1 2000 Paper 6 Question 11

https://www.cl.cam.ac.uk/teaching/exams/pastpapers/y2000p6q11.pdf

(a) For each of the given pairs of terms, give a most general unifier or indicate why none exists. (Here x, y, z are variables while a, b are constant symbols.)

- $h(x, y, x)$ and $h(y, z, u)$

  $$\sigma = [x/y, x/z, x/u]$$

  only if u is a variable!

- $h(x, y, z)$ and $h(f(y), z, x)$

  There is no most-general unifier. Unify the first parameter using the most general unifier $\sigma = [f(y)/x]$. Leaving $h(f(y), y, z)$ and $h(f(y), z, f(y))$. Next unify the second parameter using the most general unifier $\sigma = [y/z]$. This leaves $h(f(y), y, y)$ and $h(f(y), y, f(y))$. Since y and $f(y)$ cannot be unified, clearly the third parameter cannot be unified.

- $h(x, y, b)$ and $h(a, x, y)$

  There is no most-general unifier. To unify the first parameters, we must use the unifier $[x/a]$. This leaves the terms $h(a, y, b)$ and $h(a, a, y)$. Now y must be unified with both a and b – this is not possible and therefore there is no most general unifier.

- $h(x, y, z)$ and $h(g(y, y), g(z, z), g(u, u))$

  $$\sigma = [g(g(g(u, u), g(u, u)), g(g(u, u), g(u, u)))/x, g(g(u, u), g(u, u))/y, g(u, u)/z]$$

A standard unification algorithm takes a pair of terms t_1 and t_2 and returns a substitution θ such that $t_1\theta = t_2\theta$. Show how this algorithm can be used to find the unifier of several ($n > 2$) terms $t_1, t_2, \ldots, t_n$: a substitution θ such that $t_1\theta = t_2\theta = \cdots = t_n\theta$. Indicate how the unifier is constructed from the unifiers of $n - 1$ pairs of terms. (Assume all required unifiers exist and ignore the question of whether the unifiers are most general).

Its possible to apply unifiers to unifiers. For example, if the unifier θ unifies t_1 and t_2, and the unifier ϕ unifies $t_1\theta$ and $t_3\theta$; then $\psi = \theta\phi$ is a unifier for t_1, t_2, t_3 ie $t_1\psi = t_2\psi = t_3\psi$.

We can use this knowledge to define a unifier for all the terms inductively:

$$t_1\theta_1 = t_2\theta_1$$
$$t_1\theta_1\theta_2 \ldots \theta_i = t_{i+1}\theta_1\theta_2 \ldots \theta_i$$
$$\sigma = \theta_1\theta_2 \ldots \theta_{n-1}$$

With A as the algorithm to find the most general unifier for two terms:

```
terms = [t1, t2, ..., tn]

σ = []

for i in range(2, n + 1):
    θ = A(t1, ~~terms[1]~~ terms[i]σ)
    σ = σθ
    t1 = t1θ
```

need to apply existing unifiers to terms[i] before applying A or you risk inconsistent commitments for the variables.

(b) Prove using resolution the formula

$$(\forall x[P(x) \leftrightarrow (Q(x) \land \neg Q(f(x)))]) \rightarrow \exists y \neg P(y)$$

The first stage is to negate the formula. Note this is an implication $A \to B$ so the negation is of the form $A \land \neg B$. I will create the clauses for A and B separately.

$$
\begin{aligned}
A =& \forall x[P(x) \leftrightarrow (Q(x) \land \neg Q(f(x)))]\\
=& \forall x.(P(x) \lor \neg(Q(x) \land \neg Q(f(x)))) \land (\neg P(x) \lor (Q(x) \land \neg Q(f(x))))\\
=& \forall x.(P(x) \lor \neg Q(x) \lor Q(f(x))) \land (\neg P(x) \lor Q(x)) \lor (\neg P(x) \lor \neg Q(f(x)))
\end{aligned}
$$

