软件理论基础与实践

这是一次习题课（伪）

2022年3月17日（星期四）

Church Encoding

0: O

1: $S\ O$

2: $S\ (S\ O)$

3: $S\ (S\ (S\ O))$

…

Church Encoding

0: $\lambda f.\ O$

1: $\lambda f.\ f\ O$

2: $\lambda f.\ f\ (f\ O)$

3: $\lambda f.\ f\ (f\ (f\ O))$

…

Church Encoding

0: $\lambda f.\ \lambda x.\ x$

1: $\lambda f.\ \lambda x.\ f\ x$

2: $\lambda f.\ \lambda x.\ f\ (f\ x)$

3: $\lambda f.\ \lambda x.\ f\ (f\ (f\ x))$

…

Church Encoding

$$(0\ f\ x) = f^0(x)$$
$$(1\ f\ x) = f^1(x)$$
$$(2\ f\ x) = f^2(x)$$
$$(3\ f\ x) = f^3(x)$$

…

Church Encoding

```
Definition cnat := forall X : Type, (X -> X) -> X -> X.

Definition zero : cnat :=
  fun (X : Type) (f : X -> X) (x : X) => x.
Definition one : cnat :=
  fun (X : Type) (f : X -> X) (x : X) => f x.
Definition two : cnat :=
  fun (X : Type) (f : X -> X) (x : X) => f (f x).

...
```

Church Encoding

后继

$$f^{n+1}(x) = f(f^n(x)) = f(\mathfrak{n}\ f\ x)$$

```
Definition succ (n : cnat) : cnat :=
  fun (X : Type) (f : X -> X) (x : X) => f (n X f x).
```

Church Encoding

加法 $n + m$

$$f^{n+m}(x) = f^m f^n(x) = f^m(\mathfrak{n}\ f\ x) = \mathfrak{m}\ f\ (\mathfrak{n}\ f\ x)$$

```
Definition plus (n m : cnat) : cnat :=
  fun (X : Type) (f : X -> X) (x : X) => m X f (n X f x).
```

Church Encoding

乘法 $n \times m$

$$f^{n \times m}(x) = (f^n)^m(x) = (\mathfrak{n}\ f)^m(x) = \mathfrak{m}\ (\mathfrak{n}\ f)\ x$$

```
Definition mult (n m : cnat) : cnat :=
  fun (X : Type) (f : X -> X) (x : X) => m X (n X f) x.
```

Church Encoding

乘法 $n \times m$

$$f^{n\times m}(x) = (f^n)^m(x) = (\mathfrak{n}\ f)^m(x) = \mathfrak{m}\ (\mathfrak{n}\ f)\ x$$

```
Definition mult (n m : cnat) : cnat :=
  fun (X : Type) (f : X -> X) => m X (n X f).
```

Church Encoding

乘方 n^m

$$(\mathfrak{m}\ f\ \_) = f^m(\_) \longrightarrow (\mathfrak{m}\ \mathfrak{n}\ \_) = n^m(\_)$$

```
Definition exp (n m : cnat) : cnat :=
  := fun (X : Type) => m (X -> X) (n X).
```

Church Encoding

布尔值

True: $\lambda x.\ \lambda y.\ x$

False: $\lambda x.\ \lambda y.\ y$

```
Definition true  (X : Type) := fun (x y : X) => x.
Definition false (X : Type) := fun (x y : X) => y.
```

Church Encoding

条件语句

`true then-expr else-expr` = `then-expr`

`false then-expr else-expr` = `else-expr`

逻辑运算

and: $\lambda p.\ \lambda q.\ p\ q\ p$
or: $\lambda p.\ \lambda q.\ p\ p\ q$

Church Encoding

- 减法
- 序对 pair: $\lambda x.\ \lambda y.\ \lambda z.\ z\ x\ y$
- 分数 $q = \frac{k}{1+a}$
- 实数 $|x - q| < 2^{-k}, k \in \mathbb{N}$

软件理论基础与实践

这是一次习题课（真）

2022年3月17日（星期四）

Poly.mumble_grumble

```
Inductive mumble : Type :=
  | a
  | b (x : mumble) (y : nat)
  | c.

Inductive grumble (X:Type) : Type :=
  | d (m : mumble)
  | e (x : X).
```

Which of the following are well-typed elements of `grumble X` for some type `X`.

