择；出现频率越高的变元，选择后满足的子句数更多，因此应该优先选择。

#### 变量选取策略

本实验中在优化前的变量决策方法为在变元中顺次选择没有进行变元赋值的变量。为了对于变量决策阶段进行优化，主要采用以下方法：

* + - 1. 最大频率优先：在 CNF 文件中选择变元出现频率最高的变元。
      2. 最短子句出现频率最大优先：在 CNF 文件中选择含变元数最少的子句，根据变元统计，选择所选子句中出现频率最高的变元，再根据变元正负文字比例中所占比例更大的文字作为所选文字。
      3. 变元加权计算得分：在 CNF 文件中计算变元得分，在选择变元时优先选择得分较高的变元。由于变元出现频率与变元得分正相关，变元所在子句长度与变元得分负相关，因此通过多次尝试，最终确定计算公式如下：

∑

##### score =

*literal*\_*num*

##### clause\_num

*y*

*i*

*length*2

(3.1)

*i*=1

其中 clause\_num 表示子句数目，literal\_num 表示变元数目，*yi* 表示决策变量，若变元存在于子句中，*yi* = 1，反之 *yi* = 0，length 代表子句所含变元数。

* + - 1. 独立变量衰减和策略修改版：首先通过变量加权计算得分。为了减少决策时间，可以在遇到冲突时将冲突的变元放入冲突栈中，在选择变元时，只更

## 华 中 科 技 大 学 课 程 实 验 报 告

新与冲突相关的变元得分（冲突变元所在子句的变元），进而选取得分高的变元。

说明：因为在遍历 CNF 文件，对变量计算得分的过程中，分数不断累加，有可能导致分数差距过大，因此设定某一阈值，存在变量得分达到或者超过阈值的时候，对于全体变元的得分进行等比例降低，有效避免某一变元分数过高，与其他变元差距较大的情况。

#### 冲突分析

当 BCP 过程中遇到冲突时，通过对于冲突原因进行有效的分析可以对于变量决策有积极影响。本实验在面对冲突时采取的策略时对于冲突相关变量进行 双倍计算得分，即对于所有冲突变元所在子句的所有变元重复一次得分的计算， 这样可以在变量选择的过程中更容易选到这些变元，让不容易满足的子句所在 变元被优先选择，可以更好地避免重复的冲突再次发生。

### 推理阶段

SAT 求解器的 80%～90% 时间都用来执行BCP 过程，所以 BCP 的执行效率很重要。为此本文主要做了以下方面的努力：

#### 数据结构优化

通过对于数据结构的优化，有效减少 BCP 执行的事件。

首先，增加邻接表数据结构。为每一个变元增加邻接表，在单子句传播过程中往往需要遍历整个 *CNF* 文件，然而这样极大地降低了BCP 过程所用的时间。通过邻接表的数据结构，能够有效的遍历含有某一变元的所有子句，便于更方便定位每一个变元对应 *CNF* 文件中变元的位置，能够有效加快BCP 执行的时间。

其次，对于 *CNF* 文件中的每一个变元，都增加一个指针指向子句的头指针， 以便在更新变元的过程中，更加方便地获取加权公式中的参数值。

最终对应的文字结点对应参数和参照下图：

## 华 中 科 技 大 学 课 程 实 验 报 告

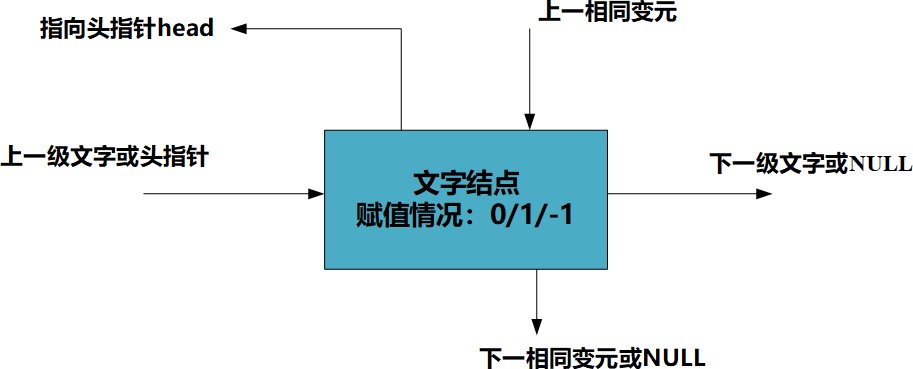


图 3-2 文字结点示意图

#### 传播方式优化

采用双字面量监督算法通过减少单子句传播后回溯时操作次数，加速 BCP

过程。

依据性质一个子句只要有两个不同的字面量都不为 0（即都不判定为假）， 该子句就不会成为单位子句或冲突。因此，我们主要通过监视两个不同的字面量watch\_literals，来判断每一个子句的具体情况。两个字面量可以被初始化为子句中任意的一对位置不同的自由字面量。

当进行赋值的时候，对于每一个包含赋值变元 p 的子句进行检查，如果该变元并不是被监视的变元，则跳过该子句进行下一子句的检查；如果该变元是被监视的变元，则对于以下情况进行处理：

* + - 1. 在子句中如果能够找到与被监视的字面量不同的非零字面量，则更改监视变量 p 为此变量；
      2. 如果唯一的非零字面量为另一监视变量 q，则需要分成两种情况进行讨论。如果 q 的字面量为 1，则不需要进行任何操作；如果 q 的字面量为 0，则需要对此变量进行赋值为真的操作，同时进行 BCP 过程。
      3. 如果所有的字面量的值均为 0，则该子句发生冲突，需要进行回溯。

## 华 中 科 技 大 学 课 程 实 验 报 告

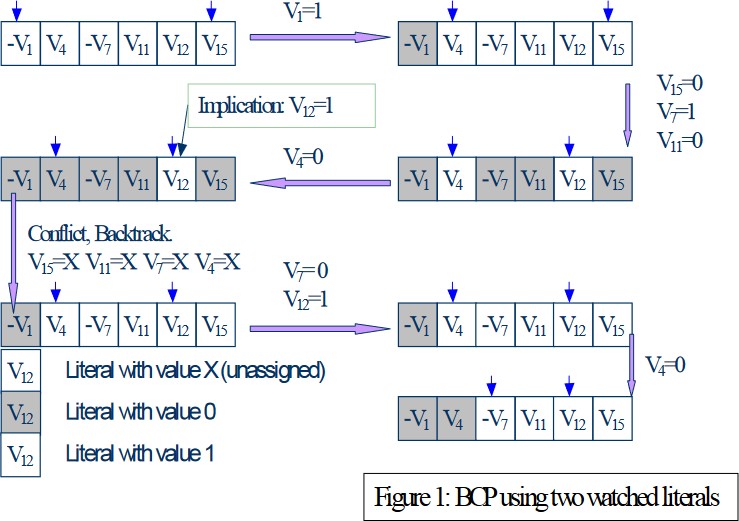


图 3-3 双字面量监督算法结构图

参考前人的算法结构图如上。因为在回溯过程中，被监督变量依然有效，因此只需调整变量的赋值，就可以重新进行变量选择，会节省大量时间。

### 回溯阶段

回溯是指在 BCP 过程中发生冲突，因此需要回溯到某一决策层进行重新赋值，进而重新进行 BCP 过程。

#### 智能回溯

在 BCP 过程中，将单子句传播的变元放入回溯栈中，同时标注决策层的深度，便于在回溯的过程中进行同一决策层的回溯。

在回溯过程中，通过将同一决策层的变元出栈，进行变元赋值的取消，即将变元转变成未赋值的状态，实现回溯到上一决策层的目的，可以进一步倒转决策变量的赋值，继续进行 BCP 过程。

#### 周期性重启策略

在 *SAT* 问题的求解过程中，由于最初的变量决策顺序可能不是最优的，这就会导致搜索陷入某些子空间中而白白耗费时间。重新启动机制，就是清除现在所有变量的赋值状态，重新选择一组决策变量进行赋值，然后进行正常的搜索过程。

周期性重启策略主要是为了对抗 *CNF* 文件的随机性，由于最初的变量赋值可能会导致 *CNF* 文件的求解陷入某一困境当中，重新根据当前得分进行赋值变

## 华 中 科 技 大 学 课 程 实 验 报 告

量选取，有利于打破僵局，更好地找到答案。该周期性重启策略，对于可满足的

*CNF* 文件求解效率提高具有显著作用。

### 数独问题的规约和求解

SAT 问题是理论计算机科学要研究的重要问题，在许多的实际领域都有应用，比如图论、数理逻辑、模型检测、EDA 领域等。这些问题都可以通过一些转换表示为 SAT 问题的形式进而运用 SAT 算法来进行求解。本文主要将其应用在求解数独问题中，通过求解器生成并求解数独。

#### 数独终局的生成

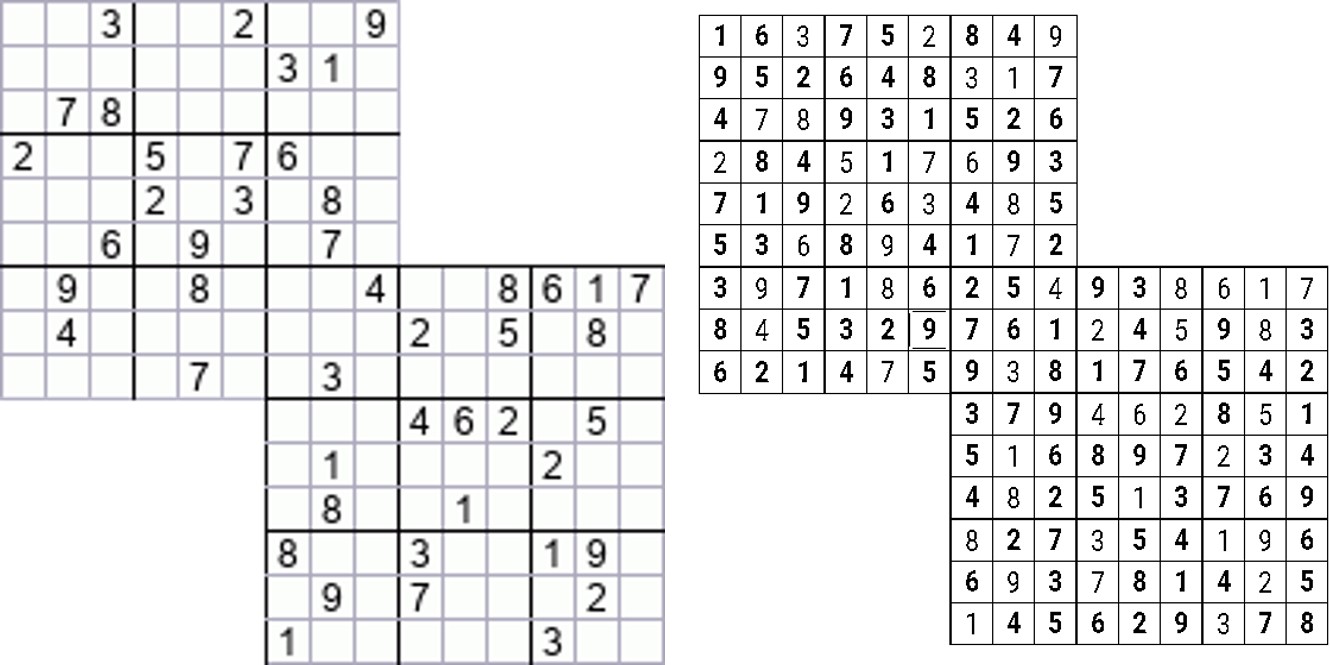
数独的要求是在 9×9 的格子内，每一行和每一列和每一宫内都包含不重复的九个元素。双数独则要求左上数独的最后一宫和右下数独的第一宫完全相同。

图 3-4 双数独样式及终局图

通过排列法生成终局是一种简单而有效的办法，通过随机数生成符合数独 要求的第一行，通过行平移可以发现，在接下来的 8 行中分别对第一行移动 3、6、1、4、7、2、5、8 格时，就能生成一个符合要求的数独。我们还可以发现，对于任何一个数独终局的 1-3 行，我们任意交换这三行的顺序，得到的仍然是一个合法的终局，4-6 行和 7-9 行同理，列也同理。这样我们又可以在刚刚的基础上， 按照这种方法扩展出很多种终局。

按照最后一宫将数独对称就可以得到双数独的终局，同时交换数字也可以增加终局生成的可能性。

## 华 中 科 技 大 学 课 程 实 验 报 告

#### 数独问题规约

变元可按语义编码为 1-9 之间数字构成的四位整数 pijk，i, j, k*∈*{1,2,…,9}， p*∈*{0,1}, 其中 i 表示单元格的行号，j 表示单元格的列号，k 表示单元格 <i, j> 填入的数字为k，p 代表数独盘的序号。如 0163 变元表示在左上数独盘第 1 行 6 列填入 3；负文字-1452 表示在右下数独盘第 4 行 5 列不填入 2。这样编码共有 1458 个变元。

约束条件按照 *CNF* 文件的格式要求写入 *CNF* 文件中，具体约束条件包括：

* + - 1. 每个格内只能含有 1-9 中的一个
      2. 每一行 1-9 只出现一次
      3. 每一列 1-9 只出现一次
      4. 每一宫 1-9 只出现一次
      5. 每一格一定要有 1-9 中的一个数字
      6. 双数独重叠部分等价关系

最终空数独的 *CNF* 文件中包含的变元数为 1458，子句数为 20736。

#### 挖洞法生成数独终局

通过挖洞法生成数独的算法效率较高，而且可以通过挖洞的顺序、数量等控制生成数独的难度，因此本实验主要通过挖洞法生成数独，同时通过控制挖洞数量进而控制生成数独的难度。

挖洞法的思想主要是通过挖洞的办法，不断生成含有空格的数独，同时在挖洞的过程中通过 SAT 求解器确保数独解的唯一性和确定性。

本实验主要采用挖洞法的剪枝优化策略，即通过由左至右，由上至下的顺序依次挖洞，对每一个格子只进行一次挖洞判断操作。例如对第一个格子进行挖洞后，通过填入其他数字，调用 SAT 求解器，判断数独问题是否有解，如果该数独填入其他数字后仍然有解，则说明该格子的挖去使得数独的为一街被破坏，因此应该将此格子恢复挖洞前的状态，同时标记此格不能被挖洞；如果没有其他解， 则说明此洞可以被挖。

## 华 中 科 技 大 学 课 程 实 验 报 告

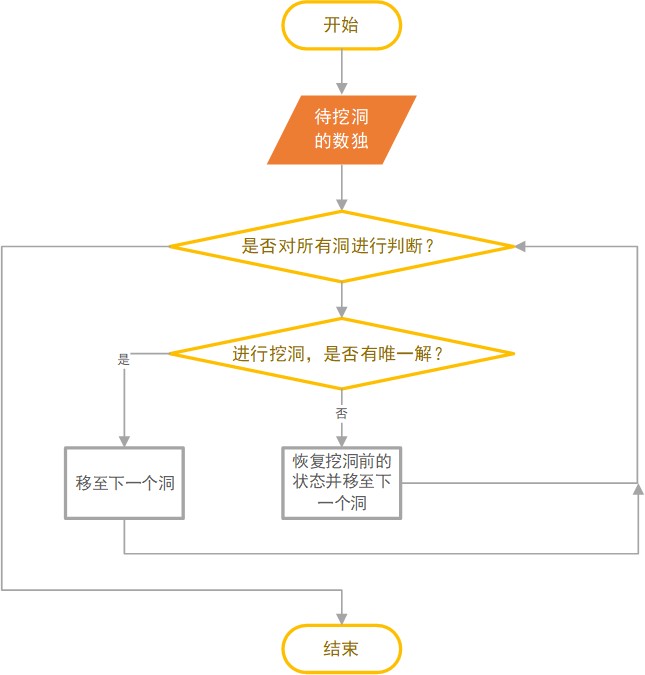


图 3-5 挖洞法算法流程图

#### 数独问题求解

将已知的单子句写入单子句栈中，调用 SAT 求解器进行求解，对对应的语义编码转换为自然顺序编码，公式为：pijn → p\*729+(i-1)\*81+(j-1)\*9+n，因此可以将此进行逆变换，通过分别对 9，81，729 进行取模操作，最终可以得到数独问题的语义编码。

#### 数独问题的交互和可视化

为了提高游戏的可玩性和交互性，本实验设计将数独问题进行可视化交互， 主要通过 *EasyX* 进行图形化绘制。

## 华 中 科 技 大 学 课 程 实 验 报 告

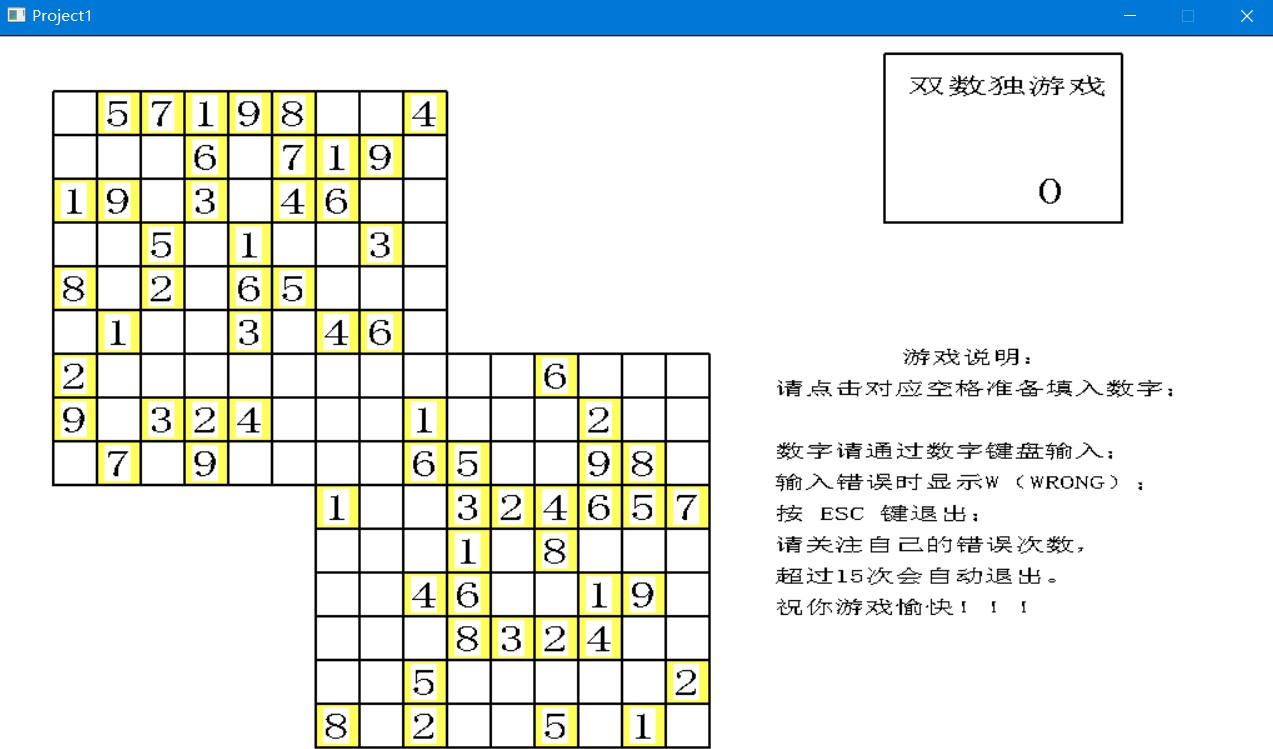


图 3-6 数独界面

数独问题的交互逻辑主要包括进行难度选择、绘制数独盘、进行方格选择、填充数字、判断正误、计算输入错误次数等，绘制好的数独界面如上图。

华 中 科 技 大 学 课 程 实 验 报 告

# 4 系统实现与测试

### 系统实现

#### 软硬件环境

* + - 1. 硬件环境
         1. 处理器：Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz 2.21 GHz

(b) 机带：16.0 GB (15.8 GB 可用)

(c) 系统类型：64 位操作系统, 基于 x64 的处理器

* + - 1. 软件环境
         1. Windows 规格：Windows 10 专业版
         2. 编译器：Clion 2021.3.1 / Visual Studio 2022

#### 各模块主要函数

#### CNF 文件的读取和解析

对于 *CNF* 文件的读取需要建立相应的抽象数据结构，包括创建 *CNF* 结构CreateCNF、创建子句CreateClause、销毁 *CNF* 结构 DestoryCNF、销毁子句DestroyClause、销毁文字 DestoryLiteral、删除 DeleteLiteral、判断是否有单子句HasUnitClause 等运算。具体函数及实现如下：

## 华 中 科 技 大 学 课 程 实 验 报 告



图 4-1 CNF 文件解析函数组成图

#### DPLL 算法

实现 DPLL 算法需要建立相应的抽象数据结构，同时由于不同的优化策略或者算法，抽象数据结构会有相应的修改，主要包括恢复文字结点 Recover- Literal，更新变元信息 update\_storevalue，算法主体 *DPLL*，变元选取策略 Se- lect\_Literal\_DPLL，变元分支处理 DPLL1\_Partition，结点恢复 DPLL1\_Recover 等运算。具体函数及实现如下：

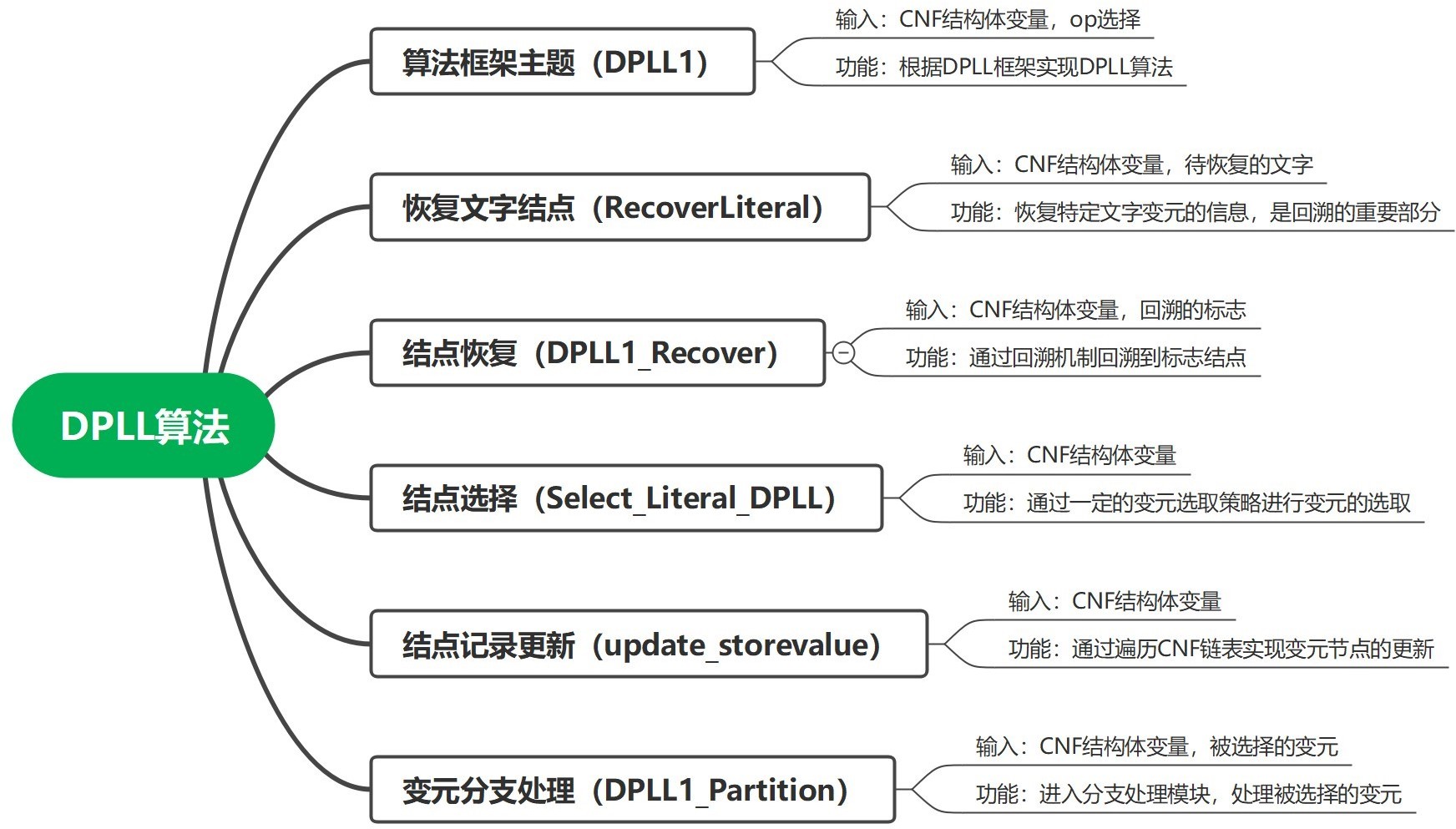


图 4-2 DPLL 算法框架函数组成图

## 华 中 科 技 大 学 课 程 实 验 报 告

#### 数独问题的生成和规约

数独问题的生成和规约需要建立相应的抽象数据结构，主要包括创造数独终局CreateSudoku，数独解析为 CNF 文件CreatePreSudokuFile，通过挖洞法生成数独 Dig\_Hole\_Easy，求解挖洞数独并记录 sudokusat，调用 SAT 求解器求解数独DPLL2\_SUDOKU 等运算。具体函数及实现如下：

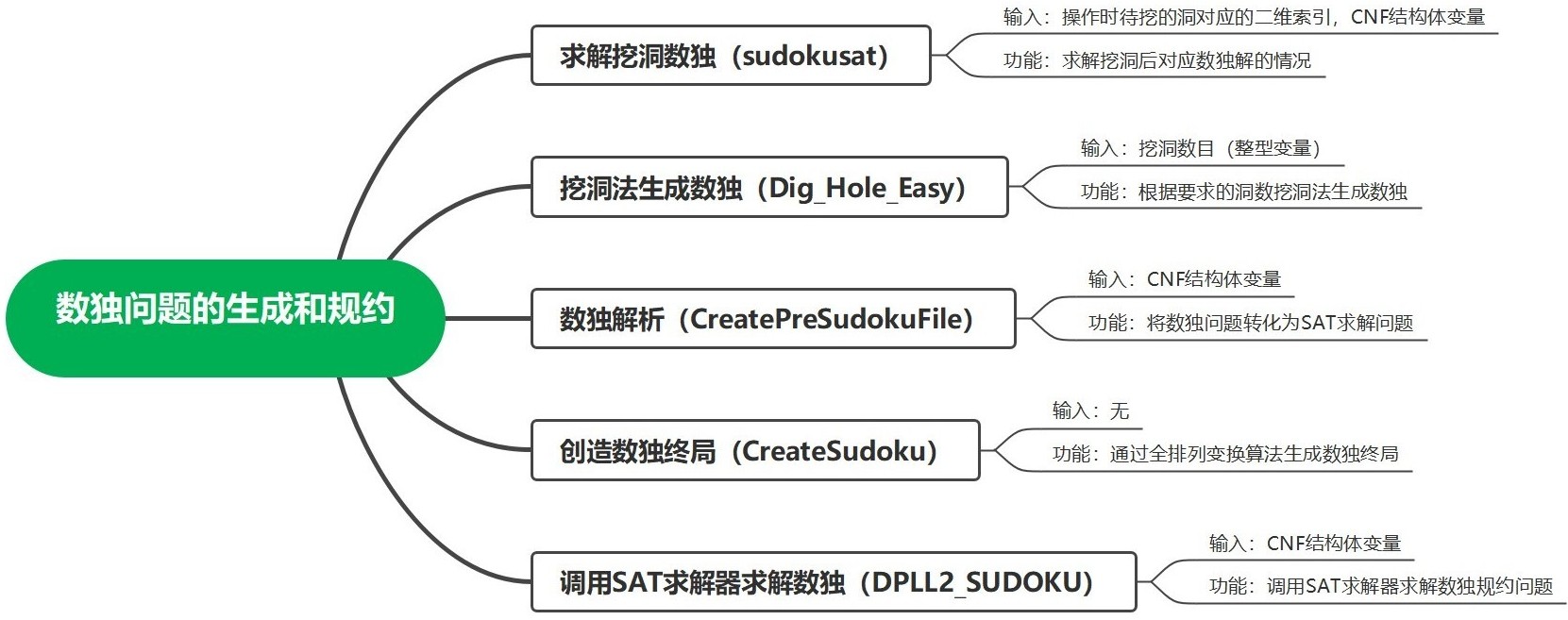


图 4-3 数独生成和规约函数组成图

#### 程序输入输出

程序的输入输出需要建立相应的抽象数据结构，主要包括存储 CNF 求解文件 store\_document，遍历 CNF 文件并输出 traveser\_cnf，验证 CNF 解的正确性prove\_cnf，展示 CNF 解 show\_cnf，输出数独终局 SudokuFinalPrint，输出数独求解时间 print\_time 等运算。具体函数及实现如下：

## 华 中 科 技 大 学 课 程 实 验 报 告

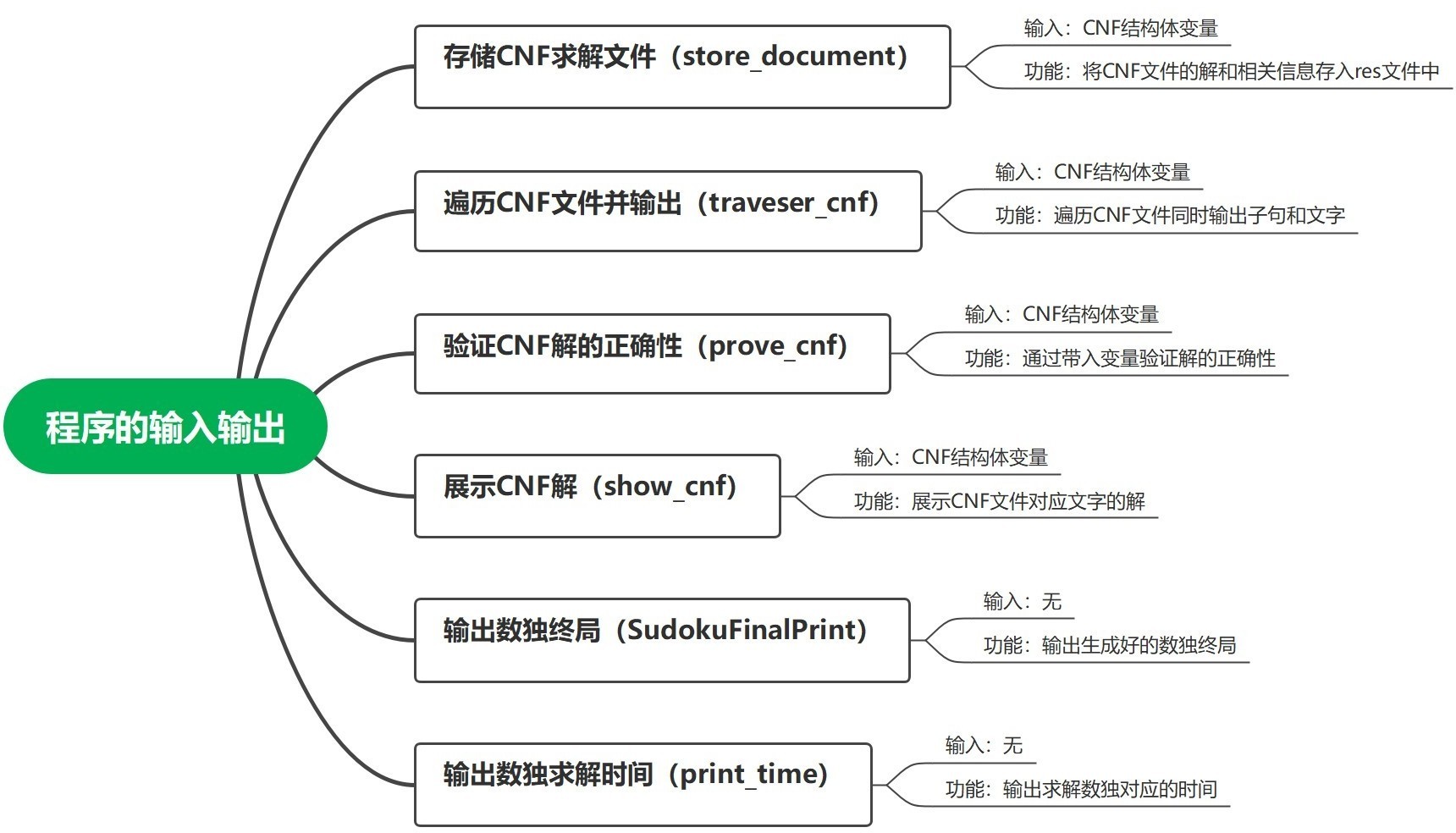


图 4-4 程序输入输出函数组成图

### 系统测试

#### 交互页面展示

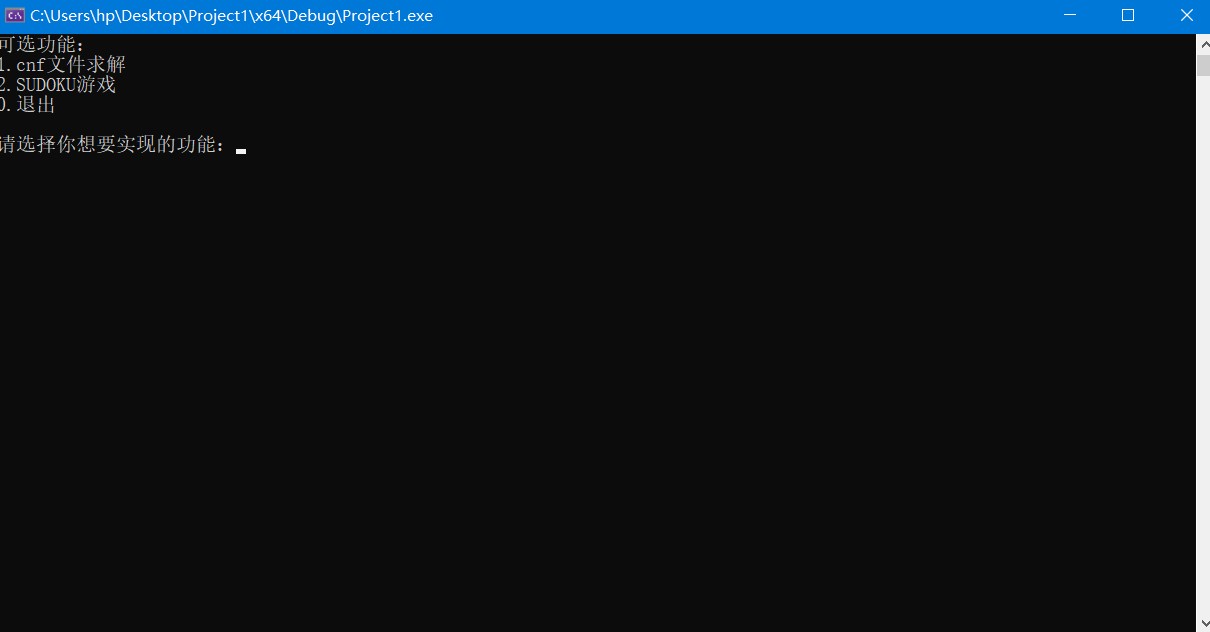
主要包括 *SAT* 求解过程中的一系列框型交互页面，以及数独问题的交互可视化页面。

