COMP30026 Models of Computation
Assignment 1

XuLin Yang, 904904

September 9, 2019

Challenge 1

$$
\begin{array}{lcl}
F & = & \forall x\ (\neg P(x,x)) \\
G & = & \forall x\ \forall y\ \forall z\ (P(x,y) \wedge P(y,z) \Rightarrow P(x,z)) \\
G' & = & \forall x\ \forall y\ \forall z\ (P(x,y) \wedge P(y,z) \Rightarrow \neg P(x,z)) \\
H & = & \forall x\ \forall y\ (P(x,y) \Rightarrow \neg P(y,x))
\end{array}
$$

a. Show that $F \vee G \vee G' \vee H$ is not valid.

note: $\mathbf{f}$ = false, $\mathbf{t}$ = true in the following answers.

domain	$P(a,b)$	formula			
		F	G	G'	H
{-1, 0, 1}	$P(a,b) = 2a + b \geq 0$	$x = 1$	$x = 0, y = 1, z = -1$	$x = 0, y = 0, z = 0$	$x = 0, y = 0$
		f	**f**	**f**	**f**

So we have one intepretation gives $F \vee G \vee G' \vee H = \mathbf{f}$.
It shows $F \vee G \vee G' \vee H$ is not valid.

b. Show that $F \wedge G' \wedge H$ is satisfiable.

domain	$P(a,b)$	formula			
		F	G'	H	$F \wedge G' \wedge H$
{0}	$P(a,b) = \mathbf{f}$	**t**	**t**	**t**	**t**

The listed intepretation shows $F \wedge G' \wedge H$ is satisfiable.

c. Show that $(F \wedge G) \Rightarrow H$ is valid.
Let $K = (F \wedge G) \Rightarrow H$,

Proof. K is valid

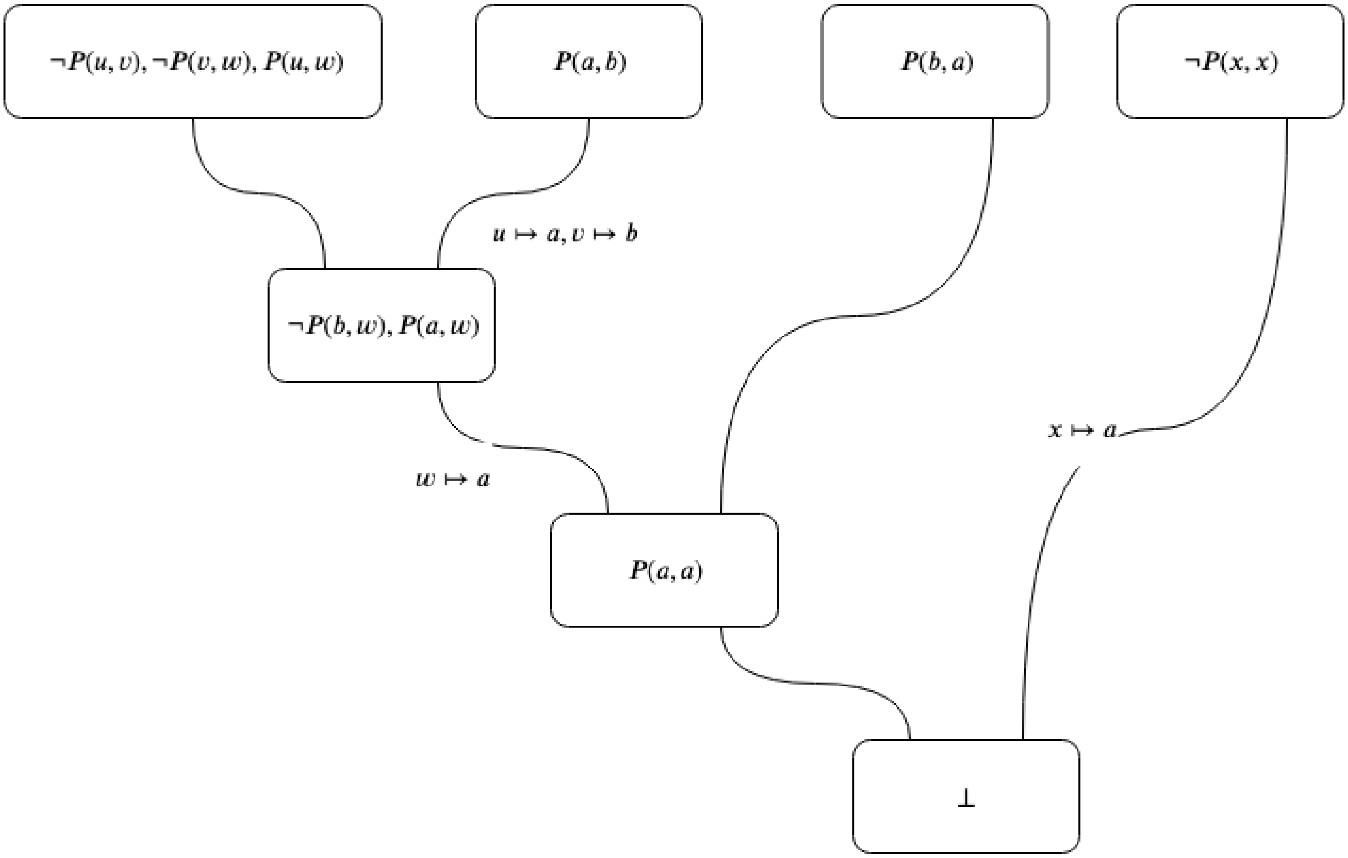
is equivalent to prove $\neg K$ is unsatisfiable.