- `c` : NO

Lists.rev_injective

```
Theorem rev_injective : forall (l1 l2 : natlist),
    rev l1 = rev l2 -> l1 = l2.
```

There is a hard way and an easy way to do this.

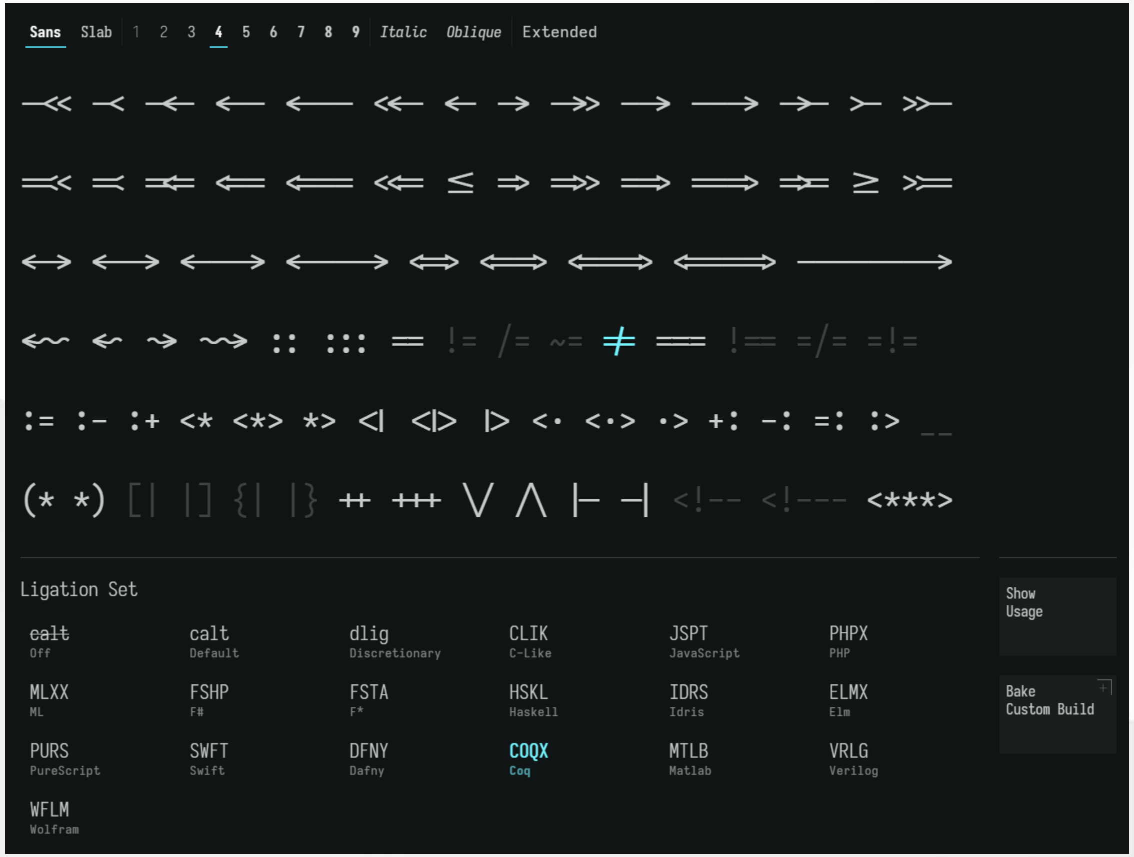
```
Proof.
  intros l1 l2 H.
  replace (l1) with (rev (rev l1)).
  replace (l2) with (rev (rev l2)).
  rewrite H. reflexivity.
  - rewrite rev_involutive. reflexivity.
  - rewrite rev_involutive. reflexivity.
Qed.
```

一些其他的事情

1. 提交的邮件名
   ~~2022MMDD-~~ 2000000000-张三
2. 提交的文件
   只提交更改的文件对**我**而言最简单
3. 记得做手写题!