图 4-5 初始交互界面

刚开始运行 exe 文件可以看到如上初始交互界面，可以进行基本功能的选择，也可以选择退出。

## 华 中 科 技 大 学 课 程 实 验 报 告

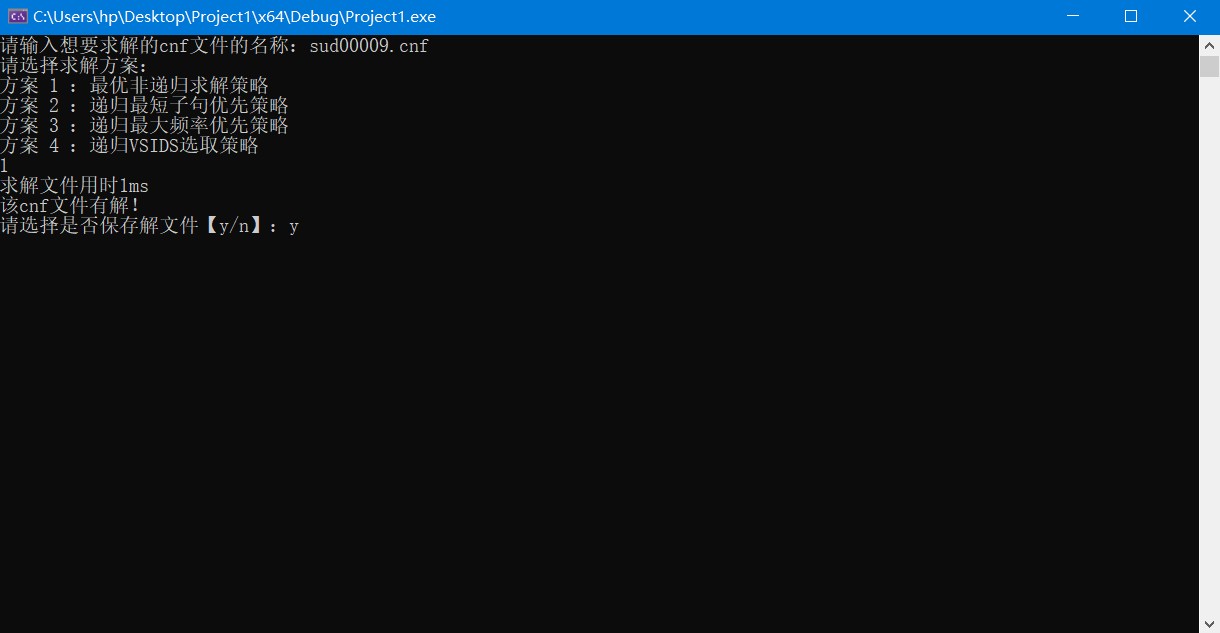


图 4-6 CNF 文件求解界面

上图为CNF 求解界面，输入CNF 文件名称后，选取方案策略进行求解，最后保存解文件可选。

图 4-7 数独游戏首页

上图为数独游戏的首页。

## 华 中 科 技 大 学 课 程 实 验 报 告

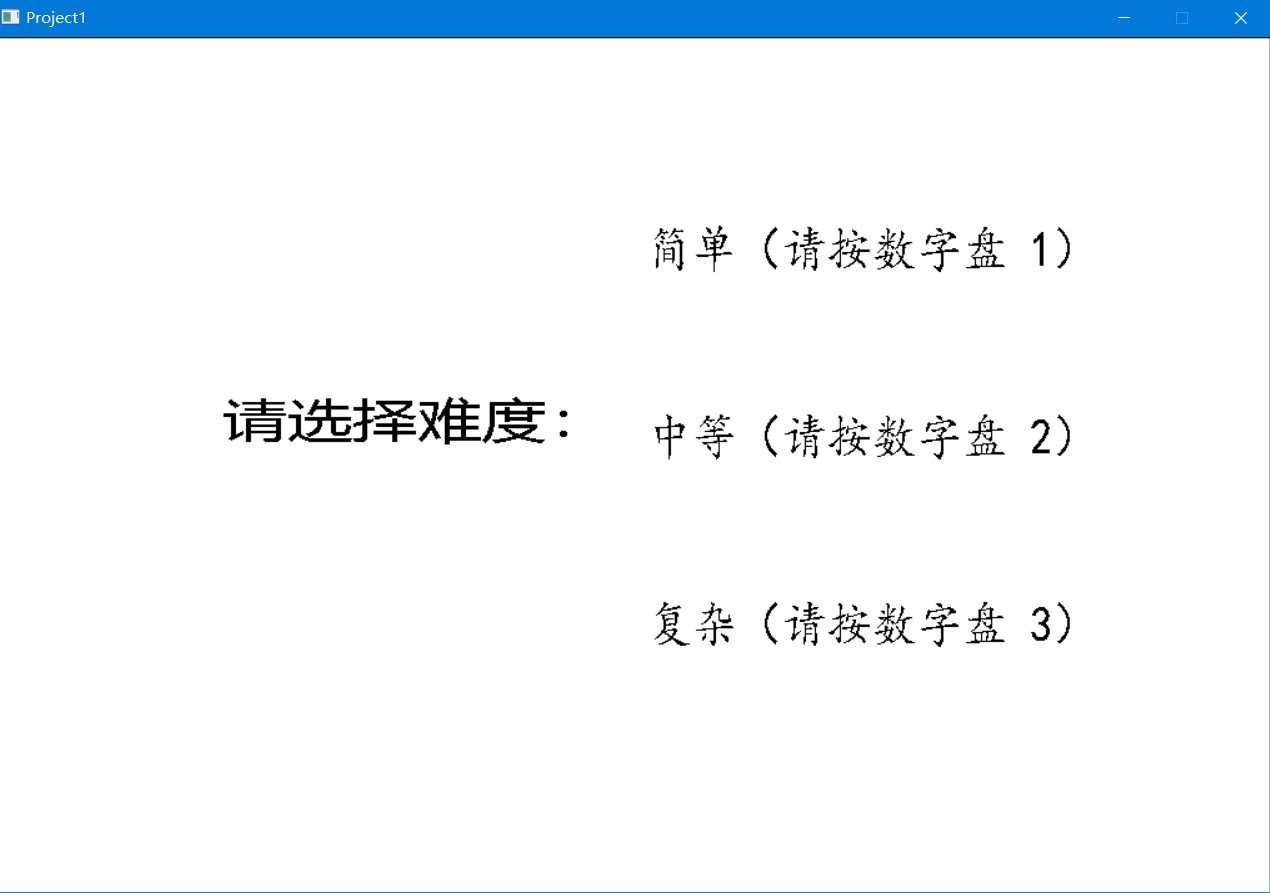


图 4-8 难度选择界面

上图为图形界面的难度选择界面，可以进行三个难度的选择。具体游戏界面已经在前文展示过，因此在此不再进行展示。

#### CNF 文件的读取和解析以及程序输入输出测试

首先简述一下过程：通过首页面选择 *CNF* 文件的读取和解析功能即可进入 *CNF* 文件读取解析模块，输入 *CNF* 文件名称后，可以通过选择求解方案对*CNF* 文件进行求解，输出求解时间和求解判断，同时提供选项是否保存解文件， 之后可以提供辅助功能进行 *CNF* 文件和解文件信息的查看。

该模块的测试主要为以下四个部分：

* + - 1. CNF 文件能否成功读取并解析
      2. CNF 文件能否正确求解
      3. 能否得出解的结构
      4. 能否保存解的信息

测试算例 **1**：**sud00009.cnf**

首先选择读取 CNF 文件，然后通过方案 1 的 SAT 求解器对 CNF 文件进行求解并保存解文件，求解 CNF 文件的视图如下：

## 华 中 科 技 大 学 课 程 实 验 报 告

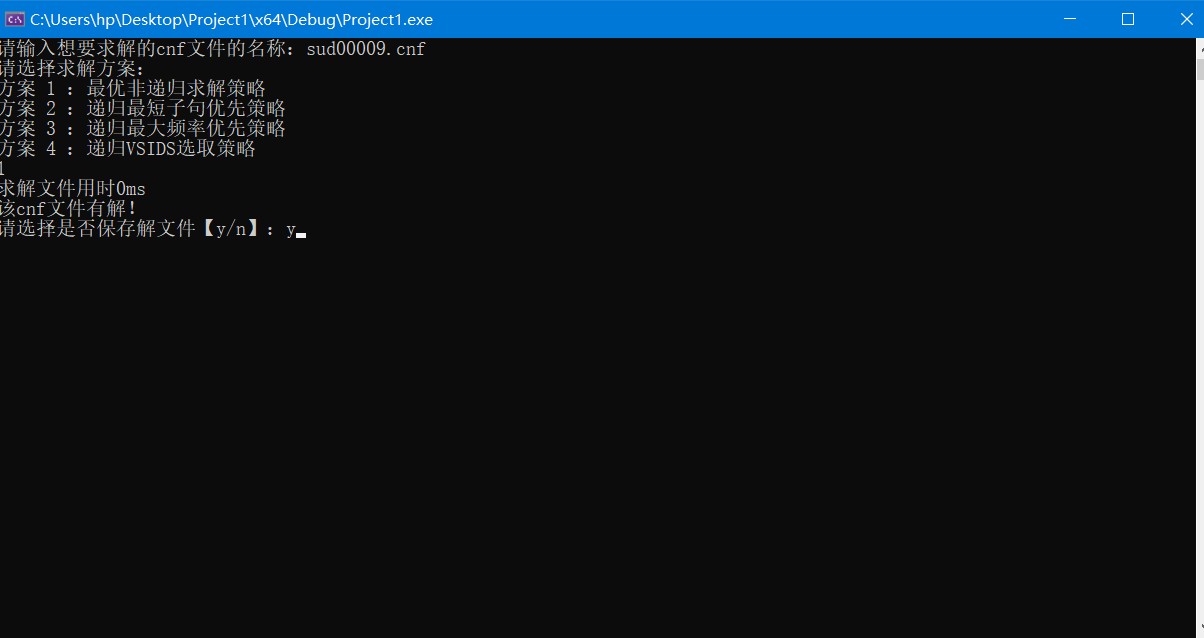


图 4-9 求解 CNF 文件的视图

然后通过一系列可以查看的信息对于 CNF 文件的读取和解析进行测试。通过遍历CNF 文件对CNF 文件能否成功读取并解析进行测试，测试结果如

下：

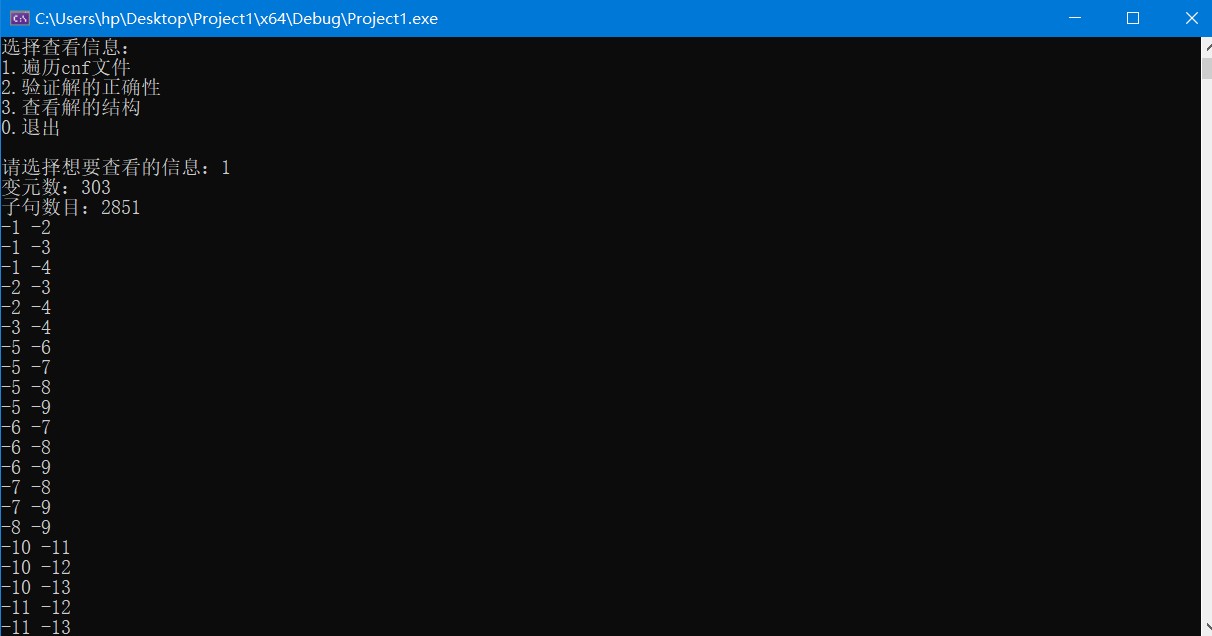


图 4-10 遍历 CNF 文件视图

通过验证解的正确性对 CNF 文件能否正确求解进行测试，测试结果如下：

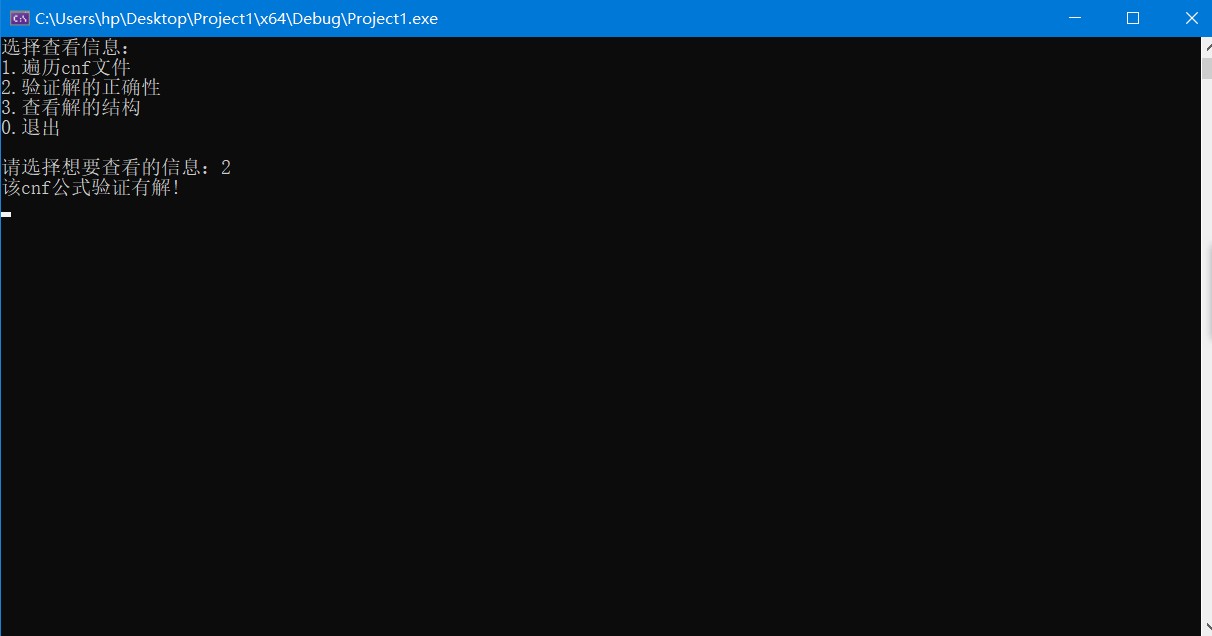


图 4-11 验证解的正确性视图

通过查看解的结构对CNF 文件能否得出解的结构进行测试，测试结果如下：

## 华 中 科 技 大 学 课 程 实 验 报 告

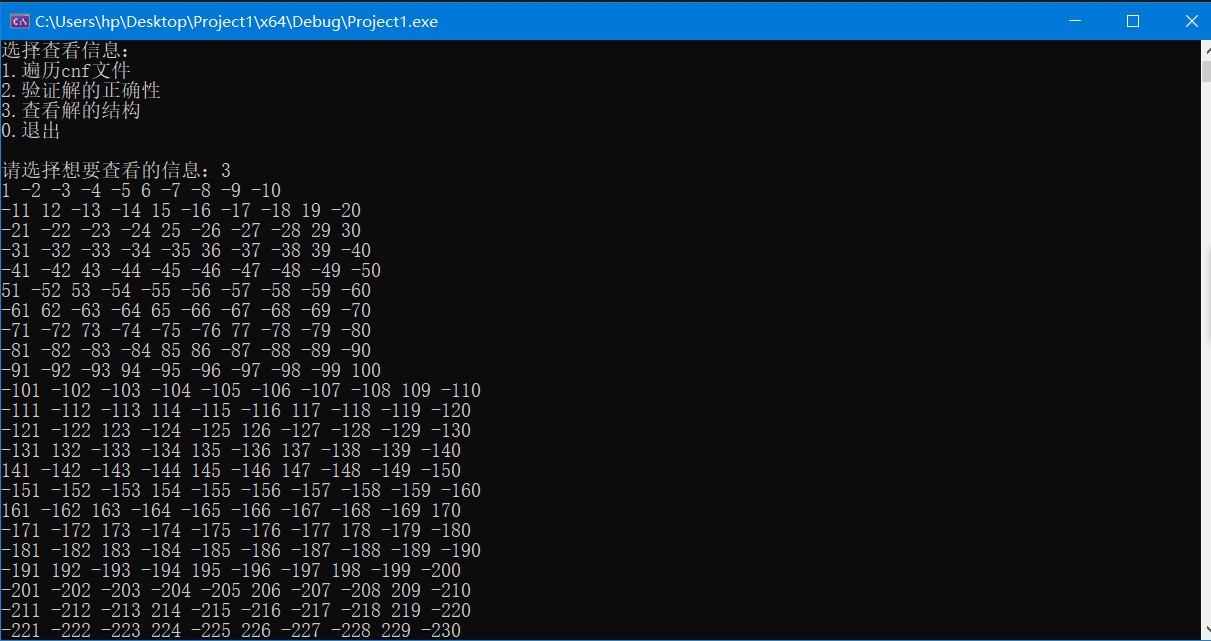


图 4-12 查看解的结构视图

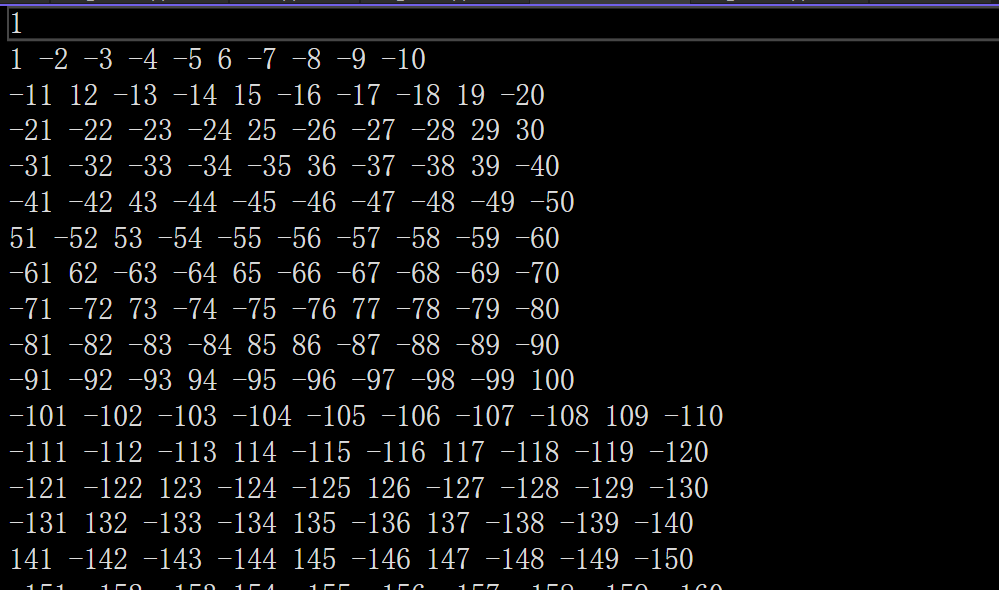
通过查看RES 文件对CNF 文件能否保存解的信息进行测试，测试结果如下：

图 4-13 查看 RES 文件视图测试算例 **2**：**ais10.cnf**

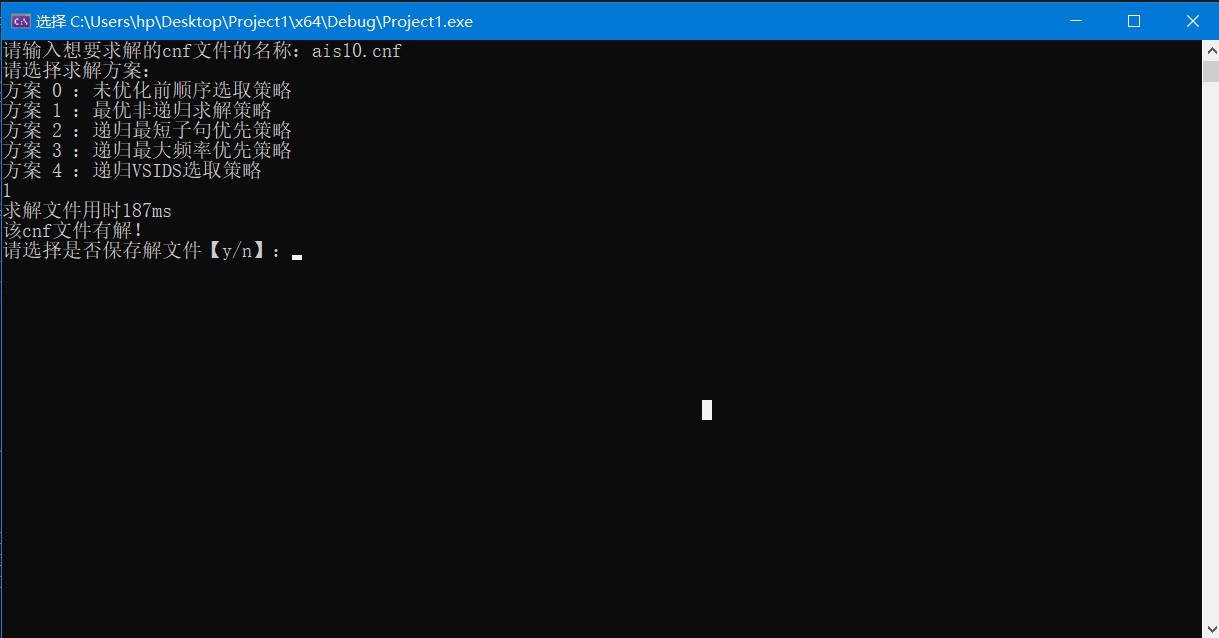
首先选择读取 CNF 文件，然后通过方案 1 的 SAT 求解器对 CNF 文件进行求解并保存解文件，求解 CNF 文件的视图如下：

图 4-14 求解 CNF 文件的视图

然后通过一系列可以查看的信息对于 CNF 文件的读取和解析进行测试。通过遍历CNF 文件对CNF 文件能否成功读取并解析进行测试，测试结果如

下：

## 华 中 科 技 大 学 课 程 实 验 报 告

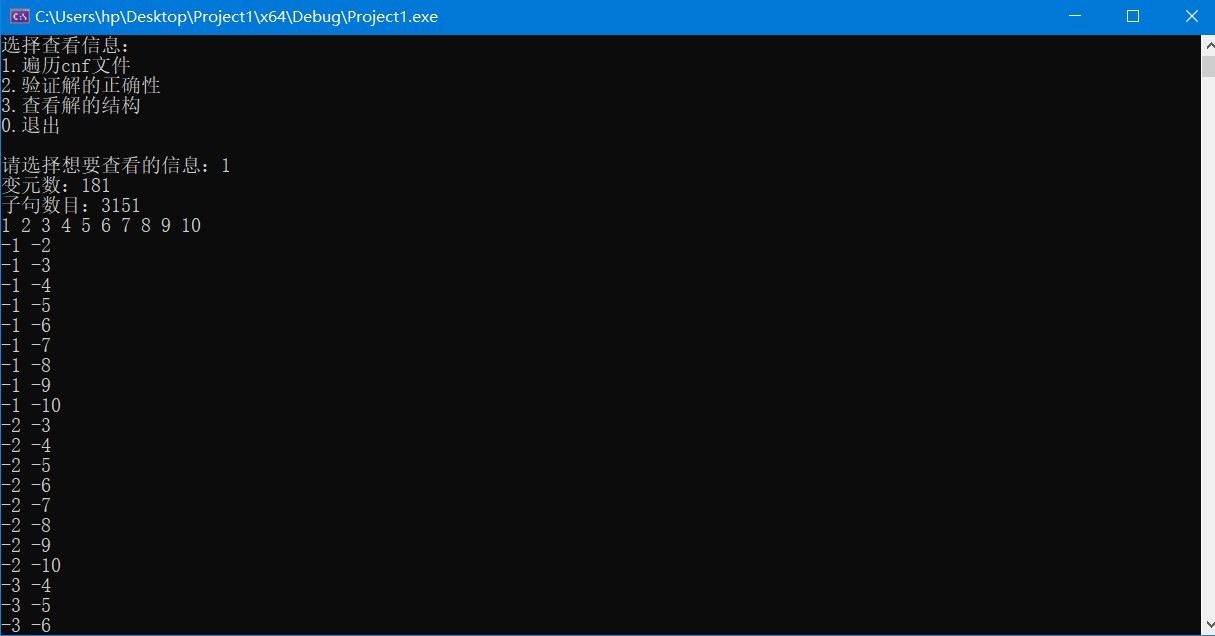


图 4-15 遍历 CNF 文件视图

通过验证解的正确性对 CNF 文件能否正确求解进行测试，测试结果如下：

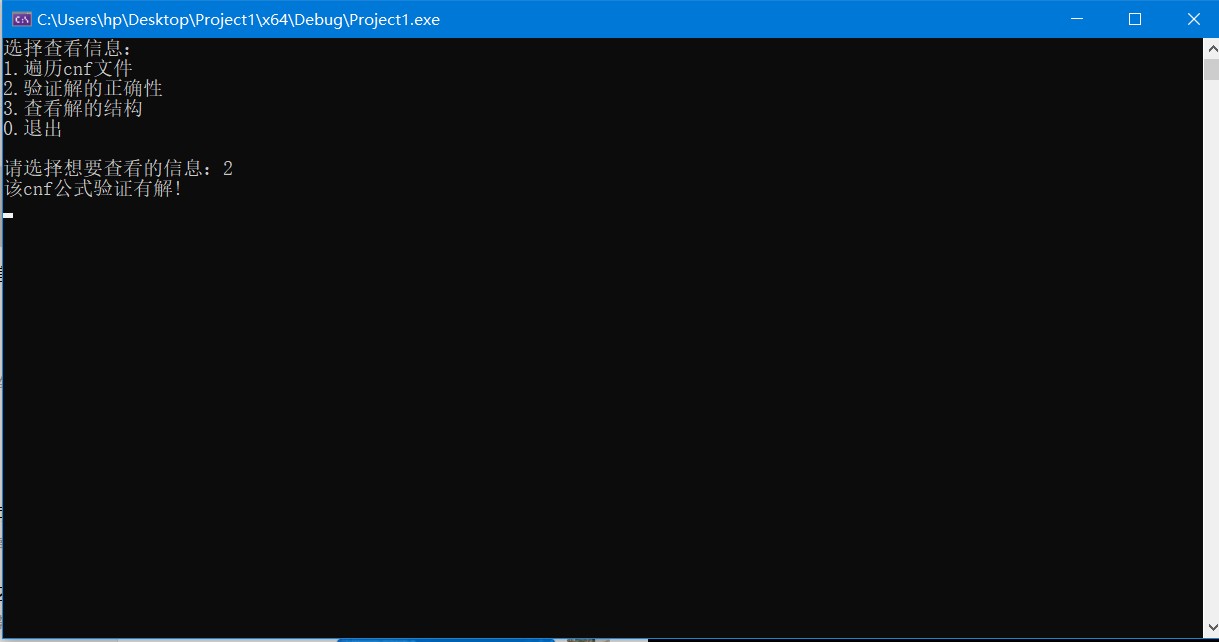


图 4-16 验证解的正确性视图

通过查看解的结构对CNF 文件能否得出解的结构进行测试，测试结果如下：

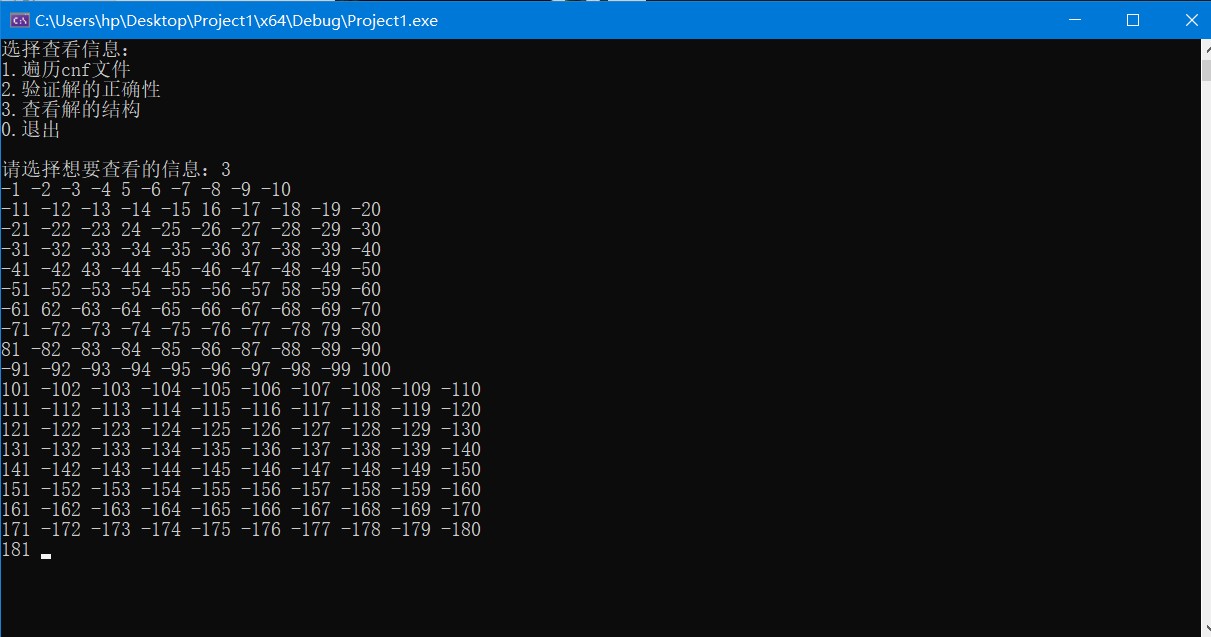


图 4-17 查看解的结构视图

通过查看RES 文件对CNF 文件能否保存解的信息进行测试，测试结果如下：

## 华 中 科 技 大 学 课 程 实 验 报 告

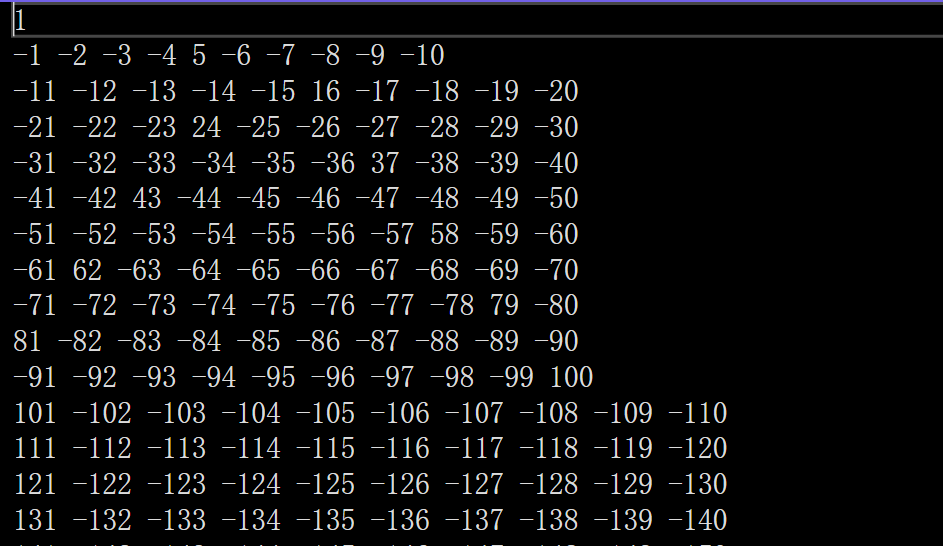


图 4-18 查看 RES 文件视图

#### DPLL 算法测试

说明：为了测试方便，在测试求解时间时，将这一模块抽取出来在Clion 编译器中进行测试。

输入CNF 文件后，通过不同的方案对 CNF 文件进行求解，同时比较求解的时间和优化率。

该模块的测试主要为获取求解时间进而计算优化率。计算优化率的公式如下：

*t − t*0 100% (4.1)

