栈与队列

调用栈：尾递归

04-B4

《星期评论》问我“女子解放从那里做起？”我的答案是：
“女子解放当从女子解放做起。此外更无别法。”

邓俊辉
deng@tsinghua.edu.cn

定义

- 在递归实例中，作为最后一步的递归调用
- fac(n) {

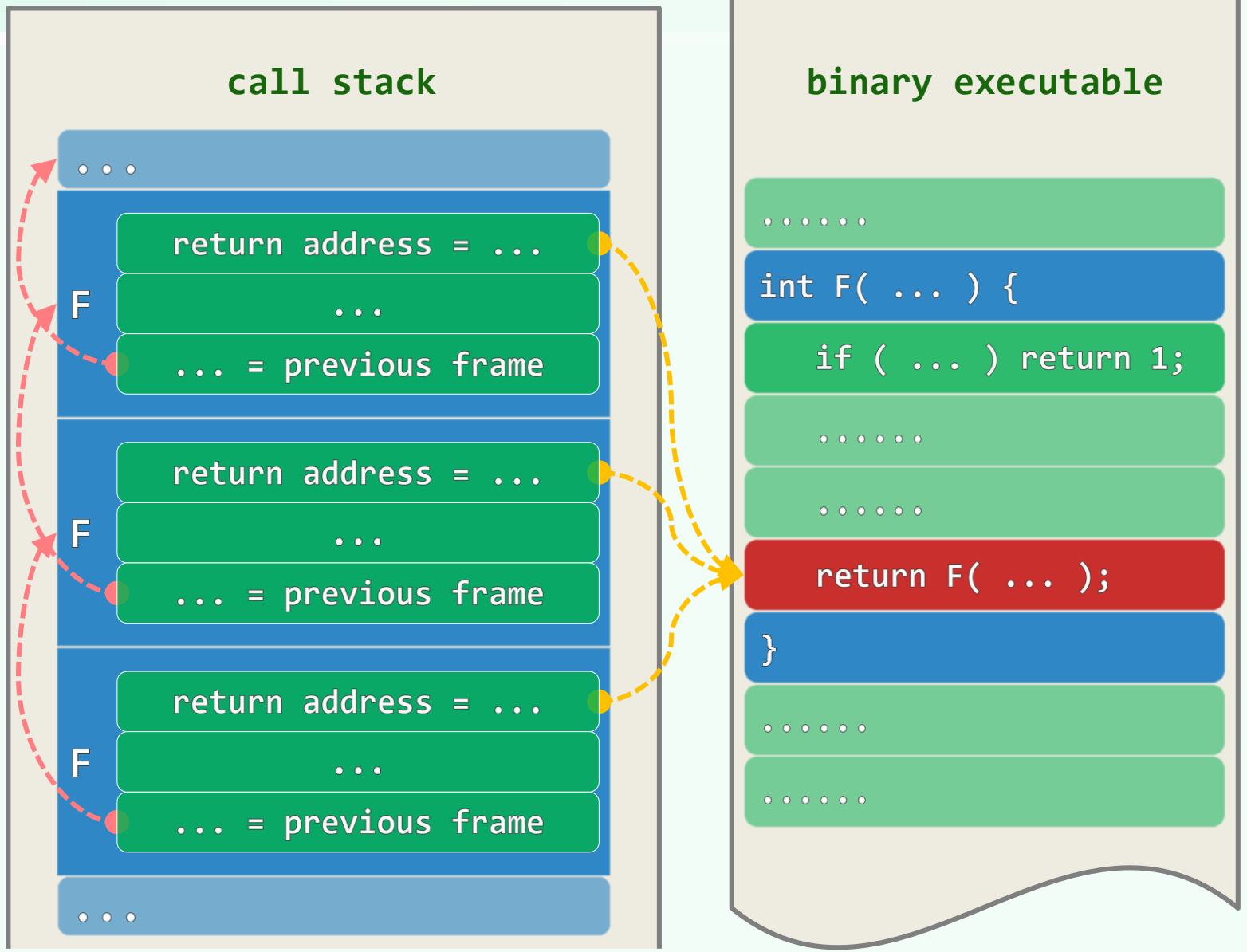
```
    if (1 > n) return 1; //base

    return n * fac( n-1 ); //tail recursion
}
```

- 严格地讲，上例在递归返回前的最后一步是乘法

  但这不是本质问题

性质

- 系最简单的递归模式
- 一旦抵达递归基，便会
  - 引发一连串的return
    （且返回地址相同）
  - 调用栈相应地连续pop
- 故不难改写为迭代形式
- 越来越多的编译器可以
  自动识别并代为改写
- 时间复杂度有常系数改进
  空间复杂度或有渐近改进



消除

```
fac(n) { //尾递归
   if (1 > n) return 1;
   return n * fac( n-1 );
} //O(n)时间 + O(n)空间
```

```
fac(n) { //统一转换为迭代
   int f = 1; //记录子问题的解
next: //转向标志，模拟递归调用
   if (1 > n) return f;
   f *= n--;
   goto next; //模拟递归返回
} //O(n)时间 + O(1)空间
```

```
fac(n) { //简捷
   int f = 1;
   while (1 < n)
      f *= n--;
   return f;
} //O(n)时间 + O(1)空间
```