

## Seaman.h.zhang

博客园 :: 首页 :: 新随笔 :: 联系 :: 订阅  :: 管理 34 Posts :: 0 Stories :: 2 Comments :: 0 Trackbacks

### 公告

昵称: seaman.kingfall

园龄: 4年3个月

粉丝: 4

关注: 1

+加关注

### 搜索

找找看

谷歌搜索

### 常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

### 我的标签

练习题(6)

合一(3)

递归(3)

中断(2)

类型变量(2)

数字(2)

列表(2)

Haskell(2)

recursive(2)

比较(2)

更多

### 随笔分类

Haskell(2)

Prolog(32)

### 随笔档案

2015年8月 (7)

2015年7月 (22)

2015年6月 (5)

### 最新评论

1. Re:Learn Prolog Now  
翻译 - 第一章 - 事实, 规则  
和查询 - 第一节, 一些简单  
的例子  
学习!

--深蓝医生

2. Re:Learn Prolog Now  
翻译 - 第一章 - 事实, 规则  
和查询 - 第一节, 一些简单  
的例子  
翻译了这么多了, 而且每天一  
篇, 不能望其项背啊。

## Learn Prolog Now 翻译 - 第十二章 - 文件相关操作 - 第二节, 文件的读写

# 从文件中读取内容

本节将学习如何从文件中读取内容。如果文件中的内容是以Prolog的语句形式存在的, 那么在Prolog中读取这样的文件内容是很容易的。比如文件houses.txt的内容如下:

```
gryffindor.  
hufflepuff.  
ravenclaw.  
slytherin.
```

下面是Prolog打开文件, 读取内容, 并且将内容显示在屏幕上的代码:

```
main :-  
    open('houses.txt', read, Str),  
    read(Str, House1),  
    read(Str, House2),  
    read(Str, House3),  
    read(Str, House4),  
    close(Str),  
    write([House1, House2, House3, House4]), nl.
```

上面代码将会以只读模式打开文件, 然后使用Prolog内置谓词read/2读取Prolog语句, 然后关闭流, 最后打印信息到屏幕上去。

这种方式是很直接和简单的。但是, read/2谓词需要谨慎地使用。首先, 它只能处理Prolog的语句(我们将会在后面讨论更多这方面的内容), 第二, 如果流没有任何内容了, 它可能会导致运行时错误。有没有一种更加优雅的方式可以克服第二个问题呢?

当然有的。内置谓词at\_end\_of\_stream/1能够检查stream是否已经到达了尾端, 而且是以一种安全的方式进行使用。对于一个流X, at\_end\_of\_stream(X)当流X已经到达了其尾端时为真(换种说法, 文件中所有的语句都已经被读取了)。

下面的代码是经过修改后的版本, 展示了如何使用at\_end\_of\_stream/1这个谓词:

```
main :-  
    open('houses.txt', read, Str),  
    read_houses(Str, Houses),  
    close(Str),  
    write(Houses), nl.  
  
read_houses(Stream, []) :-  
    at_end_of_stream(Stream).  
  
read_houses(Stream, [X|L]) :-
```

--Benjamin Yan

## 阅读排行榜

1. Learn Prolog Now 翻译 - 第三章 - 递归 - 第一节, 递归的定义(1168)
2. Learn Prolog Now 翻译 - 第一章 - 事实, 规则和查询 - 第一节, 一些简单的例子(1087)
3. Learn Prolog Now 翻译 - 第一章 - 事实, 规则和查询 - 第二节, Prolog语法介绍(781)
4. Haskell学习笔记二: 自定义类型(767)
5. Learn Prolog Now 翻译 - 第六章 - 列表补遗 - 第一节, 列表合并(753)

## 评论排行榜

1. Learn Prolog Now 翻译 - 第一章 - 事实, 规则和查询 - 第一节, 一些简单的例子(2)

## 推荐排行榜

1. Haskell学习笔记二: 自定义类型(1)
2. Learn Prolog Now 翻译 - 第三章 - 递归 - 第四节, 更多的实践和练习(1)

```
\+ at_end_of_stream(Stream),
read(Stream, X),
read_houses(Stream, L).
```

现在来解决更加麻烦的问题。上面提及read/2只能读取Prolog语句。如果你想要读取任意文件内容, 情况可能会变得比较复杂, 因为Prolog会迫使读入的内容以字符级别来进行, 内置的谓词get\_code/2从流中读取下一个存在的字符。字符在Prolog中是使用其整数数字来代替的。比如, get\_code/2将会在流中读取字符a, 然后返回结果是97。