*×*

##### t

测试算例：**sud00009.cnf**

对应五种策略的求解结果如下：

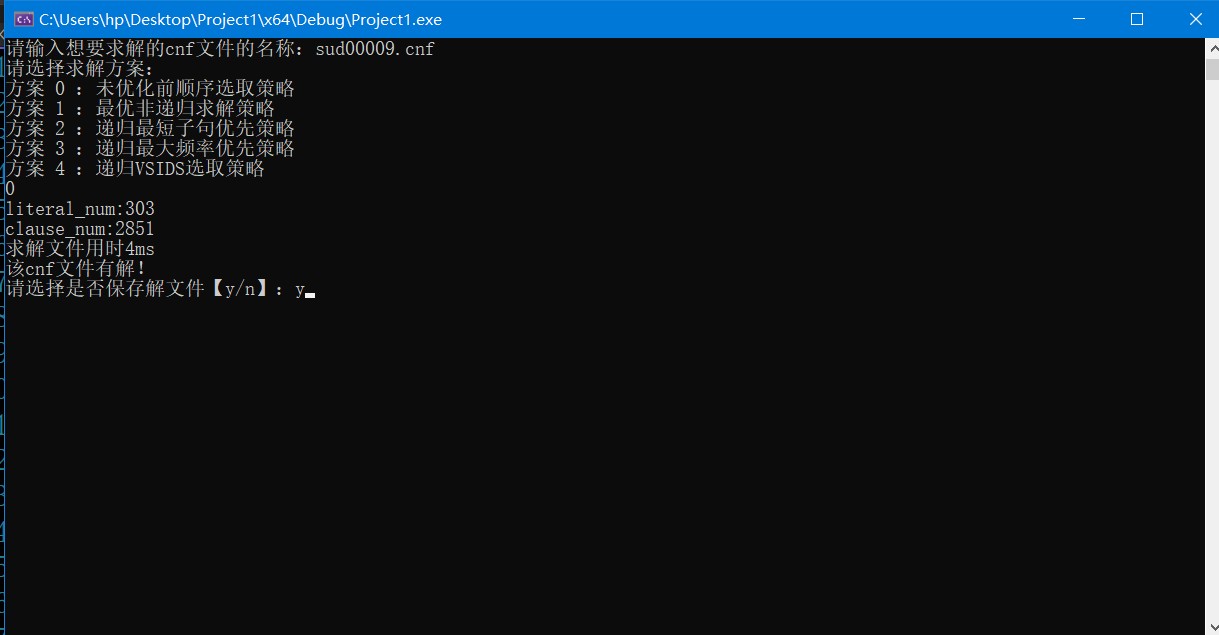


图 4-19 第一种策略求解视图

## 华 中 科 技 大 学 课 程 实 验 报 告

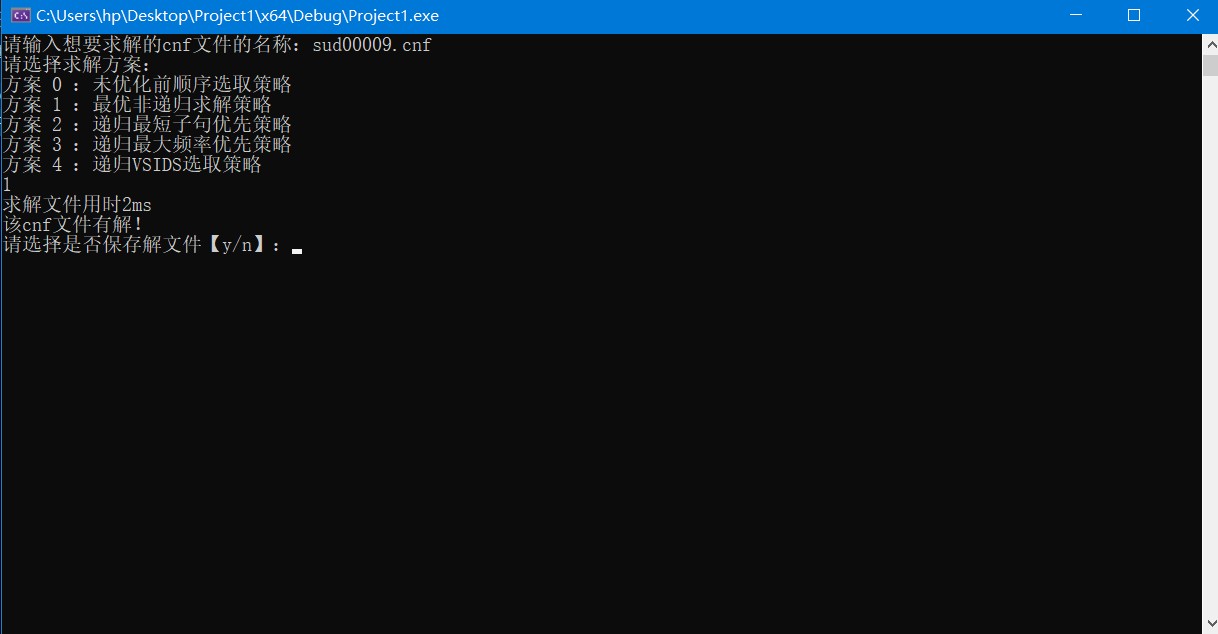


图 4-20 第二种策略求解视图

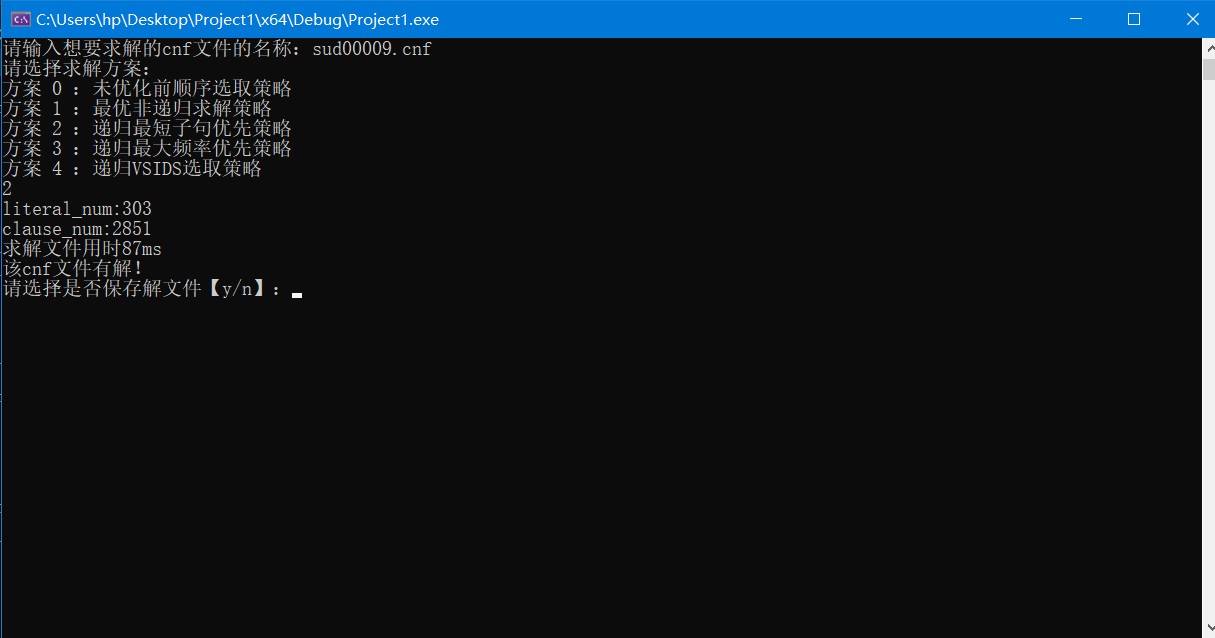


图 4-21 第三种策略求解视图

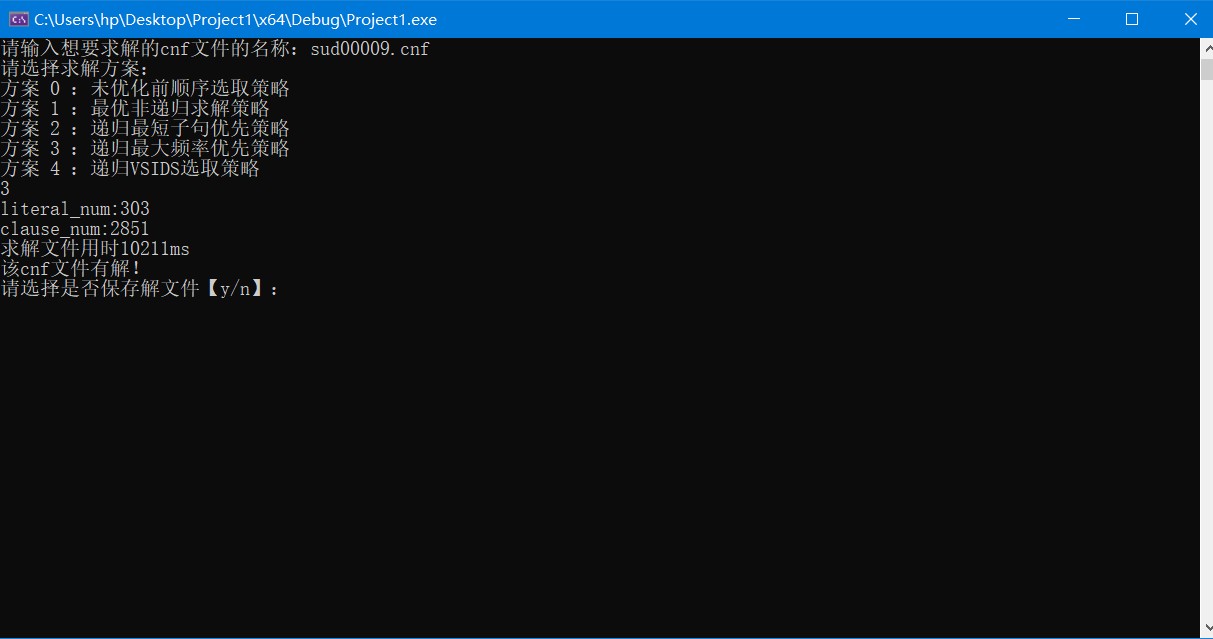


图 4-22 第四种策略求解视图

## 华 中 科 技 大 学 课 程 实 验 报 告

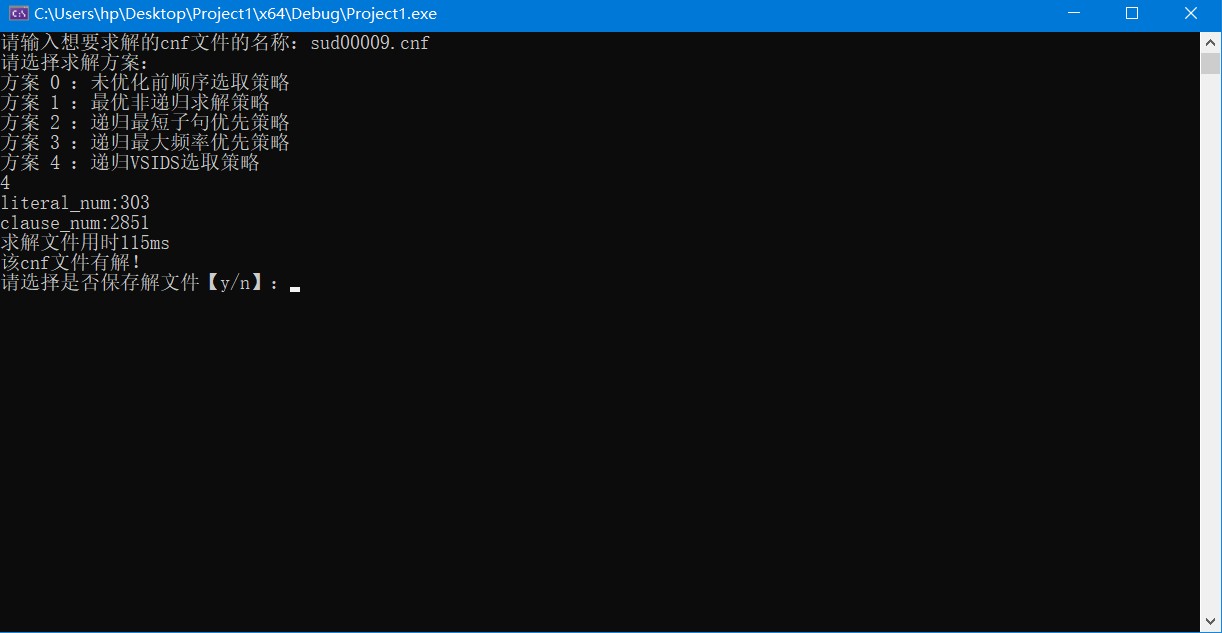


图 4-23 第五种策略求解视图

**CNF** 算例测试总结表格

不少于 18 个 SAT 算例，其中可满足的算例不少于 15 个，不满足的算例不少于 3 个，大中小算例各占三分之一。对算例规模的要求为：小型算例变元数为100 个左右；中型算例变元数介于 200-500 个；大型算例变元数 600 个以上。

小型算例部分

|  |  |  |
| --- | --- | --- |
| 表 4-1 算例性质 |  | |
| 算例名称 变元数 | 子句数 | 子句数/变元数 |
| problem2-50.cnf 50 | 80 | 1.6 |
| problem3-100.cnf 100 | 340 | 3.4 |
| problem11-100.cnf 100 | 600 | 6 |
| problem8-50.cnf 50 | 300 | 6 |
| 7cnf20\_90000\_90000\_7.shuffled-20.cnf 20 | 1532 | 76.6 |

表 4-2 算例求解结果

算例名称 优化前时间/ms 是否满足 优化后时间/ms（用 | 分隔） 最佳优化率/%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| problem2-50.cnf | 64 | 满足 | 1|0|7|0 | 100 |
| problem3-100.cnf | 754 | 满足 | 39|7|260|476 | 99.1 |
| problem11-100.cnf | 27 | 满足 | 1|5|14|6 | 96.3 |
| problem8-50.cnf | 1 | 满足 | 0|1|1|0 | 100 |
| 7cnf20\_90000\_90000\_7.shuffled-20.cnf | 49 | 满足 | 70|103|37|23 | 53.1 |

中型算例部分

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表 4-3 | 算例性质 | | | | |
| 算例名称 | | | 变元数 | 子句数 | 子句数/变元数 |
| bart17.shuffled-231.cnf | | | 231 | 1166 | 5.0 |
| problem12-200.cnf | | | 200 | 1200 | 6 |
| sud00021.cnf | | | 308 | 2911 | 9.5 |
| m-mod2c-rand3bip-sat-220-3.shuffled-as.sat05-2490-311.cnf | | | 311 | 2192 | 7.0 |
| sud00861.cnf | | | 297 | 2721 | 9.2 |
| 表 4-4 | 算例求解结果 | | | | |
| 算例名称 优化前时间/ms 是否满足 优化后时间/ms（用 | 分隔） 最佳优化率/% | | | | | |
| bart17.shuffled-231.cnf | >120000 | 满足 | 8|4|3|50 | | 100 |
| problem12-200.cnf | 10292 | 满足 | 515|1|6854|8182 | | 100 |
| sud00021.cnf | 52 | 满足 | 1|210|2784|251 | | 99.9 |
| m-mod2c-rand3bip-sat-220-3.shuffled-as.sat05-2490-311.cnf | >120000 | 满足 | 7983 | | 100 |
| sud00861.cnf | 19 | 满足 | 2|25|2580|115 | | 89.5 |

|  |  |  |
| --- | --- | --- |
| 大型算例部分  表 4-5 算例性质 |  | |
| 算例名称 变元数 | 子句数 | 子句数/变元数 |
| ec-iso-ukn009.shuffled-as.sat05-3632-1584.cnf 1584 | 16587 | 10.5 |
| eh-dp04s04.shuffled-1075.cnf 1075 | 3152 | 2.9 |
| e-par32-3.shuffled-3176.cnf 3176 | 10297 | 3.2 |

表 4-6 算例求解结果

算例名称 优化前时间/ms 是否满足 优化后时间/ms（用| 分隔） 最佳优化率/%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ec-iso-ukn009.shuffled-as.sat05-3632-1584.cnf | >120000 | 满足 | 659 | 100 |
| eh-dp04s04.shuffled-1075.cnf | >120000 | 满足 | 21 | 100 |
| e-par32-3.shuffled-3176.cnf | >120000 | 满足 | 1855 | 100 |

说明：由于处理器的限制，仅能跑出三个大型算例。

不满足算例部分

## 华 中 科 技 大 学 课 程 实 验 报 告

表 4-7 算例性质

|  |  |  |  |
| --- | --- | --- | --- |
| 算例名称 | 变元数 | 子句数 | 子句数/变元数 |
| php-010-008.shuffled-as.sat05-1171.cnf | 80 | 370 | 4.6 |
| u-dp04u03.shuffled-825.cnf | 825 | 2411 | 2.9 |
| u-problem7-50.cnf | 50 | 100 | 2 |
| u-5cnf\_3900\_3900\_060.shuffled-60.cnf | 60 | 936 | 15.6 |
| u-5cnf\_3500\_3500\_30f1.shuffled-30.cnf | 30 | 420 | 14 |

表 4-8 算例求解结果

算例名称 优化前时间/ms 是否满足 优化后时间/ms（用 | 分隔） 最佳优化率/%

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| php-010-008.shuffled-as.sat05-1171.cnf | | | 2666 | 不满足 | 856|1613|1205|2522 | 71.8 |
| u-dp04u03.shuffled-825.cnf | | | 465475 | 不满足 | 70|14553|481 | 100 |
| u-problem7-50.cnf | | | 44 | 不满足 | 1|2|45|71 | 97.7 |
| u-5cnf\_3900\_3900\_060.shuffled-60.cnf | | | 102739 | 不满足 | 19431|10602|42769 | 89.7 |
| u-5cnf\_3500\_3500\_30f1.shuffled-30.cnf | | | 25 | 不满足 | 21|18|31|28 | 19.4 |
| 基准算例部分 | |  | |  |  | |
|  | | 表 4-9 | | 算例性质 |  | |
| 算例名称 | | 变元数 | | 子句数 | 子句数/变元数 | |
| ais10.cnf | | 181 | | 3151 | 17.4 | |
| sud00009.cnf | | 303 | | 2851 | 9.4 | |
|  | | 表 4-10 | | 算例求解结果 |  | |
| 算例名称 优化前时间/ms 是否满足 优化后时间/ms（用 | 分隔） 最佳优化率/% | | | | | | |
| ais10.cnf | 1430 | 满足 | | 20|1831|3090|2068 | | 98.6 |
| sud00009.cnf | 10 | 满足 | | 0|40|120|60 | | 100 |

#### 数独游戏求解及简易游玩模块测试

首先简介一下数独部分：基于挖洞法随机生成数独游戏盘进行交互游玩，并且有选择难度的棋盘生成，游玩过后可以查看答案.

该模块的测试主要为以下四个部分：

## 华 中 科 技 大 学 课 程 实 验 报 告

* + - 1. 生成双数独游戏
      2. 进行数字填入，包括正确填入和错误填入
      3. 获取数独求解时间

生成双数独游戏

首先进行难度的选择。

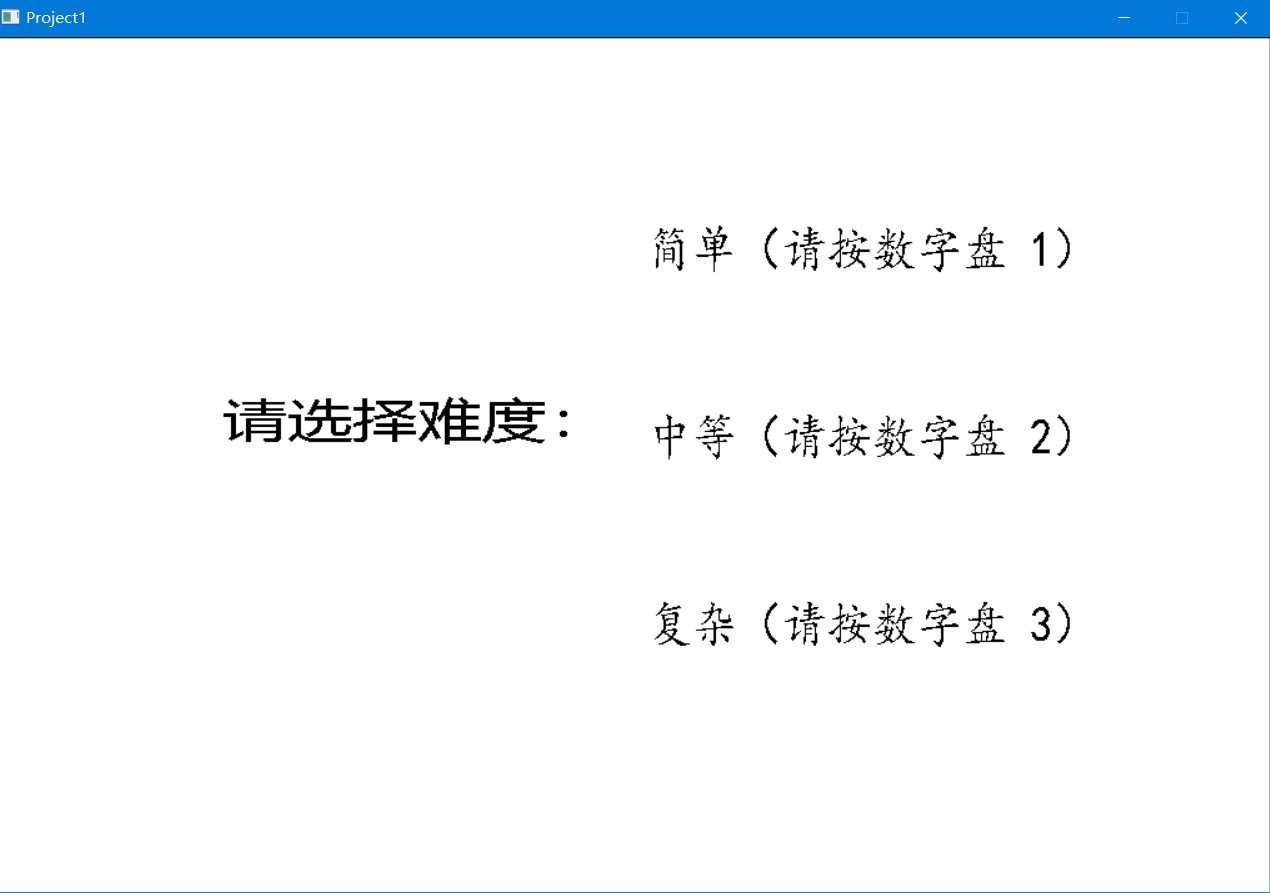


图 4-24 难度选择页面

通过正确输入难度序号可以成功进行选择，进入数独游戏界面。

进行数字填入，包括正确填入和错误填入

正确输入时可以看到，数字可以填入并且正确显示。

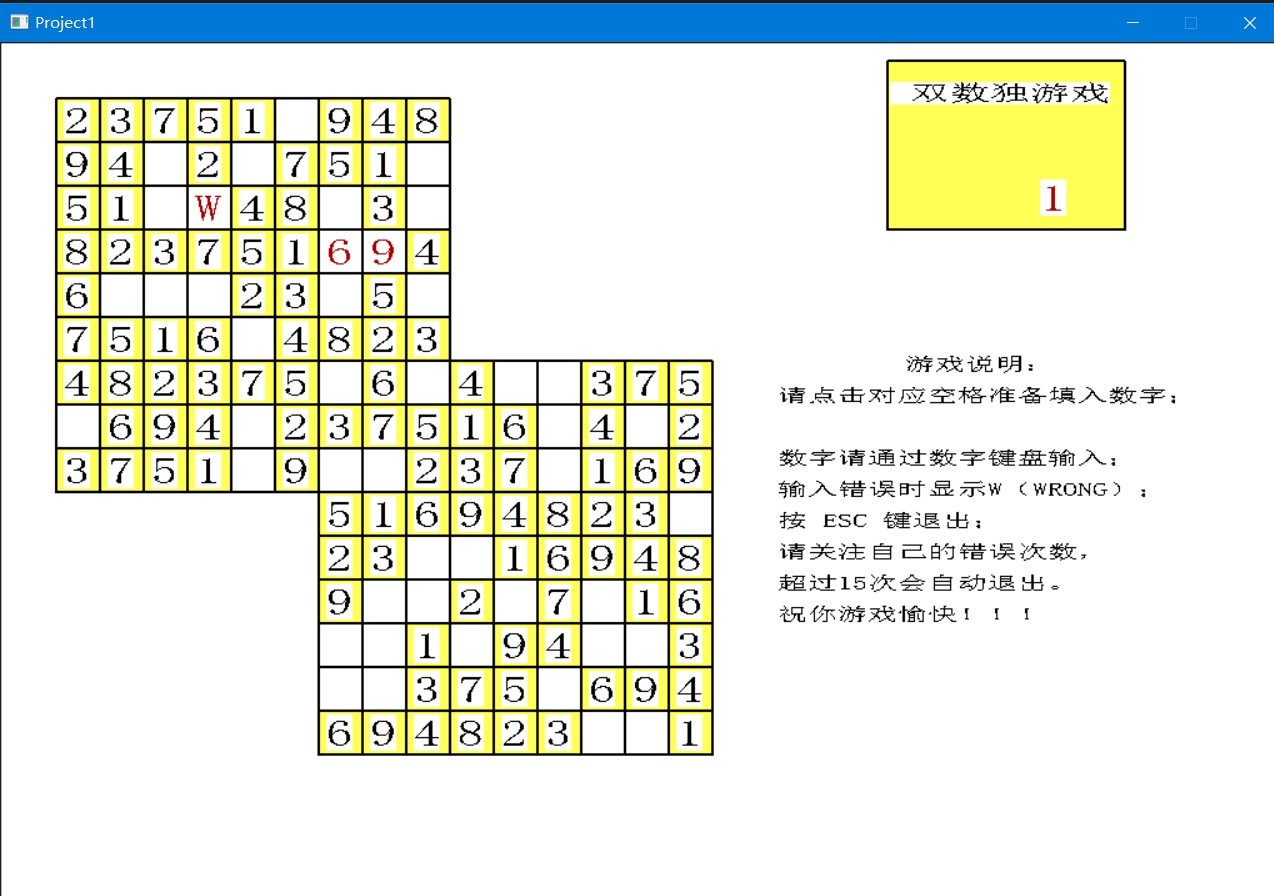


图 4-25 正确选择情形页面

错误输入时可以看到，数字错误输入时会显示 W，可以重新填入，但是错误次数累加。

## 华 中 科 技 大 学 课 程 实 验 报 告

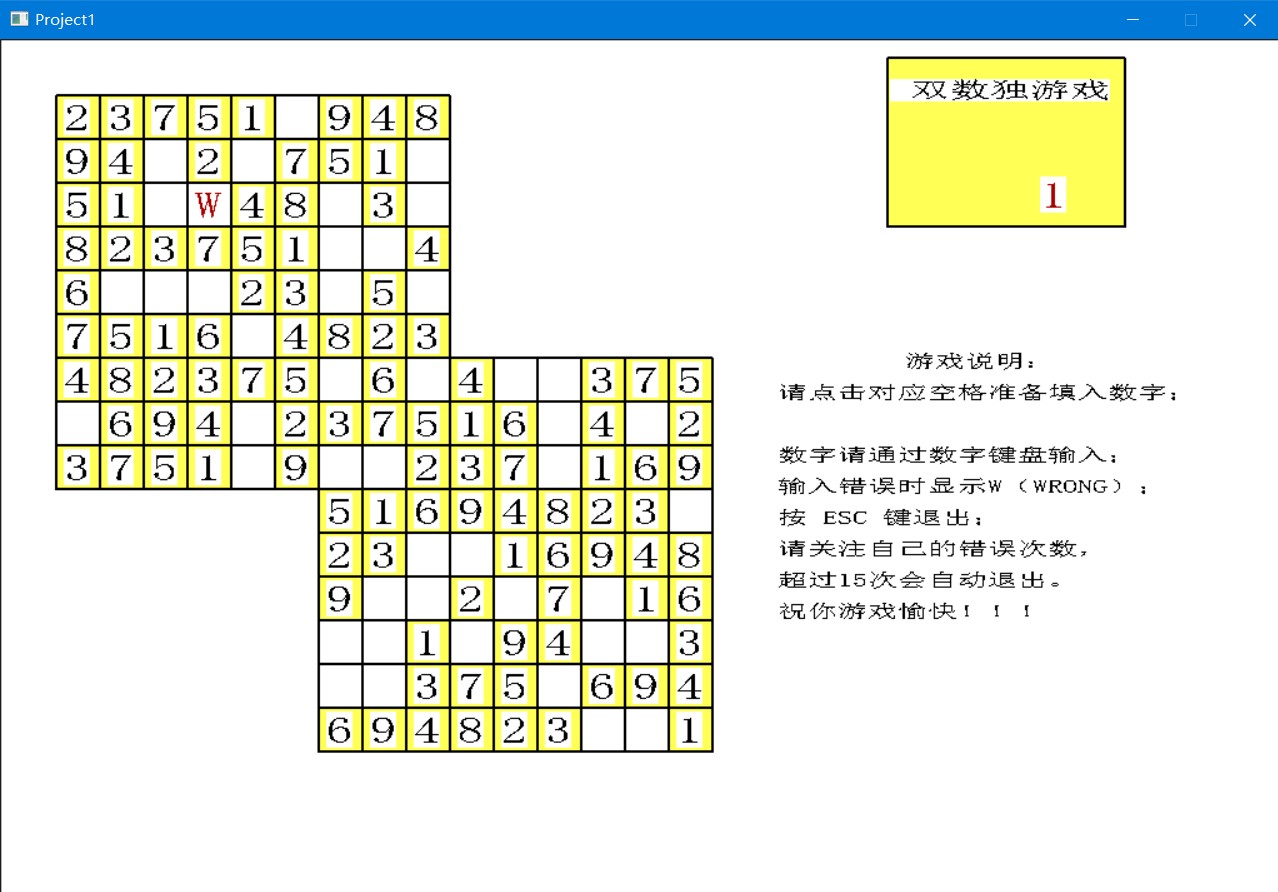


图 4-26 错误选择情形页面

获取数独求解时间

生成数独的求解时间如下：

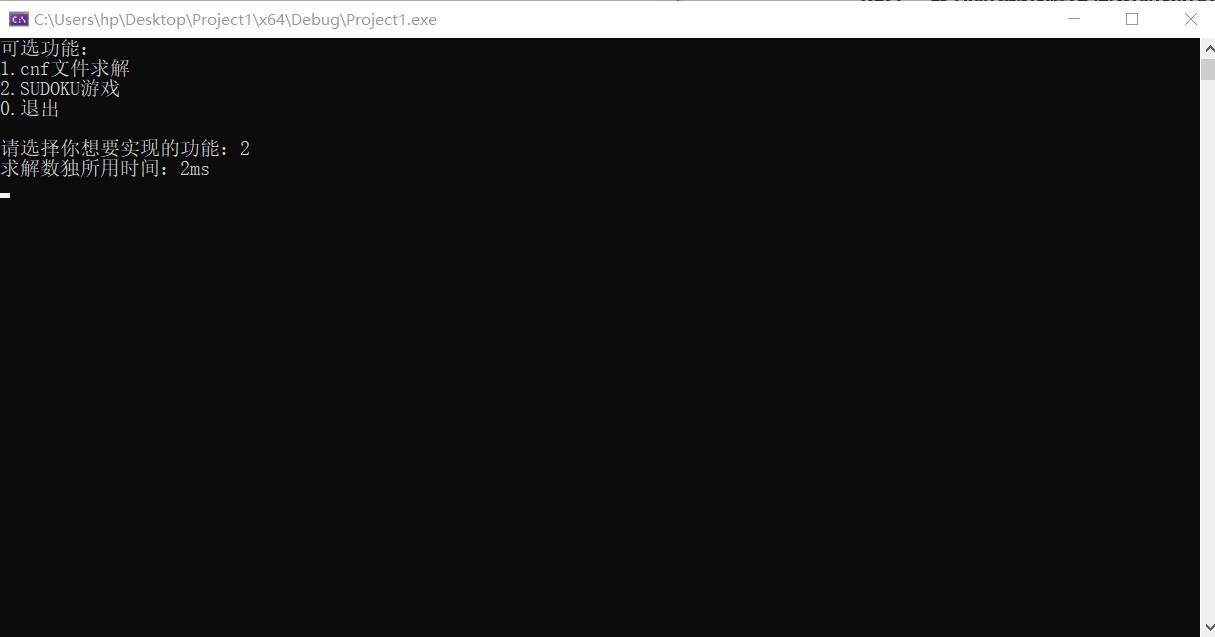


图 4-27 数独求解时间页面

由于 SAT 求解器对于数独规约后的CNF 文件良好的求解能力，求解时间在10ms 以内。

华 中 科 技 大 学 课 程 实 验 报 告

# 5 总结与展望

### 全文总结

对自己的工作做个总结，主要工作如下：

* + 1. 实现了一个基于 DPLL 算法的 SAT 求解器；
    2. 通识多篇论文，了解 SAT 求解器的发展历程；
    3. 分别在变量决策阶段、单子句传播阶段和回溯阶段进行优化；
    4. 实现了一定程度上论文方法的复现，包括重启策略，冲突分析等；
    5. 完成了数独游戏的规约和求解，并通过 EasyX 实现了游戏的可交互性；
    6. 设计了测试方案并完成了多个算例的测试；
    7. 完善了各种部分的衔接，构建一个简易系统。

