



软件科学基础

HoareAsLogic: Hoare Logic as a Logic

熊英飞
北京大学



复习

- 操作语义的7条规则以及规则名称
- 霍尔逻辑的6条规则



复习：霍尔逻辑的性质

- 正确性Soundness: 所有用霍尔逻辑规则推导出来的霍尔三元组在IMP的语义下都是正确的，即给定霍尔三元组 $\{P\}c\{Q\}$
 - 给定任意满足P的状态，执行c后，Q一定满足
- 完备性Completeness: 所有在IMP语义下正确的霍尔三元组都可以用霍尔逻辑推导出来
- 本课程后续我们将证明这两个性质



复习：Coq中的霍尔逻辑

- 基于IMP的语法和语义，将霍尔逻辑规则证明成定理
 - 即模型论的方法
 - 基于IMP的语法，将霍尔逻辑规则定义成归纳定义命题的constructor
 - 即逻辑的方法
- 接下来我们首先用模型论的方法定义霍尔逻辑。



逻辑的方法

- 将霍尔三元组定义为归纳定义的关系
- 将霍尔逻辑规则定义为该关系的constructor
- 即，该关系包括且仅包括所有用霍尔逻辑规则可以推出的三元组



关系：可推导三元组

```
Inductive derivable : Assertion -> com -> Assertion -> Type :=
| H_Skip : forall P,
  derivable P <{skip}> P
| H_Asgn : forall Q V a,
  derivable (Q [V |-> a]) <{V := a}> Q
| H_Seq  : forall P c Q d R,
  derivable P c Q -> derivable Q d R -> derivable P <{c;d}> R
| H_If   : forall P Q b c1 c2,
  derivable (fun st => P st /\ bassn b st) c1 Q ->
  derivable (fun st => P st /\ ~(bassn b st)) c2 Q ->
  derivable P <{if b then c1 else c2 end}> Q
```



关系：可推导三元组

```
| H_While : forall P b c,  
  derivable (fun st => P st /\ bassn b st) c P ->  
  derivable P <{while b do c end}>  
    (fun st => P st /\ ~ (bassn b st))  
| H_Consequence : forall (P Q P' Q' : Assertion) c,  
  derivable P' c Q' ->  
  (forall st, P st -> P' st) ->  
  (forall st, Q' st -> Q st) ->  
  derivable P c Q.
```



两个方便使用的引理

```
Lemma H_Consequence_pre : forall (P Q P' : Assertion) c,  
  derivable P' c Q ->  
  (forall st, P st -> P' st) ->  
  derivable P c Q.
```

```
Proof. eauto using H_Consequence. Qed.
```

```
Lemma H_Consequence_post : forall (P Q Q' : Assertion) c,  
  derivable P c Q' ->  
  (forall st, Q' st -> Q st) ->  
  derivable P c Q.
```

```
Proof. eauto using H_Consequence. Qed.
```




正确性

```
Definition valid (P : Assertion)
  (c : com) (Q : Assertion) : Prop :=
  forall st st',
    st =[ c ]=> st' ->
    P st ->
    Q st'.
```

```
Theorem hoare_sound : forall P c Q,
  derivable P c Q -> valid P c Q.
```

只需要重复上一章的证明即可，留作作业。



完备性

Theorem `hoare_complete`: `forall` P c Q,
valid P c Q \rightarrow derivable P c Q.

Proof.

```
unfold valid. intros P c. generalize dependent P.  
induction c; intros P Q HT.
```

```
1:{  
  (* HT: forall st st' : state, st =[ skip ]=> st' -> P st -> Q st'  
    Goal: derivable P <{ skip }> Q *)  
  apply H_Consequence with (P' := P) (Q' := P).  
  (* Goal1: derivable P <{ skip }> P  
    Goal2: forall st : state, P st -> P st  
    Goal3: forall st : state, P st -> Q st *)  
  * apply H_Skip.  
  all: eauto.  
}
```

