Task: 人工确认语义等价label

任务内容

每个人会分得一组数据，组id取值范围为[1,46]，每组数据中含有10个文件夹(第46组数据中包含6个）。对于每个文件夹（文件夹中含有n个文件），每次取其中的任意2个文件，判断二者的源代码是否等价，并记录在csv文件中。若等价，则记录在"组id_equal_pairs.csv"中，若不等价，则记录在""组id_inequal_pairs.csv"中。(例如分到的是第3组数据，则提交的文件名为3_equal_pairs.csv和3_inequal_pairs.csv。）每个文件夹产生的总数据量（包括等价和不等价）应为 C_n^2 。（在没有重复数据的情况下。重复数据的情况处理下文会提到。）

csv文件中的每个单元格的内容由“文件夹名/文件名“组成，具体格式如下：

	A	B
1	file1	file2
2	3400/receive_pack_config_132-770902_252-826032.foo.c	3368/receive_pack_config_132-770902_257-826176.foo.c
3	3400/receive_pack_config_132-770902_252-826032.foo.c	3368/receive_pack_config_134-770964_257-826176.foo.c
4	3400/receive_pack_config_134-770964_252-826032.foo.c	3368/receive_pack_config_132-770902_257-826176.foo.c
5	3400/receive_pack_config_134-770964_252-826032.foo.c	3368/receive_pack_config_134-770964_257-826176.foo.c
6	311/blk_SHA256_Transform_65-519364_117-524591.foo.c	1381/blk_SHA256_Transform_89-520659_136-527378.foo.c
7	311/blk_SHA256_Transform_65-519364_117-524591.foo.c	1381/blk_SHA256_Transform_98-521944_145-528665.foo.c
8	311/blk_SHA256_Transform_65-519364_117-524591.foo.c	1381/blk_SHA256_Transform_89-520702_142-528236.foo.c
9	311/blk_SHA256_Transform_69-519415_144-528452.foo.c	1381/blk_SHA256_Transform_89-520659_136-527378.foo.c
10	311/blk_SHA256_Transform_69-519415_144-528452.foo.c	1381/blk_SHA256_Transform_98-521944_145-528665.foo.c
11	311/blk_SHA256_Transform_69-519415_144-528452.foo.c	1381/blk_SHA256_Transform_89-520702_142-528236.foo.c
12	311/blk_SHA256_Transform_65-519364_122-525306.foo.c	1381/blk_SHA256_Transform_89-520659_136-527378.foo.c
13	311/blk_SHA256_Transform_65-519364_122-525306.foo.c	1381/blk_SHA256_Transform_98-521944_145-528665.foo.c
14	311/blk_SHA256_Transform_65-519364_122-525306.foo.c	1381/blk_SHA256_Transform_89-520702_142-528236.foo.c
15	1417/blk_SHA256_Transform_90-520732_143-528309.foo.c	571/blk_SHA256_Transform_65-519364_134-527092.foo.c
16	1417/blk_SHA256_Transform_90-520732_143-528309.foo.c	571/blk_SHA256_Transform_69-519415_118-524804.foo.c
17	1417/blk_SHA256_Transform_90-520732_143-528309.foo.c	571/blk_SHA256_Transform_65-519364_130-526520.foo.c
18	1417/blk_SHA256_Transform_90-520732_143-528309.foo.c	571/blk_SHA256_Transform_69-519415_138-527664.foo.c
19	1417/blk_SHA256_Transform_96-521658_143-528309.foo.c	571/blk_SHA256_Transform_65-519364_134-527092.foo.c

如何判断语义等价

以“__dyc_read”开头的函数表示读取不同格式的输入。在下图中，红框中的内容即为该函数的输入。

```
{
ut_dbg_null_ptr = __dyc_read_ptr__typdef_ulint();
dict_sys = __dyc_read_ptr__typdef_dict_sys_t();
name_len = (ulint )__dyc_readpre_byte();
is_sys_table = (ulint )__dyc_readpre_byte();
table = __dyc_read_ptr__typdef_dict_table_t();
__dyc_funcallvar_6 = __dyc_read_ptr__typdef_dtuple_t();
__dyc_funcallvar_7 = __dyc_read_ptr__typdef_dfield_t();
__dyc_funcallvar_8 = __dyc_read_ptr__void();
__dyc_funcallvar_9 = (ulint )__dyc_readpre_byte();
__dyc_funcallvar_10 = __dyc_read_ptr__typdef_rec_t();
__dyc_funcallvar_11 = (unsigned char *)__dyc_read_ptr__char();
__dyc_funcallvar_12 = __dyc_readpre_byte();
__dyc_funcallvar_13 = (ulint )__dyc_readpre_byte();
__dyc_funcallvar_14 = (unsigned char *)__dyc_read_ptr__char();
__dyc_funcallvar_15 = __dyc_read_comp_100dulint_struct();
__dyc_funcallvar_16 = __dyc_read_ptr__typdef_dict_field_t();
__dyc_funcallvar_17 = __dyc_read_ptr__typdef_dict_col_t();
__dyc_funcallvar_18 = __dyc_readpre_byte();
```