### 工作展望

在今后的研究中，围绕着如下几个方面开展工作：

1. 由于设备和算法原因，设计的 SAT 求解器较为低级。测试过程中，当变元数量较大时，对于有些大型算例可以求解但时间效率往往有点低下。在今后的工作中，希望可以得到更多的编程和项目经验；
2. 在 DPLL 算法实现中，最主要的优化方向是变元选取策略。因为变元选取策略没有最优化的结果，因此再下来还应该不断去寻找更好的选取策略。
3. 对于数据结构的选择有待完善，本次使用的链表结构存储各变元并进行相关操作。数据结构的选择的简单导致后续设计相应操作时较为复杂。接下来考虑使用懒惰数据结构进行更好地存储和操作。
4. 在数独游戏模块，主要采用了 EasyX 进行交互界面的设计，但仍然有很多的不足，接下来考虑使用 Qt 进行界面的设计。

华 中 科 技 大 学 课 程 实 验 报 告

# 6 体会

在本次课程设计的过程中我阅读了很多篇文献，提高了自己的编程能力，也认识到了自己的很多不足。以下是我按照完成整个课设的时间顺序写的一些体会：

1、由于是第一次完成项目，因此在最开始的时候会有些对自己的不自信，也查找了很多资料，阅读理解很多次任务书才能较为清晰的感受到大致的方向。然后就开始一个步骤一个步骤去完成，当不断上手过后，才开始有自信去不断重构，去完成这个项目。

1. 在实现基于 DPLL 的 SAT 求解器的过程中，由于自身代码能力和经验的限制，因此在实现 SAT 求解器的过程中遇到了很多 bug，开始时对于 bug 还是感到很无力的，但是在不断修改的过程中，我也开始不断提高自己修复 bug 的能力，最终面对错误也能够以正确的心态进行调整。
2. 在数独创建中，学习了挖洞法生成数独的算法，在学习过程中，开始了解如何真正学会一个算法，从理解到实现，最终应用到课程设计的实验中，我觉得自己对于算法的学习能力有了很大的成长。我相信在进一步的学习过程中，我一定能更快地掌握算法，同时也能更好地进行实现。
3. 最后就是整合为项目的问题，这个问题确实困扰了我很久。因为一开始使用的是Clion 编译器，Clion 对于项目的组织结构要求很高，需要极大的时间配置CMakeLists.txt，因此我在网上不断地进行学习；后来因为 EasyX 只兼容 Visual Studio，因此使用其作为最终的编译器来完成全部的项目。

总而言之，在这次课程设计中收获到了很多知识，在这个过程中，不仅要感谢纪老师的关心和解答，也要感谢许助教的帮助，同时也要感谢张晋铭同学对我思路的启发以及看文献的习惯。我相信我会在以后的学习中更加成长。

华 中 科 技 大 学 课 程 实 验 报 告

# 参考文献

1. 张健著. 逻辑公式的可满足性判定—方法、工具及应用. 科学出版社，2000
2. Carsten Sinz. Visualizing SAT Instances and Runs of the DPLL Algorithm.[J]. J. Autom. Reasoning,2007,39(2).
3. Tjark Weber. A sat-based sudoku solver. In 12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2005, pages 11–15, 2005.
4. M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang and S. Malik, ”Chaff: engi- neering an efficient SAT solver,” Proceedings of the 38th Design Automation Confer- ence (IEEE Cat. No.01CH37232), 2001, pp. 530-535, doi: 10.1145/378239.379017.
5. Robert Ganian and Stefan Szeider. Community Structure Inspired Algorithms for SAT and #SAT. SAT 2015, 223-237360
6. 薛源海, 蒋彪彬, 李永卓, 闫桂峰, 孙华飞. 基于“挖洞”思想的数独游戏生成算法 [J]. 数学的实践与认识,2009,39(21):1-7.

华 中 科 技 大 学 课 程 实 验 报 告

# 附录

文件名：define.h 功能：项目头文件

1

*/ /*

*/ / Created by hp on 2022 −09 −04.*

*/ /*

# i f n d e f SUMIT\_DEFINE\_H

# d e f i n e SUMIT\_DEFINE\_H

*/ /* 引入头文件

# i n c l u d e < g r a p h i c s . h> # i n c l u d e < c o n i o . h>

# i n c l u d e < i o s t r e a m > # i n c l u d e < c s t d i o >

# i n c l u d e < a l g o r i t h m > # i n c l u d e < l i s t >

# i n c l u d e < c s t r i n g > # i n c l u d e < s s t r e a m > # i n c l u d e <queue >

# i n c l u d e < v e c t o r > # i n c l u d e <map>

# i n c l u d e < f s t r e a m > # i n c l u d e < s e t >

# i n c l u d e < b i t s e t > # i n c l u d e <cmath >

# i n c l u d e <unordered\_map > # i n c l u d e <random >

# i n c l u d e < ctime >

# i n c l u d e < s y s / utime . h>

*/ /* 引用图形库头文件

*/ /* 有效定义

# d e f i n e TRUE 1 # d e f i n e FALSE 0

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

## 华 中 科 技 大 学 课 程 实 验 报 告

1. # d e f i n e OK 1
2. # d e f i n e ERROR 0
3. # d e f i n e INFEASTABLE −1
4. # d e f i n e INCREASEMENT 100
5. # d e f i n e MAXN 0 x7FFFFF
6. # d e f i n e \_CRT\_SECURE\_NO\_WARNINGS 38
7. s t a t i c s t r u c t t i m e v a l s t ;
8. s t a t i c s t r u c t t i m e v a l ed ;
9. s t a t i c d o u b l e t i m e \_ t o t a l ; 42

43

1. */ /* 结构体相互引用交叉定义
2. s t r u c t Clause\_Node ;
3. s t r u c t L i t e r a l \_ N o d e ;
4. t y p e d e f s t r u c t Clause\_Node Clause\_Node , \* C l a u s e \_ L i s t ;
5. t y p e d e f s t r u c t L i t e r a l \_ N o d e L i t e r a l \_ N o d e , \* L i t e r a l \_ L i s t ;
6. t y p e d e f i n t s t a t u s ; 50
7. */\** 定义子句链式结构节点 *\*/*
8. s t r u c t Clause\_Node {
9. i n t number ; */ /* 定义子句中的文字数， 可判定是否为单子句
10. i n t t a g ; */ /* 定义字句是否被处理
11. L i t e r a l \_ L i s t f i r s t \_ l i t e r a l ; */ /* 定义文字节点， 指向第一文字
12. i n t w a t c h \_ l i t e r a l [ 2 ] ; */ /* 定义被监视节点
13. s t r u c t Clause\_Node\* n e x t \_ c l a u s e ; */ /* 定义字句节点， 指向下一子句

58 } ;

59

1. */\** 定义文字链式结构节点 *\*/*
2. s t r u c t L i t e r a l \_ N o d e {

|  |  |  |
| --- | --- | --- |
| 62 | i n t l i t | e r a l ; */ /* 定义整型文字 |
| 63 | i n t t a g | ; */ /* 定义文字是否被处理 |
| 64 | s t r u c t | L i t e r a l \_ N o d e\* n e x t \_ l i t e r a l ; */ /* 定义下一文字节点 |
| 65 | s t r u c t | L i t e r a l \_ N o d e\* n e x t \_ s a m e \_ l i t e r a l ; */ /* 定义下一相同节点 |
| 66 | s t r u c t | Clause\_Node\* head ; */ /* 定义该文字对应子句句首 |

## 华 中 科 技 大 学 课 程 实 验 报 告

67 } ;

68

1. */\** 定义*CNF*链式结构结点， 存储*CNF*信息 *\*/*
2. t y p e d e f s t r u c t Conjunctive\_Normal\_Form {
3. i n t l i t e r a l \_ n u m ; */ /* 定义文字数目
4. i n t clause\_num ; */ /* 定义子句数目
5. C l a u s e \_ L i s t f i r s t \_ c l a u s e ; */ /* 定义子句头节点， 指向*CNF*的第一个子句
6. } Conjunctive\_Normal\_Form , \* C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t ; 75

76 t y p e d e f s t r u c t Argue Value {

|  |  |  |
| --- | --- | --- |
| 77 | i n t | t a g ; */ /* 是否被判断真假 |
| 78 | i n t | i s \_ v a l u e ; */ /* 判断真值情况 |
| 79 | i n t | dep ; */ /* 栈的深度 |
| 80 | i n t | s c o r e ; */ /* 该变量的得分 |
| 81 | i n t | num ; */ /* 文字总数 |
| 82 | i n t | pos ; */ /* 正文字数目 |
| 83 | i n t | nev ; */ /* 负文字数目 |

1. L i t e r a l \_ L i s t f i r s t ; */ /* 首子句指针
2. } Argue Value ; 86
3. */\*CNF*文件读取处理函数 *\*/*
4. s t a t u s D e s t r o y C l a u s e ( C l a u s e \_ L i s t& s e n t e n c e ) ; */ /* 定义 *Destroy Clause* 函数， 销毁子句结点
5. s t a t u s D e s t r o y L i t e r a l ( L i t e r a l \_ L i s t& word ) ; */ /* 定义 *D e s t r o y L i t e r a l* 函数， 销毁文字结点
6. s t a t u s DestroyCNF ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义

*DestroyCNF* 函数， 销毁*CNF*文件

91

1. */ / func\_DPLL1* 所用函数
2. s t a t u s CreateCNF ( FILE\* fp , C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /*

定义 *CreateCNF* 函数， 创建*CNF*链表结果

1. s t a t u s C r e a t e C l a u s e ( FILE\* fp , C l a u s e \_ L i s t& s e n t e n c e , i n t f i r s t ) ;

*/ /* 定义 *Create Clause* 函数， 创建子句链表

1. s t a t u s D e l e t e L i t e r a l ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , i n t

## 华 中 科 技 大 学 课 程 实 验 报 告

l i t e r a l ) ; */ /* 定义 *D e l e t e L i t e r a l* 函数， 删除 *c n f* 结构中所有的 *l i t e r a l* 文字

1. s t a t u s R e c o v e r L i t e r a l ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , i n t l i t e r a l ) ; */ /* 定义 *R e c o v e r L i t e r a l* 函数， 回复特定 *l i t e r a l* 文字
2. s t a t u s u p d a t e \_ s t o r e v a l u e ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定

义 *u p d a t e \_ s t o r e v a l u e* 函数， 更新存储文字

1. s t a t u s u p d a t e \_ s t o r e v a l u e 2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义 *u p d a t e \_ s t o r e v a l u e* 函数， 更新存储文字
2. s t a t u s DPLL1 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t cnf , i n t op ) ; */ /* 定义

*DPLL1* 函数， 作为处理 *c n f* 文件的第一个*DPLL* 文件

1. i n t S e l e c t \_ L i t e r a l \_ D P L L 0 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义 *S e l e c t \_ L i t e r a l* 函数， 变量决策策略
2. i n t S e l e c t \_ L i t e r a l \_ D P L L 1 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义 *S e l e c t \_ L i t e r a l* 函数， 变量决策策略
3. i n t S e l e c t \_ L i t e r a l \_ D P L L 2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义 *S e l e c t \_ L i t e r a l* 函数， 变量决策策略
4. i n t S e l e c t \_ L i t e r a l \_ D P L L 3 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义 *S e l e c t \_ L i t e r a l* 函数， 变量决策策略
5. s t a t u s D P L L 1 \_ P a r t i t i o n ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , i n t l i t e r a l , i n t op ) ; */ /* 定义 *DPLL 1 \_Partition* 函数， 变量分裂规则
6. s t a t u s DPLL1\_Recover ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , i n t

l i t e r a l ) ; */ /* 定义 *DPLL1\_Recover* 函数， 变量回溯规则

1. C l a u s e \_ L i s t H as U n i t Cl a us e ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义 *Has Unit Clause* 函数， 评估*CNF*是否含有单子句
2. s t a t u s p r i n t ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义 *p r i n t* 函数， 输出真值表， 用于验证和调试
3. */ / func\_DPLL2* 所用函数
4. s t a t u s Create CNF2 ( FILE\* fp , C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /*

定义 *CreateCNF* 函数， 创建*CNF*链表结果

1. s t a t u s C r e a t e C l a u s e 2 ( FILE\* fp , C l a u s e \_ L i s t& s e n t e n c e , i n t f i r s t ) ;

*/ /* 定义 *Create Clause* 函数， 创建子句链表

1. s t a t u s D e s t r o y C l a u s e 2 ( C l a u s e \_ L i s t& s e n t e n c e ) ; */ /* 定义 *Destroy Clause*

函数， 销毁子句结点

## 华 中 科 技 大 学 课 程 实 验 报 告

1. s t a t u s D e s t r o y L i t e r a l 2 ( L i t e r a l \_ L i s t& word ) ; */ /* 定义 *D e s t r o y L i t e r a l* 函数， 销毁文字结点
2. s t a t u s DestroyCNF2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义

*DestroyCNF* 函数， 销毁*CNF*文件

1. s t a t u s p r i n t 2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义 *p r i n t* 函数， 输出真值表， 用于验证和调试
2. s t a t u s DPLL2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) ; */ /* 定义*DPLL1* 函数， 作为处理 *c n f* 文件的第一个*DPLL* 文件
3. s t a t u s c h e c k \_ p r i n t 2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) ; */ /* 定义

*c h e c k \_ p r i n t 2* 函数， 进行检查

1. s t a t u s Create Sudoku ( ) ; */ /* 定义 *Create Sudoku* 函数， 创建数独终盘
2. s t a t u s Dig\_Hole\_Easy ( i n t d i g ) ; */ /* 定义 *Dig\_Hole\_Easy* 函数， 挖洞法生成数独
3. s t a t u s s u d o k u s a t ( i n t m, i n t n , C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) ;

*/ /* 定义 *s udokusat* 函数， 求解数独问题

1. s t a t u s S u d o k u F i n a l P r i n t ( voi d ) ; */ /* 定义 *Su d o k u F i n a l P r i n t* 函数， 输出数独游戏
2. s t a t u s DPLL2\_SUDOKU( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t cnf , i n t t [ ] , i n t t o p ) ; */ /* 定义*DPLL2\_SUDOKU*函数， 调用*SAT* 求解器
3. i n t p r i n t \_ t i m e ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) ; */ /* 定义 *p r i n t \_ t i m e*

函数， 输出求解时间

|  |  |  |
| --- | --- | --- |
| 126 |  | |
| 127 | voi d | s o l v e ( i n t op ) ; */ /* 定义 *s o l v e* 函数， 输出交互界面 |
| 128 | voi d | s o l v e \_ c n f ( ) ; */ /* 定义 *s o l v e \_ c n f* 函数， 输出 *c n f* 文件求解交互界面 |
| 129 | voi d | s o l v e \_ s u d o k u ( ) ; */ /* 定义 *s o l v e\_sudoku* 函数， 输出数独求解交互界面 |
| 130 | voi d | s t o r e \_ d o c u m e n t ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , c h a r\* |

f i l e n a m e , i n t d , i n t t ime 0 ) ; */ /* 定义 *s t ore\_document* 函数， 存储文件

1. voi d t r a v e s e r \_ c n f ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) ; */ /* 定义

*t r a v e s e r \_ c n f* 函数， 进行*CNF*文件遍历

1. voi d p r o v e \_ c n f ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) ; */ /* 定义 *p r ove\_cnf*

函数， 进行验证

1. voi d show\_cnf ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) ; */ /* 定义 *show\_cnf* 函数， 输出 *c n f* 文件求解结果
2. # e n d i f */ / SUMIT\_DEFINE\_H*

## 华 中 科 技 大 学 课 程 实 验 报 告

文件名：func\_DPLL.cpp

功能：DPLL 框架 1 及多种变量选取策略的实现

1

# i n c l u d e *” d e f i n e . h”*

*/\** 定义有效的全局变量 *\*/*

Argue Value\* V a l u e L i s t ; */ /* 定义变元真值表

i n t b a c k t r a c k i n g \_ s t a c k \_ D P L L 1 [MAXN] , t op 1 ; */ /* 定义回溯栈

i n t c o n f l i c t \_ s t a c k [MAXN] , c t o p ; i n t ans 1 ;

*/ /* 定义*DPLL1* 函数， 作为处理 *c n f* 文件的第一个*DPLL* 文件

s t a t u s DPLL1 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t cnf , i n t op ) { ans 1 = 0 ;

t op 1 = 0 ; c t o p = 0 ;

C l a u s e \_ L i s t p ; L i t e r a l \_ L i s t q ;

w h i l e ( ( p = H a s U n i t C l a u se ( c n f ) ) ! = NULL) { q = p−> f i r s t \_ l i t e r a l ;

w h i l e ( q−> t a g == 1 ) q = q−> n e x t \_ l i t e r a l ; V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . t a g = 1 ;

i f ( q−> l i t e r a l > 0 )

V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . i s \_ v a l u e = 1 ;

e l s e

V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . i s \_ v a l u e = 0 ; b a c k t r a c k i n g \_ s t a c k \_ D P L L 1 [ t op 1 ++] = q−> l i t e r a l ; i f ( ! D e l e t e L i t e r a l ( cnf , q−> l i t e r a l ) )

r e t u r n FALSE ;

i f ( ans 1 == cnf −> clause\_num ) r e t u r n TRUE;

}

i n t p a r t \_ l i t e r a l ; i f ( op == 0 )

p a r t \_ l i t e r a l = S e l e c t \_ L i t e r a l \_ D P L L 0 ( c n f ) ;

e l s e i f ( op == 1 )

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

## 华 中 科 技 大 学 课 程 实 验 报 告

34

35

36

37

38

39

40

41

42

43

44

45 }

46

p a r t \_ l i t e r a l = S e l e c t \_ L i t e r a l \_ D P L L 1 ( c n f ) ; e l s e i f ( op == 2 )

p a r t \_ l i t e r a l = S e l e c t \_ L i t e r a l \_ D P L L 2 ( c n f ) ; e l s e i f ( op == 3 )

p a r t \_ l i t e r a l = S e l e c t \_ L i t e r a l \_ D P L L 3 ( c n f ) ; i f ( D P L L 1 \_ P a r t i t i o n ( cnf , p a r t \_ l i t e r a l , op ) )

r e t u r n TRUE;

DPLL1\_Recover ( cnf , p a r t \_ l i t e r a l ) ;

i f ( D P L L 1 \_ P a r t i t i o n ( cnf , − p a r t \_ l i t e r a l , op ) ) r e t u r n TRUE;

r e t u r n FALSE ;

1. */ /* 定义 *DPLL 1 \_Partition* 函数， 变量分裂规则
2. s t a t u s D P L L 1 \_ P a r t i t i o n ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , i n t l i t e r a l , i n t op ) {
3. V a l u e L i s t [ abs ( l i t e r a l ) ] . t a g = 1 ;
4. i f ( l i t e r a l > 0 )
5. V a l u e L i s t [ abs ( l i t e r a l ) ] . i s \_ v a l u e = 1 ;
6. e l s e
7. V a l u e L i s t [ abs ( l i t e r a l ) ] . i s \_ v a l u e = 0 ;
8. b a c k t r a c k i n g \_ s t a c k \_ D P L L 1 [ t op 1 ++] = l i t e r a l ;
9. i f ( ! D e l e t e L i t e r a l ( cnf , l i t e r a l ) ) {
10. c o n f l i c t \_ s t a c k [ c t o p ++] = l i t e r a l ;
11. r e t u r n FALSE ; 58 }
12. i f ( ans 1 == cnf −> clause\_num ) r e t u r n TRUE;
13. C l a u s e \_ L i s t p ;
14. L i t e r a l \_ L i s t q ;
15. w h i l e ( ( p = H a s U n i t C l a u se ( c n f ) ) ! = NULL) {
16. q = p−> f i r s t \_ l i t e r a l ;
17. w h i l e ( q && q−> t a g ) q = q−> n e x t \_ l i t e r a l ;
18. V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . t a g = 1 ;
19. i f ( q−> l i t e r a l > 0 )
20. V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . i s \_ v a l u e = 1 ;

## 华 中 科 技 大 学 课 程 实 验 报 告

1. e l s e

69

V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . i s \_ v a l u e = 0 ;

70

71

72

73

74

75

76 }

b a c k t r a c k i n g \_ s t a c k \_ D P L L 1 [ t op 1 ++] = q−> l i t e r a l ; i f ( ! D e l e t e L i t e r a l ( cnf , q−> l i t e r a l ) ) {

c o n f l i c t \_ s t a c k [ c t o p ++] = q−> l i t e r a l ; r e t u r n FALSE ;

}

i f ( ans 1 == cnf −> clause\_num ) r e t u r n TRUE;

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92 }

93

i n t p a r t \_ l i t e r a l ; i f ( op == 0 )

p a r t \_ l i t e r a l = S e l e c t \_ L i t e r a l \_ D P L L 0 ( c n f ) ; e l s e i f ( op == 1 )

p a r t \_ l i t e r a l = S e l e c t \_ L i t e r a l \_ D P L L 1 ( c n f ) ; e l s e i f ( op == 2 )

p a r t \_ l i t e r a l = S e l e c t \_ L i t e r a l \_ D P L L 2 ( c n f ) ; e l s e i f ( op == 3 )

p a r t \_ l i t e r a l = S e l e c t \_ L i t e r a l \_ D P L L 3 ( c n f ) ; i f ( D P L L 1 \_ P a r t i t i o n ( cnf , p a r t \_ l i t e r a l , op ) )

r e t u r n TRUE;

DPLL1\_Recover ( cnf , p a r t \_ l i t e r a l ) ;

i f ( D P L L 1 \_ P a r t i t i o n ( cnf , − p a r t \_ l i t e r a l , op ) ) r e t u r n TRUE;

r e t u r n FALSE ;

1. */ /* 定义 *DPLL1\_Recover* 函数， 变量回溯规则
2. s t a t u s DPLL1\_Recover ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , i n t l i t e r a l ) {
3. i n t t m p \_ l i t e r a l ;
4. w h i l e ( ( t m p \_ l i t e r a l = b a c k t r a c k i n g \_ s t a c k \_ D P L L 1 [ −− t o p 1 ] ) ! = l i t e r a l ) {
5. V a l u e L i s t [ abs ( t m p \_ l i t e r a l ) ] . t a g = 0 ;
6. R e c o v e r L i t e r a l ( cnf , t m p \_ l i t e r a l ) ;

100 }

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 101 | R e c o v e r L i t e r a l ( cnf , t m p \_ l i t e r a l ) ; | |  |
| 102 | r e t u r n OK; | |
| 103 | } | |
| 104 |  | |
| 105 | */ /* 定义 *R e c o v e r L i t e r a l* 函数， 回复特定 *l i t e r a l* 文字 | |
| 106 | s t a t u s R e c o v e r L i t e r a l ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , | | i n t |
|  | l i t e r a l ) { | | |
| 107 |  | i f ( ! c n f ) r e t u r n ERROR; | |
| 108 |  | L i t e r a l \_ L i s t tmp = V a l u e L i s t [ abs ( l i t e r a l ) ] . f i r s t ; | |
| 109 |  | w h i l e ( tmp ) { | |
| 110 |  | i f ( l i t e r a l == −tmp−> l i t e r a l ) { | |
| 111 |  | tmp−>head −>number ++; | |
| 112 |  | i f ( tmp−> t a g == −1) { | |
| 113 |  | tmp−> t a g = 0 ; | |
| 114 |  | b r e a k ; | |
| 115 |  | } | |
| 116 |  | tmp−> t a g = 0 ; | |
| 117 |  | } | |
| 118 |  | e l s e { | |
| 119 |  | tmp−>head −> t ag − −; | |
| 120 |  | i f ( ! tmp−>head −> t a g ) | |
| 121 |  | ans 1 − −; | |
| 122 |  | tmp−>head −>number ++; | |
| 123 |  | i f ( tmp−> t a g == −1) { | |
| 124 |  | tmp−> t a g = 0 ; | |
| 125 |  | b r e a k ; | |
| 126 |  | } | |
| 127 |  | tmp−> t a g = 0 ; | |
| 128 |  | } | |
| 129 |  | tmp = tmp−> n e x t \_ s a m e \_ l i t e r a l ; | |
| 130 |  | } | |
| 131 |  | r e t u r n OK; | |
| 132 | } |  | |
| 133 |  |  | |
| 134 | */ /* 定 | 义 *D e l e t e L i t e r a l* 函数， 删除 *c n f* 结构中所有的 *l i t e r a l* 文字 | |

华 中 科 技 大 学 课 程 实 验 报 告

135 s t a t u s D e l e t e L i t e r a l ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , i n t l i t e r a l ) {

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 136 | L i t e r a l \_ L i s t tmp = | | | | | V a l u e L i s t [ abs ( l i t e r a l ) ] . f i r s t ; | |
| 137 | w h i l e ( tmp ) { | | | | |  | |
| 138 | i f ( l i t e r a l == | | | | | −tmp−> l i t e r a l ) { | |
| 139 | tmp−> t a g = | | | | | 1 ; | |
| 140 tmp−>head −>number − −; | | | | | | | |
| 141 | i f | | | | ( ! tmp−>head −>number && ! tmp−>head −> t a g ) | | { |
| 142 |  | | | | tmp−> t a g = − 1 ; | |  |
| 143 |  | | | | r e t u r n FALSE ; | |  |
| 144 | } | | | |  | |  |
| 145 | } | | | |  | |  |
| 146 | e l s e { | | | |  | |  |
| 147 |  | tmp−>head −> t a g ++; | | | | | |
| 148 |  | i f ( tmp−>head −> t a g == 1 ) { | | | | | |
| 149 |  | ans 1 ++; | | | | | |
| 150 |  | } | | | | | |
| 151 |  | tmp−>head −>number − −; | | | | | |
| 152 |  | tmp−> t a g = 1 ; | | | | | |
| 153 |  | } | | | | | |
| 154 |  | tmp = tmp−> n e x t \_ s a m e \_ l i t e r a l ; | | | | | |
| 155 |  | } | | | | | |
| 156 |  | r e t u r n OK; | | | | | |
| 157 | } |  | | | | | |
| 158 |  |  | | | | | |
| 159 | */ /* 定 | 义 *Select\_Literal\_DPLL 0* 函数， 变量决策策略 | | | | | |
| 160 | i n t | S e l e c t \_ L i t e r a l \_ D P L L 0 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) { | | | | | |
| 161 |  | */ / c l o c k \_ t s t a r t = 0 , f i n i s h = 0 ; / /* 记录*DPLL* 函数调用的起始和 | | | | | |
|  |  | 终止时间 | | | |  |  |
| 162 | */ /* | |  | *i n t d u r a t i o n = 0 ; / /* 记录*SAT* 求解时间 | | | |
| 163 | */ /* | |  | *s t a r t = c l o c k ( ) ;* | | | |
| 164 | */ /* | |  | *p r i n t f (” 1 ”) ;* | | | |
| 165 | i n t | | i , | n u m \_ l i t e r a l ; | | | |
| 166 | f o r | | ( i | = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) { | | | |
| 167 |  | | i f | ( ! V a l u e L i s t [ i ] . t a g ) { | | | |

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 168 |  |  |  | r e t u r n i ; | | |
| 169 |  |  | } |  | | |
| 170 |  | } |  |  | | |
| 171 | } |  |  |  | | |
| 172 |  |  |  |  | | |
| 173 | */ /* 定义 *Select\_Literal\_DPLL 1* 函数， 变量决策策略 | | | |  |  |
| 174 | i n t S e l e c t \_ L i t e r a l \_ D P L L 1 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& | | | | c n f ) | { |
| 175 | */ / p r i n t f (” 1 ”) ;* | | | |  |  |
| 176 | i f ( ! c n f ) r e t u r n 0 ; | | | |  |  |
| 177 | C l a u s e \_ L i s t p = cnf −> f i r s t \_ c l a u s e , q = NULL; | | | |  |  |
| 178 | i n t n u m \_ l i t e r a l = cnf −> l i t e r a l \_ n u m ; | | | |  |  |
| 179 | u p d a t e \_ s t o r e v a l u e ( c n f ) ; | | | |  |  |
| 180 |  | | | |  |  |
| 181 | w h i l e ( p ) { | | | |  |  |
| 182 | i f ( p−> t a g | | ! p−>number ) { | | | |  |  |
| 183 | p = p−> n e x t \_ c l a u s e ; | | | |  |  |
| 184 | c o n t i n u e ; | | | |  |  |
| 185 | } | | | |  |  |
| 186 | i f ( p−>number < n u m \_ l i t e r a l ) { | | | |  |  |
| 187 | n u m \_ l i t e r a l = p−>number ; | | | |  |  |
| 188 | q = p ; | | | |  |  |
| 189 | } | | | |  |  |
| 190 | p = p−> n e x t \_ c l a u s e ; | | | |  |  |
| 191 | } | | | |  |  |
| 192 | L i t e r a l \_ L i s t m = q−> f i r s t \_ l i t e r a l , n = NULL; | | | |  |  |
| 193 |  | | | |  |  |
| 194 | i n t num\_count = 0 ; | | | |  |  |
| 195 | w h i l e (m) { | | | |  |  |
| 196 | i f (m−> t a g ) { | | | |  |  |
| 197 | m = m−> n e x t \_ l i t e r a l ; | | | |  |  |
| 198 | c o n t i n u e ; | | | |  |  |
| 199 | } | | | |  |  |
| 200 | i f ( V a l u e L i s t [ abs (m−> l i t e r a l ) ] . num > num\_count ) | | | | { |  |
| 201 | num\_count = V a l u e L i s t [ abs (m−> l i t e r a l ) ] . num ; | | | |  |  |
| 202 | n = m; | | | |  |  |

华 中 科 技 大 学 课 程 实 验 报 告

|  |  |
| --- | --- |
| 203 | } |
| 204 | m = m−> n e x t \_ l i t e r a l ; |
| 205 | } |
| 206 | i f ( V a l u e L i s t [ abs ( n−> l i t e r a l ) ] . pos >= V a l u e L i s t [ abs ( n−> |

207

208

209

210 } 211

l i t e r a l ) ] . nev )

r e t u r n abs ( n−> l i t e r a l ) ;

e l s e

r e t u r n − abs ( n−> l i t e r a l ) ;