Skolemizing gives:

$$
\begin{aligned}
=& \boxed{(P(x) \lor \neg Q(x) \lor Q(f(x)))} \land (\neg P(x) \lor Q(x)) \lor (\neg P(x) \lor \neg Q(f(x)))\\
=& \{P(x), \neg Q(x)\}\ \{\neg P(x), Q(x)\}\ \{\neg P(x), \neg Q(f(x))\}
\end{aligned}
$$

7 B!

$$
\begin{aligned}
B =& \neg\exists y \neg P(y)\\
=& \forall y \neg\neg P(y)\\
=& \forall y P(y)
\end{aligned}
$$

Skolemizing gives:

$$
\begin{aligned}
=& P(y)\\
=& \{P(y)\}
\end{aligned}
$$

So the clauses for the whole expression are:

$$\{P(x), \neg Q(x)\}\ \{\neg P(x), Q(x)\}\ \{\neg P(x), \neg Q(f(x))\}\ \{P(y)\}$$

Variables are scoped only within their clause – for clarity I will rename to give unique names. Note this does not change the semantic meaning.

$$\{P(w), \neg Q(w)\}\ \{\neg P(x), Q(x)\}\ \{\neg P(y), \neg Q(f(y))\}\ \{P(z)\}$$

$$\{P(w), \neg Q(w)\}\ \{\neg P(x), Q(x)\}\ \{\neg P(y), \neg Q(f(y))\}\ \{P(z)\}$$

Use the unifiers $\sigma_1 = [x/z]$ and $\sigma_2 = [y/z]$ to create two new clauses

$$
\begin{aligned}
=& \{P(w), \neg Q(w)\}\ \{\neg P(x), Q(x)\}\ \{\neg P(y), \neg Q(f(y))\}\ \{P(z)\}\ \{P(x)\}\ \{P(y)\}\\
=& \{P(w), \neg Q(w)\}\ \{P(x)\}\ \{\neg P(x), Q(x)\}\ \{P(y)\}\ \{\neg P(y), \neg Q(f(y))\}\ \{P(z)\}
\end{aligned}
$$

Use resolution to create two new clauses

$$=\{P(w), \neg Q(w)\}\ \{P(x)\}\ \{\neg P(x), Q(x)\}\ \{P(y)\}\ \{\neg P(y), \neg Q(f(y))\}\ \{P(z)\}\ \{Q(u)\}\ \{\neg Q(f(v))\}$$

Use the unifier $\sigma_3 = [f(v)/u]$ to create a new clause

$$=\{P(w), \neg Q(w)\}\ \{P(x)\}\ \{\neg P(x), Q(x)\}\ \{P(y)\}\ \{\neg P(y), \neg Q(f(y))\}\ \{P(z)\}\ \{Q(u)\}\ \{Q(f(v))\}\ \{\neg Q(f(v))\}$$

Resolve to derive the empty clause $\square$

$$=\{P(w), \neg Q(w)\}\ \{P(x)\}\ \{\neg P(x), Q(x)\}\ \{P(y)\}\ \{\neg P(y), \neg Q(f(y))\}\ \{P(z)\}\ \{Q(u)\}\ \{Q(f(v))\}\ \{\neg Q(f(v))\}\ \square$$

Since the empty clause $\square$ has been derived, the clauses are inconsistent. Since the clauses were formed from the Skolemization of the negation of the original formula, the original formula must be valid

How should we lay this out? This feels verbose – and would becomes unreasonable in difficult questions. However, removal of clauses for clarity (even when explicitly stated) looks erroneous.

a tree is enough.
draw some lines!

2 2004 Paper 6 Question 9

https://www.cl.cam.ac.uk/teaching/exams/pastpapers/y2004p6q9.pdf

For each of the following statements, briefly justify whether it is true or false.

(a) Given any propositional logic formula ψ that is a tautology, converting ψ to CNF will result in $\mathbf{t}$.