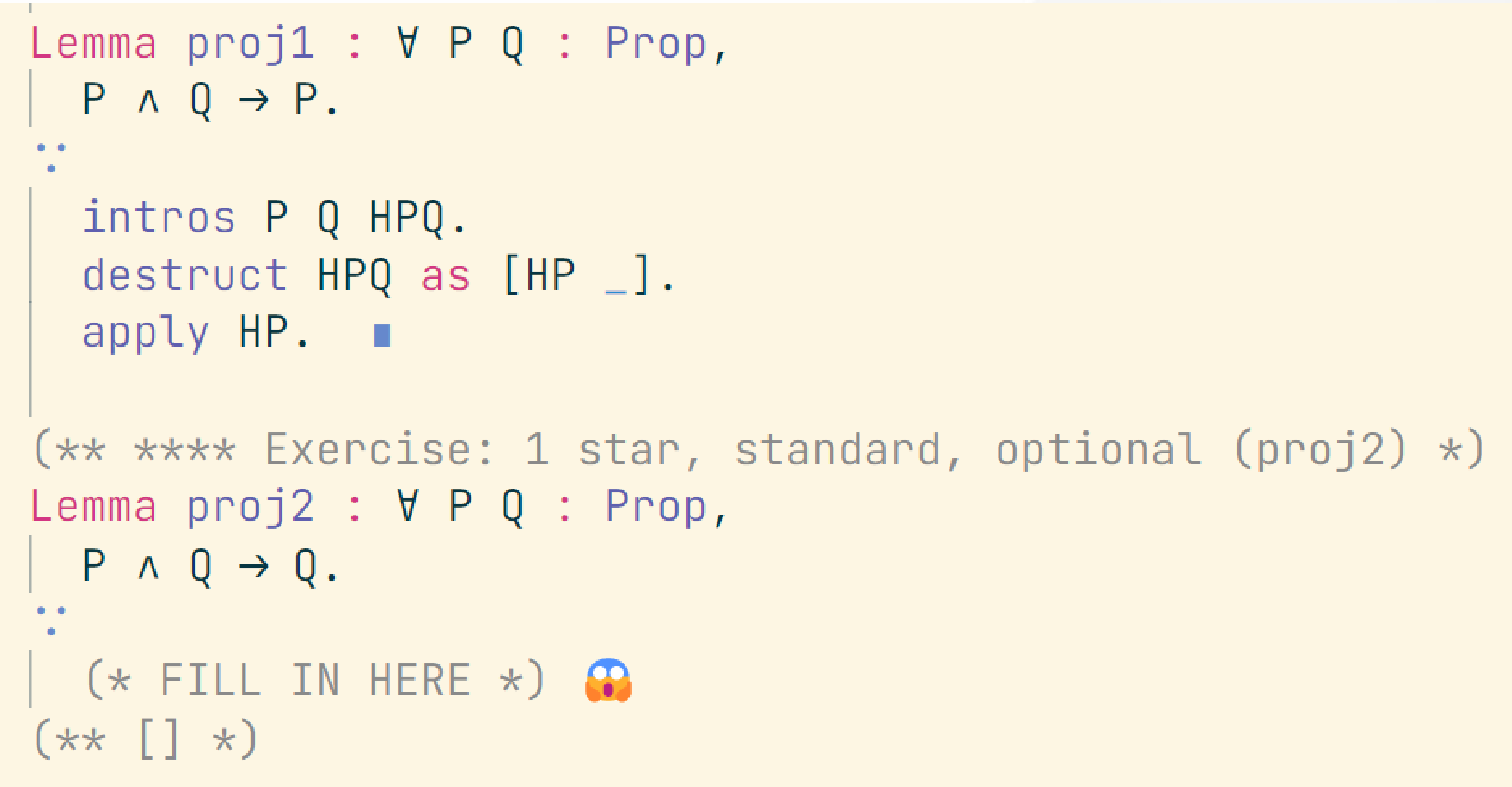
一些其他的推荐

字体：Iosevka `"editor.fontLigatures": "'calt' off, 'COQX' on",`



一些其他的推荐

VSCode 插件：Conceal



```
Lemma proj1 : ∀ P Q : Prop,
  P ∧ Q → P.
∵
  intros P Q HPQ.
  destruct HPQ as [HP _].
  apply HP.  ■

(** **** Exercise: 1 star, standard, optional (proj2) *)
Lemma proj2 : ∀ P Q : Prop,
  P ∧ Q → Q.
∵
  (* FILL IN HERE *) 😱
(** [] *)
```

谢谢大家