

PrograMet: A Program Similarity Metric

Justin Wong
Columbia University

PrograMet: A Program Similarity Metric

Motivating Example: Dead Code

```
def IF1(x,y,z):  
    if (x < 0):  
        y = 1  
    else:  
        y = 2  
    if (y > 1):  
        z = -z  
    return z  
      
def IF1_dead(x,y,z):  
    if (x < 0):  
        y = 1  
    else:  
        y = 2  
    if (y > 2):  
        z = -z  
    return z  
      
def IF1_clean(x,y,z):  
    if (x < 0):  
        y = 1  
    else:  
        y = 2  
    //if (y > 2):  
        //z = -z  
    return z
```

The diagram illustrates the transformation of a program through mutation and cleaning. The original function `IF1` is mutated into `IF1_dead`, where the condition `y > 1` is changed to `y > 2`. This mutation introduces dead code, as the branch `if (y > 2): z = -z` is never executed. The `IF1_dead` function is then cleaned to produce `IF1_clean`, where the dead code is removed, leaving only the original logic of `IF1`.

PrograMet: A Program Similarity Metric

Question: Can we capture both dynamic and static features?