1. */ /* 定义 *Select\_Literal\_DPLL 2* 函数， 变量决策策略
2. i n t S e l e c t \_ L i t e r a l \_ D P L L 2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) {
3. */ / c l o c k \_ t s t a r t = 0 , f i n i s h = 0 ; / /* 记录*DPLL* 函数调用的起始和终止时间

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 215 | */ /* | | *i n t d u r a t i o n = 0 ; / /* 记录*SAT* 求解时间 | |
| 216 | */ /* | | *s t a r t = c l o c k ( ) ;* | |
| 217 | */ /* | | *p r i n t f (” 1 ”) ;* | |
| 218 | i n t | | n u m \_ l i t e r a l = 0 , num = 0 , i ; | |
| 219 u p d a t e \_ s t o r e v a l u e ( c n f ) ; | | | | |
| 220 | f o r | | ( i | = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) { |
| 221 |  | | i f | ( ! V a l u e L i s t [ i ] . t a g && V a l u e L i s t [ i ] . num >= num ) { |
| 222 |  | |  | n u m \_ l i t e r a l = i ; |
| 223 |  | |  | num = V a l u e L i s t [ i ] . num ; |
| 224 |  | | } |  |
| 225 |  | } | | |
| 226 |  | i f ( V a l u e L i s t [ n u m \_ l i t e r a l ] . pos > V a l u e L i s t [ n u m \_ l i t e r a l ] . nev ) | | |
| 227 |  | r e t u r n n u m \_ l i t e r a l ; | | |
| 228 |  | e l s e | | |
| 229 |  | r e t u r n − n u m \_ l i t e r a l ; | | |
| 230 | } |  | | |
| 231 |  |  | | |
| 232 | */ /* 定 | 义 *Select\_Literal\_DPLL 3* 函数， 变量决策策略 | | |
| 233 | i n t | S e l e c t \_ L i t e r a l \_ D P L L 3 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) { | | |
| 234 |  | */ / p r i n t f (” 1 ”) ;* | | |
| 235 |  | i f ( ! c n f ) r e t u r n 0 ; | | |

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 236 | C l a u s e \_ L i s t p = cnf −> f i r s t \_ c l a u s e , q = NULL; | |  |
| 237 | i n t s c o r e = 0 , i , l i t e r a l ; | |
| 238 | u p d a t e \_ s t o r e v a l u e 2 ( c n f ) ; | |
| 239 | f o r ( i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) { | |
| 240 | i f ( ! V a l u e L i s t [ i ] . t a g && V a l u e L i s t [ i ] . s c o r e >= s c o r e ) | | { |
| 241 | s c o r e = V a l u e L i s t [ i ] . s c o r e ; | |  |
| 242 | l i t e r a l = i ; | |  |
| 243 | } | |  |
| 244 | } | |  |
| 245 | i f ( V a l u e L i s t [ l i t e r a l ] . pos > V a l u e L i s t [ l i t e r a l ] . nev ) | |  |
| 246 | r e t u r n l i t e r a l ; | |  |
| 247  248 | e l s e  r e t u r n − l i t e r a l ; | |  |
| 249 | } |  |  |
| 250 |  |  |  |
| 251 | */ /* 定义 *u p d a t e \_ s t o r e v a l u e 2* 函数， 更新存储文字 |  |  |
| 252 | s t a t u s u p d a t e \_ s t o r e v a l u e 2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& | c n f ) | { |
| 253 | i n t tmp0 ; |  |  |
| 254 | f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) { |  |  |
| 255 | V a l u e L i s t [ tmp0 ] . num = 0 ; |  |  |
| 256 | V a l u e L i s t [ tmp0 ] . nev = 0 ; |  |  |
| 257 | V a l u e L i s t [ tmp0 ] . pos = 0 ; |  |  |
| 258 | V a l u e L i s t [ tmp0 ] . s c o r e = 0 ; |  |  |
| 259 | } |  |  |
| 260 | i n t l i t e r a l ; |  |  |
| 261 | L i t e r a l \_ L i s t q , r ; |  |  |
| 262 | w h i l e ( c t o p ) { |  |  |
| 263 | l i t e r a l = c o n f l i c t \_ s t a c k [ −− c t o p ] ; |  |  |
| 264 |  |  |  |
| 265 | q = V a l u e L i s t [ abs ( l i t e r a l ) ] . f i r s t ; |  |  |
| 266 | w h i l e ( q ) { |  |  |
| 267 | i f ( q−>head −> t a g ) { |  |  |
| 268 | q = q−> n e x t \_ s a m e \_ l i t e r a l ; |  |  |
| 269 | c o n t i n u e ; |  |  |
| 270 | } |  |  |

|  |  |  |
| --- | --- | --- |
|  | 华 | 中 科 技 大 学 课 程 实 验 报 告 |
| 271 |  | r = q−>head −> f i r s t \_ l i t e r a l ; |
| 272 |  | w h i l e ( r ) { |
| 273 |  | i f ( r −> t a g ) { |
| 274 |  | r = r −> n e x t \_ l i t e r a l ; |
| 275 |  | c o n t i n u e ; |
| 276 |  | } |
| 277 |  | V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . s c o r e += ( cnf −> |

clause\_num / ( r −>head −>number \* r −>head −> number ) ) ;

1. r −> l i t e r a l > 0 ? V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . pos ++

: V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . nev ++;

1. r = r −> n e x t \_ l i t e r a l ;

280 }

281 q = q−> n e x t \_ s a m e \_ l i t e r a l ;

282 }

283 }

1. C l a u s e \_ L i s t p = cnf −> f i r s t \_ c l a u s e ;
2. w h i l e ( p ) {
3. i f ( p−> t a g ) {
4. p = p−> n e x t \_ c l a u s e ;
5. c o n t i n u e ;

289 }

1. q = p−> f i r s t \_ l i t e r a l ;
2. w h i l e ( q ) {
3. i f ( q−> t a g ) {
4. q = q−> n e x t \_ l i t e r a l ;
5. c o n t i n u e ;

295 }

1. V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . s c o r e += ( cnf −> c lause\_num

/ ( p−>number \* p−>number ) ) ;

1. i f ( V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . s c o r e > 100 ) {
2. f o r ( i n t i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) {
3. V a l u e L i s t [ i ] . s c o r e / = 1 0 ;

300 }

301 }

## 华 中 科 技 大 学 课 程 实 验 报 告

302 q−> l i t e r a l > 0 ? V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . pos ++ : V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . nev ++;

|  |  |  |  |
| --- | --- | --- | --- |
| 303 | q = q−> n e x t \_ l i t e r a l ; |  | |
| 304 | } |
| 305 | p = p−> n e x t \_ c l a u s e ; |
| 306 | } |
| 307 | r e t u r n OK; |
| 308 | } |
| 309 |  |
| 310 | */ /* 定义 *u p d a t e \_ s t o r e v a l u e* 函数， 更新存储文字 |
| 311 | s t a t u s u p d a t e \_ s t o r e v a l u e ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& | c n f ) | { |
| 312 | i n t tmp0 ; |  |  |
| 313 | f o r ( tmp0 = 0 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) { |  |  |
| 314 | V a l u e L i s t [ tmp0 ] . num = 0 ; |  |  |
| 315 | V a l u e L i s t [ tmp0 ] . nev = 0 ; |  |  |
| 316 | V a l u e L i s t [ tmp0 ] . pos = 0 ; |  |  |
| 317 | } |  |  |
| 318 | C l a u s e \_ L i s t p = cnf −> f i r s t \_ c l a u s e ; |  |  |
| 319 | L i t e r a l \_ L i s t q ; |  |  |
| 320 | w h i l e ( p ) { |  |  |
| 321 | i f ( p−> t a g ) { |  |  |
| 322 | p = p−> n e x t \_ c l a u s e ; |  |  |
| 323 | c o n t i n u e ; |  |  |
| 324 | } |  |  |
| 325 | q = p−> f i r s t \_ l i t e r a l ; |  |  |
| 326 | w h i l e ( q ) { |  |  |
| 327 | i f ( q−> t a g ) { |  |  |
| 328 | q = q−> n e x t \_ l i t e r a l ; |  |  |
| 329 | c o n t i n u e ; |  |  |
| 330 | } |  |  |
| 331 | V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . num++; |  |  |
| 332 | i f ( V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . num > 100 ) { |  |  |
| 333 | f o r ( i n t i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) | | { |
| 334 | V a l u e L i s t [ i ] . num / = 1 0 ; | |  |
| 335 | } | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 华 | | 中 科 技 大 学 课 程 实 验 报 告 |
| 336 |  | | } |
| 337 |  | | q−> l i t e r a l > 0 ? V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . pos ++ :  V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . nev ++; |
| 338 |  | q = q−> n e x t \_ l i t e r a l ; | |
| 339 |  | } | |
| 340 |  | p = p−> n e x t \_ c l a u s e ; | |
| 341 |  | } | |
| 342 |  | r e t u r n OK; | |
| 343 | } |  | |
| 344 |  |  | |
| 345 | */ /* 定 | 义 *p r i n t* 函数， 输出真值表， 用于验证和调试 | |

1. s t a t u s p r i n t ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) {
2. f o r ( i n t i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) {
3. p r i n t f ( *”{%d }”* , V a l u e L i s t [ i ] . t ag , V a l u e L i s t [ i ] . i s \_ v a l u e

) ;

352 } 353

}

p r i n t f ( *” \ n”* ) ; r e t u r n OK;

1. */ /* 定义 *CreateCNF* 函数， 创建*CNF*链表结果
2. s t a t u s CreateCNF ( FILE\* fp , C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) {
3. c h a r r e a d f i l e [ 2 0 ] ; */ /* 定义字符类型数组记录在文件中
4. i n t i , tmp ; */ /* 整型变量定义
5. C l a u s e \_ L i s t c l ause\_tmp 1 , c l a u s e \_ t m p 2 ; */ /* 定义临时结点变量
6. */ /* 初始化*CNF*
7. c n f = ( Conjunctive\_Normal\_Form \*) m a l l o c ( s i z e o f ( Conjunctive\_Normal\_Form ) ) ;
8. cnf −> f i r s t \_ c l a u s e = NULL; 363
9. */ / CNF*文件读取
10. w h i l e ( f s c a n f ( fp , *”%s”* , r e a d f i l e ) ! = EOF) { */ /* 循环读文件
11. i f ( s t r c m p ( r e a d f i l e , *”p”* ) == 0 ) */ /* 持续读取文件直至开始标志’ *p* ‘

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |
| --- | --- |
| 367 | b r e a k ; |
| 368 | } |
| 369 | w h i l e ( f s c a n f ( fp , *”%s”* , r e a d f i l e ) ! = EOF) { |
| 370 | i f ( s t r c m p ( r e a d f i l e , *” c n f ”* ) == 0 ) { */ /* 从文件中读到字符串“ |
|  | *c n f* ” |
| 371 | f s c a n f ( fp , *”%d”* , &cnf −> l i t e r a l \_ n u m ) ; */ /* 读取*CNF*文件变元 |

数并存入 *l i t e r a l \_ n u m*

1. f s c a n f ( fp , *”%d”* , &cnf −> clause\_num ) ; */ /* 读取*CNF*文件子句总数并存入 *clause\_num*
2. b r e a k ;

374 }

375 }

376

377 p r i n t f ( *” l i t e r a l \_ n u m :%d \ nclause\_num:%d \ n”* , cnf −> l i t e r a l \_ n u m , cnf −> c lause\_num ) ; */ /* 输出基本信息

378

1. */ /* 定义并初始化变元真值表
2. V a l u e L i s t = ( Argue Value \*) m a l l o c ( ( cnf −> l i t e r a l \_ n u m + 2 ) \*

s i z e o f ( Argue Value ) ) ;

|  |  |  |
| --- | --- | --- |
| 381 |  | i f ( ! V a l u e L i s t ) r e t u r n OVERFLOW; */ /* 没分配成功， 返回*OVERFLOW* |
| 382 |  | f o r ( i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) { |
| 383 |  | V a l u e L i s t [ i ] . i s \_ v a l u e = 0 ; |
| 384 |  | V a l u e L i s t [ i ] . t a g = 0 ; |
| 385 |  | V a l u e L i s t [ i ] . num = 0 ; |
| 386 |  | V a l u e L i s t [ i ] . nev = 0 ; |
| 387 |  | V a l u e L i s t [ i ] . pos = 0 ; |
| 388 |  | V a l u e L i s t [ i ] . s c o r e = 0 ; |
| 389 |  | V a l u e L i s t [ i ] . f i r s t = NULL; |
| 390 |  | } |
| 391 |  |  |
| 392 |  | */ /* 创建*CNF*链式结构 |
| 393 |  | i f ( cnf −> clause\_num ) { |
| 394 |  | i f ( f s c a n f ( fp , *”%d”* , &tmp ) ! = EOF && tmp ! = 0 ) { |
| 395 |  | */ /* 创建*CNF*子句的头指针 |
| 396 | */ /* | *p r i n t f (”%d ” , tmp ) ;* |

## 华 中 科 技 大 学 课 程 实 验 报 告

1. c l a u s e \_ t m p 1 = ( C l a u s e \_ L i s t ) m a l l o c ( s i z e o f ( Clause\_Node )

) ;

1. i f ( ! c l a u s e \_ t m p 1 ) r e t u r n OVERFLOW;
2. cnf −> f i r s t \_ c l a u s e = c l a u s e \_ t m p 1 ; */ /* 定义头节点
3. c l a u s e \_ t m p 2 = c l a u s e \_ t m p 1 ;
4. C r e a t e C l a u s e ( fp , c l ause\_tmp 1 , tmp ) ; */ /* 创建其对应子句链表

|  |  |
| --- | --- |
| 402 | c l ause\_tmp 1 −> n e x t \_ c l a u s e = NULL; |
| 403 | c l ause\_tmp 1 −> t a g = 0 ; |
| 404 | i = 2 ; |
| 405 | */ /* 创建完整的*CNF*子句链式结构 |
| 406 | w h i l e ( i ++ <= cnf −> clause\_num ) { |
| 407 | f s c a n f ( fp , *”%d”* , &tmp ) ; |
| 408 | */ / p r i n t f (”%d ” , tmp ) ;* |
| 409 | c l a u s e \_ t m p 1 = ( C l a u s e \_ L i s t ) m a l l o c ( s i z e o f ( |

Clause\_Node ) ) ;

410 i f ( ! c l a u s e \_ t m p 1 ) r e t u r n OVERFLOW;

411 C r e a t e C l a u s e ( fp , c l ause\_tmp 1 , tmp ) ; */ /* 创建其对应子句链表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 412 |  |  |  | c l ause\_tmp 2 −> n e x t \_ c l a u s e = c l a u s e \_ t m p 1 ; | |
| 413 |  |  |  | c l a u s e \_ t m p 2 = c l a u s e \_ t m p 1 ; | |
| 414 |  |  |  | c l ause\_tmp 2 −> t a g = 0 ; | |
| 415 |  |  |  | c l ause\_tmp 1 −> n e x t \_ c l a u s e = NULL; | |
| 416 |  |  | } |  | |
| 417 |  | } |  |  | |
| 418 |  | } |  |  | |
| 419 |  | r e t u r n | OK; |  | |
| 420 | } | | | |  |
| 421 |  | | | |  |
| 422 | */ /* 定义 *Create Clause* 函数， 创建子句链表 | | | |  |
| 423 | */ /* 输入： 文件指针， 子句链式结点， 首值 | | | |  |
| 424 | */ /* 输出： 状态 | | | |  |
| 425 | s t a t u s C r e a t e C l a u s e ( FILE\* fp , C l a u s e \_ L i s t& s e n t e n c e , i n t | | | | f i r s t ) { |
| 426 | i n t num = 1 , tmp ; */ /* 定义文字数目， 临时变量 | | | |  |
| 427 | i f ( ! f i r s t ) r e t u r n ERROR; | | | |  |

## 华 中 科 技 大 学 课 程 实 验 报 告

1. L i t e r a l \_ L i s t l i t e r a l \_ t m p 1 , l i t e r a l \_ t m p 2 ; */ /* 定义临时结点变量
2. l i t e r a l \_ t m p 1 = ( L i t e r a l \_ L i s t ) m a l l o c ( s i z e o f ( L i t e r a l \_ N o d e ) ) ;
3. i f ( ! l i t e r a l \_ t m p 1 ) r e t u r n OVERFLOW;
4. */ /* 创建文字链式结构头节点
5. l i t e r a l \_ t m p 1 −> l i t e r a l = f i r s t ;
6. l i t e r a l \_ t m p 1 −> t a g = 0 ;
7. V a l u e L i s t [ abs ( f i r s t ) ] . num++;
8. f i r s t > 0 ? V a l u e L i s t [ abs ( f i r s t ) ] . pos ++ : V a l u e L i s t [ abs ( f i r s t

) ] . nev ++;

|  |  |  |
| --- | --- | --- |
| 436 | i f | ( ! V a l u e L i s t [ abs ( f i r s t ) ] . f i r s t ) { |
| 437 |  | l i t e r a l \_ t m p 1 −>head = s e n t e n c e ; |
| 438 |  | V a l u e L i s t [ abs ( f i r s t ) ] . f i r s t = l i t e r a l \_ t m p 1 ; |
| 439 |  | V a l u e L i s t [ abs ( f i r s t ) ] . f i r s t −> n e x t \_ s a m e \_ l i t e r a l = NULL; |
| 440 | } | |
| 441 | e l s e { | |
| 442 | l i t e r a l \_ t m p 1 −>head = s e n t e n c e ; | |
| 443 | l i t e r a l \_ t m p 1 −> n e x t \_ s a m e \_ l i t e r a l = V a l u e L i s t [ abs ( f i r s t ) ] . | |
|  |  | f i r s t ; |
| 444 | V a l u e L i s t [ abs ( f i r s t ) ] . f i r s t = l i t e r a l \_ t m p 1 ; | |
| 445 | } | |
| 446 | l i t e r a l \_ t m p 1 −> n e x t \_ l i t e r a l = NULL; | |
| 447 | l i t e r a l \_ t m p 2 = l i t e r a l \_ t m p 1 ; | |
| 448 | l i t e r a l \_ t m p 2 −> t a g = 0 ; | |
| 449 | s e n t e n c e −> f i r s t \_ l i t e r a l = l i t e r a l \_ t m p 1 ; | |
| 450 |  | |
| 451 | */ /* 创建文字链表结点 | |
| 452 | f s c a n f ( fp , *”%d”* , &tmp ) ; | |
| 453 | w h i l e ( tmp ! = 0 ) { | |
| 454 | V a l u e L i s t [ abs ( tmp ) ] . num++; | |
| 455 | tmp > 0 ? V a l u e L i s t [ abs ( tmp ) ] . pos ++ : V a l u e L i s t [ abs ( tmp ) | |

] . nev ++;

1. l i t e r a l \_ t m p 1 = ( L i t e r a l \_ L i s t ) m a l l o c ( s i z e o f ( L i t e r a l \_ N o d e ) )

;

1. i f ( ! l i t e r a l \_ t m p 1 ) r e t u r n OVERFLOW;
2. l i t e r a l \_ t m p 1 −> l i t e r a l = tmp ;

## 华 中 科 技 大 学 课 程 实 验 报 告

1. l i t e r a l \_ t m p 1 −> n e x t \_ l i t e r a l = NULL;
2. l i t e r a l \_ t m p 2 −> n e x t \_ l i t e r a l = l i t e r a l \_ t m p 1 ;
3. l i t e r a l \_ t m p 2 = l i t e r a l \_ t m p 1 ;

|  |  |  |
| --- | --- | --- |
| 462 | i f | ( ! V a l u e L i s t [ abs ( tmp ) ] . f i r s t ) { |
| 463 |  | l i t e r a l \_ t m p 1 −>head = s e n t e n c e ; |
| 464 |  | V a l u e L i s t [ abs ( tmp ) ] . f i r s t = l i t e r a l \_ t m p 1 ; |
| 465 |  | V a l u e L i s t [ abs ( tmp ) ] . f i r s t −> n e x t \_ s a m e \_ l i t e r a l = NULL; |
| 466 | } |  |
| 467 | e l s | e { |
| 468 |  | l i t e r a l \_ t m p 1 −>head = s e n t e n c e ; |
| 469 |  | l i t e r a l \_ t m p 1 −> n e x t \_ s a m e \_ l i t e r a l = V a l u e L i s t [ abs ( tmp ) |
|  | ] . f i r s t ; | |
| 470 | V a l u e L i s t [ abs ( tmp ) ] . f i r s t = l i t e r a l \_ t m p 1 ; | |
| 471 | } | |
| 472 | l i t e r a l \_ t m p 2 −> t a g = 0 ; | |
| 473 | num++; | |
| 474 | f s c a n f ( fp , *”%d”* , &tmp ) ; */ /* 读取下一文字 | |
| 475 |  | |
| 476 | } | |
| 477 | s e n t e n c e −>number = num ; | |
| 478 | r e t u r n OK; | |
| 479 | } | |
| 480 |  | |
| 481 | */ /* 注意传入的是前一个结点 | |
| 482 | */ /* 定义 *Destroy Clause* 函数， 销毁子句结点 | |
| 483 | */ /* 输入： 字句链式节点 | |
| 484 | */ /* 输出： 状态 | |
| 485 | s t a t u s D e s t r o y C l a u s e ( C l a u s e \_ L i s t& s e n t e n c e ) { | |
| 486 | C l a u s e \_ L i s t p ; | |
| 487 | p = s e n t e n c e −> n e x t \_ c l a u s e ; | |
| 488 | i f ( ! p ) r e t u r n ERROR; */ /* 判断合理性 | |
| 489 | s e n t e n c e −> n e x t \_ c l a u s e = p−> n e x t \_ c l a u s e ; | |
| 490 | w h i l e ( p−> f i r s t \_ l i t e r a l −> n e x t \_ l i t e r a l ) D e s t r o y L i t e r a l ( p−> | |
|  | f i r s t \_ l i t e r a l ) ; | |
| 491 | f r e e ( p−> f i r s t \_ l i t e r a l ) ; | |

华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 492 | p−> f i r s t \_ l i t e r a l = NULL; |  | |
| 493 | f r e e ( p ) ; |
| 494 | p = NULL; |
| 495 | r e t u r n OK; |
| 496 | } |
| 497 |  |
| 498 | */ /* 注意传入的是前一个结点 |
| 499 | */ /* 定义 *D e s t r o y L i t e r a l* 函数， 销毁文字结点 |
| 500 | */ /* 输入： 文字链式结点 |
| 501 | */ /* 输出： 状态 |
| 502 | s t a t u s D e s t r o y L i t e r a l ( L i t e r a l \_ L i s t& word ) | { | |
| 503 | L i t e r a l \_ L i s t p ; |  | |
| 504 | p = word −> n e x t \_ l i t e r a l ; |  | |
| 505 | i f ( ! p ) r e t u r n ERROR; */ /* 判断合理性 |  | |
| 506 | word −> n e x t \_ l i t e r a l = p−> n e x t \_ l i t e r a l ; |  | |
| 507 | f r e e ( p ) ; |  | |
| 508 | p = NULL; |  | |
| 509 | r e t u r n OK; |  | |
| 510 | } |  | |
| 511 |  |  | |
| 512 | */ /* 定义 *DestroyCNF* 函数， 销毁*CNF*文件 | |  |
| 513 | */ /* 输入： *c n f* 指针 | |  |
| 514 | */ /* 输出： 状态 | |  |
| 515 | s t a t u s DestroyCNF ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& | | c n f ) { |
| 516 | i f ( ! c n f ) r e t u r n ERROR; */ /* 判断合理性 | |  |
| 517 | f o r ( i n t i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) | | { |
| 518 | V a l u e L i s t [ i ] . i s \_ v a l u e = 0 ; | |  |
| 519 | V a l u e L i s t [ i ] . t a g = 0 ; | |  |
| 520 | V a l u e L i s t [ i ] . num = 0 ; | |  |
| 521 | V a l u e L i s t [ i ] . nev = 0 ; | |  |
| 522 | V a l u e L i s t [ i ] . pos = 0 ; | |  |
| 523 | V a l u e L i s t [ i ] . s c o r e = 0 ; | |  |
| 524 | V a l u e L i s t [ i ] . f i r s t = NULL; | |  |
| 525 | } | |  |
| 526 | C l a u s e \_ L i s t c l a u s e \_ t m p 1 ; | |  |

华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |
| --- | --- | --- |
| 527 | w h i l e ( cnf −> f i r s t \_ c l a u s e −> n e x t \_ c l a u s e ) D e s t r o y C l a u s e ( cnf −>  f i r s t \_ c l a u s e ) ; | |
| 528 | c l a u s e \_ t m p 1 = ( C l a u s e \_ L i s t ) m a l l o c ( s i z e o f ( Clause\_Node ) ) ; |  |
| 529 | c l ause\_tmp 1 −> n e x t \_ c l a u s e = cnf −> f i r s t \_ c l a u s e ; |  |
| 530 | D e s t r o y C l a u s e ( c l a u s e \_ t m p 1 ) ; |  |
| 531 | f r e e ( c l a u s e \_ t m p 1 ) ; |  |
| 532 | c l a u s e \_ t m p 1 = NULL; |  |
| 533 | f r e e ( c n f ) ; |  |
| 534 | c n f = NULL; |  |
| 535 | r e t u r n OK; |  |
| 536 | } |  |
| 537 |  |  |
| 538 |  |  |
| 539 | */ /* 定义 *Has Unit Clause* 函数， 评估*CNF*是否含有单子句 |  |
| 540 | */ /* 输入： *c n f* 指针 |  |
| 541 | */ /* 输出： 文字链式结点 |  |
| 542 | C l a u s e \_ L i s t H as U n i t Cl a us e ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) | { |
| 543 | i f ( ! c n f ) r e t u r n NULL; |  |
| 544 | C l a u s e \_ L i s t p ; |  |
| 545 | f o r ( p = cnf −> f i r s t \_ c l a u s e ; p ; p = p−> n e x t \_ c l a u s e ) { |  |
| 546 | i f ( ! p−> t a g && p−>number == 1 ) { |  |
| 547 | r e t u r n p ; |  |
| 548 | } |  |
| 549 | } |  |
| 550 | r e t u r n NULL; |  |
| 551 | } |  |
| 552 |  |  |
| 553 | s t a t u s c h e c k \_ p r i n t 2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) { |  |
| 554 | C l a u s e \_ L i s t p = cnf −> f i r s t \_ c l a u s e ; |  |
| 555 | L i t e r a l \_ L i s t q = NULL; |  |
| 556 | i n t f l a g ; |  |
| 557 | w h i l e ( p ) { |  |
| 558 | q = p−> f i r s t \_ l i t e r a l ; |  |
| 559 | f l a g = 0 ; |  |
| 560 | w h i l e ( q ) { |  |

华 中 科 技 大 学 课 程 实 验 报 告

561 i f ( q−> l i t e r a l \* V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . i s \_ v a l u e

> 0 ) {

|  |  |  |
| --- | --- | --- |
| 562 | f l a g = 1 ; |  |
| 563 | b r e a k ; |  |
| 564 | } |  |
| 565 | q = q−> n e x t \_ l i t e r a l ; |  |
| 566 | } |  |
| 567 | p r i n t f ( *”%d \ n”* , f l a g ) ; |  |
| 568 | p = p−> n e x t \_ c l a u s e ; |  |
| 569 | } |  |
| 570 | r e t u r n OK; |  |
| 571 | } |  |
| 572 |  |  |
| 573 | */ /* 定义 *s t ore\_document* 函数， 存储文件 |  |
| 574 | voi d s t o r e \_ d o c u m e n t ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& cnf , c h a r\* f i l e n a m e , i n t d , i n t t ime 0 ) { |  |
| 575 | i n t i = 0 ; |  |
| 576 | w h i l e ( 1 ) { |  |
| 577 | i f ( f i l e n a m e [ i ] == *’ c ’* && f i l e n a m e [ i + 1 ] == *’ n ’* && |  |
|  | f i l e n a m e [ i + 2 ] == *’ f ’* ) |  |
| 578 | b r e a k ; |  |
| 579 | i ++; |  |
| 580 | } |  |
| 581 | f i l e n a m e [ i ] = *’ r ’* , f i l e n a m e [ i + 1 ] = *’ e ’* , f i l e n a m e [ i + 2 ] | = *’* |
|  | *s ’* ; |  |
| 582 | FILE\* fp ; |  |
| 583 | fp = fopen ( f i l e n a m e , *”w”* ) ; |  |
| 584 | f p r i n t f ( fp , *”%d \ n”* , d ) ; |  |
| 585 | f o r ( i n t tmp0 = 1 ; tmp0 < cnf −> l i t e r a l \_ n u m ; tmp0 ++) { |  |
| 586 | i f ( V a l u e L i s t [ tmp0 ] . i s \_ v a l u e == 1 ) |  |
| 587 | f p r i n t f ( fp , *”%d* , tmp0 ) ; |  |
| 588 | e l s e |  |
| 589 | f p r i n t f ( fp , *”%d* , −tmp0 ) ; |  |
| 590 | i f ( tmp0 % 10 == 0 ) |  |
| 591 | f p r i n t f ( fp , *” \ n”* ) ; |  |

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 592 | } |  | |
| 593 | f p r i n t f ( fp , *”%d”* , t ime 0 ) ; |
| 594 | f c l o s e ( fp ) ; |
| 595 | r e t u r n ; |
| 596 | } |
| 597 |  |
| 598 | voi d t r a v e s e r \_ c n f ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) { |
| 599 | C l a u s e \_ L i s t p = cnf −> f i r s t \_ c l a u s e ; |
| 600 | L i t e r a l \_ L i s t q = NULL; |
| 601 | i n t f l a g ; |
| 602 | p r i n t f ( *”* 变元数：*%d \ n* 子句数目：*%d \ n”* , cnf −> l i t e r a l \_ n u m | , | cnf −> |
|  | clause\_num ) ; |  |  |
| 603 | w h i l e ( p ) { |  | |
| 604 | q = p−> f i r s t \_ l i t e r a l ; |  | |
| 605 | w h i l e ( q ) { |  | |
| 606 | p r i n t f ( *”%d* , q−> l i t e r a l ) ; |  | |
| 607 | q = q−> n e x t \_ l i t e r a l ; |  | |
| 608 | } |  | |
| 609 | p r i n t f ( *” \ n”* ) ; |  | |
| 610 | p = p−> n e x t \_ c l a u s e ; |  | |
| 611 | } |  | |
| 612 | r e t u r n ; |  | |
| 613 | } |  | |
| 614 |  |  | |
| 615 | */ /* 定义 *p r ove\_cnf* 函数， 进行验证 |  | |
| 616 | voi d p r o v e \_ c n f ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t | c n f ) { | |
| 617 | C l a u s e \_ L i s t p = cnf −> f i r s t \_ c l a u s e ; |  | |
| 618 | L i t e r a l \_ L i s t q = NULL; |  | |
| 619 | i n t f l a g ; |  | |
| 620 | w h i l e ( p ) { |  | |
| 621 | q = p−> f i r s t \_ l i t e r a l ; |  | |
| 622 | f l a g = 0 ; |  | |
| 623 | w h i l e ( q ) { |  | |

624 i f ( q−> l i t e r a l \* V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . i s \_ v a l u e

> 0 ) {

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 625 |  |  | f l a g = 1 ; | | |
| 626 |  |  | b r e a k ; | | |
| 627 |  | } |  | | |
| 628 |  | q = | q−> n e x t \_ l i t e r a l ; | | |
| 629 | } |  |  | | |
| 630 | i f | ( ! f l a g ) { | | | |
| 631 |  | p r i n t f ( *”* 该 *c n f* 公式验证无解 *! \ n”* ) ; | | | |
| 632 |  | r e t u r n ; | | | |
| 633 | } | | |  |  |
| 634 | p = p−> n e x t \_ c l a u s e ; | | |  |  |
| 635 | } | | |  |  |
| 636 | p r i n t f ( *”* 该 *c n f* 公式验证有解 *! \ n”* ) ; | | |  |  |
| 637 | r e t u r n ; | | |  |  |
| 638 | } | | |  |  |
| 639 |  | | |  |  |
| 640 | */ /* 定义 *show\_cnf* 函数， 输出 *c n f* 文件求解结果 | | |  |  |
| 641 | voi d show\_cnf ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t | | | c n f ) | { |
| 642 | f o r ( i n t i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; | | | i ++) | { |
| 643 | i f ( V a l u e L i s t [ i ] . i s \_ v a l u e == 1 ) | | |  |  |
| 644 | p r i n t f ( *”%d* , i ) ; | | |  |  |
| 645 | e l s e | | |  |  |
| 646 | p r i n t f ( *”%d*, − i ) ; | | |  |  |
| 647 | i f ( ! ( i % 10 ) ) | | |  |  |
| 648 | p r i n t f ( *” \ n”* ) ; | | |  |  |
| 649 | } | | |  |  |
| 650 | r e t u r n ; | | |  |  |
| 651 | } | | |  |  |