$$
\begin{aligned}
\neg K &\equiv \neg[(F \wedge G) \Rightarrow H] \\
&\equiv \neg[\neg(F \wedge G) \vee H] && \text{(by implication)} \\
&\equiv F \wedge G \wedge \neg H && \text{(by deriving negation in)} \\
&\equiv [\forall x\ (\neg P(x,x))] \wedge \{\forall x\ \forall y\ \forall z\ [(\neg(P(x,y) \wedge P(y,z)) \vee P(x,z)]\} \wedge [\neg \forall x\ \forall y\ (\neg P(x,y) \vee \neg P(y,x))] \\
& && \text{(by replacing occurrence of } \Rightarrow\text{)} \\
&\equiv [\forall x\ (\neg P(x,x))] \wedge [\forall x\ \forall y\ \forall z\ (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z))] \wedge [\exists x\ \exists y\ (P(x,y) \wedge P(y,x))] \\
& && \text{(by deriving negation in)} \\
&\equiv [\forall x\ (\neg P(x,x))] \wedge [\forall u\ \forall v\ \forall w\ (\neg P(u,v) \vee \neg P(v,w) \vee P(u,w))] \wedge [\exists z\ \exists y\ (P(z,y) \wedge P(y,z))] \\
& && \text{(by standarizing apart)} \\
&\rightsquigarrow [\forall x\ (\neg P(x,x))] \wedge [\forall u\ \forall v\ \forall w\ (\neg P(u,v) \vee \neg P(v,w) \vee P(u,w))] \wedge P(a,b) \wedge P(b,a) \\
& && \text{(by skolemization)} \\
&\equiv \neg P(x,x) \wedge (\neg P(u,v) \vee \neg P(v,w) \vee P(u,w)) \wedge P(a,b) \wedge P(b,a) \\
& && \text{(by dropping universal quantifiers)}
\end{aligned}
$$

and written as a set of sets of literals:

$$
\left\{
\begin{array}{l}
\{\neg P(x,x)\}, \\
\{\neg P(u,v), \neg P(v,w), P(u,w)\}, \\
\{P(a,b)\}, \\
\{P(b,a)\}
\end{array}
\right\}
$$

and do the resolution:



As resolution shows: $\neg K$ is unsatisfiable.

$\therefore K$ is valid. $\square$

Challenge 2

- $F(x)$, which stands for "the force is with x"
- $J(x)$, which stands for "x is a Jedi master"
- $E(x, y)$, which stands for "x exceeds y"
- $P(x, y)$, which stands for "x is a pupil of y"
- $V(x)$, which stands for "x is venerated"

and use the constant 'a' to denote Yoda.

a. For each of the statements S_1, ..., S_4, express it *as a formula in first-order predicate logic* (not clausal form):

$S_1 = \forall x(J(x) \Rightarrow F(x))$

$S_2 = \exists y(J(y) \wedge E(a, y))$

$S_3 = \forall z\{\{J(z) \Rightarrow [F(z) \wedge (\forall u(P(u, z) \Rightarrow E(u, z)))]\} \Rightarrow V(z)\}$

$S_4 = \forall v\{[J(v) \Rightarrow (\neg\exists w(P(w, v)))] \Rightarrow V(v)\}$

b. Translate S_1-S_3 to clausal form.

