



```
const fact = f => n => n < 2 ? 1 : n * f(f)(n - 1);
```

调用变化

fact

$\Rightarrow (f \Rightarrow n \Rightarrow n < 2 ? 1 : n * f(f)(n - 1))$

夕提

$$= \lambda (f = \lambda (g = \lambda n = \lambda n < 2 ? 1 : n * g(n - 1))) (f(f))$$

这不就是函数映射P吗

$$= \Rightarrow (\neg \Rightarrow f \Rightarrow \neg (x \Rightarrow f)(x))(P)$$

$$= \sup ( \bigcup_{f \in \bigcup (f(F))} (P) )$$



# 延迟计算防止爆栈

$$(\forall x \Rightarrow f(x) \Rightarrow f(f(x)))(P) \text{ 等价于 } \text{fact}$$

`const fact = f => n => n < 2 ? 1 : n * f(f)(n - 1);`  
调用变化

`fact`

`=> (f => n => n < 2 ? 1 : n * f(f)(n - 1))`  
外提

`=> (f => (g => n => n < 2 ? 1 : n * g(n - 1))(f(f)))`  
这不就是函数映射P吗

`=> (l => f => l(f(f)))(P)`  
延迟计算防止爆栈

`=> (l => f => l(x => f(f)(x)))(P)`

`(l => f => l(x => f(f)(x)))(P)` 等价于 `fact`

```
fact(fact)(n)
```