1.

2. - $f : X \to Y$. Если Y - открыто $\Rightarrow X -$ открыто
   - Требуется доказать: $\forall x \in X : \exists \varepsilon_1 : (x_0 - \varepsilon_1; x_0 + \varepsilon_1) \subseteq X$
   - Т.к Y открытое $\Rightarrow f(x_0) : \exists \varepsilon_2 : (f(x_0) - \varepsilon_2; f(x_0) + \varepsilon_2) \subseteq Y$
   - Т.к выполняется свойство лимита: $\varepsilon_2 : \exists \delta : |x - x_0| < \delta \Rightarrow |f(x) - f(x_0)| \Rightarrow \varepsilon_1 = \delta$

3.

4.

5. **a)**
   - $A \to B \to A$
   - ```
     def foo(a: A): B => A = {
       return (b: B): A => a
     }
     ```

   b)
   - $A \& B \to A \vee B$
   - ```
     def foo(ab: (A, B)): A | B = {
         return ab._1
       }
     ```

   c) NO
   - $(A \& (B \vee C)) \to ((A \& B) \vee (A \& C))$
   - ```
     def foo(value: (A, B | C)): (A, B) | (A, C) = {
       return value
     }
     ```
   - ```
     def foo(value: (A, B | C)): (A, B) | (A, C) = {
       value._2 match {
         case b: B => (value._1, b)
         case c: C => (value._1, c)
       }
     }
     ```

   d)
   - $(A \to C) \& (B \to C) \& (A \vee B) \to C$
   - ```
     def foo(
       value: ((A => C), (B => C), A | B)
     ) = {
       return value._3 match {
         case x: A => value._1(x)
         case x: B => value._2(x)
       }
     }
     ```

   e)
   - $(B \vee C \to A) \to ((B \to A) \& (C \to A))$
   - ```
     def foo(f: B | C => A): (B => A, C => A) = {
       return ((b: B) => f(b), (c: C) => f(c))
     }
     ```

   f) NO
   - $(A \to B) \to (\neg B \to \neg A)$
   - ```
     def foo1(f: A => B): (B => Nothing) => (A => Nothing) = {
       return (g: B => False) => (a: A) => System.exit()
     }
     ```
   - ```
     def foo2(f: A => B): (B => Nothing) => (A => Nothing) = func => (a =>
     func(f(a)))
     ```

g)

- $((A \to B) \to C) \to (A \to (B \to C))$
-
  ```
  def foo(f: (A => B) => C): A => (B => C) = {
    return (a: A) => (b: B) => {
      val func: A => B = (_: A) => b
      f(func)
    }
  }
  ```

h)

- $(A \to B) \& (A \to \neg B) \to \neg A$
- ```
  def popa(
    value: (f: A => B, g: A => B => Nothing)
  ): A => Nothing = (a: A) => System.exit()
  ```

```
  def foo(
  value: (A => B, A => B => Nothing)
): A => Nothing = (a: A) => {
  val b: B = value._1(a)       // Apply f to a to get b
  val notB: B => Nothing = value._2(a) // Get the function g(a): B => Nothing
  notB(b)                      // Apply notB to b to get Nothing
}
```

i)

- $(\neg A \lor B) \Rightarrow (A \Rightarrow B)$
- ```
  def foo(
    value: (A => Nothing) | B
  ): A => B = {
    value match {
      case b: B => (a: A) => b
      case g: (A => Nothing) => (a: A) => {
        return g(a)
      }
    }

  }
  ```
- $(A \Rightarrow B) \Rightarrow (\neg A \lor B)$

```
def foo(f: A => B): ((A -> Nothing) | B) = {
  return (a: A) => throw new Exception()
}
```

6.

7. **a)**

- $a \leq b \leq b + d$
- $c \leq d \leq b + d$
- $a + c \leq b + d$
- ====================
- $a \cdot c \leq a \leq b$
- $a \cdot c \leq c \leq d$
- $a \cdot c \leq b \cdot d$

b)

- $a \cdot (a + b) = a$

- $a + b = \sup\{a, b\} = x \Rightarrow a \le x \& b \le x$
- $a \cdot x = \inf\{a, x\} = d$
- $d \le a \& d \le x \Rightarrow d = a$
- ====================
- $a + (a \cdot b) = a$
- $a \cdot b = \inf\{a, b\} = x \Rightarrow x \le a \& x \le b$
- $a + x = \sup\{a, x\} = d \Rightarrow a \le d \& x \le d$
- $x = a$

b.1) Авторское решение от Димы Ч

- $a \cdot (a + b) = \inf\{a, \sup\{a + b\}\}$
- $x = \sup\{a, b\} \ge a$
- $a \cdot (a + b) = \inf\{a, x \ge a\}\} = a$
- ====================
- $a + a \cdot b = \sup\{a, \inf\{a + b\}\}$
- $x = \inf\{a, b\} \le a$
- $a \cdot (a + b) = \sup\{a, x \le a\}\} = a$

c)