证明思路：对任意Valid的三元组，构造相应的霍尔逻辑规则应用序列



完备性

```
2: {  
  (* IHc1: forall P Q : Assertion,  
    valid P c1 Q -> derivable P c1 Q  
    IHc2: forall P Q : Assertion,  
    valid P c2 Q -> derivable P c2 Q  
    HT: forall st st' : state,  
        st =[ c1; c2 ]=> st' -> P st -> Q st'  
    Goal: derivable P <{ c1; c2 }> Q  
  *)  
  .....
```

问题：某些情况的应用序列不容易构造，需要找到合适中间断言



如何证明Seq

- 需要c1和c2的中间断言
- 该中间断言只需要满足如下两个条件
 - 弱于 $\{P\}c1$ 的最强后条件
 - 强于 $c2\{Q\}$ 的最弱前条件
- 取任意一个作为中间断言都可以



定义最弱前条件证明Seq

Definition wp (c:com) (Q:Assertion) : Assertion :=
 fun s => forall s', s =[c]=> s' -> Q s'.

Hint Unfold wp : core.

基于wp可以完成sequence的证明

```
2: {  
  (* IHc1: forall P Q : Assertion,  
    valid P c1 Q -> derivable P c1 Q  
    IHc2: forall P Q : Assertion,  
    valid P c2 Q -> derivable P c2 Q  
    HT: forall st st' : state,  
        st =[ c1; c2 ]=> st' -> P st -> Q st'  
    Goal: derivable P <{ c1; c2 }> Q *)  
  apply H_Seq with (Q:=(wp c2 Q)).  
  (* Goal1: derivable P c1 (wp c2 Q)  
    Goal2: derivable (wp c2 Q) c2 Q *)  
  all: eauto.
```



如何证明While

- 证明While需要循环不变式
 - 能被前条件推出
 - 在循环执行过程中一直保持
 - 能推出后条件
- 如何选择循环不变式?
- 即对应的状态集合为循环执行过程中的所有状态
- 执行过程中的任意状态在经过循环之后都满足Q
 - 即循环的最弱前条件



最弱前条件作为循环不变式

Lemma `wp_invariant` : `forall` `b c Q`,
 `valid` (`wp` `<{while b do c end}>` `Q /\ b`)
 `c` (`wp` `<{while b do c end}>` `Q`).

Proof.

`unfold` `valid`, `wp`.

`intros`.

`(* WHILE = <{while b do c end}>`
 `H: st =[c]=> st'`
 `H0: (wp WHILE Q /\ b) st`
 `H1: st'=[WHILE]=>s`
 `Goal: Q s *)`

`apply` `H0`.

`eapply` `E_WhileTrue`.

`* (* beval st b = true *) apply` `H0`.

`* (* st =[c]=> ?st' *) apply` `H`.

`* (* st'=[WHILE]=> s' *) apply` `H1`.

Qed.



证明While

```
3: {  
  (* WHILE = <{while b do c end}>  
    Inv = wp WHILE Q  
    IHc: forall P Q, valid P c Q -> derivable P c Q  
    HT: valid P WHILE Q  
    Goal: derivable P WHILE Q  
  *)  
  eapply H_Consequence.  
  eapply H_While with (b:=b) (c:=c)  
    (P:=wp <{while b do c end}> Q).  
  * (* derivable (Inv /\ b) c Inv *)  
    apply IHc.  
    apply wp_invariant.  
  * (* P ->> Inv *)  
    intros. eauto.  
  * (* Inv /\ ~b ->> Q *)  
    intros. eapply H. apply E_WhileFalse.  
    simpl in H. apply Bool.not_true_is_false. apply H.  
}
```

剩下证明留
作作业



霍尔逻辑的可判定性

- 和我们预期相同，霍尔逻辑是不可判定的
- 证明：
 - 将停机问题规约为霍尔三元组
 - 假设当前要判断 c 的停机问题
 - 等价于判断 $\{True\}c\{False\}$ 是否成立



作业

- 完成HoareAsLogic中的6道习题
 - 部分证明课上已经给出
 - 请使用最新英文版教材