# 人工智能-第五次课程作业报告

授课教师：张宇 作者：蒋雨初-58121102

## 1 问题描述

### 1.1题目介绍

结合11.3偏序规划中的穿鞋穿袜问题，基于状态空间搜索得到规划。

目标状态：

)

动作列表：

算法1：前向搜索：可基于广度优先、深度有限、或启发式搜索。得到有4个动作的一个规划。

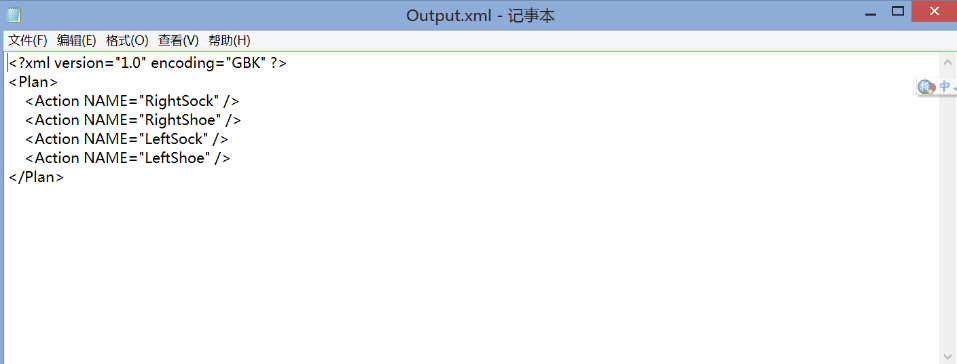
算法2：反向搜索：可基于广度优先、深度有限、或启发式搜索。得到有4个动作的一个规划。

### 1.2任务说明

完成Assignment5里的2个搜索函数：（1）前向搜索的CSearchPlan::RunForward()函数、以及它调用的CSearchPlan::Expand()函数,(2)后向搜索的CSearchPlan:: RunBackward ()函数、以及它调用的CSearchPlan::Expand\_Reverse()函数。

完成前向搜索/逆向搜索后，会根据操作系统的设置自动打开查询结果，可能是浏览器、或记事本等。

规划结果：\Data\Output.xml



### 1.3实验环境

### 设备规格

设备名称 LAPTOP-TFMBQKQ8

处理器 AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz

机带 RAM 16.0 GB (15.4 GB 可用)

设备 ID 6CC35513-8821-49C9-A60B-C44F31F302A7

产品 ID 00342-35891-56086-AAOEM

系统类型 64 位操作系统, 基于 x64 的处理器

笔和触控 没有可用于此显示器的笔或触控输入

### 系统规格

版本 Windows 11 家庭中文版

版本 22H2

安装日期 ‎2022/‎9/‎28

操作系统版本 22621.521

体验 Windows Feature Experience Pack 1000.22634.1000.0

### 开发环境

Microsoft Visual Studio Professional 2022

Version 17.2.4

VisualStudio.17.Release/17.2.4+32602.215

Microsoft .NET Framework

Version 4.8.09032

Installed Version: Professional

Visual C++ 2022 00483-00000-00004-AA929

Microsoft Visual C++ 2022

### 语言标准

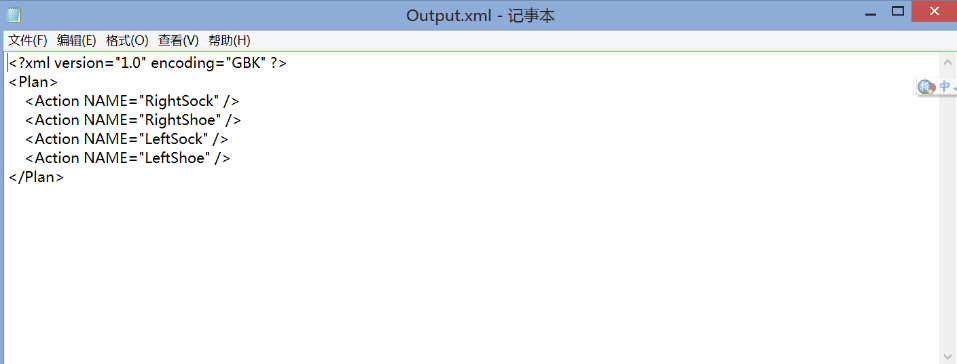
C++20 or latest.

