**用途与运行环境：**

本程序平台采用C#实现我们所提出的大部分基于Petri网的类A\*搜索算法，详细算法描述请参见我的英文专著及期刊论文。

本平台基于Visual Studio 2008开发，安装VS2008后直接双击后缀名为.sln或.csproj文件即可打开运行。如果采用的不同版本的VS则可能无法直接打开，此时需要新建空白C# Console工程，将Main.cs, AStar.cs和Heap.cs三个文件，以及bin文件夹和帮助文档拷贝至新建工程的所在目录，然后在工程中Project->add existing item将三个cs文件加入即可。（将老的程序备份，并创建新的程序也是此流程）

运行：打开main.cs，点击Debug->Start Debugging(F5)

**平台组成：**

Main.cs: 主程序，输入一些用户参数

AStar.cs: 搜索程序，定义了两个类AStarNode和AStar。

Heap.cs: 堆栈操作函数，用于处理OPEN, CLOSED之类的堆栈操作，通常不用改动，直接调用即可

Bin/Debug文件夹：存放输入和输出文件。输入文件，对于一个Petri网，需要给出两个输入文件，其中xxx\_matrix.txt放置Petri网的关联矩阵的转置矩阵（即行为变迁，列为托肯），xxx\_init.txt放置起始标识，库所操作时间信息，和目的标识。输出文件为0result.txt。

0result.txt为输出文件。

**注意：**

* 如果FindPath\_AStar()为A\*搜索程序及其直接变种，如果实现差异较大的搜索程序，例如Anytime A\*或基于强化学习或蒙特卡洛树搜索放的A\*时，建议另外定义此函数，并命名为FindPath\_anytimeAstar(), FindPath\_RL (), FindPath\_MCTS ();
* 如果采用非A\*搜索且需要扩展节点的内容，可以继承AStarNode类
* 命名要规范，名字要有意义，且变量名尽量首字母小写，函数名首字母大写且为动词短语，常量首字母大写等，命名规则具体见https://static.app.yinxiang.com/embedded-web/profile/#/join?guid=c092ee66-b605-493e-baf8-6b018bc2bc0a&sharedNotebookGuid=&channel=copylink&shardId=s34&ownerId=25641086；
* 多写注释，方便自己和他人以后理解；新增句子和函数要注释姓名、日期和作用，如//ZS20221023: xxxxxxx；
* 程序出现bug时，采用单步执行查看变量、设置断点、设置输出变量、打印中间结果等方法进行Debug；

**版本更新说明：**

<https://static.app.yinxiang.com/embedded-web/profile/#/join?guid=0f885707-2f1a-4ea6-9fab-f2479b6dd6b5&sharedNotebookGuid=&channel=copylink&shardId=s34&ownerId=25641086>