**PJ3报告**

我们选择了从ast生成目标代码。由于mips指令非常简单易懂，非常适合学习使用，所以我们选择生成mips汇编代码。在生成汇编代码的过程中主要有下面几个阶段。

1. **赋值语句**

在赋值语句，我们很简单地顺着语法分析树从底向上的办法逐个计算。需要注意的是，mips的所有操作语法都是在寄存器中进行的，所以不论是不是中间变量都需要通过放到寄存器后再计算。这里我设计了find\_unused\_register和release\_register两个函数，来管理正在使用和没有使用的寄存器。需要注意的地方是record，array和闭包。对于array，我们可能需要算一些数和赋值语句来得到地址，以及record的key-value的储存。对于闭包我们需要根据local, global的顺序计算出每一个identity是属于local还是global。同时栈空间变量/参数需要额外的设计（5.）

1. **选择语句**

在循环语句中，我们要考虑跳转的问题。我们一共有If then else/elseif 3个段，我们只需要设置最多两个跳转点。先用普通赋值语句计算cond的状态，使用条件跳转指令来进行跳转。之后再根据顺序编译then/else/elseif后的block段。

1. **循环语句**

循环语句实际上和选择语句差不多。对于for/while，等价于一些赋值语句+cond计算语句，如果条件满足则进入for包裹的block；如果不满足条件则立即跳转到代码block末。在block末再设置跳转到循环句首。（如果有until则将cond计算放在block末，如果条件满足则跳转至循环句首。）

1. **声明段**

声明段的第一个麻烦点在于record和array。我们需要计算出size大小，对于array我们可以直接通过乘得到。而record则带来的递归结构，我们需要递归计算size大小。第二个麻烦点在于直接量，直接量是可以直接计算出来的。我们需要模拟运行直接得到一些和变量无关的直接数（例如 T := 3\*5+8）我们需要直接计算出3\*5+8然后将值赋给T。

1. **函数段**

函数段最为麻烦。调用函数的过程：1.将对应参数和当前代码行（栈帧）压入栈空间 2.跳转到对应函数段 3.将函数内的local变量翻译成相对地址（栈空间地址） 4.翻译函数段代码 5.将栈帧清空，把返回行保留到专用栈寄存器中 5.计算返回值（如果有的话），放入专用栈寄存器中 6.跳回原来运行的代码行。