以“__dyc_print”开头的函数表示输出不同格式的数据，其参数即为函数输出。在下图中，红框中的变量值即为函数的输出。

```
__dyc_print_ptr__typdef_ulint(ut_dbg_null_ptr);
__dyc_print_ptr__typdef_dtuple_t(tuple);
__dyc_print_ptr__typdef_dfield_t(dfield);
__dyc_print_ptr__typdef_rec_t(rec);
__dyc_print_ptr__char(field);
__dyc_print_ptr__char(name_buf);
__dyc_printpre_byte(type);
__dyc_printpre_byte(space);
__dyc_printpre_byte(n_fields);
__dyc_print_ptr__char(buf);
__dyc_print_comp_100dulint_struct(id);
__dyc_print_ptr__typdef_dict_field_t(tmp___7);
__dyc_print_ptr__typdef_dict_col_t(tmp___8);
__dyc_printpre_byte(tmp___10);
__dyc_print_ptr__typdef_dict_field_t(tmp___13);
__dyc_print_ptr__typdef_dict_col_t(tmp___14);
__dyc_printpre_byte(tmp___16);
}
```

函数的所有输入值都是随机生成的，且相互独立。

判断等价时两个函数的输入（或输出）个数可以不同，顺序也可以不同。

将输入和输出都看作【集合】的概念，**比较两个函数在输入相同的时候，输出值的集合是不是呈现包含关系。当呈现包含关系时，进一步查看输出变量的状态trace（即变量值的变化过程）是否一样。若一样，则二者等价**，记录到"组id_equal_pairs.csv"中，否则为不等价，记录到"组id_inequal_pairs.csv"中。

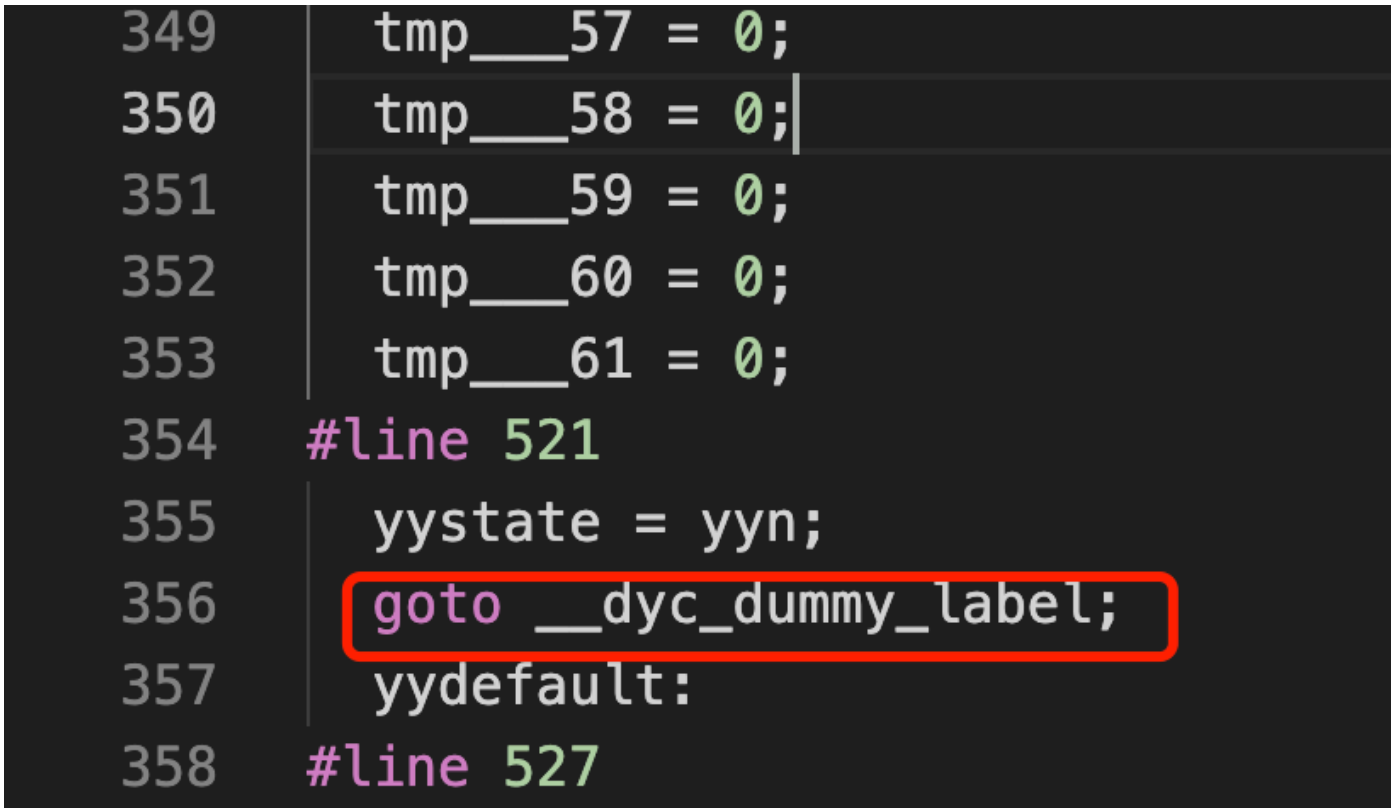
当输入个数不同时（例如一个为i，一个为j，且i<j），输入相同即表示两个函数拥有相同的i个输入，且输入个数为j的那个函数会有额外的j-i个随机生成的输入。输入值也是集合的概念，两个函数中相同的变量名不一定具有相同的输入值。