This is false. If ϕ is converted to CNF then all clauses are tautological – of the form

Or (T V..)>

$P \vee \cdots \vee \neg P$ for some proposition P. If the formula is converted to CNF and all tautological clauses are deleted *then* the formula will be $\mathbf{t}$.

Consider for example the tautological formula below, in CNF:

$$(P \vee Q \vee \neg P) \wedge (Q \vee R \vee \neg P \vee \neg Q)$$

"Provided we canonicalise it, then "yes""

so you're saying that CNF might not be canonicalised?

(b) Executing the DPLL method on the clauses

$$\{P, Q, \neg S\}\ \{\neg P, Q, \neg R\}\ \{P\}\ \{\neg Q, R\}\ \{S, \neg Q\}$$

produces a result without needing any case split steps.

This is false.

The DPLL method is as follows:

- Delete all tautological clauses (clauses of the form $\{P, \ldots, \neg P\}$)
- Delete all Unit clauses (clauses of the form $\{P\}$) and propagate the truth value of the propositional formula in the unit clause.
- Propagate any pure literals – these are literals which only occur either negated or non-negated – it's possible to satisfy all clauses containing them without restricting any other formulae. So shoud be done.
- If none of the above are present then perform a case split – try case $P \mapsto 0$, if you fail then try case $P \mapsto 1$.
- Repeat

Applying this method to the clauses above requires a case split:

$$\{P, Q, \neg S\}\ \{\neg P, Q, \neg R\}\ \{P\}\ \{\neg Q, R\}\ \{S, \neg Q\}$$

Unit P

$$\{Q, \neg R\}\ \{\neg Q, R\}\ \{S, \neg Q\}$$

Pure S

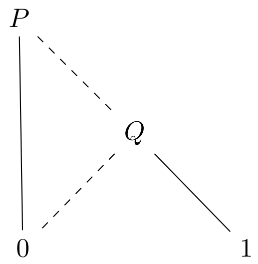
$$\{Q, \neg R\}\ \{\neg Q, R\}$$

Case split!

(c) The OBDD corresponding to the propositional logical formula $(P \vee Q) \wedge \neg P$ does not have any decision nodes for the propositional letter P.

The statement is false.

The OBDD for the propositional formula is given below:



Clearly there is a decision node for the propositional letter P.

(d) Skolemizing the first order logic formula $\exists x(\phi(x))$ results in a logically equivalent formula $\phi(a)$ (where a is a fresh constant).

This is false. Skolemization does not preserve meaning – it only preserves consistency. Therefore Skolemizing the formula $\exists x(\phi(x))$ does not preserve logical meaning.

(e) The Herbrand Universe that is generated from the clauses $\{P(a)\}$, $\{Q(x, b), \neg P(x)\}$ and $\{\neg Q(a, y)\}$ contains two elements.

This is true. The Herbrand Universe for these clauses is defined as follows:

$$\begin{aligned} H_0 &= \{a, b\} \\ H_{i+1} &= \{f_n(x_1, \ldots x_n) | x_1, \ldots x_n \in H_i\} \\ \mathcal{H} &= \bigcup_{i \in \mathbb{N}} H_i \end{aligned}$$

Note, however that there are no functions – therefore the sets $H_1, \ldots H_n$ are empty. So the Herbrand Universe $\mathcal{H}$ contains only the two elements $\{a, b\}$.

(f) The two terms $f(x, y, z)$ and $f(g(y, y), g(z, z), g(a, a))$ can be unified.

This is true! The most general unifier is:

$$\sigma = [g(g(g(a, a), g(a, a)), g(g(a, a), g(a, a)))/x, g(g(a, a), g(a, a))/y, g(a, a)/z]$$

(g) It is not possible to remove the clauses $\{P(x)\}$ and $\{\neg P(f(x))\}$ because the *occurs check* prevents the literals being unified.