文件名：func\_DPLL2.cpp 功能：DPLL2 框架的实现

1

*/ /*

*/ / Created by hp on 2022 −09 −01.*

*/ /*

# i n c l u d e *” d e f i n e . h”*

2

3

4

5

## 华 中 科 技 大 学 课 程 实 验 报 告

6

1. */\** 定义有效的全局变量 *\*/*
2. e x t e r n Argue Value\* V a l u e L i s t ; */ /* 定义变元真值表
3. boo l i s C o n f l i c t ;

|  |  |  |
| --- | --- | --- |
| 10 | i n t | p r e h a n d l e \_ s t a c k [MAXN] , p r e \_ t o p ; |
| 11 | i n t | d e c i s i o n \_ s t a c k [MAXN] , d e c \_ t o p ; |
| 12 | i n t | decision\_num [MAXN] ; |
| 13 | i n t | c o n f l i c t \_ s t a c k \_ [MAXN] , con\_top ; |

14 d o u b l e t ime\_sum ; 15

16 i n t c n t = 0 , t i m e s = 1 , c n t 0 = 0 ; 17

1. */ /* 定义 *CreateCNF2* 函数， 创建*CNF*链表结果
2. s t a t u s Create CNF2 ( FILE\* fp , C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) {
3. i s C o n f l i c t = FALSE ;
4. c h a r r e a d f i l e [ 2 0 ] ; */ /* 定义字符类型数组记录在文件中
5. i n t i , tmp ; */ /* 整型变量定义
6. C l a u s e \_ L i s t c l ause\_tmp 1 , c l a u s e \_ t m p 2 ; */ /* 定义临时结点变量

24

1. */ /* 初始化*CNF*
2. c n f = ( Conjunctive\_Normal\_Form \*) m a l l o c ( s i z e o f ( Conjunctive\_Normal\_Form ) ) ;
3. cnf −> f i r s t \_ c l a u s e = NULL; 28
4. */ / CNF*文件读取
5. w h i l e ( f s c a n f ( fp , *”%s”* , r e a d f i l e ) ! = EOF) { */ /* 循环读文件
6. i f ( s t r c m p ( r e a d f i l e , *”p”* ) == 0 ) */ /* 持续读取文件直至开始标志’ *p* ‘
7. b r e a k ;

33 }

1. w h i l e ( f s c a n f ( fp , *”%s”* , r e a d f i l e ) ! = EOF) {
2. i f ( s t r c m p ( r e a d f i l e , *” c n f ”* ) == 0 ) { */ /* 从文件中读到字符串“

*c n f* ”

1. f s c a n f ( fp , *”%d”* , &cnf −> l i t e r a l \_ n u m ) ; */ /* 读取*CNF*文件变元数并存入 *l i t e r a l \_ n u m*

## 华 中 科 技 大 学 课 程 实 验 报 告

37

38

39 }

40 }

41

f s c a n f ( fp , *”%d”* , &cnf −> clause\_num ) ; */ /* 读取*CNF*文件子句总数并存入 *clause\_num*

b r e a k ;

42 */ / p r i n t f (” l i t e r a l \_ n u m :%d \ nclause\_num:%d \ n ” , cnf −>l i t e r a l \_ n u m , cnf −>clause\_num ) ; / /* 输出基本信息

43

1. */ /* 定义并初始化变元真值表
2. V a l u e L i s t = ( Argue Value \*) m a l l o c ( ( cnf −> l i t e r a l \_ n u m + 2 ) \*

s i z e o f ( Argue Value ) ) ;

|  |  |  |
| --- | --- | --- |
| 46 |  | i f ( ! V a l u e L i s t ) r e t u r n OVERFLOW; */ /* 没分配成功， 返回*OVERFLOW* |
| 47 |  | f o r ( i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) { |
| 48 |  | V a l u e L i s t [ i ] . i s \_ v a l u e = 0 ; |
| 49 |  | V a l u e L i s t [ i ] . t a g = 0 ; |
| 50 |  | V a l u e L i s t [ i ] . dep = 0 ; |
| 51 |  | V a l u e L i s t [ i ] . num = 0 ; |
| 52 |  | V a l u e L i s t [ i ] . nev = 0 ; |
| 53 |  | V a l u e L i s t [ i ] . pos = 0 ; |
| 54 |  | V a l u e L i s t [ i ] . s c o r e = 0 ; |
| 55 |  | V a l u e L i s t [ i ] . f i r s t = NULL; |
| 56 |  | } |
| 57 |  |  |
| 58 |  | */ /* 创建*CNF*链式结构 |
| 59 |  | i f ( cnf −> clause\_num ) { |
| 60 |  | i f ( f s c a n f ( fp , *”%d”* , &tmp ) ! = EOF && tmp ! = 0 ) { |
| 61 |  | */ /* 创建*CNF*子句的头指针 |
| 62 | */ /* | *p r i n t f (”%d ” , tmp ) ;* |

1. c l a u s e \_ t m p 1 = ( C l a u s e \_ L i s t ) m a l l o c ( s i z e o f ( Clause\_Node )

) ;

1. i f ( ! c l a u s e \_ t m p 1 ) r e t u r n OVERFLOW;
2. cnf −> f i r s t \_ c l a u s e = c l a u s e \_ t m p 1 ; */ /* 定义头节点
3. c l a u s e \_ t m p 2 = c l a u s e \_ t m p 1 ;
4. C r e a t e C l a u s e 2 ( fp , c l ause\_tmp 1 , tmp ) ; */ /* 创建其对应子句

## 华 中 科 技 大 学 课 程 实 验 报 告

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82 }

链表

c l ause\_tmp 1 −> n e x t \_ c l a u s e = NULL; i = 2 ;

*/ /* 创建完整的*CNF*子句链式结构

w h i l e ( i ++ <= cnf −> clause\_num ) { f s c a n f ( fp , *”%d”* , &tmp ) ;

*/ / p r i n t f (”%d ” , tmp ) ;*

c l a u s e \_ t m p 1 = ( C l a u s e \_ L i s t ) m a l l o c ( s i z e o f ( Clause\_Node ) ) ;

i f ( ! c l a u s e \_ t m p 1 ) r e t u r n OVERFLOW;

C r e a t e C l a u s e 2 ( fp , c l ause\_tmp 1 , tmp ) ; */ /* 创建其对应子句链表

c l ause\_tmp 2 −> n e x t \_ c l a u s e = c l a u s e \_ t m p 1 ; c l a u s e \_ t m p 2 = c l a u s e \_ t m p 1 ;

c l ause\_tmp 1 −> n e x t \_ c l a u s e = NULL;

}

}

83

84 }

85

r e t u r n OK;

1. */ /* 定义 *Create Clause 2* 函数， 创建子句链表
2. */ /* 输入： 文件指针， 子句链式结点， 首值
3. */ /* 输出： 状态
4. s t a t u s C r e a t e C l a u s e 2 ( FILE\* fp , C l a u s e \_ L i s t& s e n t e n c e , i n t f i r s t )

{

1. i n t num = 1 , tmp ; */ /* 定义文字数目， 临时变量
2. */ / p r i n t f (”%d ” , f i r s t ) ;*
3. i f ( ! f i r s t ) r e t u r n ERROR;
4. L i t e r a l \_ L i s t l i t e r a l \_ t m p 1 , l i t e r a l \_ t m p 2 ; */ /* 定义临时结点变量
5. l i t e r a l \_ t m p 1 = ( L i t e r a l \_ L i s t ) m a l l o c ( s i z e o f ( L i t e r a l \_ N o d e ) ) ;
6. i f ( ! l i t e r a l \_ t m p 1 ) r e t u r n OVERFLOW;
7. */ /* 创建文字链式结构头节点
8. l i t e r a l \_ t m p 1 −> l i t e r a l = f i r s t ;
9. V a l u e L i s t [ abs ( f i r s t ) ] . num++;

## 华 中 科 技 大 学 课 程 实 验 报 告

1. f i r s t > 0 ? V a l u e L i s t [ abs ( f i r s t ) ] . pos ++ : V a l u e L i s t [ abs ( f i r s t

) ] . nev ++;

|  |  |  |
| --- | --- | --- |
| 100 | i f | ( ! V a l u e L i s t [ abs ( f i r s t ) ] . f i r s t ) { |
| 101 |  | l i t e r a l \_ t m p 1 −>head = s e n t e n c e ; |
| 102 |  | V a l u e L i s t [ abs ( f i r s t ) ] . f i r s t = l i t e r a l \_ t m p 1 ; |
| 103 |  | V a l u e L i s t [ abs ( f i r s t ) ] . f i r s t −> n e x t \_ s a m e \_ l i t e r a l = NULL; |
| 104 | } | |
| 105 | e l s e { | |
| 106 | l i t e r a l \_ t m p 1 −>head = s e n t e n c e ; | |
| 107 | l i t e r a l \_ t m p 1 −> n e x t \_ s a m e \_ l i t e r a l = V a l u e L i s t [ abs ( f i r s t ) ] . | |
|  |  | f i r s t ; |
| 108 | V a l u e L i s t [ abs ( f i r s t ) ] . f i r s t = l i t e r a l \_ t m p 1 ; | |
| 109 | } | |
| 110 | l i t e r a l \_ t m p 1 −> n e x t \_ l i t e r a l = NULL; | |
| 111 | l i t e r a l \_ t m p 2 = l i t e r a l \_ t m p 1 ; | |
| 112 | s e n t e n c e −> f i r s t \_ l i t e r a l = l i t e r a l \_ t m p 1 ; | |
| 113 | s e n t e n c e −> w a t c h \_ l i t e r a l [ 0 ] = l i t e r a l \_ t m p 1 −> l i t e r a l ; | |
| 114 |  | |
| 115 | */ /* 创建文字链表结点 | |
| 116 | f s c a n f ( fp , *”%d”* , &tmp ) ; | |
| 117 | i f ( ! tmp ) { | |
| 118 | p r e h a n d l e \_ s t a c k [ p r e \_ t o p ++] = f i r s t ; | |
| 119 | i n t op = ( f i r s t > 0 ? 1 : −1) ; | |
| 120 | i n t o l d l i t e r a l = V a l u e L i s t [ abs ( f i r s t ) ] . i s \_ v a l u e ; | |
| 121 | i f ( o l d l i t e r a l && o l d l i t e r a l ! = op ) { | |
| 122 | i s C o n f l i c t = TRUE; | |
| 123 | } | |
| 124 | V a l u e L i s t [ abs ( f i r s t ) ] . i s \_ v a l u e = ( f i r s t > 0 ? 1 : −1) ; | |
| 125 | } | |
| 126 | w h i l e ( tmp ! = 0 ) { | |
| 127 | */ / p r i n t f (”%d ” , tmp ) ;* | |
| 128 | V a l u e L i s t [ abs ( tmp ) ] . num++; | |
| 129 | tmp > 0 ? V a l u e L i s t [ abs ( tmp ) ] . pos ++ : V a l u e L i s t [ abs ( tmp ) | |

] . nev ++;

1. l i t e r a l \_ t m p 1 = ( L i t e r a l \_ L i s t ) m a l l o c ( s i z e o f ( L i t e r a l \_ N o d e ) )

## 华 中 科 技 大 学 课 程 实 验 报 告

;

1. i f ( ! l i t e r a l \_ t m p 1 ) r e t u r n OVERFLOW;
2. l i t e r a l \_ t m p 1 −> l i t e r a l = tmp ;
3. l i t e r a l \_ t m p 1 −> n e x t \_ l i t e r a l = NULL;
4. l i t e r a l \_ t m p 2 −> n e x t \_ l i t e r a l = l i t e r a l \_ t m p 1 ;
5. l i t e r a l \_ t m p 2 = l i t e r a l \_ t m p 1 ;

|  |  |  |
| --- | --- | --- |
| 136 | i f | ( ! V a l u e L i s t [ abs ( tmp ) ] . f i r s t ) { |
| 137 |  | l i t e r a l \_ t m p 1 −>head = s e n t e n c e ; |
| 138 |  | V a l u e L i s t [ abs ( tmp ) ] . f i r s t = l i t e r a l \_ t m p 1 ; |
| 139 |  | V a l u e L i s t [ abs ( tmp ) ] . f i r s t −> n e x t \_ s a m e \_ l i t e r a l = NULL; |
| 140 | } |  |
| 141 | e l s | e { |
| 142 |  | l i t e r a l \_ t m p 1 −>head = s e n t e n c e ; |
| 143 |  | l i t e r a l \_ t m p 1 −> n e x t \_ s a m e \_ l i t e r a l = V a l u e L i s t [ abs ( tmp )  ] . f i r s t ; |
| 144 | V a l u e L i s t [ abs ( tmp ) ] . f i r s t = l i t e r a l \_ t m p 1 ; | |
| 145 | } | |
| 146 | num++; | |
| 147 | f s c a n f ( fp , *”%d”* , &tmp ) ; */ /* 读取下一文字 | |
| 148 | } i f ( s e n t e n c e −> f i r s t \_ l i t e r a l −> n e x t \_ l i t e r a l ) | |
| 149 | s e n t e n c e −> w a t c h \_ l i t e r a l [ 1 ] = s e n t e n c e −> f i r s t \_ l i t e r a l −> | |

n e x t \_ l i t e r a l −> l i t e r a l ;

1. s e n t e n c e −>number = num ;
2. */ / p r i n t f (”%d ” , num ) ;*

152 */ / p r i n t f ( ” \ n ”) ;*

153 r e t u r n OK;

|  |  |
| --- | --- |
| 154 | } |
| 155 |  |
| 156 | */ /* 注意传入的是前一个结点 |
| 157 | */ /* 定义 *Destroy Clause 2* 函数， 销毁子句结点 |
| 158 | */ /* 输入： 字句链式节点 |
| 159 | */ /* 输出： 状态 |
| 160 | s t a t u s D e s t r o y C l a u s e 2 ( C l a u s e \_ L i s t& s e n t e n c e ) { |
| 161 | C l a u s e \_ L i s t p ; |
| 162 | p = s e n t e n c e −> n e x t \_ c l a u s e ; |

## 华 中 科 技 大 学 课 程 实 验 报 告

1. i f ( ! p ) r e t u r n ERROR; */ /* 判断合理性
2. s e n t e n c e −> n e x t \_ c l a u s e = p−> n e x t \_ c l a u s e ;
3. w h i l e ( p−> f i r s t \_ l i t e r a l −> n e x t \_ l i t e r a l ) D e s t r o y L i t e r a l 2 ( p−> f i r s t \_ l i t e r a l ) ;

|  |  |  |
| --- | --- | --- |
| 166 | f r e e ( p−> f i r s t \_ l i t e r a l ) ; |  |
| 167 | p−> f i r s t \_ l i t e r a l = NULL; |
| 168 | f r e e ( p ) ; |
| 169 | p = NULL; |
| 170 | r e t u r n OK; |
| 171 | } |
| 172 |  |
| 173 | */ /* 注意传入的是前一个结点 |
| 174 | */ /* 定义 *D e s t r o y L i t e r a l* 函数， 销毁文字结点 |
| 175 | */ /* 输入： 文字链式结点 |
| 176 | */ /* 输出： 状态 |
| 177 | s t a t u s D e s t r o y L i t e r a l 2 ( L i t e r a l \_ L i s t& word ) | { |
| 178 | L i t e r a l \_ L i s t p ; |  |
| 179 | p = word −> n e x t \_ l i t e r a l ; |  |
| 180 | i f ( ! p ) r e t u r n ERROR; */ /* 判断合理性 |  |
| 181 | word −> n e x t \_ l i t e r a l = p−> n e x t \_ l i t e r a l ; |  |
| 182 | f r e e ( p ) ; |  |
| 183 | p = NULL; |  |
| 184 | r e t u r n OK; |  |
| 185 | } |  |
| 186 |  |  |

1. */ /* 定义 *DestroyCNF2* 函数， 销毁*CNF*文件
2. */ /* 输入： *c n f* 指针
3. */ /* 输出： 状态
4. s t a t u s DestroyCNF2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) {
5. i f ( ! c n f ) r e t u r n ERROR; */ /* 判断合理性
6. C l a u s e \_ L i s t c l a u s e \_ t m p 1 ;
7. w h i l e ( cnf −> f i r s t \_ c l a u s e −> n e x t \_ c l a u s e ) D e s t r o y C l a u s e 2 ( cnf −> f i r s t \_ c l a u s e ) ;
8. c l a u s e \_ t m p 1 = ( C l a u s e \_ L i s t ) m a l l o c ( s i z e o f ( Clause\_Node ) ) ;
9. c l ause\_tmp 1 −> n e x t \_ c l a u s e = cnf −> f i r s t \_ c l a u s e ;

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |
| --- | --- | --- |
| 196 |  | D e s t r o y C l a u s e 2 ( c l a u s e \_ t m p 1 ) ; |
| 197 |  | f r e e ( c l a u s e \_ t m p 1 ) ; |
| 198 |  | c l a u s e \_ t m p 1 = NULL; |
| 199 |  | f r e e ( c n f ) ; |
| 200 |  | c n f = NULL; |
| 201 |  | r e t u r n OK; |
| 202 | } |  |
| 203 |  |  |
| 204 | */ /* 定 | 义 *S e l e c t \_ L i t e r a l 3* 函数， 变量决策策略 |
| 205 | i n t | s e l e c t \_ l i t e r a l 3 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) { |
| 206 |  |  |
| 207 |  | */ / g e t t i m e o f d a y (& s t , NULL) ;* |
| 208 |  |  |
| 209 |  | i n t tmp0 , f l a g = 0 , t m p \_ l i t e r a l , s t p ; |
| 210 |  | f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) { |
| 211 |  | i f ( ! V a l u e L i s t [ tmp0 ] . i s \_ v a l u e ) { |
| 212 |  | f l a g = 1 ; |
| 213 |  | } |
| 214 |  | } |
| 215 |  | s r a n d ( i n t ( t i me ( 0 ) ) ) ; |
| 216 |  | i n t a ; |
| 217 |  | i f ( ! f l a g ) r e t u r n 0 ; |

1. i f ( c n t 0 > 5000 \* t i m e s ) {
2. t i m e s ++;
3. f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) {
4. V a l u e L i s t [ tmp0 ] . i s \_ v a l u e = 0 ;
5. V a l u e L i s t [ tmp0 ] . dep = 0 ;

223 }

224 */ / p r i n t f (” c o n f l i c t :%d \ n \ n \ n \ n \ n \ n \ n \ n ” , t i m e s ) ;*

225 }

226

1. C l a u s e \_ L i s t p ;
2. L i t e r a l \_ L i s t q = NULL, r ;
3. i n t l i t e r a l ;
4. w h i l e ( con\_top ) {

## 华 中 科 技 大 学 课 程 实 验 报 告

1. l i t e r a l = c o n f l i c t \_ s t a c k \_ [ −− con\_top ] ;
2. q = V a l u e L i s t [ abs ( l i t e r a l ) ] . f i r s t ;
3. w h i l e ( q ) {
4. i f ( q−>head −> t a g ) {
5. q = q−> n e x t \_ s a m e \_ l i t e r a l ;
6. c o n t i n u e ;

237 }

1. r = q−>head −> f i r s t \_ l i t e r a l ;
2. w h i l e ( r ) {
3. i f ( r −> t a g ) {
4. r = r −> n e x t \_ l i t e r a l ;
5. c o n t i n u e ;

243 }

1. V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . s c o r e += cnf −>

l i t e r a l \_ n u m / ( r −>head −>number \* r −>head −> number ) ;

1. i f ( V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . s c o r e > 5000 ) {
2. */ / p r i n t f (” s core :%d \ n ” , V a l u e L i s t [ abs ( r−> l i t e r a l ) ] . s core ) ;*
3. f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0

++) {

1. V a l u e L i s t [ tmp0 ] . s c o r e / = 5 0 ;

249 }

250 }

1. r −> l i t e r a l > 0 ? V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . pos ++

: V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . nev ++;

1. r = r −> n e x t \_ l i t e r a l ;

253 }

254 q = q−> n e x t \_ s a m e \_ l i t e r a l ;

255 }

256 }

1. s t p = 0 ;
2. f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) {
3. i f ( ! V a l u e L i s t [ tmp0 ] . i s \_ v a l u e && V a l u e L i s t [ tmp0 ] . s c o r e >= s t p ) {

## 华 中 科 技 大 学 课 程 实 验 报 告

1. s t p = V a l u e L i s t [ tmp0 ] . s c o r e ;
2. l i t e r a l = tmp0 ;

262 }

|  |  |  |
| --- | --- | --- |
| 263 | } |  |
| 264 | c n t ++; |
| 265 | */ / g e t t i m e o f d a y (& ed , NULL) ;* |
| 266 |  |
| 267 | */ / t i m e \_ t o t a l = ( ed . t v \_ s e c − s t . t v \_ s e c )* | *+ ( ed . t v \_ u s e c − s t .* |
|  | *t v \_ u s e c ) / 1000000 . 0 ;* | |
| 268 | */ / t ime\_sum += t i m e \_ t o t a l ;* | |
| 269 | */ / p r i n t f (”[ t i m e \_ t o t a l%d]:% l f S \ n ” , cnt , t i m e \_ t o t a l ) ;* | |
| 270 | i f ( V a l u e L i s t [ abs ( l i t e r a l ) ] . pos >= V a l u e L i s t [ abs ( l i t e r a l ) ] . | |
|  | nev ) | |
| 271 | r e t u r n abs ( l i t e r a l ) ; | |
| 272 | e l s e | |
| 273 | r e t u r n − abs ( l i t e r a l ) ; | |
| 274 | } | |
| 275 |  | |
| 276 | */ /* 定义 *S e l e c t \_ L i t e r a l 2* 函数， 变量决策策略 | |
| 277 | i n t s e l e c t \_ l i t e r a l 2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) { | |
| 278 | i n t tmp0 , f l a g = 0 ; | |
| 279 | f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) { | |
| 280 | V a l u e L i s t [ tmp0 ] . num = 0 ; | |
| 281 | V a l u e L i s t [ tmp0 ] . pos = 0 ; | |
| 282 | V a l u e L i s t [ tmp0 ] . nev = 0 ; | |
| 283 | V a l u e L i s t [ tmp0 ] . s c o r e = 0 ; | |
| 284 | i f ( ! V a l u e L i s t [ tmp0 ] . i s \_ v a l u e ) | |
| 285 | f l a g = 1 ; | |
| 286 | } | |
| 287 | i f ( ! f l a g ) r e t u r n 0 ; | |
| 288 | C l a u s e \_ L i s t p ; | |
| 289 | L i t e r a l \_ L i s t q = NULL, r ; | |
| 290 | i n t l i t e r a l ; | |
| 291 | w h i l e ( con\_top ) { | |
| 292 | l i t e r a l = c o n f l i c t \_ s t a c k \_ [ −− con\_top ] ; | |

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |
| --- | --- | --- |
| 293 | q = V a l u e L i s t [ abs ( l i t e r a l ) ] . f i r s t ; |  |
| 294 | w h i l e ( q ) { |
| 295 | r = q−>head −> f i r s t \_ l i t e r a l ; |
| 296 | w h i l e ( r ) { |
| 297 | i f ( V a l u e L i s t [ r −> l i t e r a l ] . i s \_ v a l u e ) | { |
| 298 | r = r −> n e x t \_ l i t e r a l ; |  |
| 299 | c o n t i n u e ; |  |
| 300 | } |  |
| 301 | V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . s c o r e += | ( cnf −> |

l i t e r a l \_ n u m / ( r −>head −>number \* r −>head −> number ) ) ;

1. r −> l i t e r a l > 0 ? V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . pos ++

: V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . nev ++;

1. r = r −> n e x t \_ l i t e r a l ;

304 }

305 q = q−> n e x t \_ s a m e \_ l i t e r a l ;

306 }

307 }

1. C l a u s e \_ L i s t h = NULL;
2. f o r ( p = cnf −> f i r s t \_ c l a u s e ; p ; p = p−> n e x t \_ c l a u s e ) {
3. f l a g = 0 ;
4. q = p−> f i r s t \_ l i t e r a l ;
5. w h i l e ( q ) {
6. i f ( q−> l i t e r a l \* V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . i s \_ v a l u e

> 0 ) {

1. f l a g = 1 ;
2. b r e a k ;

316 }

317 q = q−> n e x t \_ l i t e r a l ;

318 }

1. i f ( f l a g ) c o n t i n u e ;
2. r = p−> f i r s t \_ l i t e r a l ;
3. w h i l e ( r ) {
4. V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . num++;
5. V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . s c o r e += ( cnf −> l i t e r a l \_ n u m

## 华 中 科 技 大 学 课 程 实 验 报 告

/ ( r −>head −>number \* r −>head −>number ) ) ;

1. r −> l i t e r a l > 0 ? V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . pos ++ :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . nev ++; | | |
| 325 | i f | | | ( V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . s c o r e > 5000 ) { | | |
| 326 |  | | | */ / p r i n t f (” s core :%d \ n ” , V a l u e L i s t [*  *abs ( r−> l i t e r a l ) ] . s core ) ;* | | |
| 327 | f o r ( i n t i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) | | | | | { |
| 328 | V a l u e L i s t [ i ] . s c o r e / = 5 0 ; | | | | |  |
| 329 | } | | | | |  |
| 330 |  | | } | |  |  |
| 331 |  | | r = r −> n e x t \_ l i t e r a l ; | |  |  |
| 332 | } | |  | |  |  |
| 333 | } | |  | |  |  |
| 334 | i n t i , | | s c o r e = 0 ; | |  |  |
| 335 | l i t e r a l | | = 0 ; | |  |  |
| 336 | f o r ( i | | = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) { | |  |  |
| 337 | i f | | ( ! V a l u e L i s t [ i ] . i s \_ v a l u e && V a l u e L i s t [ i ] . s c o r e | | >= | s c o r e |
|  |  | | ) { | |  |  |
| 338 |  | | s c o r e = V a l u e L i s t [ i ] . s c o r e ; | |  |  |
| 339 |  | | l i t e r a l = i ; | |  |  |
| 340 | } | |  | |  |  |
| 341 | } | | | | | |
| 342 | i f ( V a l u e L i s t [ l i t e r a l ] . pos >= V a l u e L i s t [ l i t e r a l ] . nev ) | | | | | |
| 343 | r e t u r n abs ( l i t e r a l ) ; | | | | | |
| 344 | e l s e | | | | | |
| 345  346 | } | r e t u r n − abs ( l i t e r a l ) ; | | | | |
| 347 |  |  | | | | |
| 348 | */ /* 定 | 义 *u p d a t e \_ s t o r e v a l u e* 函数， 更新存储文字 | | | | |
| 349 | i n t | u p d a t e \_ v a l u e ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) { | | | | |
| 350 |  | i n t tmp0 , t m p \_ l i t e r a l ; | | | | |
| 351 |  | i n t s c o r e ; | | | | |
| 352 |  | L i t e r a l \_ L i s t p = NULL; | | | | |
| 353 |  | f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) { | | | | |
| 354 |  | p = V a l u e L i s t [ tmp0 ] . f i r s t ; | | | | |

## 华 中 科 技 大 学 课 程 实 验 报 告

\* p−>

|  |  |  |  |
| --- | --- | --- | --- |
| 355 | s c o r e = 0 ; | | |
| 356 | w h i l e ( p ) { | | |
| 357 | s c o r e += ( cnf −> l i t e r a l \_ n u m ) / ( p−>head −>number | | |
|  |  | head −>number ) ; |  |
| 358 |  | p = p−> n e x t \_ l i t e r a l ; |  |
| 359 |  | } |  |
| 360 |  | V a l u e L i s t [ tmp0 ] . s c o r e = s c o r e ; |  |
| 361 |  | } |  |
| 362 |  | r e t u r n OK; |  |
| 363 | } |  |  |
| 364 |  |  |  |
| 365 | */ /* 定 | 义 *S e l e c t \_ L i t e r a l* 函数， 变量决策策略 |  |
| 366 | i n t | s e l e c t \_ l i t e r a l ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) | { |
| 367 |  |  |  |
| 368 |  | */ / g e t t i m e o f d a y (& s t , NULL) ;* |  |
| 369 |  |  |  |
| 370 |  | i n t tmp0 , f l a g = 0 , t m p \_ l i t e r a l , s t p ; |  |
| 371 |  | f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) | { |
| 372 |  | i f ( ! V a l u e L i s t [ tmp0 ] . i s \_ v a l u e ) { |  |
| 373 |  | f l a g = 1 ; |  |
| 374 |  | } |  |
| 375 |  | } |  |
| 376 |  | s r a n d ( i n t ( t i me ( 0 ) ) ) ; |  |
| 377 |  | i n t a ; |  |
| 378 |  | i f ( ! f l a g ) r e t u r n 0 ; |  |
| 379 |  | i f ( c n t 0 > 100 \* t i m e s ) { |  |

380

t i m e s ++;

|  |  |
| --- | --- |
| 381 | f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) { |
| 382 | V a l u e L i s t [ tmp0 ] . i s \_ v a l u e = 0 ; |
| 383 | } |
| 384 | */ / p r i n t f (” c o n f l i c t :%d \ n \ n \ n \ n \ n \ n \ n \ n ” , t i m e s ) ;* |
| 385 | } |
| 386 |  |
| 387 | C l a u s e \_ L i s t p ; |
| 388 | L i t e r a l \_ L i s t q = NULL, r ; |

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |
| --- | --- | --- |
| 389 | i n t l i t e r a l ; | |
| 390 | w h i l e ( con\_top ) { | |
| 391 | l i t e r a l = c o n f l i c t \_ s t a c k \_ [ −− con\_top ] ; | |
| 392 | q = V a l u e L i s t [ abs ( l i t e r a l ) ] . f i r s t ; | |
| 393 | w h i l e ( q ) { | |
| 394 | i f ( q−>head −> t a g ) { | |
| 395 | q = q−> n e x t \_ s a m e \_ l i t e r a l ; | |
| 396 | c o n t i n u e ; | |
| 397 | } | |
| 398 | r = q−>head −> f i r s t \_ l i t e r a l ; | |
| 399 | w h i l e ( r ) { | |
| 400 | i f ( r −> t a g ) { | |
| 401 | r = r −> n e x t \_ l i t e r a l ; | |
| 402 | c o n t i n u e ; | |
| 403 | } | |
| 404 | V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . s c o r e += cnf −> | |
|  | l i t e r a l \_ n u m / ( r −>head −>number \* r −>head −> | |
| 405 | i f ( V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . s c o r e > 500 ) { | |
| 406 | */ / p r i n t f (” s core :%d \ n ” ,*  *V a l u e L i s t [ abs ( r−> l i t e r a l ) ] . s core ) ;* | |
| 407 | f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 | |
|  | ++) { | |
| 408 | V a l u e L i s t [ tmp0 ] . s c o r e / = 1 0 ; | |
| 409 | } | |
| 410 | } | |
| 411 | r −> l i t e r a l > 0 ? V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . pos ++ | |
|  | : V a l u e L i s t [ abs ( r −> l i t e r a l ) ] . nev ++; | |
| 412 |  | r = r −> n e x t \_ l i t e r a l ; |
| 413 |  | } |
| 414 |  | q = q−> n e x t \_ s a m e \_ l i t e r a l ; |
| 415 | } |  |
| 416 | } | |
| 417 | s t p = 0 ; | |
| 418 | f o r ( tmp0 = 1 ; tmp0 <= cnf −> l i t e r a l \_ n u m ; tmp0 ++) { | |

number ) ;

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 419 | i f | ( ! V a l u e L i s t [ tmp0 ] . i s \_ v a l u e && V a l u e L i s t [ tmp0 ] . s c o r e | >= |
|  |  | s t p ) { |  |
| 420 |  | s t p = V a l u e L i s t [ tmp0 ] . s c o r e ; |  |
| 421 |  | l i t e r a l = tmp0 ; |  |
| 422 | } |  |  |
| 423 | } | | |
| 424 | c n t ++; | | |
| 425 | */ / g e t t i m e o f d a y (& ed , NULL) ;* | | |
| 426 |  | | |
| 427 | t i m e \_ t o t a l = ( ed . t v \_ s e c − s t . t v \_ s e c ) + ( ed . t v \_ u s e c − s t . | | |

t v \_ u s e c ) / 1 0 0 0 0 0 0 . 0 ;

1. t ime\_sum += t i m e \_ t o t a l ;
2. */ / p r i n t f (”[ t i m e \_ t o t a l%d]:% l f S \ n ” , cnt , t i m e \_ t o t a l ) ;*
3. i f ( V a l u e L i s t [ abs ( l i t e r a l ) ] . pos >= V a l u e L i s t [ abs ( l i t e r a l ) ] . nev )
4. r e t u r n abs ( l i t e r a l ) ;
5. e l s e
6. r e t u r n − abs ( l i t e r a l ) ;

|  |  |  |
| --- | --- | --- |
| 434 | } | |
| 435 |  | |
| 436 | */ /* 定义 *decide* 函数， 进行变量决策 | |
| 437 | s t a t u s d e c i d e ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) { | |
| 438 | i f ( p r e \_ t o p ) | |
| 439 | r e t u r n TRUE; | |
| 440 | i n t p a r t \_ l i t e r a l = s e l e c t \_ l i t e r a l 3 ( c n f ) ; | |
| 441 |  | i f ( ! p a r t \_ l i t e r a l ) { |
| 442 |  | r e t u r n FALSE ; |
| 443 |  | } |
| 444 |  | e l s e { |
| 445 |  | d e c i s i o n \_ s t a c k [ d e c \_ t o p ] = p a r t \_ l i t e r a l ; |
| 446 |  | decision\_num [ d e c \_ t o p ] = 1 ; |
| 447 |  | d e c \_ t o p ++; |
| 448 |  | r e t u r n TRUE; |
| 449 |  | } |
| 450 | } |  |

