

理论任务选题列表

• 【已选】正则表达式化简的算法验证

- 输入的正则表达式中可能包含的语法结构有：空集、字符的集合（如 `[a-z0-9]`）、可选的情形（即 $?r$ ）、多种循环（即 r^* 与 r^+ ）、字符串（如 `"ab\n"`）、单字符、并集与连接；
- 输出的正则表达式中可能包含的语法结构有字符的集合、空字符串、星号表示的循环（即 r^* ）、并集与连接；
- 本选题将提供输出的正则表达式的语法与语义；
- 要求 2 与要求 3 中，输出的正则表达式不应包含空字符串与其他正则表达式的连接；

要求 1：定义输入正则表达式的语法和语义，定义正则表达式的转换算法，并证明转换前后的正则表达式的指称语义相同；要求 2：重新定义正则表达式的转换算法（可以分阶段），保证输出的正则表达式不包含空字符串与其他正则表达式的连接，证明转换前后的正则表达式语义相同；要求 3：证明输出的正则表达式不包含空字符串与其他正则表达式的连接。相关材料详见 `Reg2Reg.v`。

• 【已选】基于正则表达式生成 NFA 的算法验证

- 输入的正则表达式中可能包含的语法结构有字符的集合、空字符串、星号表示的循环（即 r^* ）、并集与连接；
- 输出是一个 NFA 图与起始节点编号、终止节点编号；
- 本选题将提供正则表达式的语义与 NFA 的语义；
- 本选题将提供一个在 Coq 中定义的生成 NFA 的算法，该算法将使用两个全局变量记录当前已经使用的最大节点编号和最大边编号。

要求 1：证明空字符串的情况、字符串集合的情况、并集与连接有关的归纳步骤；要求 23：证明星号有关的归纳步骤，并完成完整的归纳法证明。相关材料详见 `RegExpNFA.v`。

• 【已选】二叉堆操作的算法验证

- 本选题将提供二叉堆的 Coq 定义；
- 本选题将提供二叉堆（小根堆）的合法性 Coq 定义；
- 本选题将提供二叉堆与其所表示集合之间关系的定义；
- 本选题中需要学生定义局部破坏的二叉堆合法性性质：某节点需要向下移时的全树满足的性质、某节点需要向上移时的全树满足的性质。

要求 1：定义局部破坏的二叉堆合法性性质，定义二叉树中向上移动节点与向下移动节点的动作，并证明其正确性，正确性应包含集合不变性与合法性；要求 2：定义二叉堆（小根堆）中的插入操作与删除最小元操作，并证明其正确性，除了需要证明的性质之外，这些操作应当能保持“满二叉树”性质；要求 3：定义“满二叉树”性质，并证明前述操作能保持该性质。相关材料详见 `Heap.v`。

- 【已选】快速排序算法的算法验证

要求 1: 定义快速排序中的分区 (partition) 操作, 基于此定义广义的快速排序算法, 并证明这两者的正确性; 要求 2: 定义 Mediam of Mediams 算法, 证明该算法选出的一定是原序列中的元素, 即可以用作快速排序中的选 pivot 节点的方法; 要求 3: 证明上述 Mediam of Mediams 算法选出的元素在大致 $3/10$ - $7/10$ 的范围内。本选题无附件材料。

- 【已选】PRIM 图论算法的算法验证

要求 1: 定义 PRIM 算法最小生成树算法, 定义何种无向图是树, 基于此证明 PRIM 算法每一步生成的子图都是树; 要求 2: 证明如果原图连通, 那么 PRIM 算法最终找到的是树; 要求 3: 证明如果原图连通, 那么 PRIM 算法找到的一定是最小生成树。相关材料详见Prim.v。

- 【已选】string 库的程序验证

定义字符串的表示谓词, 实现并证明下面 C 函数的正确性。本选题无附件材料。

- strcat (要求 1);
- strncat (要求 1);
- strcmp (要求 2);
- strncmp (要求 2);
- strcpy (要求 1);
- strncpy (要求 1);
- strlen (要求 1);
- strstr (要求 3)。

- PRIM 图论算法实现的程序验证

暂无详细信息。

- 基于单子理论定义概率程序并证明基本性质

- 本选题将提供概率分布的 Coq 定义;
- 本选题将提供概率单子的 Coq 定义框架;

要求 1: 证明概率分布的基本性质; 要求 2: 证明概率单子的整体基本性质; 要求 3: 证明概率单子算子的基本性质。详见ProbMonad.v, 具体内容以此附件为准。

- 带 IO 程序的指称语义与基本性质

要求 1: 定义新语言的语法要包含 IO 指令与非确定性的程序指令, 定义考虑该语言有穷行为的指称语义, 证明每个语义算子能保持语义等价, 并证明 seq-assoc、seq-skip、skip-seq、if-seq 与 while-unroll1 性质; 要求 2: 在考虑程序运行不终止的情况下重新定义该语言的指称语义, 证明每个语义算子能保持语义等价; 要求 3: 在考虑不终止行为的前提下, 证明 seq-assoc、seq-skip、skip-seq、if-seq 与 while-unroll1。本选题无附件材料。

- 简单程序内联的正确性证明

要求 1: 在 while 语言的表达式语法中添加函数调用, 定义其指称语义, 证明所有语法算子能保持语义等价。要求 23: 假设某函数调用仅仅通过一条表达式完成计算就返回, 请定义该函数的内联, 并证明内联前后表达式语义等价。本选题无附件材料。