

Discussion #6: Type Systems

1 Type Checking

For this problem consider that you have the following Cool class declarations. Notice that the body of method `a` is not yet filled in. Do not worry about syntactic errors when writing Cool code in this problem.

```
class A {  
  a() : SELF_TYPE {  
  
  }  
}  
  
class B inherits A {  
  b() : Int { 0 }  
}
```

- (a) Consider the following modified typing rule for Cool conditionals.

$$\frac{O, M, C \vdash e_1 : \text{Bool} \quad O, M, C \vdash e_2 : T_2 \quad O, M, C \vdash e_3 : T_3 \quad T_2 \leq T_3}{O, M, C \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 \text{ fi} : T_2}$$

Write a very short cool program fragment that uses the classes defined above and typechecks with the modified rule for conditionals, but attempts to invoke the method `b` on an object of dynamic type `A` at run time.

- (b) Exactly one of the following Cool subtyping rules is sound for all classes `C`. Circle the **unsound** rule: $C \leq \text{SELF_TYPE}_C$ or $\text{SELF_TYPE}_C \leq C$.

To demonstrate that the rule you **circled** is **unsound**, fill in the body of method `a` above and write a short Cool program fragment that typechecks with the **unsound** subtyping rule and generates a type error at run-time.

2 Operational Semantics

Consider the following **modified** operational semantics rules for **while** in Cool.

$$\frac{so, S_1, E \vdash e_1 : \text{Bool}(\text{false}), S_2}{so, S_1, E \vdash \text{while } e_1 \text{ loop } e_2 \text{ pool} : \text{Int}(0), S_2}$$

$$\frac{so, S_1, E \vdash e_1 : \text{Bool}(\text{true}), S_2 \quad so, S_2, E \vdash e_2 : v_2, S_3 \quad so, S_3, E \vdash \text{while } e_1 \text{ loop } e_2 \text{ pool} : v_3, S_4}{so, S_1, E \vdash \text{while } e_1 \text{ loop } e_2 \text{ pool} : v_2, S_4}$$

- (a) In one or two sentences, explain the return value of this new **while** expression.
- (b) Give a typing rule for this new **while** expression. The rule must preserve type safety of Cool and must accept as many programs as possible.

- (c) Now imagine that we want a **while** loop that counts its number of iterations. The value of a **while** loop should be an integer object that holds the number of times the body has been evaluated. Give the typing rule for this new **while** expression.
- (d) Give the operational semantic rules for the “counting” **while** described in part (c).