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |
| --- | --- | --- |
| 451 |  |  |
| 452 | */ /* 定义 *bcp* 过程， 实现单子句传播 |
| 453 | s t a t u s bcp ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) { |  |
| 454 | i n t l i t e r a l ; |  |
| 455 | i f ( d e c \_ t o p ) { |  |
| 456 | l i t e r a l = d e c i s i o n \_ s t a c k [ d e c \_ t o p − 1 ] ; |  |
| 457 | p r e h a n d l e \_ s t a c k [ p r e \_ t o p ++] = l i t e r a l ; |  |
| 458 | } |  |
| 459 | L i t e r a l \_ L i s t x = NULL; |  |
| 460 | C l a u s e \_ L i s t p = NULL; |  |
| 461 | w h i l e ( p r e \_ t o p ) { |  |
| 462 | i n t p r e \_ l i t e r a l = p r e h a n d l e \_ s t a c k [ −− p r e \_ t o p ] ; |  |
| 463 | */ / p r i n t f (”(% d ) ” , p r e \_ l i t e r a l ) ;* |  |
| 464 | V a l u e L i s t [ abs ( p r e \_ l i t e r a l ) ] . i s \_ v a l u e = ( p r e \_ l i t e r a l | > 0 ? |

1 : −1) ;

465 V a l u e L i s t [ abs ( p r e \_ l i t e r a l ) ] . dep = d e c \_ t o p ;

466 f o r ( x = V a l u e L i s t [ abs ( p r e \_ l i t e r a l ) ] . f i r s t ; x ; x = x−>

|  |  |  |
| --- | --- | --- |
|  | n e x t \_ s a m e \_ l i t e r a l ) { |  |
| 467 | p = x−>head ; |
| 468 | i f ( p−>number == 1 ) { |
| 469 | c o n t i n u e ; |
| 470 | } |
| 471 | i f ( ( abs ( p−> w a t c h \_ l i t e r a l [ 0 ] ) == | abs ( p r e \_ l i t e r a l ) ) | | |
|  | ( abs ( p−> w a t c h \_ l i t e r a l [ 1 ] ) == | abs ( p r e \_ l i t e r a l ) ) ) { |
| 472 | i n t up , s t ; |  |
| 473 | i f ( abs ( p−> w a t c h \_ l i t e r a l [ 0 ] ) | == abs ( p r e \_ l i t e r a l ) ) |

{

474 up = 0 ;

475 s t = 1 ;

476 }

477 e l s e {

478 up = 1 ;

479 s t = 0 ;

480 }

481 i f ( p−> w a t c h \_ l i t e r a l [ up ] == p r e \_ l i t e r a l ) {

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 482 | c o n t i n u e ; | | |
| 483 | } | | |
| 484 | L i t e r a l \_ L i s t q = NULL; | | |
| 485 | f o r ( q | | = p−> f i r s t \_ l i t e r a l ; q ; q = q−> n e x t \_ l i t e r a l |
|  | ) { | |  |
| 486 | i f | | ( q−> l i t e r a l == p−> w a t c h \_ l i t e r a l [ 0 ] | | q−> |
|  |  | | l i t e r a l == p−> w a t c h \_ l i t e r a l [ 1 ] ) { |
| 487 |  | | c o n t i n u e ; |
| 488 | } | |  |
| 489 | i f | | ( q−> l i t e r a l \* V a l u e L i s t [ abs ( q−> l i t e r a l ) ] . i s \_ v a l u e < 0 ) { |
| 490 |  | c o n t i n u e ; | |
| 491 |  | } | |
| 492 |  | b r e a k ; | |
| 493 | } |  | |
| 494 | i f | ( q == NULL) { | |

495 i f ( ! V a l u e L i s t [ abs ( p−> w a t c h \_ l i t e r a l [ s t ] ) ] . i s \_ v a l u e ) {

496 p r e h a n d l e \_ s t a c k [ p r e \_ t o p ++] = p−> w a t c h \_ l i t e r a l [ s t ] ;

497 }

498 e l s e i f ( p−> w a t c h \_ l i t e r a l [ s t ] \* V a l u e L i s t [ abs ( p−> w a t c h \_ l i t e r a l [ s t ] ) ] . i s \_ v a l u e < 0 ) {

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 499 |  | | | | c o n f l i c t \_ s t a c k \_ [ con\_top ++] = p r e \_ l i t e r a l ; |
| 500 |  | | | | r e t u r n FALSE ; |
| 501 | } | | | |  |
| 502 | } | | | |  |
| 503 | e l s e { | | | |  |
| 504 |  |  |  | p−> w a t c h \_ l i t e r a l [ up ] = q−> l i t e r a l ; | |
| 505 |  |  |  | } | |
| 506 |  |  | } |  | |
| 507 |  | } |  |  | |
| 508 |  | } |  |  | |
| 509 |  | r e t u r n | TRUE; |  | |
| 510 | } |  |  |  | |

## 华 中 科 技 大 学 课 程 实 验 报 告

511

512 */ /* 定义 *r e s o l v e C o n f l i c t* 函数， 解决冲突

513 s t a t u s r e s o l v e C o n f l i c t ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) {

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 514 | */ /* | | *p r i n t f (” a ”) ;* | | |
| 515 | i n t i ; | |  | | |
| 516 |  | w h i l e ( d e c \_ t o p > 0 && decision\_num [ d e c \_ t o p − 1 ] == 2 ) | | { |  |
| 517 |  | decision\_num [ d e c \_ t o p − 1 ] = 0 ; | |  |  |
| 518 |  | dec\_top − −; | |  |  |
| 519 |  | } | |  |  |
| 520 |  | i f ( ! d e c \_ t o p ) { | |  |  |
| 521 |  | r e t u r n FALSE ; | |  |  |
| 522 |  | } | |  |  |
| 523 |  | decision\_num [ d e c \_ t o p − 1 ] + + ; | |  |  |
| 524 |  | d e c i s i o n \_ s t a c k [ d e c \_ t o p − 1 ] = − d e c i s i o n \_ s t a c k [ d e c \_ t o p | | − | 1 ] ; |
| 525 |  | p r e \_ t o p = 0 ; | |  |  |
| 526 |  | f o r ( i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) { | |  |  |
| 527 |  | i f ( V a l u e L i s t [ i ] . dep >= d e c \_ t o p ) { | |  |  |
| 528 |  | V a l u e L i s t [ i ] . i s \_ v a l u e = 0 ; | |  |  |
| 529 |  | V a l u e L i s t [ i ] . dep = 0 ; | |  |  |
| 530 |  | } | |  |  |
| 531 |  | } | |  |  |
| 532 |  | r e t u r n TRUE; | |  |  |
| 533 | } |  | |  |  |
| 534 |  |  | |  |  |

1. */ /* 定义*DPLL1* 函数， 作为处理 *c n f* 文件的第一个*DPLL* 文件
2. s t a t u s DPLL2 ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) {
3. u p d a t e \_ v a l u e ( c n f ) ;
4. i f ( i s C o n f l i c t ) {
5. */ / c o n f l i c t i n s i n g l e l i t e r a l c l a u s e s*
6. r e t u r n FALSE ;

541 }

542 w h i l e ( TRUE) {

543 i f ( ! d e c i d e ( c n f ) ) {

544 */ / p r i n t f (”[ t o t a l \_ t i m e \_ s u m ]:% l f ” , t ime\_sum*

*\* 1000 ) ;*

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 545 | r e t u r n t r u e ; | | | | | |
| 546 | } | | | | | |
| 547 | w h i l e ( ! bcp ( c n f ) ) { | | | | | |
| 548 | c n t 0 ++; | | | | | |
| 549 | i f ( ! r e s o l v e C o n f l i c t ( c n f ) ) { | | | | | |
| 550 |  |  |  |  | */ /* | *c h e c k \_ p r i n t 2 ( c n f ) ;* |
| 551 |  |  |  |  | r e t u r n f a l s e ; |  |
| 552 |  |  |  | } |  |  |
| 553 |  |  | } |  |  |  |
| 554 |  | } |  |  |  |  |
| 555 | } |  |  |  |  |  |

文件名：func\_SUDOKU.cpp

功能：处理数独问题相关函数实现

1

*/ /*

*/ / Created by hp on 2022 −09 −04.*

*/ /*

# i n c l u d e *” d e f i n e . h”*

i n t s u d o k u \_ t a b l e [ 1 8 ] [ 9 ] ; */ /* 定义全局变量 *i n t* 类型二维数组存储双数独终盘

i n t u s e r s \_ s u d o k u [ 1 8 ] [ 9 ] ; */ /* 存储输出的含空格数独问题格局

i n t s h u f f l e \_ v a l u e [ 1 6 4 ] ; */ /* 存储被打乱的变元

e x t e r n Argue Value\* V a l u e L i s t ; */ /* 定义变元真值表

enum { e as y = 1 , medium , d i f f i c u l t } ;

i n t easynum = 50 , midnum = 80 , difnum = 11 0 ;

*/ /* 定义 *Create Pre Sudoku File* 函数， 规约数独为*CNF*文件

FILE\* C r e a t e P r e S u d o k u F i l e ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t& c n f ) { i n t x , y , z , i , j , k , l ;

FILE\* fp ;

fp = fopen ( *” Sudoku Table Base . c n f ”* , *”w”* ) ;

i f ( fp == NULL) {

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

## 华 中 科 技 大 学 课 程 实 验 报 告

21

22

23 }

p r i n t f ( *” F i l e r r o r ! \ n”* ) ; e x i t ( 0 ) ;

24 f p r i n t f ( fp , *”p n f \ n”* ) ; */ /* 共有 *729* 个变元， *9\*9* 个数独空格每个格对应*9* 个变元， 填入*1* ～ *9* 中某一值则对应变元为真， 其他为假

25

26 */ /* 每个格内只能含有*1−9* 中的一个

27 f o r ( x = 0 ; x < 9 ; x ++) { 28 f o r ( y = 0 ; y < 9 ; y ++)

29 f o r ( z = 1 ; z < 9 ; z ++)

30 f o r ( i = z + 1 ; i <= 9 ; i ++)

31 f p r i n t f ( fp , *”%d \ n”* , −(81 \* x + 9 \* y + z

) , −(81 \* x + 9 \* y + i ) ) ;

32 }

33 f o r ( x = 9 ; x < 1 8 ; x ++) { 34 f o r ( y = 0 ; y < 9 ; y ++)

35 f o r ( z = 1 ; z < 9 ; z ++)

36 f o r ( i = z + 1 ; i <= 9 ; i ++)

37 f p r i n t f ( fp , *”%d \ n”* , −(81 \* x + 9 \* y + z

) , −(81 \* x + 9 \* y + i ) ) ;

38 }

39 */ /* 每一行*1−9* 只出现一次

40 f o r ( x = 0 ; x < 9 ; x ++) {

41 f o r ( z = 1 ; z <= 9 ; z ++)

42 f o r ( y = 0 ; y < 8 ; y ++)

43 f o r ( i = y + 1 ; i <= 8 ; i ++)

44 f p r i n t f ( fp , *”%d \ n”* , −(81 \* x + 9 \* y + z

) , −(81 \* x + 9 \* i + z ) ) ;

45 }

46

47 f o r ( x = 9 ; x < 1 8 ; x ++) { 48 f o r ( z = 1 ; z <= 9 ; z ++)

49 f o r ( y = 0 ; y < 8 ; y ++)

50 f o r ( i = y + 1 ; i <= 8 ; i ++)

## 华 中 科 技 大 学 课 程 实 验 报 告

51

52 }

f p r i n t f ( fp , *”%d \ n”* , −(81 \* x + 9 \* y + z

) , −(81 \* x + 9 \* i + z ) ) ;

53 */ /* 每一列*1−9* 只出现一次

54 f o r ( y = 0 ; y < 9 ; y ++) {

55 f o r ( z = 1 ; z <= 9 ; z ++)

56 f o r ( x = 0 ; x < 8 ; x ++)

57 f o r ( i = x + 1 ; i <= 8 ; i ++)

58 f p r i n t f ( fp , *”%d \ n”* , −(81 \* x + 9 \* y + z

) , −(81 \* i + 9 \* y + z ) ) ;

59 }

60

61 f o r ( y = 0 ; y < 9 ; y ++) {

62 f o r ( z = 1 ; z <= 9 ; z ++)

63 f o r ( x = 9 ; x < 1 7 ; x ++)

64 f o r ( i = x + 1 ; i <= 1 7 ; i ++)

65 f p r i n t f ( fp , *”%d \ n”* , −(81 \* x + 9 \* y + z

) , −(81 \* i + 9 \* y + z ) ) ;

66 }

67 */ /* 每一宫*1−9* 只出现一次

68 f o r ( i = 0 ; i < 3 ; i ++) { 69 f o r ( j = 0 ; j < 3 ; j ++)

70 f o r ( z = 1 ; z <= 9 ; z ++)

71 f o r ( x = 0 ; x < 3 ; x ++)

72 f o r ( y = 0 ; y < 3 ; y ++)

73 f o r ( k = x + 1 ; k < 3 ; k ++)

74 f o r ( l = 0 ; l < 3 ; l ++)

75 i f ( y ! = l )

76 f p r i n t f ( fp , *”%d \ n”* , −(81

\* ( 3 \* i + x ) + 9 \* ( 3 \*

j + y ) + z ) ,

77 −(81 \* ( 3 \* i + k ) + 9 \*

( 3 \* j + l ) + z ) ) ;

78 }

79

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 华 中 科 技 大 学 课 程 实 | 验 报 告 |
| 80 | f o r | ( i = 0 ; i < 3 ; i ++) { |  |
| 81 |  | f o r ( j = 0 ; j < 3 ; j ++) |  |
| 82 |  | f o r ( z = 1 ; z <= 9 ; z ++) |  |
| 83 |  | f o r ( x = 0 ; x < 3 ; x ++) |  |
| 84 |  | f o r ( y = 0 ; y < 3 ; y ++) |  |
| 85 |  | f o r ( k = x + 1 ; k < 3 ; | k ++) |
| 86 |  | f o r ( l = 0 ; l < 3 ; | l ++) |
| 87 |  | i f ( y ! = l ) |  |
| 88 | f p r i n t f ( fp , *”%d \ n”* , | | |
|  | −( 729 + 81 \* ( 3 \* i + x ) + | | |

89

90 }

91 */ /* 每一格一定要有*1−9* 中的一个数字

92 f o r ( x = 0 ; x < 9 ; x ++) {

93 f o r ( y = 0 ; y < 9 ; y ++) {

94 f o r ( z = 1 ; z <= 9 ; z ++)

9 \* ( 3 \* j + y ) + z ) ,

−( 729 + 81 \* ( 3 \* i + k )

+ 9 \* ( 3 \* j + l ) + z )

) ;

95

96

97 }

98 }

99

f p r i n t f ( fp , *”%d* , 81 \* x + 9 \* y + z ) ; f p r i n t f ( fp , *” 0 \ n”* ) ;

100 f o r ( x = 9 ; x < 1 8 ; x ++) {

101 f o r ( y = 0 ; y < 9 ; y ++) {

102 f o r ( z = 1 ; z <= 9 ; z ++)

103 f p r i n t f ( fp , *”%d*, 81 \* x + 9 \* y + z ) ;

104 f p r i n t f ( fp , *” 0 \ n”* ) ;

105 }

106 }

107 */ /* 双数独重叠部分等价关系108 f o r ( x = 6 ; x < 9 ; x ++) { 109 f o r ( y = 6 ; y < 9 ; y ++)

110 f o r ( z = 1 ; z <= 9 ; z ++) {

## 华 中 科 技 大 学 课 程 实 验 报 告

111

112

113

114 }

f p r i n t f ( fp , *”%d \ n%d \ n”* , −(81 \* x + 9 \*

y + z ) , ( 729 + 81 \* ( x − 6 ) + 9 \* ( y − 6 ) + z )

,

( 81 \* x + 9 \* y + z ) , −( 729 + 81 \* ( x − 6 ) + 9 \* ( y − 6 ) + z ) ) ;

}

115

116

117

118 } 119

f c l o s e ( fp ) ;

fp = fopen ( *” Sudoku Table Base . c n f ”* , *” r”* ) ; r e t u r n fp ;

1. voi d swap ( i n t& x , i n t& y ) {
2. i n t tmp = x ;
3. x = y ;
4. y = tmp ;
5. r e t u r n ;

125 } 126

1. */ /* 定义 *Create Sudoku* 函数， 创建数独终盘
2. s t a t u s Create Sudoku ( ) {
3. s r a n d ( i n t ( t i me ( 0 ) ) ) ;
4. i n t i , j , k ;

131 f o r ( i = 0 ; i < 9 ; i ++) {

132 s u d o k u \_ t a b l e [ 0 ] [ i ] = i + 1 ;

133 }

134 f o r ( i = 0 ; i < 9 ; i ++) {

135 j = r a nd ( ) % 9 ;

136 i f ( i ! = j ) {

137 swap ( s u d o k u \_ t a b l e [ 0 ] [ i ] , s u d o k u \_ t a b l e [ 0 ] [ j ] ) ;

138 }

139 }

140 i n t move [ ] = { 0 , 3 , 6 , 1 , 4 , 7 , 2 , 5 , 8 } ; 141 f o r ( j = 1 ; j < 9 ; j ++) {

142 i n t tmp [ 9 ] , t a g = 0 ;

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 143 | f o r | | | ( i = 8 − move [ j ] + 1 ; i <= 8 ; i ++) { | |
| 144 |  | | | tmp [ t a g ++] = s u d o k u \_ t a b l e [ 0 ] [ i ] ; | |
| 145 | } | | |  | |
| 146 | f o r | | | ( i = 0 ; i <= 8 − move [ j ] ; i ++) { | |
| 147 |  | | | s u d o k u \_ t a b l e [ j ] [ i + move [ j ] ] = s u d o k u \_ t a b l e [ 0 ] [ i ] ; | |
| 148 | } | | |  | |
| 149 | f o r | | | ( i = 0 ; i < t a g ; i ++) { | |
| 150 |  | | | s u d o k u \_ t a b l e [ j ] [ i ] = tmp [ i ] ; | |
| 151 | } | | |  | |
| 152 | } | | | |  |
| 153 | s u d o k u \_ t a b l e [ 9 ] [ 0 ] = s u d o k u \_ t a b l e [ 6 ] [ 6 ] , s u d o k u \_ t a b l e [ 9 ] [ 1 ] | | | | = |
|  | s u d o k u \_ t a b l e [ 6 ] [ 7 ] , s u d o k u \_ t a b l e [ 9 ] [ 2 ] = s u d o k u \_ t a b l e | | | |  |
|  | [ 6 ] [ 8 ] ; | | | |  |
| 154 | s u d o k u \_ t a b l e [ 9 ] [ 3 ] = s u d o k u \_ t a b l e [ 8 ] [ 6 ] , s u d o k u \_ t a b l e [ 9 ] [ 4 ] | | | | = |
|  | s u d o k u \_ t a b l e [ 8 ] [ 7 ] , s u d o k u \_ t a b l e [ 9 ] [ 5 ] = s u d o k u \_ t a b l e | | | |  |
|  | [ 8 ] [ 8 ] ; | | | |  |
| 155 | s u d o k u \_ t a b l e [ 9 ] [ 6 ] = s u d o k u \_ t a b l e [ 7 ] [ 6 ] , s u d o k u \_ t a b l e [ 9 ] [ 7 ] | | | | = |
|  | s u d o k u \_ t a b l e [ 7 ] [ 7 ] , s u d o k u \_ t a b l e [ 9 ] [ 8 ] = s u d o k u \_ t a b l e | | | |  |
|  | [ 7 ] [ 8 ] ; | | | |  |
| 156 | f o r | | ( j = | 1 0 ; j < 1 8 ; j ++) { | |
| 157 |  | | i n t | tmp [ 9 ] , t a g = 0 ; | |
| 158 |  | | f o r | ( i = 8 − move [ j − 9 ] + 1 ; i <= 8 ; i ++) { | |
| 159 |  | |  | tmp [ t a g ++] = s u d o k u \_ t a b l e [ 9 ] [ i ] ; | |
| 160 |  | | } |  | |
| 161 |  | | f o r | ( i = 0 ; i <= 8 − move [ j − 9 ] ; i ++) { | |
| 162 |  | |  | s u d o k u \_ t a b l e [ j ] [ i + move [ j − 9 ] ] = s u d o k u \_ t a b l e [ 9 ] [ i | |
|  |  | |  | ] ; | |
| 163 |  | | } |  | |
| 164 |  | | f o r | ( i = 0 ; i < t a g ; i ++) { | |
| 165 |  | |  | s u d o k u \_ t a b l e [ j ] [ i ] = tmp [ i ] ; | |
| 166 |  | | } |  | |
| 167 |  | } | | | |
| 168 |  | r e t u r n OK; | | | |
| 169 | } |  | | | |
| 170 |  |  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 华 | | 中 科 技 大 学 课 程 实 验 报 告 |
| 171 | e x t e r n i n t | | p r e h a n d l e \_ s t a c k [MAXN] ; |
| 172 | e x t e r n i n t | | p r e \_ t o p , d e c \_ t o p ; |
| 173 |  | |  |
| 174 | */ /* 定义*DPLL2\_SUDOKU*函数， 调用*SAT* 求解器 | | |
| 175 | s t a t u s DPLL2\_SUDOKU( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t cnf , i n t t [ ] , | | |
|  | i n t t o p ) { | | |
| 176 | */ / p r i n t f (”%d ” , t op ) ;* | | |
| 177 | p r e \_ t o p = 0 ; | | |
| 178 | d e c \_ t o p = 0 ; | | |
| 179 | i n t i ; | | |
| 180 | f o r ( i = 0 ; i < t o p ; i ++) { | | |
| 181 | p r e h a n d l e \_ s t a c k [ p r e \_ t o p ++] = t [ i ] ; | | |
| 182 | } | | |
| 183 | */ / p r i n t f (” pre\_top%d ” , pre\_top ) ;* | | |
| 184 | */ / p r i n t f (” l l l l l % d l l l l ” , cnf −>f i r s t \_ c l a u s e −>w a t c h \_ l i t e r a l* | | |
| *[ 0 ] ) ;* | | | |
| 185 |  | r e t u r n DPLL2 ( c n f ) ; | |
| 186 | } |  | |
| 187 |  |  | |
| 188 | */ /* 定 | 义 *s udokusat* 函数， 求解数独问题 | |

1. s t a t u s s u d o k u s a t ( i n t m, i n t n , C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f )

{

1. i n t x , y , z , t , i ;
2. i n t tmp [ 2 0 0 0 ] , tmp0 , ans = 0 ; 192 f o r ( t = 1 ; t <= 9 ; t ++) {

193 tmp0 = 0 ;

194 p r e \_ t o p = 0 , d e c \_ t o p = 0 ;

195 f o r ( x = 0 ; x < 1 8 ; x ++)

196 f o r ( y = 0 ; y < 9 ; y ++) {

197 i f ( u s e r s \_ s u d o k u [ x ] [ y ] ! = 1 ) {

198 f o r ( z = 1 ; z <= 9 ; z ++) {

199 i f ( z == s u d o k u \_ t a b l e [ x ] [ y ] ) {

200 tmp [ tmp0 ++] = 81 \* x + 9 \* y + z ;

201 }

202 e l s e {

## 华 中 科 技 大 学 课 程 实 验 报 告

203 tmp [ tmp0 ++] = −(81 \* x + 9 \* y + z ) ;

204 }

205 }

206 }

207 }

208 f o r ( z = 1 ; z <= 9 ; z ++) {

209 i f ( z == t ) {

210 tmp [ tmp0 ++] = 81 \* m + 9 \* n + z ;

211 }

212 e l s e {

213 tmp [ tmp0 ++] = −(81 \* m + 9 \* n + z ) ;

214 }

215 }

1. ans += DPLL2\_SUDOKU( cnf , tmp , tmp0 ) ;
2. f o r ( i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) {
3. V a l u e L i s t [ i ] . i s \_ v a l u e = 0 ;
4. V a l u e L i s t [ i ] . t a g = 0 ;
5. V a l u e L i s t [ i ] . dep = 0 ;
6. V a l u e L i s t [ i ] . num = 0 ;
7. V a l u e L i s t [ i ] . nev = 0 ;
8. V a l u e L i s t [ i ] . pos = 0 ;
9. V a l u e L i s t [ i ] . s c o r e = 0 ;

225 }

1. */ / p r i n t f (”%d ” , ans ) ;*
2. i f ( ans > 1 ) r e t u r n FALSE ;

228 }

229

230 } 231

r e t u r n t r u e ;

1. */ /* 定义 *Dig\_Hole\_Easy* 函数， 挖洞法生成数独
2. s t a t u s Dig\_Hole\_Easy ( i n t d i g ) {
3. i n t dig\_num = d i g \* 4 5 ;
4. s r a n d ( i n t ( t i me ( 0 ) ) ) ;
5. i n t x , y , ans , randnum , d , c n t = 1 , i , b , z ;
6. C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ;

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |
| --- | --- |
| 238 | FILE\* fp = C r e a t e P r e S u d o k u F i l e ( c n f ) ; |
| 239 | Create CNF2 ( fp , c n f ) ; |
| 240 | f o r ( x = 0 ; x < 1 8 ; x ++) |
| 241 | f o r ( y = 0 ; y < 9 ; y ++) |
| 242 | u s e r s \_ s u d o k u [ x ] [ y ] = 0 ; |
| 243 | f o r ( i = 1 ; i <= 1 62 ; i ++) |
| 244 | s h u f f l e \_ v a l u e [ i ] = i ; |
| 245 | f o r ( i = 16 2 ; i > 2 ; i − −) { |
| 246 | randnum = r a n d ( ) % i + 1 ; |
| 247 | i f ( i ! = randnum ) { |
| 248 | d = s h u f f l e \_ v a l u e [ i ] ; |
| 249 | s h u f f l e \_ v a l u e [ i ] = s h u f f l e \_ v a l u e [ randnum ] ; |
| 250 | s h u f f l e \_ v a l u e [ randnum ] = d ; |
| 251 | } |
| 252 | } |
| 253 | f o r ( i = 1 ; i <= dig\_num && c n t < 1 62 ; i ++) { |
| 254 | x = ( s h u f f l e \_ v a l u e [ c n t ] − 1 ) / 9 ; |
| 255 | y = ( s h u f f l e \_ v a l u e [ c n t ] − 1 ) % 9 ; |
| 256 | */ / x = rand ( ) % 18 ;* |
| 257 | */ / y = rand ( ) % 9 ;* |
| 258 | c n t ++; |
| 259 | i f ( u s e r s \_ s u d o k u [ x ] [ y ] ! = −1 && u s e r s \_ s u d o k u [ x ] [ y ] ! = 1 ) |
|  | { |
| 260 | u s e r s \_ s u d o k u [ x ] [ y ] = 1 ; |
| 261 | i f ( x >= 6 && x <= 8 && y >= 6 && y <= 8 ) { |
| 262 | u s e r s \_ s u d o k u [ x + 3 ] [ y − 6 ] = 1 ; |
| 263 | } |
| 264 | e l s e i f ( x >= 9 && x <= 11 && y >= 0 && y <= 2 ) { |
| 265 | u s e r s \_ s u d o k u [ x − 3 ] [ y + 6 ] = 1 ; |
| 266 | } |
| 267 | ans = s u d o k u s a t ( x , y , c n f ) ; |
| 268 |  |
| 269 | i f ( ! ans ) { |
| 270 | u s e r s \_ s u d o k u [ x ] [ y ] = − 1 ; |
| 271 | i f ( x >= 6 && x <= 8 && y >= 6 && y <= 8 ) { |

华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 272 |  | u s e r s \_ s u d o k u [ x | + | 3 ] [ y − 6 ] = − 1 ; |  | | | | |
| 273 |  | } |  |  |
| 274 |  | e l s e i f ( x >= 9 && | x | <= 11 && y >= 0 | && | y | <= | 2 ) | { |
| 275 |  | u s e r s \_ s u d o k u [ x | − | 3 ] [ y + 6 ] = − 1 ; |  | | | | |
| 276 |  | } |  |  |
| 277 |  | i − −; |  |  |
| 278 |  | c o n t i n u e ; |  |  |
| 279 |  | } |  |  |
| 280 |  | c o n t i n u e ; |  |  |
| 281 | } |  |  |  |
| 282 | i − −; |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 283 |  | } |  |
| 284 |  | p r i n t \_ t i m e ( c n f ) ; |
| 285 |  | r e t u r n OK; |
| 286 | } |  |
| 287 |  |  |
| 288 | */ /* 定 | 义 *p r i n t \_ t i m e* 函数， 输出求解时间 |
| 289 | i n t | p r i n t \_ t i m e ( C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ) | { |
| 290 |  | c l o c k \_ t s t a r t , end ; |  |
| 291 |  |  |  |
| 292 |  | i n t x , y , z , t , i ; |  |
| 293 |  | i n t tmp [ 2 0 0 0 ] , tmp0 , ans = 0 ; |  |
| 294 |  | tmp0 = 0 ; |  |
| 295 |  | p r e \_ t o p = 0 , d e c \_ t o p = 0 ; |  |
| 296 |  | f o r ( x = 0 ; x < 1 8 ; x ++) |  |
| 297 |  | f o r ( y = 0 ; y < 9 ; y ++) { |  |
| 298 |  | i f ( u s e r s \_ s u d o k u [ x ] [ y ] ! = 1 ) { |  |
| 299 |  | f o r ( z = 1 ; z <= 9 ; z ++) { |  |
| 300 |  | i f ( z == s u d o k u \_ t a b l e [ x ] [ y ] ) | { |

301 tmp [ tmp0 ++] = 81 \* x + 9 \* y + z ;

302 }

303 e l s e {

304 tmp [ tmp0 ++] = −(81 \* x + 9 \* y + z ) ;

305 }

306 }

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 307 |  | | | } |
| 308 | } | | |  |
| 309 |  | s t a r t = c l o c k ( ) ; | | |
| 310 |  | ans += DPLL2\_SUDOKU( cnf , tmp , tmp0 ) ; | | |
| 311 |  | end = c l o c k ( ) ; | | |
| 312 |  | f o r ( i = 1 ; i <= cnf −> l i t e r a l \_ n u m ; i ++) { | | |
| 313 |  | V a l u e L i s t [ i ] . i s \_ v a l u e = 0 ; | | |
| 314 |  | V a l u e L i s t [ i ] . t a g = 0 ; | | |
| 315 |  | V a l u e L i s t [ i ] . dep = 0 ; | | |
| 316 |  | V a l u e L i s t [ i ] . num = 0 ; | | |
| 317 |  | V a l u e L i s t [ i ] . nev = 0 ; | | |
| 318 |  | V a l u e L i s t [ i ] . pos = 0 ; | | |
| 319 |  | V a l u e L i s t [ i ] . s c o r e = 0 ; | | |
| 320 |  | } | | |
| 321 |  | p r i n t f ( *”* 求解数独所用时间：*%dms \ n”* , end − s t a r t ) ; | | |
| 322 |  | r e t u r n t r u e ; | | |
| 323 | } |  | | |
| 324 |  |  | | |
| 325  326 | */ /* 定义 *p r i n t \_ t i m e* 函数， 输出求解时间  s t a t u s S u d o k u F i n a l P r i n t ( voi d ) { | | | |
| 327 | i n t | | x , y ; | |
| 328 | f o r | | ( x = 0 ; x < 1 8 ; x ++) { | |
| 329 |  | | p r i n t f ( *” \ n”* ) ; | |
| 330 |  | | f o r ( y = 0 ; y < 9 ; y ++) { | |
| 331 |  | | i f ( u s e r s \_ s u d o k u [ x ] [ y ] == 1 ) | |
| 332 |  | | p r i n t f ( *”0* ) ; | |
| 333 |  | | e l s e | |
| 334 |  | | p r i n t f ( *”%d* , s u d o k u \_ t a b l e [ x ] [ y ] ) ; | |
| 335 |  | | } | |
| 336 |  | } | | |
| 337 |  | r e t u r n 0 ; | | |
| 338 | } |  | | |

文件名：func\_front.cpp 功能：程序前端交互

## 华 中 科 技 大 学 课 程 实 验 报 告

1 */ /*

2 */ / Created by hp on 2022 −09 −09.*

3 */ /*

4

1. # i n c l u d e *” d e f i n e . h”*
2. C o n j u n c t i v e \_ N o r m a l \_ F o r m \_ L i s t c n f ; */ /* 定义 *c n f*
3. c h a r f i l e n a m e [ 1 0 0 ] ; */ /* 定义文件名

