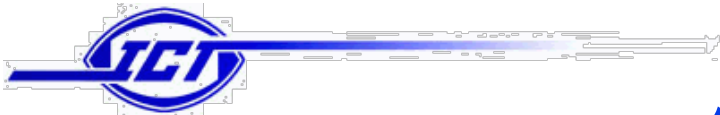


# 第十一次作业

截止日期: 2024.05.20

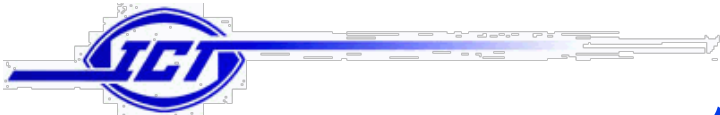
- 练习6.2.1: 假定图 6-26 中的函数 **widen** 可以处理图 6-25a 的层次结构中的所有类型, 翻译下列表达式。假定 **d** 是 **char** 型, **s** 和 **t** 是 **short** 型, **x** 是 **float** 型。
  1.  $x = s + d$
  2.  $x = (s + d) * (t + x)$
- 练习6.2.2: 在图 6-36 的语法制导定义中添加处理下列控制流构造的规则:
  1.  $S \rightarrow \text{repeat } S_1 \text{ until } B$ , 当 **B** 为真时结束循环
  2.  $S \rightarrow \text{for } (S_1 ; B ; S_2) S_3$



# 第十一次作业

- 练习7.1.1：考虑 C 语言的函数 f 和 g：按照图7-7的约定，不考虑编译器优化，讨论当 f 调用 g 而 g 即将返回时运行时栈的状态，其中 f 的参数 a = 3。只需要讨论返回值、参数、控制链和代码中体现的局部数据。指出
  1. 哪个函数在栈中为各个元素创建了所使用的空间？
  2. 哪个函数写入了各个元素的值？参数、返回值和局部变量的值是什么？
  3. 这些元素属于哪个活动记录？

```
int g(int *);  
int f(int a) {  
    int i = a + 2;  
    return g(&i);  
}  
int g(int *b) {  
    int j = *b;  
    return j + 2;  
}
```

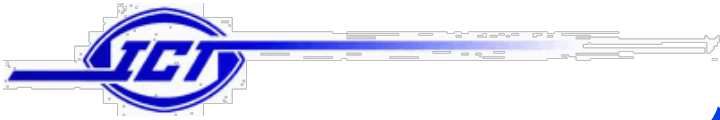


## 第十一次作业

- 练习7.1.2：考虑下面的Fibonacci函数：

嵌套在fib0中的是fib1，它假设 $n \geq 2$ 并计算第 $n$ 个Fibonacci数。嵌套在fib1中的是fib2，它假设 $n \geq 4$ 。请注意，fib1和fib2都不需要检查基本情况。我们考虑从对main的调用开始，直到（对fib0(1)的）第一次调用即将返回的时段。

1. 请描述出当时的活动记录栈，并给出栈中的各个活动记录的访问链。
2. 假设我们使用display表来实现下图中的函数。请给出fib0(1) 的第一次调用即将返回时的display表。同时指明那时在栈中的各个活动记录中保存的display表条目。



# 第十一次作业

---

```
fun main() {  
  let  
    fun fib0(n) =  
      let  
        fun fib1(n) =  
          let  
            fun fib2(n) = fib1(n-1) + fib1(n-2)  
          in  
            if n >= 4 then fib2(n)  
            else fib0(n-1) + fib0(n-2)  
          in  
            if n >= 2 then fib1(n)  
          end  
        in  
          fib0(4)  
        end;  
      }  
}
```