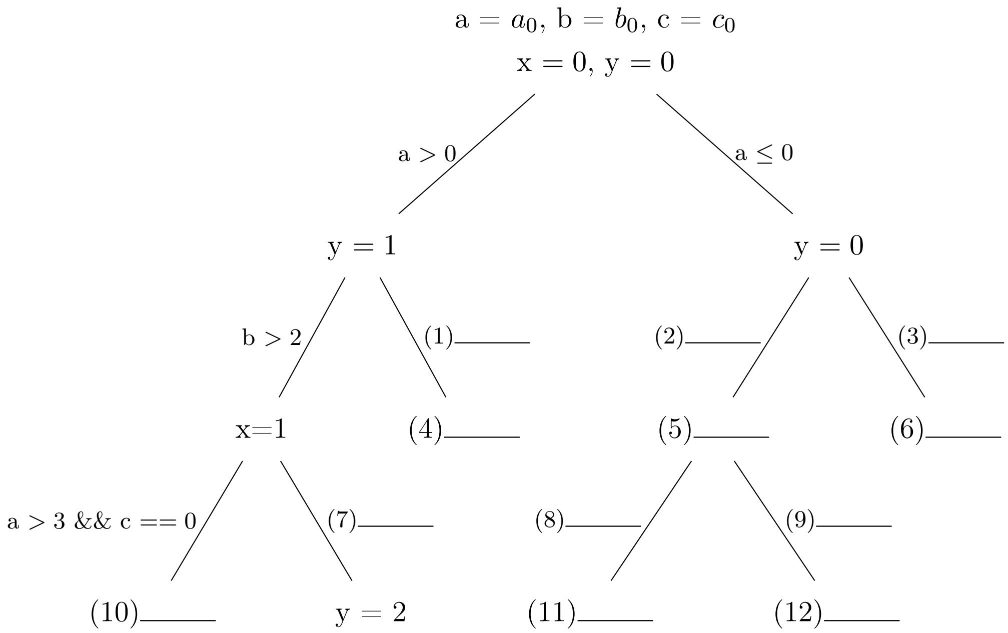
Question 1 Symbolic Execution

Consider following C program.

```c
#include <inttypes.h>
#include <assert.h>

void foo(int32_t a, int32_t b, int32_t c) {
    // program state
    int32_t x = 0;
    int32_t y = 0;

    if (a > 0) {
        y = 1;
    }

    // program state
    if (b > 2) {
        x = 1;
        // program state
        if (a > 3 && c == 0) {
            y = a;
        } else {
            y = x + 1;
        }
    }

    // program state
    assert(y == a);
}
```

(1) Below lists the Program Execution Tree (PET) of the program, where its nodes and edges show the altered variable values and conditions to check at each execution point. a_0, b_0 and c_0 are symbolic input to the function *foo*.

Complete the missing components in the graph. You should figure out the values of variables (concrete or symbolic), rather than filling in plain assignment relationships (*6 points*).



(2) Each path from the root to a leaf denotes an execution path of the function. Collect the path conditions along the path and write them down for all leaves in the PET. Use logical connectives (i.e., $\wedge$, $\neg$, etc.) rather than logic operators from C to clarify that they are logical formulas. Simplifying the formula is preferred, though not required. Keep in mind that the path conditions are playing with symbolic/concrete values rather than program variables (*6 points*).

(3) An SMT solver (of course and you as a human) can 'solve' the path conditions, meaning finding an assignment of the symbolic values present in the logical formula (namely, a_0, b_0, and c_0) to make the whole expression to be true. If such an assignment can not be found, then the formula is **UNSAT**. Write down one possible solution for each path condition you get in (2). You can either do it using your organic brain, or have a try with a real-world SMT solver Microsoft Z3 (*6 points*).

(4) Based on the above information, give the logical formula to make the assertion on line 25 successful. You are not required to simplify the formula (*3 points*).

(5) Suppose we start concolic executing the function with initial input a=4, b=3, c=0. Follow the example in the lecture slides to fill the below table for the first iteration, and give a possible set of input values for the next iteration to explore new paths. You just need to fill the changed states/path conditions only in each cell of the table (*4 points*).