$$
\begin{aligned}
S_1 &\equiv \forall x(J(x) \Rightarrow F(x)) \\
&\equiv \forall x(\neg J(x) \vee F(x)) && \text{(by replacing occurrence of } \Rightarrow\text{)} \\
&\equiv \neg J(x) \vee F(x) && \text{(by dropping universal quantifiers)}
\end{aligned}
$$

and written as a set of sets of literals:

$$
\{ \{\neg J(x), F(x)\} \quad \}
$$

$$
\begin{aligned}
S_2 &\equiv \exists y(J(y) \wedge E(a, y)) \\
&\rightsquigarrow J(b) \wedge E(a, b) && \text{(by skolemization)}
\end{aligned}
$$

and written as a set of sets of literals:

$$
\left\{ \begin{array}{l} \{J(b)\}, \\ \{E(a,b)\} \end{array} \right\}
$$

$$
\begin{aligned}
S_3 &\equiv \forall z\{\{J(z) \Rightarrow [F(z) \wedge (\forall u(P(u,z) \Rightarrow E(u,z)))]\} \Rightarrow V(z)\} \\
&\equiv \forall z\{\neg\{\neg J(z) \vee [F(z) \wedge (\forall u(\neg P(u,z) \vee E(u,z)))]\} \vee V(z)\} \\
& && \text{(by replacing occurrence of } \Rightarrow\text{)} \\
&\equiv \forall z\{\{J(z) \wedge \neg[F(z) \wedge (\forall u(\neg P(u,z) \vee E(u,z)))]\} \vee V(z)\} && \text{(by deriving negation in)} \\
&\equiv \forall z\{\{J(z) \wedge [\neg F(z) \vee \neg(\forall u(\neg P(u,z) \vee E(u,z)))]\} \vee V(z)\} && \text{(by deriving negation in)} \\
&\equiv \forall z\{\{J(z) \wedge [\neg F(z) \vee (\exists u(P(u,z) \wedge \neg E(u,z)))]\} \vee V(z)\} && \text{(by deriving negation in)} \\
&\rightsquigarrow \forall z\{\{J(z) \wedge [\neg F(z) \vee (P(f(z),z) \wedge \neg E(f(z),z))]\} \vee V(z)\} && \text{(by skolemization)} \\
&\equiv \{J(z) \wedge [\neg F(z) \vee (P(f(z),z) \wedge \neg E(f(z),z))]\} \vee V(z) && \text{(by dropping universal quantifiers)} \\
&\equiv (J(z) \vee V(z)) \wedge [(V(z) \vee \neg F(z)) \vee (P(f(z),z) \wedge \neg E(f(z),z))] && \text{(by distributivity)} \\
&\equiv (J(z) \vee V(z)) \wedge [(V(z) \vee \neg F(z) \vee P(f(z),z)) \wedge (V(z) \vee \neg F(z) \vee \neg E(f(z),z))] \\
& && \text{(by distributivity)}
\end{aligned}
$$

and written as a set of sets of literals:

$$
\left\{ \begin{array}{l} \{J(z), V(z)\}, \\ \{V(z), \neg F(z), P(f(z),z)\}, \\ \{V(z), \neg F(z), \neg E(f(z),z)\} \end{array} \right\}
$$

c. Translate *the negation of* S_4 to clausal form.

$$
\begin{aligned}
\neg S_4 &\equiv \neg\forall v\{[J(v) \Rightarrow (\neg\exists w(P(w,v)))] \Rightarrow V(v)\} \\
&\equiv \neg\forall v\{\neg[\neg J(v) \vee (\neg\exists w(P(w,v)))] \vee V(v)\} && \text{(by replacing occurrence of } \Rightarrow\text{)} \\
&\equiv \exists v\{[\neg J(v) \vee (\forall w(\neg P(w,v)))] \wedge \neg V(v)\} && \text{(by deriving negation in)} \\
&\rightsquigarrow \{[\neg J(c) \vee (\forall w(\neg P(w,c)))] \wedge \neg V(c)\} && \text{(by skolemization)} \\
&\equiv (\neg J(c) \vee \neg P(w,c)) \wedge \neg V(c) && \text{(by dropping universal quantifiers)}
\end{aligned}
$$

