

数独游戏基本策略求解器 - 需求分析文档

1. 引言

1.1 目的

本文档旨在定义数独游戏基本策略求解器的功能需求和非功能需求，明确系统边界和开发目标。

1.2 范围

本系统将实现一个能够应用两种基本策略（Last Remaining Cell和Possible Number）自动解决数独问题的程序，不包括图形用户界面和高级解题策略。

1.3 定义和缩写

- **Last Remaining Cell**: 唯一候选数法，当某单元格在行、列或宫中只有一个可能数字时填入该数字
- **Possible Number**: 显性唯一候选数法，当某数字在某行、列或宫中只有一个可能位置时填入该数字
- 数独板: 9x9的网格，表示数独游戏的当前状态

2. 总体描述

2.1 产品前景

本产品是一个数独解题辅助工具，旨在帮助用户理解基本解题策略，验证解题思路，并作为更复杂数独求解器的基础。

2.2 用户特征

- 初级数独爱好者：学习基本解题策略
- 教育工作者：用于教学演示
- 开发者：作为更复杂求解器的基础组件

2.3 假设与依赖

- 假设输入的数独问题有解且可通过两种基本策略解决
- 依赖标准输入输出环境

3. 详细需求

3.1 功能需求

3.1.1 输入功能

- **FR1.1**：系统应能接受9x9数独板的输入
 - 输入格式：文本文件或命令行输入，空单元格用0或点号表示
 - 示例格式：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 0 | 0 | 7 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 1 | 9 | 5 | 0 | 0 | 0 |
| 0 | 9 | 8 | 0 | 0 | 0 | 0 | 6 | 0 |
| 8 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 3 |
| 4 | 0 | 0 | 8 | 0 | 3 | 0 | 0 | 1 |
| 7 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 6 |
| 0 | 6 | 0 | 0 | 0 | 0 | 2 | 8 | 0 |
| 0 | 0 | 0 | 4 | 1 | 9 | 0 | 0 | 5 |
| 0 | 0 | 0 | 0 | 8 | 0 | 0 | 7 | 9 |

- **FR1.2:** 系统应验证输入的数独板是否符合规则
 - 检查行、列、宫中是否有重复数字
 - 检查是否为9x9网格
 - 检查是否包含非法字符

3.1.2 解题功能

- **FR2.1:** 系统应实现Last Remaining Cell策略
 - 对于每个空单元格，检查其所在行、列和宫已存在的数字
 - 如果只有一个可能的数字，则填入该数字
- **FR2.2:** 系统应实现Possible Number策略
 - 对于每个数字1-9，检查每行/列/宫
 - 如果该数字在某行/列/宫中只有一个可能位置，则填入该数字
- **FR2.3:** 系统应能交替应用两种策略直到无法继续或问题解决
 - 记录应用策略的顺序和次数
 - 当两种策略都无法继续应用时终止

3.1.3 输出功能

- **FR3.1:** 系统应能显示解题后的数独板
 - 格式清晰的9x9网格
 - 区分原始数字和新填入的数字
- **FR3.2:** 系统应能提供解题过程摘要
 - 应用的策略类型和次数
 - 填入的数字总数
 - 解题是否完成

3.2 非功能需求

3.2.1 性能需求

- **NR1.1:** 对于简单数独（仅需基本策略即可解决），解题时间应小于1秒
- **NR1.2:** 系统应能处理至少1000次策略应用循环

3.2.2 可靠性需求

- **NR2.1:** 系统不应改变原始数独板中已填数字
- **NR2.2:** 当输入无解时，系统应明确提示而非无限循环

3.2.3 可用性需求

- **NR3.1:** 命令行界面应提供清晰的用法说明
- **NR3.2:** 错误信息应明确指示问题所在

3.2.4 可维护性需求

- **NR4.1:** 代码应有适当注释，特别是策略实现部分
- **NR4.2:** 系统应模块化设计，便于添加新策略

3.3 约束

- **C1**: 初始版本仅实现两种基本策略，不处理需要猜测的高级数独
- **C2**: 使用Python或Java等高级语言实现
- **C3**: 不依赖外部数独解题库

4. 用例模型

4.1 主要用例：解决数独

参与者：用户

前置条件：用户提供有效的数独问题

主成功场景：

1. 用户启动程序并输入数独问题
2. 系统验证输入有效性
3. 系统应用Last Remaining Cell策略
4. 系统应用Possible Number策略
5. 重复3-4步直到无法继续或问题解决
6. 系统显示解题结果和过程摘要

扩展：

- 2a. 输入无效：
 1. 系统显示错误信息
 2. 返回步骤1
- 5a. 无法完全解题：

1. 系统显示部分解题结果
2. 注明哪些单元格未解决

5. 未来可能的扩展

1. 实现更高级的解题策略
2. 添加图形用户界面
3. 支持不同尺寸的数独变体
4. 添加解题步骤可视化
5. 生成可解的数独题目

6. 验收标准

1. 能正确解决仅需两种基本策略的数独问题
2. 对于无法完全解决的问题，能正确识别并报告
3. 处理标准输入格式并产生标准输出
4. 代码结构清晰，有适当文档和注释
5. 包含测试用例验证所有主要功能

附录A：示例输入输出

示例输入：

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 0 | 0 | 7 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 1 | 9 | 5 | 0 | 0 | 0 |
| 0 | 9 | 8 | 0 | 0 | 0 | 0 | 6 | 0 |
| 8 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 3 |
| 4 | 0 | 0 | 8 | 0 | 3 | 0 | 0 | 1 |
| 7 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 6 |
| 0 | 6 | 0 | 0 | 0 | 0 | 2 | 8 | 0 |
| 0 | 0 | 0 | 4 | 1 | 9 | 0 | 0 | 5 |
| 0 | 0 | 0 | 0 | 8 | 0 | 0 | 7 | 9 |

预期输出：

解题结果：

| | | | | | | | | | | |
|-------------------|---|---|--|---|---|---|--|---|---|---|
| 5 | 3 | 4 | | 6 | 7 | 8 | | 9 | 1 | 2 |
| 6 | 7 | 2 | | 1 | 9 | 5 | | 3 | 4 | 8 |
| 1 | 9 | 8 | | 3 | 4 | 2 | | 5 | 6 | 7 |
| -----+-----+----- | | | | | | | | | | |
| 8 | 5 | 9 | | 7 | 6 | 1 | | 4 | 2 | 3 |
| 4 | 2 | 6 | | 8 | 5 | 3 | | 7 | 9 | 1 |
| 7 | 1 | 3 | | 9 | 2 | 4 | | 8 | 5 | 6 |
| -----+-----+----- | | | | | | | | | | |
| 9 | 6 | 1 | | 5 | 3 | 7 | | 2 | 8 | 4 |
| 2 | 8 | 7 | | 4 | 1 | 9 | | 6 | 3 | 5 |
| 3 | 4 | 5 | | 2 | 8 | 6 | | 1 | 7 | 9 |

解题摘要：

应用Last Remaining Cell策略：15次

应用Possible Number策略：12次

填入数字总数：27

解题状态：完全解决

附录B：术语表

- 单元格：数独板中的单个格子，可填入数字1-9
- 行：数独板中的水平9个单元格
- 列：数独板中的垂直9个单元格
- 宫：数独板中的3x3粗线框内的9个单元格