Line Number	Concrete State	Symbolic State	Path condition
5	a=4,b=3,c=0	a=a_0, b=b_0, c=c_0	N/A
13	x=0, y=1	________	________
16	________	________	________
24	________	________	________

Question 2 Data flow analysis

Consider the following C code. Your task is to perform *Available Expression* analysis on this code snippet.

```
int y = x - 2;
int z = x + 3;

while (z > x - 2) {
  if (x > y) {
    y = x - 2;
  }
  x++;
}
y = x + 3;
```

(1) Draw the CFG of the given program (*7 points*).

(2) Find the *gen* and *kill* sets for each basic block. They should be sets of expressions and are in the form of $\{x\ op\ y, ...\}$ (*12 points*).

(3) Use the iterative algorithm from the lecture to get the final OUT[B] for each basic block. Again, it should be the set of expressions in the form of $\{x\ op\ y, ...\}$, denoting the available expressions after each basic block (*6 points*).

Question 3 Class Diagram (*25 points*)

Design a class diagram given the following descriptions, please involve as much information as possible.

1. To design a class diagram for an online shopping system, we need to identify the classes, attributes, and methods required to meet the system's requirements. The system must allow customers to view products, add them to their cart, and checkout, while administrators must be able to manage products and orders.
2. We can begin by identifying the classes needed for the system. These include Customer, Product, Cart, Order, and Administrator. Each of these classes will have their own attributes and methods.
3. The Customer class will have attributes such as id, name, email, and password, as well as methods to view products, add products to their cart, remove products from their cart, view their cart, and checkout and pay for their order.
4. The Product class will have attributes such as id, name, description, and price, as well as methods to create a new product and delete an existing product.
5. The Cart class will have an attribute to hold a list of products, and a customer_id attribute to associate the cart with a specific customer. The Cart class will also have methods to add a product to the cart and remove a product from the cart.
6. The Order class will have attributes to hold a list of products, a customer_id attribute to associate the order with a specific customer, a status attribute to indicate whether the order has been shipped or not, and a total_price attribute to hold the total cost of the order. The Order class will also have a method to mark an order as shipped.
7. Finally, the Administrator class will have attributes such as id, name, email, and password, as well as methods to add a new product, remove an existing product, and view all orders.

The relationships between the classes will include a 1-to-1 relationship between a Customer and a Cart, a 1-to-many relationship between a Customer and an Order, and a 1-to-many relationship between an Order and a Product. Additionally, there will be a 1-to-many relationship between an Administrator and a Product.

It's important to note that the system may also include additional components, such as a database or storage component to store and retrieve data, and may have additional security measures to protect customer and administrator data.

Question 4 Sequence Diagram (*25 points*)

Design a sequence diagram given the following descriptions, please involve as much information as possible.

1. The sequence diagram describes the process of purchasing a book from an online bookstore, as represented by a sequence diagram. The diagram outlines

the interactions between five participants: the Customer, the WebApp, the InvSys, the PayGateway, and the ShipSys. The process involves browsing for books, selecting a book, adding it to the cart, providing shipping and payment information, processing the payment, and updating inventory and shipping details.

2. The Customer starts by browsing the book selection on the WebApp, which requests the book list from the InvSys and displays it to the Customer. The Customer selects a book, and the WebApp checks its availability with the InvSys. The Customer adds the book to the cart, and the WebApp updates the cart accordingly. The Customer then proceeds to checkout, providing shipping and payment information to the WebApp.

3. The WebApp sends a payment request to the PayGateway, which sends back a payment confirmation. If the payment is successful, the WebApp updates the book inventory by sending a request to the InvSys, which sends back an inventory update confirmation. The WebApp creates a shipping order with the ShipSys, which sends back a shipping order confirmation. Finally, the WebApp sends an order confirmation and shipping details to the Customer.

4. If the payment fails, the WebApp sends a payment failure message to the Customer. The sequence diagram includes an alt condition to handle this potential issue. It also includes procedures to handle potential issues such as low stock levels and shipment delays.

Overall, the sequence diagram provides a clear representation of the book purchasing process, including the interactions between the various components involved and potential issues that may arise. It helps to identify potential issues and optimize the system for better performance and efficiency.