This is false!

The domain of a variable is within its own clause – so the x in each clause is a *different* universally bound variable.

The clauses are logically equivalent to $\{P(x)\}$, $\{\neg P(f(y))\}$ – which can be unified to $\square$ using the unifier $\sigma = [f(y)/x]$.

(h) The clause $\{P(x, x), P(x, a)\}$ can be factored to give the new clause $\{P(x, a)\}$.

This is false. The most-general unifier for $\{P(x, x)\}$ and $\{P(x, a)\}$ is $\sigma = [a/x]$. This unifies the terms to $\{P(a, a)\}$ and $\{P(a, a)\}$. This clause can then be factored to form the new clause $\{P(a, a)\}$. This is **note** equivalent to $\{P(x, a)\}$.

(i) The empty clause can be derived from the clauses $\{P(x), P(a)\}$, $\{P(x), \neg P(a)\}$, $\{\neg P(b), Q\}$ and $\{\neg P(c), \neg Q\}$.

This is true.

- The clauses $\{\neg P(b), Q\}$ and $\{\neg P(c), \neg Q\}$ can be unified, creating the clause $\{\neg P(b), \neg P(c)\}$.
- The clauses $\{P(x), P(a)\}$ and $\{P(x), \neg P(a)\}$ can be unified forming the clause $\{P(x)\}$

- The clauses $\{P(x)\}$ and $\{\neg P(b), \neg P(c)\}$ can be unified with the unifier $[b/x]$ then resolved to form the new clause $\{\neg P(c)\}$
- The clauses $\{P(x)\}$ and $\{\neg P(c)\}$ can be unified with the unifier $[c/x]$ then resolved to form the empty clause $\square$

(j) Because in the modal logic S4 the equivalence $\square\square\phi \simeq \square\phi$ holds for every formula ϕ, it follows that $\diamond\diamond\phi \simeq \diamond\phi$

This is true. We can derive $\diamond\diamond\phi \simeq \diamond\phi$ using the equivalence $\diamond\phi \simeq \neg\square\neg\phi$

$$\begin{gathered}
\square\square\phi \simeq \square\phi \iff \\
\square\square\neg\phi \simeq \square\neg\phi \iff \\
\neg\square\square\neg\phi \simeq \neg\square\neg\phi \iff \\
\diamond\neg\square\neg\phi \simeq \diamond\neg\neg\phi \iff \\
\diamond\diamond\neg\neg\phi \simeq \diamond\neg\neg\phi \iff \\
\diamond\diamond\phi \simeq \diamond\phi
\end{gathered}$$

3 1998 Paper 5 Question 10

https://www.cl.cam.ac.uk/teaching/exams/pastpapers/y1998p5q10.pdf

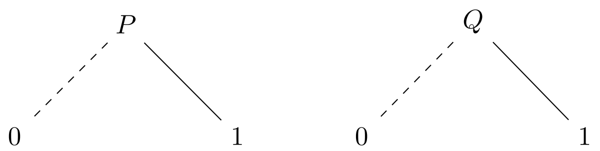
(a) Construct an ordered binary decision diagram (OBDD) for the formula

$$[(P \to Q) \land (\neg R \lor \neg Q)] \to \neg R$$

showing each step carefully. What does the OBDD tell us about whether the formula is (a) valid and (b) satisfiable and (c) inconsistent?

Note that although I show logically equivalent expressions separately, they point to the same BDD. I do this for clarity.