8

9

1. e x t e r n i n t s u d o k u \_ t a b l e [ 1 8 ] [ 9 ] ; */ /* 定义全局变量 *i n t* 类型二维数组存储双数独终盘
2. e x t e r n i n t u s e r s \_ s u d o k u [ 1 8 ] [ 9 ] ; */ /* 存储输出的含空格数独问题格局
3. e x t e r n i n t s h u f f l e \_ v a l u e [ 1 6 4 ] ; */ /* 存储被打乱的变元

13

14 e x t e r n Argue Value\* V a l u e L i s t ; */ /* 定义变元真值表

15

1. */ /* 定义 *s o l v e* 函数， 输出交互界面
2. voi d s o l v e ( i n t op ) {
3. s w i t c h ( op ) {
4. c a s e 1 : s o l v e \_ c n f ( ) ; b r e a k ;
5. c a s e 2 : s o l v e \_ s u d o k u ( ) ; b r e a k ;
6. d e f a u l t : b r e a k ;

22 }

23 r e t u r n ; 24 }

25

1. */ /* 定义 *s o l v e \_ c n f* 函数， 输出 *c n f* 文件求解交互界面
2. voi d s o l v e \_ c n f ( ) {
3. i n t c r e a t e \_ f l a g = 0 , d ;
4. c l o c k \_ t s t a r t = 0 , f i n i s h = 0 ; */ /* 记录*DPLL* 函数调用的起始和终止时间
5. i n t d u r a t i o n = 0 ; */ /* 记录*SAT* 求解时间
6. system ( ( *” c l s ”* ) ) ;
7. p r i n t f ( *”* 请输入想要求解的 *c n f* 文件的名称： *”* ) ;
8. s c a n f ( *”%s”* , f i l e n a m e ) ;

## 华 中 科 技 大 学 课 程 实 验 报 告

1. FILE\* fp ; 35
2. i f ( ( fp = fopen ( f i l e n a m e , *” r”* ) ) == NULL) {
3. p r i n t f ( *” F a i l i l e ! \ n”* ) ;
4. e x i t ( 0 ) ; */ /* 退出程序（ 结束程序）

39 }

40

41 i n t op1 ; 42

1. p r i n t f ( *”* 请选择求解方案： *\ n”* ) ;
2. p r i n t f ( *”* 方案*0*： 未优化前顺序选取策略 *\ n”* ) ;
3. p r i n t f ( *”* 方案*1*： 最优非递归求解策略 *\ n”* ) ;
4. p r i n t f ( *”* 方案*2*： 递归最短子句优先策略 *\ n”* ) ;
5. p r i n t f ( *”* 方案*3*： 递归最大频率优先策略 *\ n”* ) ;
6. p r i n t f ( *”* 方案*4*： 递归*VSIDS* 选取策略 *\ n”* ) ; 49

50 s c a n f ( *”%d”* , &op1 ) ; 51

1. s w i t c h ( op1 ) {
2. c a s e 0 :
3. c r e a t e \_ f l a g = CreateCNF ( fp , c n f ) ;
4. i f ( ! c r e a t e \_ f l a g ) {
5. p r i n t f ( *” F a i l r e a t e n f ! ”* ) ;
6. e x i t ( 0 ) ; */ /* 退出程序（ 结束程序）

58 }

1. s t a r t = c l o c k ( ) ;
2. d = DPLL1 ( cnf , 0 ) ;
3. */ / c h e c k \_ p r i n t 2 ( c n f ) ;*
4. */ / p r i n t 2 ( c n f ) ;*
5. f i n i s h = c l o c k ( ) ;
6. d u r a t i o n = ( f i n i s h − s t a r t ) ; 65

66 p r i n t f ( *”* 求解文件用时*%dms \ n”* , d u r a t i o n ) ;

67 i f ( d == 1 ) {

68 p r i n t f ( *”* 该 *c n f* 文件有解！ *\ n”* ) ;

## 华 中 科 技 大 学 课 程 实 验 报 告

69 }

70 e l s e

71

p r i n t f ( *”* 该 *c n f* 文件无解！ *\ n”* ) ;

1. b r e a k ;
2. c a s e 1 :
3. c r e a t e \_ f l a g = Create CNF2 ( fp , c n f ) ;
4. i f ( ! c r e a t e \_ f l a g ) {
5. p r i n t f ( *” F a i l r e a t e n f ! ”* ) ;
6. e x i t ( 0 ) ; */ /* 退出程序（ 结束程序）

78 }

1. s t a r t = c l o c k ( ) ;
2. d = DPLL2 ( c n f ) ;
3. */ / c h e c k \_ p r i n t 2 ( c n f ) ;*
4. */ / p r i n t 2 ( c n f ) ;*
5. f i n i s h = c l o c k ( ) ;
6. d u r a t i o n = ( f i n i s h − s t a r t ) ; 85

86 p r i n t f ( *”* 求解文件用时*%dms \ n”* , d u r a t i o n ) ;

87 i f ( d == 1 ) {

88 p r i n t f ( *”* 该 *c n f* 文件有解！ *\ n”* ) ; 89 }

1. e l s e
2. p r i n t f ( *”* 该 *c n f* 文件无解！ *\ n”* ) ;
3. b r e a k ;
4. c a s e 2 :
5. c r e a t e \_ f l a g = CreateCNF ( fp , c n f ) ;
6. i f ( ! c r e a t e \_ f l a g ) {
7. p r i n t f ( *” F a i l r e a t e n f ! ”* ) ;
8. e x i t ( 0 ) ; */ /* 退出程序（ 结束程序）

98 }

1. s t a r t = c l o c k ( ) ;
2. d = DPLL1 ( cnf , 1 ) ;
3. */ / c h e c k \_ p r i n t 2 ( c n f ) ;*
4. */ / p r i n t 2 ( c n f ) ;*
5. f i n i s h = c l o c k ( ) ;

## 华 中 科 技 大 学 课 程 实 验 报 告

1. d u r a t i o n = ( f i n i s h − s t a r t ) ; 105

106 p r i n t f ( *”* 求解文件用时*%dms \ n”* , d u r a t i o n ) ;

107 i f ( d == 1 ) {

108 p r i n t f ( *”* 该 *c n f* 文件有解！ *\ n”* ) ;

109 }

1. e l s e
2. p r i n t f ( *”* 该 *c n f* 文件无解！ *\ n”* ) ;
3. b r e a k ;
4. c a s e 3 :
5. c r e a t e \_ f l a g = CreateCNF ( fp , c n f ) ;
6. i f ( ! c r e a t e \_ f l a g ) {
7. p r i n t f ( *” F a i l r e a t e n f ! ”* ) ;
8. e x i t ( 0 ) ; */ /* 退出程序（ 结束程序）

118 }

1. s t a r t = c l o c k ( ) ;
2. d = DPLL1 ( cnf , 2 ) ;
3. */ / c h e c k \_ p r i n t 2 ( c n f ) ;*
4. */ / p r i n t 2 ( c n f ) ;*
5. f i n i s h = c l o c k ( ) ;
6. d u r a t i o n = ( f i n i s h − s t a r t ) ; 125

126 p r i n t f ( *”* 求解文件用时*%dms \ n”* , d u r a t i o n ) ;

127 i f ( d == 1 ) {

128 p r i n t f ( *”* 该 *c n f* 文件有解！ *\ n”* ) ;

129 }

1. e l s e
2. p r i n t f ( *”* 该 *c n f* 文件无解！ *\ n”* ) ;
3. b r e a k ;
4. c a s e 4 :
5. c r e a t e \_ f l a g = CreateCNF ( fp , c n f ) ;
6. i f ( ! c r e a t e \_ f l a g ) {
7. p r i n t f ( *” F a i l r e a t e n f ! ”* ) ;
8. e x i t ( 0 ) ; */ /* 退出程序（ 结束程序）

138 }

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |
| --- | --- | --- |
| 139 | s t a r | t = c l o c k ( ) ; |
| 140 | d = | DPLL1 ( cnf , 3 ) ; |
| 141 | */ /* | *c h e c k \_ p r i n t 2 ( c n f ) ;* |
| 142 | */ /* | *p r i n t 2 ( c n f ) ;* |

1. f i n i s h = c l o c k ( ) ;
2. d u r a t i o n = ( f i n i s h − s t a r t ) ; 145

146 p r i n t f ( *”* 求解文件用时*%dms \ n”* , d u r a t i o n ) ;

147 i f ( d == 1 ) {

148 p r i n t f ( *”* 该 *c n f* 文件有解！ *\ n”* ) ;

149 }

1. e l s e
2. p r i n t f ( *”* 该 *c n f* 文件无解！ *\ n”* ) ;
3. b r e a k ;
4. d e f a u l t :
5. p r i n t f ( *”* 选取策略无效， 请正确选取！ *\ n”* ) ;
6. b r e a k ;

156 }

1. f c l o s e ( fp ) ;
2. p r i n t f ( *”* 请选择是否保存解文件【 *y / n* 】： *”* ) ;
3. c h a r s [ 1 0 ] ;
4. s c a n f ( *”%s”* , s ) ;

161 i f ( s [ 0 ] == *’ y ’* ) {

162 s t o r e \_ d o c u m e n t ( cnf , f i l e n a m e , d , d u r a t i o n ) ;

163 }

1. i n t op2 = 1 ;
2. w h i l e ( op2 ) {
3. system ( *” c l s ”* ) ;
4. p r i n t f ( *”* 选择查看信息： *\ n”* ) ;
5. p r i n t f ( *” 1 .* 遍历 *c n f* 文件 *\ n”* ) ;
6. p r i n t f ( *” 2 .* 验证解的正确性 *\ n”* ) ;
7. p r i n t f ( *” 3 .* 查看解的结构 *\ n”* ) ;
8. p r i n t f ( *” 0 .* 退出 *\ n \ n”* ) ; 172

173 p r i n t f ( *”* 请选择想要查看的信息： *”* ) ;

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |
| --- | --- | --- |
| 174 | s c a n f | ( *”%d”* , &op2 ) ; |
| 175 | s w i t c | h ( op2 ) { |
| 176 | c a s e | 1 : t r a v e s e r \_ c n f ( c n f ) ; g e t c h a r ( ) ; g e t c h a r ( ) ; b r e a k ; |
| 177 | c a s e | 2 : p r o v e \_ c n f ( c n f ) ; g e t c h a r ( ) ; g e t c h a r ( ) ; b r e a k ; |
| 178 | c a s e | 3 : show\_cnf ( c n f ) ; g e t c h a r ( ) ; g e t c h a r ( ) ; b r e a k ; |
| 179 | c a s e | 0 : b r e a k ; |
| 180 | d e f a u l t : p r i n t f ( *”* 请正确选择查看信息， 否则请按*0* 退出*”* ) ; | |
|  | g e t c h a r ( ) ; g e t c h a r ( ) ; b r e a k ; | |
| 181 | } | |
| 182 | } | |
| 183 | f r e e ( V a l u e L i s t ) ; | |
| 184 | DestroyCNF ( c n f ) ; | |
| 185 |  | |
| 186 | } | |
| 187 |  | |
| 188 | */ /* 定义 *s o l v e\_sudoku* 函数， 输出数独求解交互界面 | |
| 189 | voi d s o l v e \_ s u d o k u ( ) { | |
| 190 | c h a r c , v ; | |
| 191 |  | |
| 192 | i n i t g r a p h ( 1 0 2 4 , 684 ) ; */ /* 创建绘图窗口， 大小为 *640 x 480* | |
|  | 像素 | |

193

1. s e t t e x t s t y l e ( 1 6 , 8 , \_T ( *” Courier ”* ) ) ; */ /* 设置字体
2. */ /* 设置颜色
3. s e t t e x t c o l o r (BLACK) ;
4. s e t l i n e c o l o r (BLACK) ;
5. s e t l i n e s t y l e ( PS\_SOLID , 2 ) ;

199 f i l l r e c t a n g l e ( 0 , 0 , 1024 , 700 ) ; 200

1. s e t f i l l c o l o r ( WHITE) ;
2. s e t l i n e c o l o r (BLACK) ; 203

204 s e t b k c o l o r ( WHITE) ; 205

206 IMAGE p i c t u r e ;

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 207 |  | | | |
| 208 | l oadimage (& p i c t u r e , *” c . j pg ”* , | | | 1024 , 700 ) ; |
| 209 |  | | |  |
| 210 | putimage ( 0 , 0 , & p i c t u r e ) ; | | |  |
| 211 |  | | |  |
| 212 | l oadimage (& p i c t u r e , *”d . j pg ”* , | | | 250 , 254 ) ; |
| 213 |  | | |  |
| 214 | putimage ( 7 7 4 , 446 , & p i c t u r e ) ; | | | |
| 215 | */ /* 欢迎进入数据结构的数独世界 | | | |
| 216 | s e t t e x t s t y l e ( 4 5 , 20 , \_T ( *”* 微软雅黑*”* ) ) ; | | | |
| 217 | o u t t e x t x y ( 2 3 0 , 300 , \_T ( *”* 欢迎进入数据结构的数独世界*”* ) ) ; | | | |
| 218 |  | | | |
| 219 | wchar\_t s t r 0 [ ] = L*”*请 按*ENTER* 键开始*”* ; | | | |
| 220 | s e t t e x t s t y l e ( 4 5 , 20 , \_T ( *”* 微软雅黑*”* ) ) ; | | | |
| 221 | o u t t e x t x y ( 3 3 0 , 400 , \_T ( *”*请按*ENTER* 键开始*”* ) ) ; | | | |
| 222 |  | | | |
| 223 | c h a r s 0 ; | | | |
| 224 |  | | | |
| 225 | w h i l e ( TRUE) { | | | |
| 226 | s 0 = g e t c h ( ) ; | | | |
| 227 | i f ( s 0 == 13 ) | | | |
| 228 | b r e a k ; | | | |
| 229 | } | | | |
| 230 | f i l l r e c t a n g l e ( 0 , 0 , 1024 , 700 ) ; | | | |
| 231 |  | | | |
| 232 | s e t t e x t s t y l e ( 5 0 , 25 , \_T ( *”* 微软雅黑*”* ) ) ; | | | |
| 233 | o u t t e x t x y ( 1 8 5 , 280 , \_T ( *”* 请选择难度： *”* ) ) ; | | | |
| 234 | s e t t e x t s t y l e ( 4 0 , 18 , \_T ( *”* 楷体*”* ) ) ; | | | |
| 235 | o u t t e x t x y ( 5 2 5 , | 150 , | \_T ( *”* 简单（ 请按数字盘*1* ） *”* ) ) ; | |
| 236 | o u t t e x t x y ( 5 2 5 , | 300 , | \_T ( *”* 中等（ 请按数字盘*2* ） *”* ) ) ; | |
| 237 | o u t t e x t x y ( 5 2 5 , | 450 , | \_T ( *”* 复杂（ 请按数字盘*3* ） *”* ) ) ; | |
| 238 |  |  |  | |
| 239 | i n t d i g \_ s l t = 0 | ; |  | |
| 240 |  |  |  | |
| 241 | w h i l e ( TRUE) { |  |  | |

华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 242 |  | s 0 | = g e t c h ( ) ; |  | |
| 243 |  | i f | ( s 0 == 49 ) { |
| 244 |  |  | d i g \_ s l t = 1 ; |
| 245 |  |  | o u t t e x t x y ( 3 2 5 , | 550 , | \_T ( *”* 正在加载中… … *”* ) ) ; |
| 246 |  |  | b r e a k ; |  |  |
| 247 |  | } |  |  |  |
| 248 |  | i f | ( s 0 == 50 ) { |  |  |
| 249 |  |  | d i g \_ s l t = 2 ; |  |  |
| 250 |  |  | o u t t e x t x y ( 3 2 5 , | 550 , | \_T ( *”* 正在加载中… … *”* ) ) ; |
| 251 |  |  | b r e a k ; |  |  |
| 252 |  | } |  |  |  |
| 253 |  | i f | ( s 0 == 51 ) { |  |  |
| 254 |  |  | d i g \_ s l t = 3 ; |  |  |
| 255 |  |  | o u t t e x t x y ( 3 2 5 , | 550 , | \_T ( *”* 正在加载中… … *”* ) ) ; |
| 256 |  |  | b r e a k ; |  |  |
| 257 |  | } |  |  |  |
| 258 | } |  |  |  |  |
| 259 |  |  |  |  |  |

1. Create Sudoku ( ) ;
2. Dig\_Hole\_Easy ( d i g \_ s l t ) ; 262

263 f i l l r e c t a n g l e ( 0 , 0 , 1024 , 700 ) ;

264 TCHAR o = \_T ( *’◻’* ) ;

265 f o r ( i n t i = 1 ; i <= 9 ; i ++)

266 f o r ( i n t j = 1 ; j <= 9 ; j ++) {

1. i f ( u s e r s \_ s u d o k u [ i − 1 ] [ j − 1 ] ! = 1 ) {
2. s e t f i l l c o l o r (YELLOW) ;

269 f i l l r e c t a n g l e ( 10 + 35 \* j , 10 + 35 \* i , 45 + 35 \*

j , 45 + 35 \* i ) ;

270 }

1. e l s e {
2. s e t f i l l c o l o r ( WHITE) ;

273 f i l l r e c t a n g l e ( 10 + 35 \* j , 10 + 35 \* i , 45 + 35 \*

j , 45 + 35 \* i ) ;

274 }

## 华 中 科 技 大 学 课 程 实 验 报 告

275

276 RECT r = { 10 + 35 \* j , 10 + 35 \* i , 45 + 35 \* j , 45

+ 35 \* i } ;

277 TCHAR s [ 5 ] ;

278 }

279 f o r ( i n t i = 1 ; i <= 9 ; i ++)

280 f o r ( i n t j = 1 ; j <= 9 ; j ++) {

1. i f ( u s e r s \_ s u d o k u [ i + 8 ] [ j − 1 ] ! = 1 ) {
2. s e t f i l l c o l o r (YELLOW) ;

283 f i l l r e c t a n g l e ( 10 + 35 \* ( 6 + j ) , 10 + 35 \* ( 6 + i

) , 45 + 35 \* ( 6 + j ) , 45 + 35 \* ( 6 + i ) ) ;

284 }

1. e l s e {
2. s e t f i l l c o l o r ( WHITE) ;

287 f i l l r e c t a n g l e ( 10 + 35 \* ( 6 + j ) , 10 + 35 \* ( 6 + i

) , 45 + 35 \* ( 6 + j ) , 45 + 35 \* ( 6 + i ) ) ;

288 }

289 TCHAR s [ 5 ] ;

290

1. \_ s t p r i n t f ( s , \_T ( *”%d”* ) , s u d o k u \_ t a b l e [ i + 8 ] [ j − 1 ] ) ;

*/ /* 高版本 *VC* 推荐使用 *\_ s t p r i n t f \_ s* 函数

1. s e t t e x t s t y l e ( 2 9 , 21 , s ) ;

293 o u t t e x t x y ( 16 + 35 \* ( 6 + i ) , 13 + 35 \* ( 6 + j ) , s ) ;

294 o u t t e x t x y ( 16 + 35 \* ( 6 + j ) , 13 + 35 \* ( 6 + i ) , s ) ;

295 }

296

1. TCHAR s [ 5 0 ] ;
2. c l o c k \_ t end ;
3. c l o c k \_ t s t a r t , f i n i s h ;
4. i n t d u r a t i o n ;

301 f o r ( i n t i = 1 ; i <= 9 ; i ++)

302 f o r ( i n t j = 1 ; j <= 9 ; j ++) {

1. \_ s t p r i n t f ( s , \_T ( *”%d”* ) , s u d o k u \_ t a b l e [ i − 1 ] [ j − 1 ] ) ;

*/ /* 高版本 *VC* 推荐使用 *\_ s t p r i n t f \_ s* 函数

1. s e t t e x t s t y l e ( 2 9 , 21 , s ) ;

## 华 中 科 技 大 学 课 程 实 验 报 告

1. i f ( u s e r s \_ s u d o k u [ i − 1 ] [ j − 1 ] == 1 )

307 o u t t e x t x y ( 16 + 35 \* j , 13 + 35 \* i , o ) ;

308 e l s e {

309 o u t t e x t x y ( 16 + 35 \* j , 13 + 35 \* i , s ) ;

310 }

311

312 }

313 f o r ( i n t i = 1 ; i <= 9 ; i ++)

314 f o r ( i n t j = 1 ; j <= 9 ; j ++) { 315

1. \_ s t p r i n t f ( s , \_T ( *”%d”* ) , s u d o k u \_ t a b l e [ i + 8 ] [ j − 1 ] ) ;

*/ /* 高版本 *VC* 推荐使用 *\_ s t p r i n t f \_ s* 函数

1. s e t t e x t s t y l e ( 2 9 , 21 , s ) ; 318

319 i f ( u s e r s \_ s u d o k u [ i + 8 ] [ j − 1 ] == 1 )

320 o u t t e x t x y ( 16 + 35 \* ( 6 + j ) , 13 + 35 \* ( 6 + i ) , o

) ;

321 e l s e {

322 o u t t e x t x y ( 16 + 35 \* ( 6 + j ) , 13 + 35 \* ( 6 + i ) , s

) ;

323 }

324 }

325 wchar\_t s t r 1 [ ] = L*”◻◻SUDOKU”* ;

326 f i l l r e c t a n g l e ( 7 1 0 , 15 , 900 , 150 ) ;

327 s e t t e x t s t y l e ( 1 8 , 16 , s ) ;

328 o u t t e x t x y ( 7 1 2 , 32 , \_T ( *”◻*双数独游戏*”* ) ) ;

329 i n t x , y , l , r , a = 0 , b = 0 ;

1. ExMessage m, n ; */ /* 定义消息变量
2. s e t t e x t c o l o r (BLACK) ;
3. s t a r t = c l o c k ( ) ;
4. end = c l o c k ( ) ;
5. \_ s t p r i n t f ( s , \_T ( *”%d”* ) , 0 ) ; */ /* 高版本 *VC* 推荐使用

*\_ s t p r i n t f \_ s* 函数

1. s e t t e x t s t y l e ( 2 9 , 21 , s ) ;

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 336 | o u t t e x t x y ( 8 3 2 , 110 , s ) ; | | | | | | | |
| 337 | wchar\_t s t r 2 [ ] = L*”◻◻*错 误次数 *: ”* ; | | | | | | | |
| 338 | s e t t e x t s t y l e ( 1 5 , 12 , s ) ; | | | | | | | |
| 339 | o u t t e x t x y ( 7 1 2 , | | 250 , | | \_T ( *”◻*游戏说明： *”* ) ) ; | | | |
| 340 | o u t t e x t x y ( 6 1 0 , | | 275 , | | \_T ( *”◻*请点击对应空格准备填入数字； *”* ) ) ; | | | |
| 341 | o u t t e x t x y ( 6 1 0 , | | 325 , | | \_T ( *”◻*数字请通过数字键盘输入； *”* ) ) ; | | | |
| 342 | o u t t e x t x y ( 6 1 0 , | | 350 , | | \_T ( *”◻*输入错误时显示*W*（*WRONG*） ； *”* ) ) ; | | | |
| 343 | o u t t e x t x y ( 6 1 0 , | | 375 , | | \_T ( *”◻*按*◻ESC◻*键退出； *”* ) ) ; | | | |
| 344 | o u t t e x t x y ( 6 1 0 , | | 400 , | | \_T ( *”◻*请关注自己的错误次数， *”* ) ) ; | | | |
| 345 | o u t t e x t x y ( 6 1 0 , | | 425 , | | \_T ( *”◻*超过*15* 次会自动退出。*”* ) ) ; | | | |
| 346 | o u t t e x t x y ( 6 1 0 , | | 450 , | | \_T ( *”◻*祝你游戏愉 快！ ！ ！ *”* ) ) ; | | | |
| 347 | s e t t e x t s t y l e ( 2 9 , 21 , s ) ; | | | | |  | | |
| 348 | o = \_T ( *’W’* ) ; | | | | |  | | |
| 349 | s e t t e x t c o l o r (RED) ; | | | | |  | | |
| 350 | i n t wrong\_times = 0 , f l a g = 0 ; | | | | |  | | |
| 351 | w h i l e ( t r u e ) | | | | |  | | |
| 352 | { | | | | |  | | |
| 353 | */ /* 获取一条鼠标或按键消息 | | | | |  | | |
| 354 | m = g e t m e s s a g e (EX\_MOUSE | | | | | | EX\_KEY) ; | | |
| 355 | s w i t c h (m. message ) | | | | |  | | |
| 356 | { | | | | |  | | |
| 357 | c a s e WM\_LBUTTONDOWN: | | | | |  | | |
| 358 | a = m. x , b = m. y ; | | | | |  | | |
| 359 | i f | ( a | > | 45 | && a < 360 | && b > 45 | && b < 360 ) | { |
| 360 |  | x | = | 0 , | y = 0 ; |  |  |  |
| 361 |  | l | = | 0 , | r = 0 ; |  |  |  |

362 w h i l e ( x < a ) {

363 l ++;

364 x = 10 + 35 \* l ;

365 }

366 w h i l e ( y < b ) {

367 r ++;

368 y = 10 + 35 \* r ;

369 }

370 i f ( u s e r s \_ s u d o k u [ r − 2 ] [ l − 2 ] ! = 1 )

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 371 |  | | b r e a k ; | | | |  | | |
| 372 | i f | | (m. c t r l ) | | | |
| 373 |  | | */ /* 画一个大方块 | | | |
| 374 |  | | r e c t a n g l e (m. x − 5 , m. y | | | | − 5 , m. x + 5 , m. y | + | 5 ) |
|  |  | | ; | | | |  |  |  |
| 375 |  | e l s e | | | | | |  |  |
| 376 |  | */ /* 画一个小方块 | | | | | |  |  |
| 377 |  | r e c t a n g l e (m. x − 2 , m. y − 2 , m. x + 2 , m. y | | | | | | + | 2 ) |
|  |  | ; | | | | | |  |  |
| 378 | } |  | | | | | |  |  |
| 379 | i f | ( a > 255 && a < 360 && b > 255 && b < 360 ) { | | | | | |  |  |
| 380 |  | b r e a k ; | | | | | |  |  |
| 381 | } |  | | | | | |  |  |
| 382 | i f | ( a > 255 && a < 570 && b > 255 && b < 570 ) { | | | | | |  |  |
| 383 | x | | = | 0 , y | = | 0 ; | | | |
| 384 | l | | = | 6 , r | = | 6 ; | | | |

385 w h i l e ( x < a ) {

386 l ++;

387 x = 10 + 35 \* l ;

388 }

389 w h i l e ( y < b ) {

390 r ++;

391 y = 10 + 35 \* r ;

392 }

393 i f ( u s e r s \_ s u d o k u [ r − 2 − 6 + 9 ] [ l − 2 − 6 ] ! = 1 )

1. b r e a k ;
2. i f (m. c t r l )
3. */ /* 画一个大方块
4. r e c t a n g l e (m. x − 5 , m. y − 5 , m. x + 5 , m. y + 5 )

;

e l s e

}

*/ /* 画一个小方块

r e c t a n g l e (m. x − 2 , m. y − 2 , m. x + 2 , m. y + 2 )

;

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |
| --- | --- | --- |
| 402 | b r e a k ; |  |
| 403 |  |
| 404 | c a s e WM\_KEYDOWN: |
| 405 | i f (m. vkcode == VK\_ESCAPE) | { |
| 406 | f l a g = 1 ; |  |
| 407 | b r e a k ; |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 408 | } |  | | | | | | | | | |
| 409 |  |
| 410 | i f | ( a | > | 45 | && a < 360 && b | > | 45 && | b | < | 360 ) | { |
| 411 |  | x | = | 0 , | y = 0 ; |  |  |  |  |  |  |
| 412 |  | l | = | 0 , | r = 0 ; |  |  |  |  |  |  |

413 w h i l e ( x < a ) {

414 l ++;

415 x = 10 + 35 \* l ;

416 }

417 w h i l e ( y < b ) {

418 r ++;

419 y = 10 + 35 \* r ;

420 }

421 c = s u d o k u \_ t a b l e [ r − 2 ] [ l − 2 ] + 4 8 ;

422 v = \_ g e t c h ( ) ;

423 i f ( v == c ) {

424 o u t t e x t x y ( 16 + 35 \* ( l − 1 ) , 13 + 35 \* ( r − 1 ) , c ) ;

425 }

426 e l s e {

427 o u t t e x t x y ( 16 + 35 \* ( l − 1 ) , 13 + 35 \* ( r − 1 ) , o ) ;

428 wrong\_times ++;

429 \_ s t p r i n t f ( s , \_T ( *”%d”* ) , wrong\_times ) ;

*/ /* 高版本 *VC* 推荐使用 *\_ s t p r i n t f \_ s*

函数

430 o u t t e x t x y ( 8 3 2 , 110 , s ) ;

431 }

432 }

## 华 中 科 技 大 学 课 程 实 验 报 告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 433 | i f | ( a > 255 && | a | < 360 && | b > | 255 && b | < 360 ) { |
| 434 |  | b r e a k ; |  |  |  |  |  |
| 435 | } |  |  |  |  |  |  |
| 436 | i f | ( a > 255 && | a | < 570 && | b > | 255 && b | < 570 ) { |
| 437 |  | x = 0 , y = | 0 ; |  |  |  |  |
| 438 |  | l = 6 , r = | 6 ; |  |  |  |  |
| 439 |  | w h i l e ( x < | a ) | { |  |  |  |
| 440 |  | l ++; |  |  |  |  |  |

441 x = 10 + 35 \* l ;

442 }

443 w h i l e ( y < b ) {

444 r ++;

445 y = 10 + 35 \* r ;

446 }

447 c = s u d o k u \_ t a b l e [ r − 2 − 6 + 9 ] [ l − 2 − 6 ] + 4 8 ;

448 v = \_ g e t c h ( ) ;

449 i f ( v == c ) {

450 o u t t e x t x y ( 16 + 35 \* ( l − 1 ) , 13 + 35 \* ( r − 1 ) , c ) ;

451 }

452 e l s e {

453 o u t t e x t x y ( 16 + 35 \* ( l − 1 ) , 13 + 35 \* ( r − 1 ) , o ) ;

454 wrong\_times ++;

455 \_ s t p r i n t f ( s , \_T ( *”%d”* ) , wrong\_times ) ;

*/ /* 高版本 *VC* 推荐使用 *\_ s t p r i n t f \_ s*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | 函数 |
| 456 |  | | | o u t t e x t x y ( 8 3 2 , 110 , s ) ; |
| 457 | } | | |  |
| 458 | } | | |  |
| 459 | b r e a k ; | | |  |
| 460 |  | } |  | |
| 461 |  | i f | ( f l a g ) b r e a k ; | |
| 462 |  | i f | ( wrong\_times > 15 ) b r e a k ; | |
| 463 | } |  |  | |

## 华 中 科 技 大 学 课 程 实 验 报 告

464

\_ g e t c h ( ) ;

c l o s e g r a p h ( ) ;

*/ /* 按任意键继续

*/ /* 关闭绘图窗口

}

465

466

文件名：main.cpp 功能：主函数

1

# i n c l u d e *” d e f i n e . h”*

*/\** 定义有效的全局变量 *\*/*

*/ / char f i l e n a m e [ 1 0 0 ] ; / /* 定义文件名

*/ / Conjunctive\_Normal\_Form\_List c n f ; / /* 定义 *c n f* e x t e r n Argue Value\* V a l u e L i s t ; */ /* 定义变元真值表i n t op ;

e x t e r n i n t s u d o k u \_ t a b l e [ 1 8 ] [ 9 ] ; */ /* 定义全局变量 *i n t* 类型二维数组存储

双数独终盘

e x t e r n i n t u s e r s \_ s u d o k u [ 1 8 ] [ 9 ] ; */ /* 存储输出的含空格数独问题格局

e x t e r n i n t s h u f f l e \_ v a l u e [ 1 6 4 ] ; */ /* 存储被打乱的变元

u s i n g namespace s t d ;

2

3

4

5

6

7

8

9

10

11

12

13

|  |  |  |
| --- | --- | --- |
| 14 | i n t | main ( ) { |
| 15 |  |  |
| 16 |  | w h i l e ( t r u e ) { |
| 17 |  | system ( *” c l s ”* ) ; |
| 18 |  | p r i n t f ( *”* 可选功能： *\ n”* ) ; |
| 19 |  | p r i n t f ( *” 1 . c n f* 文件求解 *\ n”* ) ; |
| 20 |  | p r i n t f ( *” 2 .SUDOKU*游戏 *\ n”* ) ; |
| 21 |  | p r i n t f ( *” 0 .* 退出 *\ n \ n”* ) ; |
| 22 |  | p r i n t f ( *”* 请选择你想要实现的功能： *”* ) ; |
| 23 |  | s c a n f ( *”%d”* , &op ) ; |
| 24 |  | i f ( op == 0 ) |
| 25 |  | b r e a k ; |
| 26 |  | s o l v e ( op ) ; |
| 27 |  | g e t c h a r ( ) ; g e t c h a r ( ) ; |
| 28 |  | } |

## 华 中 科 技 大 学 课 程 实 验 报 告

29

r e t u r n 0 ;

}

30

31