### 1.4评价标准

**文档部分（2分）**

* 针对本题目撰写文档，保证文档结构完整性，文档可包括 题目、问题描述、实验方案、实验结果、分析、结论，也可参考其他标准文档或者学术论文的格式。文档内容及结构均纳入评分范围。
* **文档：1分。源代码风格：1分。**

**源代码部分（8分）**

* 2个搜索函数：（1）前向搜索的CSearchPlan::RunForward()函数、以及它调用的CSearchPlan::Expand()函数,(2)后向搜索的CSearchPlan:: RunBackward ()函数、以及它调用的CSearchPlan::Expand\_Reverse()函数。共有4个函数需要完成，各2分。
* 输出结果应为:
* 

## 2 实验方案

## 2.1 PDDL

规划问题定义：Plainning domain definition language，简称**PDDL**。

第7章的混合命题逻辑Agent没有领域相关启发知识就能找到规划，因为其使用了**基于问题**的逻辑结构的领域无关启发知识。

但是它依赖于无变量命题推理，这意味着有许多动作和状态时会忙的一笔。

因此，规划研究者选择了**要素化**（factored representation）——用一组变量表示世界的一个状态的表示方法。可使用PDDL语言。

每个状态称为**流（**fluent）（状态变量的同义词）的合取,这些流是**基元**的（无变量），无函数的原子。

一组基元（无变量）的动作能够表示为单个**动作模式**（action schema）。这个动作模式是一种提升表示。

现在，我们将规划定义成了一个**搜索问题**：有一个初始状态，一个action函数，一个result函数以及一个目标测试。

## 2.2 例子：航空货物运输

涉及装载、卸载货物，以及从一个地方飞到另一地。

这个问题用：**Load，Unload，Fly**三个动作定义。 对应有两个谓词：In（c，p）表示货物在机，At（x，a）表示x（机或物）在机场a。

基本的PDDL是没有全称量词的。

## 2.3 经典规划的复杂性

本节考虑规划的理论上的**复杂性**，并区分两个决策问题。

**PlanSAT**：询问是否存在解决一个规划问题的某个规划

**限界**（bounded）PlanSAT：询问是否存在用于找到最优规划的、长度小于k的一个解。

后者是NP-完全的，而PlanSAT属于P。换言之，最优规划通常困难，但次优规划有时是容易的。

## 2.4 前向和后向搜索算法。

**2.4.1前向状态空间搜索**

前向搜索容易探索到无关动作。考虑从在线书店购买《AI：A Modern Approach》这本书。有一个动作模式Buy（isbn），效果是Own（isbn）。ISBN是10位数，因此这个动作模式表示100亿个基元动作。一个无启发的前向搜索将需要枚举这100亿个动作，以找到通向目标的动作。

其次，规划问题经常有大的状态空间。考虑有10个机场的航空货物问题，每个机场有5架飞机和20件货物。目标是将机场A的所有货物运送到机场B。这个问题有一个简单的解：将20件货物装载到机场A的一架飞机上，飞机飞到机场B，然后卸载货物。但是找到解可能是困难的，因为平均分支因子是巨大的：50架飞机中的任何一架可以飞到9个其他的机场，200件包裹中的每一件也能一样被卸载（如果已经装载了）或者装载到机场的任何一架飞机上（如果还没装载）。因此在任何状态，至少有450个动作（当所有包裹在一个没有飞机的机场），至多有10450个（当所有包裹和飞机在同一个机场）。平均而言，我们说每个状态存在大约2000个可能的动作，所以达到明显解的深度的搜索树大约有2000＃个结点。

显然，没有精确的启发式，即使这个相对小的问题实例也是无望的。尽管规划的许多现实世界应用依赖于特定领域的启发知识，还是能自动导出非常强的独立于领域的启发知识；这使得前向搜索具有可行性。

**2.4.2后向相关状态搜索**

在后向搜索中，我们从目标开始，向后应用动作，直到找到达到初始状态的步骤序列。它被称为相关状态搜索，因为我们只考虑与目标（或当前状态）相关的动作。就像在信念-状态搜索中（4.4节），在每一步考虑一组相关状态，而不是单个状态。