通常, 我们不会关心这些整数, 而是关心字符本身——或者, 由这些字符组成的列表, 来表示的原子。我们如何处理这些字符呢? 一种方式是使用内置谓词atom\_codes/2, 可以将整数列表转换为对应的原子。我们将使用下一个例子介绍这种方式, 例子展示了如何从流中读取单词:

```
readWord(InStream, W) :-
    get_code(InStream, Char),
    checkCharAndReadRest(Char, Chars, InStream),
    atom_codes(W, Chars).

checkCharAndReadRest(10, [], _) :- !.
checkCharAndReadRest(32, [], _) :- !.
checkCharAndReadRest(-1, [], _) :- !.
checkCharAndReadRest(end_of_file, [], _) :- !.
checkCharAndReadRest(Char, [Char|Chars], InStream) :-
    get_code(InStream, NextChar),
    checkCharAndReadRest(NextChar, Chars, InStream).
```

代码是如何工作的? 它读取一个字符然后检查这个字符是否是空白(整数为32), 是否为分行符(整数为10), 或者是流的结尾(整数为-1), 以上任何一种情况下都会被当成一个完整单词的结束, 否则的话下一个字符将会进行读取。

## 将内容写入文件

许多的应用程序都需要将输出写入到文件中进行保存, 而不仅仅是显示在屏幕上。本节我们将学习如何在Prolog中将输出内容写入到文件中。

为了写文件, 我们必须创建一个(或者打开一个)文件并且将一个流与其相关联。你可以认为流就是文件的连接。在Prolog中, 流的表现形式很不友好, 名字都是类似"\$stream(1833680)"这样可读性很差的。幸运的是, 你不会直接使用流的名字, 虽然Prolog在内部为其分配了名字, 你可以通过使用Prolog的合一去匹配流名字和一个变量, 然后使用变量操作流, 而不是Prolog中内部分配给流的名字。

如果你想要输出字符串"Hogwarts"到文件hogwarts.txt中, 可以这么做:

```
...
open('hogwarts.txt', write, Stream),
write(Stream, 'Hogwarts'), nl(Stream),
close(Stream),
...
```

如何理解上面的代码？首先，内置的谓词open/3将会创建新的文件：hogwarts.txt；open/3的第二个参数指出我们想要打开一个新的文件（或者覆盖任何已经存在的，相同名字的文件）；open/3的第三个参数返回流的名字。其次，我们将“Hogwarts”写入到流中，并且加入新的一行。最后，我们使用内置谓词close/1关闭掉流。

这就是写文件的操作。正如之前承诺的，我们对流的名字不感兴趣——我们使用合一的变量操作流。还有需要注意的是，谓词write/2是一个更加基础的版本，因为在第九章中使用了write/1将内容输出到屏幕。

如果你不希望复写已经存在的文件，而是在已经存在的文件中添加新的内容呢？可以选择打开文件的方式（不再是write方式）来做到，使用append作为open/3的第二个参数。如果给定名字的文件还不存在，这种模式将会创建一个新的文件。

分类: Prolog

标签: 读文件, 写文件



seaman.kingfall

关注 - 1

粉丝 - 4

+加关注

0

0

« 上一篇: Learn Prolog Now 翻译 - 第十二章 - 文件相关操作 - 第一节, 使用不同文件组织程序

posted on 2015-08-10 13:33 seaman.kingfall 阅读(300) 评论(0) 编辑 收藏  
刷新评论 刷新页面 返回顶部

**注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。**

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【活动】看雪2019安全开发者峰会，共话安全领域焦点

【培训】Java程序员年薪40W，他1年走了别人5年的路

#### 相关博文:

- Learn Prolog Now 翻译 - 第五章 - 数字运算 - 第二节, 数字运算与列表
- Learn Prolog Now 翻译 - 第六章 - 列表补遗 - 第二节, 列表反转
- Learn Prolog Now 翻译 - 第一章 - 事实, 规则和查询 - 第二节, Prolog语法介绍

- [Learn Prolog Now 翻译 - 第十二章 - 文件相关操作 - 第一节, 使用不同文件组织程序](#)
- [Learn Prolog Now 翻译 - 第四章 - 列表 - 第二节, 列表成员](#)

#### 最新新闻:

- [微信公开课聚焦“增长”：墨迹天气小程序DAU环比增100%](#)
  - [知否 | 太空垃圾如何清理？卫星测试用鱼叉击中太空垃圾碎片](#)
  - [一线 | “美团配送”品牌发布：对外开放配送平台 共享配送能力](#)
  - [苍蝇落在食物上会发生什么？让我们说的仔细一点](#)
  - [科学家研究板块构造变化对海洋含氧量影响](#)
- » [更多新闻...](#)

Copyright © seaman.kingfall  
Powered by: .Text and ASP.NET  
Theme by: .NET Monster