重复数据 注意，当两个文件完全相同时，忽略其中一个。即构成csv文件中每一行数据的两个文件不能完全相同。例如若某文件夹中原本包含5个文件，但其中有2个文件相同，则最后生成的数据量应为$C_4^2 = 6$。

一些建议

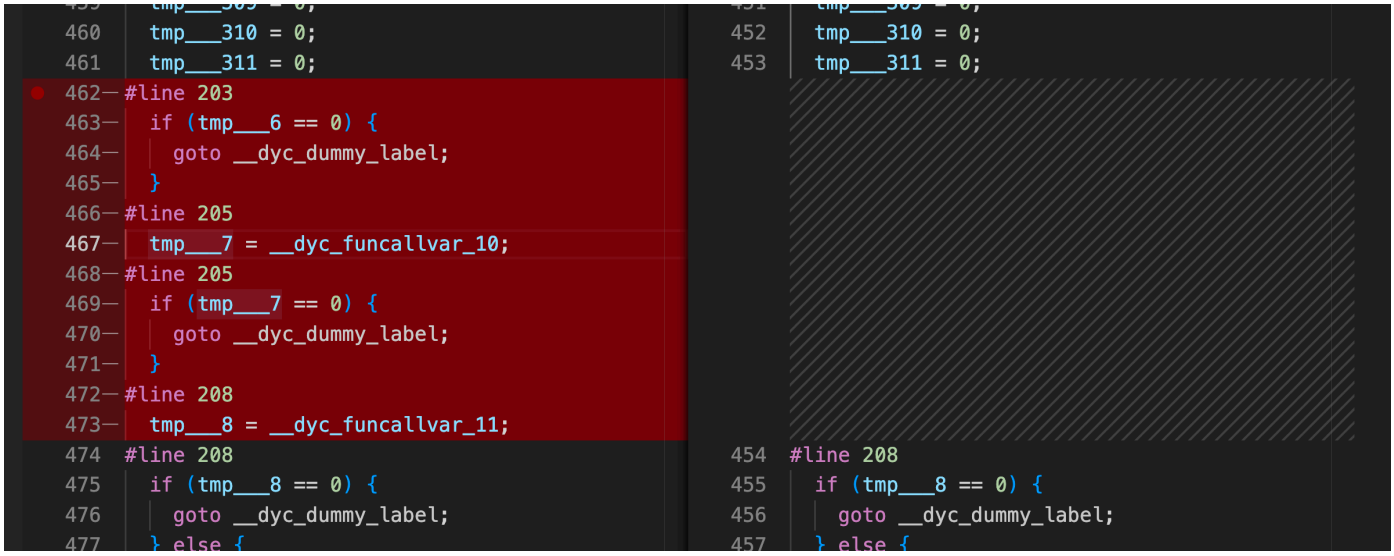
1.可以从**diff命令**开始，找出两个函数的主要差异，再基于这些差异判断等价与否。（或使用vscode的**compare**功能）

2.注意一些goto语句。例如下图中的第356行会导致从357行之后的逻辑不会被执行到，直接跳到程序输出。



一些示例

1.当差异代码涉及的变量都是新增的输入变量，但是它有goto语句，一旦进入分支就goto另一个段代码了。这种情况算作不等价。



2.存在一些差异代码是关于if条件语句的。注意看if语句的条件是否可能满足，从而判断程序执行时是否可能进入body部分，引起不等价情况的发生。

群聊二维码

若同学们还有任何不清楚的地方，欢迎进群。

该二维码7天内(5月25日前)有效，重新进入将更新