- $(a \to b) = 1 \Rightarrow a \cdot 1 \le b \Rightarrow a \le b$
- т.к $\inf\{a, 1\} = a$
- $a \le b \Rightarrow a \cdot 1 \le b \Rightarrow a \to b = 1$

d)

e)

- $a \le b \to c \Rightarrow a \cdot b \le c$
- $(b \to c) = \sup\{x \mid x \cdot b \le c\} \Rightarrow$
- $(b \to c) \in \{x \mid x \cdot b \le c\} \Rightarrow$
- $(b \to c) \cdot b \le c$
- $a \le (b \to c)$ и $b \le b \Rightarrow b \cdot a \le b(b \to c)$ - case a)
- $b \cdot a \le b(b \to c) \le c$

f)

- $b \le a \to b \Rightarrow b \cdot a \le b \Rightarrow b \in \{x \mid a \cdot x \le b\} \Rightarrow b \le a \to b$
- =============
- $a \to (b \to a) = 1$
- По пункту c): $a \le (b \to a) \Rightarrow a \to (b \to a) = 1$
- но а это предыдущий под пункт

g)

-

h)

- т. д $a \le (b \to (a \cdot b))$
- $b \cdot x \le (a \cdot b) \le a \mid b$
- $b \cdot a \le a \mid b$ но $a \le x$ т.к. мы берем наибольший x
- ==============
- пункт c)

12.

- Пусть $\exists x \in [y]_R, [z]_R$, где $[y]_R, [z]_R \in A/_R$.

- Тогда $\forall y \in [y]_R \ yRx$ и $\forall z \in [z]_R \ xRz$.
- Тогда по транзитивности $\forall y \in [y]_R, z \in [z]_R, yRz$
- Тогда классы $[y]_R$и$[z]_R$ совпадают.$_{\text{ч.т.д.}}$

13. Дано: $a \approx b$, если aRb и bRa, где R транзитивно и рефлексивно

    a) Дано:$aRb, a \approx a', b \approx b'$

    - $a \approx a' \Rightarrow a'Ra$
    - $a'Ra \ \& \ aRb \Rightarrow a'Rb$ - по транзитивности
    - $b \approx b' \Rightarrow bRb'$
    - $a'Rb \ \& \ bRb' \Rightarrow a'Rb'$ - по транзитивности.$_{\text{ч.т.д.}}$

    b) Док-ть: $[a]_{\approx} R/_{\approx} [b]_{\approx} \Leftrightarrow aRb$, где $R/_{\approx}$ отношение нестрогого порядка $A/_{\approx}$.

    1. Покажем корректность определения:
       - Пусть имеем $a \in [a]_{\approx}, b \in [b]_{\approx}, aRb$.
       - $\forall a' \in [a]_{\approx} \quad a' \approx a$ и
         $\forall b' \in [b]_{\approx} \quad b' \approx b$
       - Тогда из пункта 13.a следует,
         что $\forall a' \in [a]_{\approx}, \forall b' \in [b]_{\approx} \quad a'Rb'$
       - Тогда определение не зависит,
         на каких элементах произведены классы,
         состоящие в отношении.
       - $[a]_{\approx} R/_{\approx} [b]_{\approx} \Leftrightarrow a'Rb' \quad \forall a' \in [a]_{\approx}, \forall b' \in [b]_{\approx}$
       - Это доказывает корректность определения отношения.
         Для 2 классов либо все элементы попарно
         состоят в отношении, либо не состоят.
    2. Докажем рефлексивность

       4

       $\forall a \quad [a]_{\approx} R/_{\approx} [a]_{\approx}$ - ?:
       - Из рефлексивности R следует aRa
       - Из 13.a следует $\forall a' \in [a]_{\approx} \quad a'Ra'$
       - Тогда по определению $\forall a \quad [a]_{\approx} R/_{\approx} [a]_{\approx}$
    3. Докажем транзитивность

    $\forall a, b, c \quad [a]_{\approx} R/_{\approx} [b]_{\approx}, [b]_{\approx} R/_{\approx} [c]_{\approx} \Rightarrow [a]_{\approx} R/_{\approx} [c]_{\approx}$ - ?:
    - Знаем, что $\forall a', b' \in [a]_{\approx}, [b]_{\approx} \quad a'Rb'$
    - Знаем, что $\forall b', c' \in [b]_{\approx}, [c]_{\approx} \quad b'Rc'$
    - Тогда по транзитивности отношения R
      $\forall b', c' \in [b]_{\approx}, [c]_{\approx} \quad b'Rc'$;
    - Тогда по определению
      $\forall a, b, c \quad [a]_{\approx} R/_{\approx} [b]_{\approx}, [b]_{\approx} R/_{\approx} [c]_{\approx} \Rightarrow [a]_{\approx} R/_{\approx} [c]_{\approx}$