软件理论基础与实践

TYPECHECKING: A Typechecker for STLC

熊英飞
北京大学

类型检查

- 之前只定义了归纳定义的has_type关系
- 能否定义为函数自动检查类型正确性?
  - 能，因为之前证明了unique_types定理

```
Theorem unique_types : forall Gamma e T T',
  Gamma |- e \in T ->
  Gamma |- e \in T' ->
  T = T'.
```

辅助函数：判断类型等价


```
Fixpoint eqb_ty (T1 T2:ty) : bool :=
  match T1,T2 with
  | <{ Bool }> , <{ Bool }> =>
      true
  | <{ T11->T12 }>, <{ T21->T22 }> =>
      andb (eqb_ty T11 T21) (eqb_ty T12 T22)
  | _,_ =>
      false
  end.
```

类型检查函数


```
Fixpoint type_check (Gamma : context) (t : tm) : option ty :=
  match t with
  | tm_var x =>
      Gamma x
  | <{\x:T2, t1}> =>
      match type_check (x |-> T2 ; Gamma) t1 with
      | Some T1 => Some <{T2->T1}>
      | _ => None
      end
  | <{t1 t2}> =>
      match type_check Gamma t1, type_check Gamma t2 with
      | Some <{T11->T12}>, Some T2 =>
          if eqb_ty T11 T2 then Some T12 else None
      | _,_ => None
      end
```

类型检查函数


```
| <{true}> =>
    Some <{Bool}>
| <{false}> =>
    Some <{Bool}>
| <{if guard then t else f}> =>
    match type_check Gamma guard with
    | Some <{Bool}> =>
        match type_check Gamma t, type_check Gamma f with
        | Some T1, Some T2 =>
            if eqb_ty T1 T2 then Some T1 else None
        | _,_ => None
        end
    | _ => None
    end
end.
```

类型检查函数的性质


```
Theorem type_checking_sound : forall Gamma t T,
  type_check Gamma t = Some T -> has_type Gamma t T.
```

```
Theorem type_checking_complete : forall Gamma t T,
  has_type Gamma t T -> type_check Gamma t = Some T.
```

证明思路：在t或者has_type上做归纳，对应调用另外一边函数或者constructor即可

作业

- 无