

# 编译原理大作业实验说明

课程名:编译原理(A类)

课程号: CS308-1

授课教师: 蒋力

课程学期: 2020-2021 学年第 2 学期

# 1. 实验总体要求

本次编译原理大作业分为基础部分和加分项两部分,其中每部分各占 10 分,合计 20 分。同学们须自行组队,2 人一组,完成大作业。

大作业的截止时间为 2021 年 6 月 28 日 (周一) 24 时。为不影响同学们期末复习,请大家提前规划完成作业。

请在截止日期前在 Canvas 平台指定位置提交大作业的压缩文件,压缩文件内应包含一份实验报告和基础部分的代码,具体要求请分别参看两个部分的详细说明。

实验报告中应写明组内两位同学的姓名、学号、联系方式及分工。

# 2. 基础部分要求

#### 2.1 题目要求

从文本文件读入一个上下文无关的算符文法,构造算符优先分析表并以文本文件形式输出。要求输出的算符优先分析表能够包含所有的非终结符和\$符号,并用">"、"<"、"="或""来表示任意两个非终结符之间的关系。

程序应该能判断文法是否有二义性。如果有二义性,应该输出文字提示,不应该输出算符优先分析表。

#### 2.2 测试文法

有两个文法用于测试及评定分析器的正确性:

#### 2.3 输入输出

输入: 以文本文件的形式输入一条测试文法,示例如下:

in1.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
E -> E + T | T
T -> T \* F | F
F -> (E) | i

输出:如果文法没有二义性,应该以文本文件的形式输出算符优先分析表 (表格形式可以自定),参考示例如下:

🥘 out.txt - 记事本							
文件(F) 编	辑(E) 格式(O)	查看( <u>V</u> )	帮助( <u>H</u> )				
	+	*		(	)	i	\$
						1	
+	>	<	ı	<	>	<	>
*	>	>	· 	<	>	<	>
(	<	<		<	=	<	1
)	>	>			>		>
1 :	I .		 I		I .	1	1
	>	>	 		>	I	>
\$	<	<	1	<	1	l <	I = I

## 2.4 评分标准

报告部分				
能准确、完整说出自己的设计思路(如函数作用、具体实现方法、题目分析等)	2			
报告结构清晰完整,说明了实验过程、结果和环境本等,符合规定要求	1			
思路清晰,逻辑清楚,表达流畅				
程序部分				
程序能正确运行并生成算符优先分析表				
程序能以文本文件形式输入和输出				
两个测试文法的分析结果正确				

## 2.5 提交要求

基础部分的作业需要同学们提交一份实验报告和完整代码。

实验报告应指出实验环境(操作系统、编程语言、编译器版本等)、实验过程、实验结果,并能完整、准确地说出自己的设计思路,描述清楚各个函数的作用、具体实现方法等。

代码部分应包含完整的源代码、用于测试的两个输入文件及对应的输出文件。如果使用编译型语言,请一并上传编译后的程序及可执行文件。

推荐实验环境:

操作系统: Windows/Linux

编程语言及调试环境: C++ (gdb 8.0)

# 3. 加分项部分要求

加分项部分分为两小部分。

## 3.1 第一部分

第一部分熟悉 TVM 框架,按照官网的要求编译 TVM。将给大家的 conv2d.py 的文件导入到大家编译之后的 TVM 的框架中,然后 run 这个文件。

预期的结果: 如果大家可以正常 run 的话,输出结果应该是下面:

得到中间计算的 for 循环的描述和执行时间 (ps:这里的时间应该大家的系统不一样会有差别)。

## 3.2 第二部分

代码中的第 35 行是调用 TVM 框架中的默认的 schedule 的方式,并没有应用一些 TVM 的 schedule 优化的原语 (例如 split 等)。

```
# 这一句是调用tvm默认的schedule函数,表示不加任何优化的schedule
s = tvm.create_schedule(output.op)
```

在这一部分中,需要大家注释 35 行,然后补充 38 行中调用的这个 schedule 函数,然后得到新的调度方式。

```
# 这里需要大家调用tvm有的原语进行loop循环的优化,大家自己去补充 # s = schedule(output)
```

函数的声明在上面的第 12 行。

```
10
11 # 这个函数是需要大家自己补充的,是需要调用各种schedule的原语进行优化的
12 # def schedule(output):
13
14
```

在完成这个函数之后,大家需要获得运行的时间,看看自己优化的 conv2d

相比没有优化之前的能快多少。然后大家需要在实验报告里面填写下面这个表。

输入大小和输出大小	优化前的时间	优化后的时间	提升效率
n, ic, ih, iw = 1, 3, 32, 32			
oc, kh, kw = $32, 3, 3$			
n, ic, ih, iw = 100, 512, 32, 32			
oc, kh, kw = $1024$ , 3, 3			

优化效率=(优化前的时间-优化后的时间)/优化前的时间

#### 3.3 评分细则

提升效率<30%	В
提升效率 30%~60%	A
提升效率>60%	A+

## 3.4 参考链接

卷积是什么,参考下面的链接

卷积的操作:

https://blog.csdn.net/qq\_41398808/article/details/97925070

用 numpy 库手写卷积:

https://blog.csdn.net/free1993/article/details/108709638

TVM 是什么,参考下面的链接:

神经网络编译器: https://www.zhihu.com/question/396105855

TVM 简介:

https://blog.csdn.net/free1993/article/details/111083372

优化 For 循环的原理,参考下面的链接:

优化 For 循环的基本操作:

https://blog.csdn.net/free1993/article/details/116606081

如何编译和使用 schedule 的原语参考,参考下面的链接:

TVM 的官网: https://tvm.apache.org/

## 3.5 提交要求

- 1. 需要提交实验报告及完整代码。
- 2. 实验报告应该包括:编译成功后运行 conv2d. py 的结果的截图(有这个第一步就满分);自己优化的 schedule 函数的截图+上面的表(第二部分分数根据上述评分细则决定)。
- 3. 环境:代码中的 11 vm 是针对 cpu 版本的。所以虚拟机,本地,服务器都可以。大家在实验报告中把环境写清楚。