我们从目标开始，它是对一组状态进行描述的文字的合取-例如，目标-PoorAFamous 描述了那些Poor为假、Famous 为真、其他流（fluent）可有任何值的状态。如果

一个问题域中有n个基元流，那么就有2”个基元状态（每个流可以为真或假），但目标状

态集合有3＂种描述（每个流可以为正、负、或者并不提及）。

通常只有当我们知道如何从一个状态描述后退到前驱状态描述时，后向搜索才能工作。例如，后向搜索出n皇后问题的解是很难得，因为没有容易的方法可以描述哪些状态离目标只有一步之遥。可喜的是，PDDL 表示的设计使得后退动作很容易-如果问题域可以用PDDL表达，那么我们就可以对其进行后退搜索。给定一个基元目标描述g和一个基元动作a，从g经a回退得到状态描述g＇，定义为

g'=(g-ADD(a)) U Precond(a)

也就是说，该动作增加的效果不必在之前为真，而前提在之前必须成立，否则该动作不可能被执行。注意，公式中没有出现DEL（a）；这是因为，尽管我们知道DEL（a）中的流在动作后不再为真，但我们并不知道它们在动作之前是否真，因此不需要针对它们说什么。

**2.4.3规划的启发式**

没有好的启发式函数，无论前向后向都不高效。

可采纳的启发式。可以通过更容易求解的松弛问题导出。

将搜索问题想成一个图，节点为状态，边为动作。问题是要找一条连接初始状态到目标状态的路径。

有2个方法来松弛这个问题：

1）加入更多的边，使得路径更容易被找到

2）多个节点组合到一起，将状态空间抽象为具有更少状态的形式

忽略前提启发式

忽略删除列表启发式

定义启发式的关键思想是分解（decomposition）

## 3实验结果

## 3.1 前向状态空间搜索结果

## 3.2 后向状态空间搜索结果



## 4实验分析

## 4.1 前向空间搜索

太低效，不实用。

前向搜索：从初始状态出发，使用问题的动作，向前搜索目标状态

后向搜索：从目标的状态集出发，使用动作的逆，向后搜索初始状态。

1）前向搜索容易探索到无关动作

2）规划问题常有大的状态空间

对前向搜索来说，显然，没有精确的启发式，即使相对小的问题实例也是无望的。但是规划的许多现实应用还是能自动导出非常强的独立于领域的启发知识，这使得前向搜搜具有可行性。

## 4.2 后向空间搜索

从目标开始，向后应用动作，直到找到达到初始状态的步骤序列。在每一步考虑一组相关状态，而不是单个状态。

从目标开始，对一组状态进行描述的文字之合取。

可喜的是，PDDL表示的设计使得后退动作很容易。

最后一个问题是：决定哪些动作是后退的候选动作，在前向中我们选择适用的动作——在规划中可能是下一个步骤的那些动作

在后向中，我们需要相关的动作——导致当前目标状态的规划中可以作为最后一个步骤的那些动作。

一个动作要与一个目标相关，则必须明显对目标有贡献。

尽管后向搜索是分支因子低于前向，然而，后向使用状态集而不是单个状态的事实使得它更加难以想出好的启发知识，这就是当前主流偏向前向的主要原因。

## 5 结论

前向 (前进) 状态空间搜索：由于我们已经将规划问题转换为了一个搜索问题, 我们可以使用启发式搜索算法或其他搜索算法，从初始状态出发求解规划问题。在应用中, 我们通常结合实际人为地定义在任何一步上执行任何一种可行的新行为的成本(Cost), 结合启发式算法(如分支定界或算法)解决问题, 否则算法在每一步上都需要枚举所有可能的行为,导致搜索规模过于庞大。

后向 (后退) 状态空间搜索：在后向搜索中, 我们从目标出发, 从后往前地应用动作, 直到找到能够达到初始状态的步骤序列. 我们的搜索从目标: 对一组状态进行描述的文字的合取开始. 我们首先要将所有已知的动作“取逆”, 也就是将它们分别转换为后退动作. 由于我们此前在PDDL 中的设计, 我们只需要对每个已知动作构造一个对应的新的动作, 令这一动作增加的效果不必在前一时刻为真, 而原动作的前提在前一时刻必须成立即可。