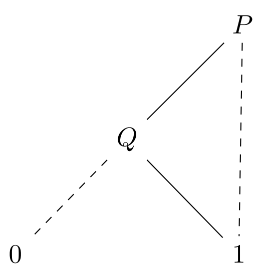
Firstly, consider the expression $P \to Q$



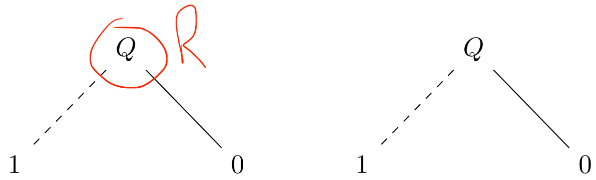
We can merge these using the $\to$ rule:

$$\begin{aligned}
IF(P,1,0) \to IF(Q,1,0) &= IF(P, 1 \to IF(Q,0,1), 0 \to IF(Q,0,1)) \\
&= IF(P, IF(Q,1,0), 1)
\end{aligned}$$

So $P \to Q$ is represented by the BDD below:



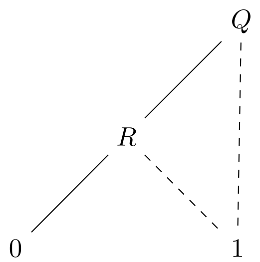
Next consider the expression $\neg R \vee \neg Q$. Starting with $\neg R$ and $\neg Q$:



We can merge these together using the $\vee$ rule:

$$\begin{aligned} IF(Q,0,1) \vee IF(R,0,1) &= IF(Q, 0 \vee IF(R,0,1), 1 \vee IF(R,0,1)) \\ &= IF(Q, IF(R,0,1), 1) \end{aligned}$$

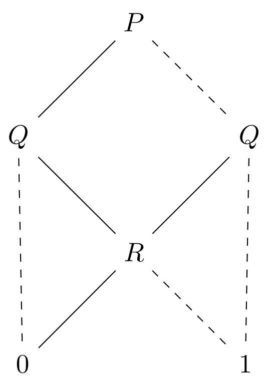
So the expression $\neg R \vee \neg Q$ forms the OBDD:



Next, merge these two expressions using the $\wedge$ rule:

$$\begin{aligned} & IF(P, IF(Q,1,0),1) \wedge IF(Q, IF(R,0,1),1) \\ =& IF(P, IF(Q,1,0) \wedge IF(Q, IF(R,0,1),1), 1 \wedge IF(Q, IF(R,0,1),1)) \\ =& IF(P, IF(Q, 1 \wedge IF(R,0,1), 0 \wedge 1), IF(Q, IF(R,0,1),1)) \\ =& IF(P, IF(Q, IF(R,0,1),0), IF(Q, IF(R,0,1),1)) \end{aligned}$$

So the LHS of the implication forms the OBDD:



We must now merge it with the RHS of the implication $\neg R$. This can be done using the $\rightarrow$ rule.

$$\begin{aligned} & IF(P, IF(Q, IF(R,0,1),0), IF(Q, IF(R,0,1),1)) \rightarrow IF(R,0,1) \\ =& IF(P, IF(Q, IF(R,0,1),0) \rightarrow IF(R,0,1), IF(Q, IF(R,0,1),1) \rightarrow IF(R,0,1)) \\ =& IF(P, IF(Q, IF(R,0,1) \rightarrow IF(R,0,1), 0 \rightarrow IF(R,0,1)), IF(Q, IF(R,0,1) \rightarrow IF(R,0,1), 1 \rightarrow IF(R,0,1))) \\ =& IF(P, IF(Q, IF(R, 0 \rightarrow 0, 1 \rightarrow 1), 1), IF(Q, IF(R, 0 \rightarrow 0, 1 \rightarrow 1), IF(R,0,1))) \\ =& IF(P, IF(Q, IF(R,1,1),1), IF(Q, IF(R,1,1), IF(R,0,1))) \end{aligned}$$

Note that the tests $IF(R, 1, 1)$ are redundant and can be replaced by 1

$=IF(P, IF(Q, 1, 1), IF(Q, 1, IF(R, 0, 1)))$

The test $IF(Q, 1, 1)$ is redundant and can be replaced by 1

$=IF(P, 1, IF(Q, 1, IF(R, 0, 1)))$

This is equivalent to the following BDD: