15 (BDD) A BDD (Binary Decision Diagram) is a binary expression that has one of the following 3 forms: $\top$, $\bot$, **if** variable **then** BDD **else** BDD **fi** . For example,

if x **then if** a **then** $\top$ **else** $\bot$ **fi else if** y **then if** b **then** $\top$ **else** $\bot$ **fi else** $\bot$ **fi fi**

is a BDD. An OBDD (Ordered BDD) is a BDD with an ordering on the variables, and in each **if then else fi** , the variable in the **if**-part must come before any of the variables in its **then**- and **else**-parts ("before" means according to the ordering). For example, using alphabetic ordering for the variables, the previous example is not an OBDD, but

if a **then if** c **then** $\top$ **else** $\bot$ **fi else if** b **then if** c **then** $\top$ **else** $\bot$ **fi else** $\bot$ **fi fi**

is an OBDD. An LBDD (Labeled BDD) is a set of definitions of the following 3 forms:

label = $\top$
label = $\bot$
label = **if** variable **then** label **else** label **fi**

The labels are separate from the variables; each label used in a **then**-part or **else**-part must be defined by one of the definitions; exactly one label must be defined but unused. The following is an LBDD.

true = $\top$
false = $\bot$
alice = **if** b **then** true **else** false **fi**
bob = **if** a **then** alice **else** false **fi**

An LOBDD is an LBDD that becomes an OBDD when the labels are expanded. The ordering prevents any recursive use of the labels. The previous example is an LOBDD. An RBDD (Reduced BDD) is a BDD such that, in each **if then else fi** , the **then**- and **else**-parts differ. An RLOBDD is reduced, labeled, and ordered. The previous example is an RLOBDD.

(a) Express $\neg a$, $a \wedge b$, $a \vee b$, $a \Rightarrow b$, $a = b$, $a \neq b$, and **if** a **then** b **else** c **fi** as BDDs.

(b) How can you conjoin two OBDDs and get an OBDD?

(c) How can you determine if two RLOBDDs are equal?

(d) How can we represent an RLOBDD in order to determine efficiently if an assignment of values to variables satisfies it (solves it, gives it value $\top$)?

After trying the question, scroll down to the solution.

(a) Express $\neg a$, $a \wedge b$, $a \vee b$, $a \Rightarrow b$, $a = b$, $a \neq b$, and **if** a **then** b **else** c **fi** as BDDs.

§ $\neg a$ is **if** a **then** $\bot$ **else** $\top$ **fi**
$a \wedge b$ is **if** a **then if** b **then** $\top$ **else** $\bot$ **fi else** $\bot$ **fi**
$a \vee b$ is **if** a **then** $\top$ **else if** b **then** $\top$ **else** $\bot$ **fi fi**
$a \Rightarrow b$ is **if** a **then if** b **then** $\top$ **else** $\bot$ **fi else** $\top$ **fi**
$a = b$ is **if** a **then if** b **then** $\top$ **else** $\bot$ **fi else if** b **then** $\bot$ **else** $\top$ **fi fi**
$a \neq b$ is **if** a **then if** b **then** $\bot$ **else** $\top$ **fi else if** b **then** $\top$ **else** $\bot$ **fi fi**
if a **then** b **else** c **fi** is **if** a **then if** b **then** $\top$ **else** $\bot$ **fi else if** c **then** $\top$ **else** $\bot$ **fi fi**

(b) How can you conjoin two OBDDs and get an OBDD?

§ If one of them is $\top$ then the answer is the other one. If one of them is $\bot$ then the answer is $\bot$. If both of them are **if**s with the same variable as **if**-part, say **if** a **then** OBDD0 **else** OBDD1 **fi** and **if** a **then** OBDD2 **else** OBDD3 **fi** , then the answer is **if** a **then** OBDD0$\wedge$OBDD2 **else** OBDD1$\wedge$OBDD3 **fi** where the same procedure is used to find the **then**- and **else**-parts. If both of them are **if**s with different variables as **if**-parts, and the one with the earlier variable is **if** a **then** OBDD0 **else** OBDD1 **fi** , and the other one is OBDD2 , then the answer is **if** a **then** OBDD0$\wedge$OBDD2 **else** OBDD1$\wedge$OBDD2 **fi** where the same procedure is used to find the **then**- and **else**-parts.

(c) How can you determine if two RLOBDDs are equal?

§ Two RLOBDDs are equal if and only if they are identical except for a one-for-one substitution of labels and reordering of equations.

(d) How can we represent an RLOBDD in order to determine efficiently if an assignment of values to variables satisfies it (solves it, gives it value $\top$)?

no solution given