

Exercise Sheet

10/14

1. 【判断题】以下说法是否正确：对于整数类型表达式 e ， $\llbracket e \rrbracket$ 表示 e 的值，是一个整数。
2. 【判断题】以下说法是否正确：对于整数类型表达式 e 与程序状态 s ， $\llbracket e \rrbracket(s)$ 表示 e 的语义，是一个整数到整数的函数。
3. 【判断题】以下说法是否正确： $\llbracket 0 \rrbracket = 0$ 。
4. 【判断题】以下说法是否正确：对于整数类型表达式 e_1 及 e_2 ，与程序状态 s ， $\llbracket e_1 + e_2 \rrbracket(s) = \llbracket e_1 \rrbracket(s) + \llbracket e_2 \rrbracket(s)$ 。
5. 【判断题】以下说法是否正确：对于整数类型表达式 e 与程序状态 s_1 及 s_2 ，如果 $s_1 \neq s_2$ ，那么 $\llbracket e \rrbracket(s_1) \neq \llbracket e \rrbracket(s_2)$ 。
6. 【判断题】以下说法是否正确：对于整数类型表达式 e 与程序状态 s_1 及 s_2 ，如果 $s_1 \neq s_2$ 并且 $\llbracket e \rrbracket(s_1) = \llbracket e \rrbracket(s_2)$ ，那么 e 中不包含变量。
7. 【判断题】以下说法是否正确：对于任意一个整数类型表达式 e ，都存在一个 $\text{state} \rightarrow \mathbb{Z}$ 的函数是 e 的语义。
8. 【判断题】以下说法是否正确：对于任意一个函数 $f : \text{state} \rightarrow \mathbb{Z}$ ，都存在唯一一个整数类型表达式 e 使得 $\llbracket e \rrbracket = f$ 。
9. 要展开下面证明目标中“行为等价”的定义

```
-----  
(1/1)  
forall a b c: expr_int,  
  [[ a + (b + c) ]] ~== [[ a + b + c ]]
```

应该使用以下哪条证明脚本：

- A. `intros.`
- B. `simpl.`
- C. `unfold iequiv.`
- D. `reflexivity.`

10. 要依据表达式指称语义的定义化简下面证明目标的结论

```
a : expr_int  
-----  
(1/1)  
forall s : state, [[ a * 0 ]] s = [[ 0 ]] s
```

应该使用以下哪条证明脚本：

- A. `intros.`
- B. `simpl.`
- C. `unfold iequiv.`

- D. reflexivity.

11. 我们知道, `iequiv_symm` 可以使用对称性 (`Symmetric`)、行为等价 (`iequiv`) 的定义进行证明。

```
#[export] Instance iequiv_symm: Symmetric iequiv.
Proof.
  unfold Symmetric, iequiv.
  intros.
  rewrite H.
  reflexivity.
Qed.
```

其中的 `rewrite H` 指令, 将证明目标由

```
x, y : expr_int
H : forall s : state, [[ x ]] s = [[ y ]] s
s : state
-----
(1/1)
[[ y ]] s = [[ x ]] s
```

变为

```
x, y : expr_int
H : forall s : state, [[ x ]] s = [[ y ]] s
s : state
-----
(1/1)
[[ y ]] s = [[ y ]] s
```

请问这步变换用到了整数相等的哪个或哪些性质?

- A. 自反性
- B. 对称性
- C. 传递性

12. 在 `iequiv_symm` 的证明中也可以将左向右 `rewrite` 改为右向左 `rewrite`, 换言之, 可以在下面证明目标中

```
x, y : expr_int
H : forall s : state, [[ x ]] s = [[ y ]] s
s : state
-----
(1/1)
[[ y ]] s = [[ x ]] s
```

使用 `rewrite <- H` 变换得到以下证明目标。

```
x, y : expr_int
H : forall s : state, [[ x ]] s = [[ y ]] s
s : state
```

(1/1)

```
[[ x ]] s = [[ x ]] s
```

请问这步变换用到了整数相等的哪个或哪些性质？

- A. 自反性
- B. 对称性
- C. 传递性