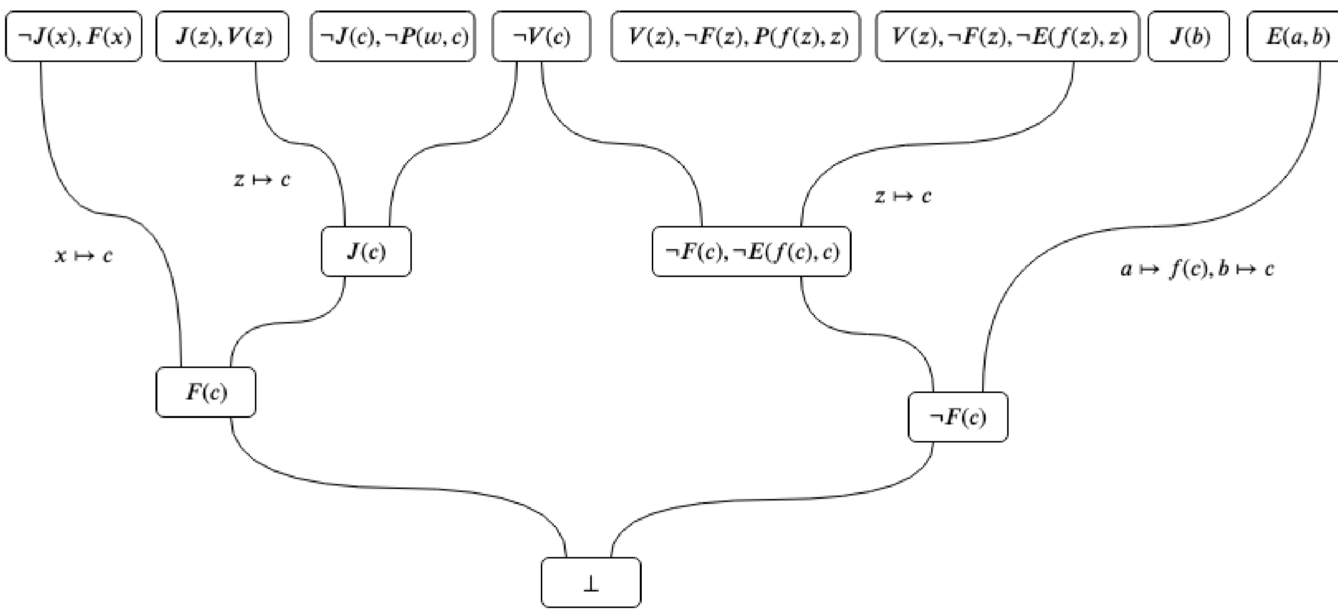
and written as a set of sets of literals:

$$
\left\{ \begin{array}{l} \{\neg J(c), \neg P(w,c)\}, \\ \{\neg V(c)\} \end{array} \right\}
$$

d. Give a proof by resolution to show that S_4 follows from the other statements.

Proof. $S_1 \wedge S_2 \wedge S_3 \models S_4$

is equivalent to prove: $S_1 \wedge S_2 \wedge S_3 \wedge \neg S_4$ is unsatisfiable.



As resolution shows: $S_1 \wedge S_2 \wedge S_3 \wedge \neg S_4$ is unsatisfiable

$\therefore S_1 \wedge S_2 \wedge S_3 \models S_4$ $\square$

Challenge 3

a. Consider the Boolean vector function $\mathbf{f}_a(b_1, b_2) = (\neg b_1, b_1 \oplus b_2)$. Show that this function is reversible.

Note: $0 =$ false, $1 =$ true in the following answers.

b_1	b_2	$\neg b_1$	$b_1 \oplus b_2$
0	0	1	0
0	1	1	1
1	0	0	1
1	1	0	0

As we can see every input has an unique output, $\mathbf{f}_a(b_1, b_2)$ is reversible.

b. Consider the Boolean vector function $\mathbf{f}_b(b_1, b_2, b_3) = (b_1 \oplus b_2, b_2 \oplus b_3, b_3 \oplus b_1)$. Show that this function is *not* reversible.

b_1	b_2	b_3	$b_1 \oplus b_2$	$b_2 \oplus b_3$	$b_3 \oplus b_1$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	0	0	0

As we can see $\mathbf{f}_b(0,0,0) = \mathbf{f}_b(1,1,1)$, $\mathbf{f}_b(b_1, b_2, b_3)$ is nor reversible.

c. Is the Boolean vector function $\mathbf{f}_c(b_1, b_2, b_3) = (b_1 \oplus b_3, \neg b_3, b_1 \oplus b_2)$ reversible? Justify your answer.

b_1	b_2	b_3	$b_1 \oplus b_3$	$\neg b_3$	$b_1 \oplus b_2$
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	1	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	1	0	1	1	0
1	1	1	0	0	0

As we can see every input has an unique output, $\mathbf{f}_c(b_1, b_2, b_3)$ is reversible.

d. Give a formula, in terms of n, for the fraction of distinct n-dimensional Boolean vector functions that are reversible.

note: # means the number of

#permutation without replacement of k input $= k!$

#permutation with replacement of k input $= k^k$

A n-dimensional boolean vector function has 2^n inputs.

To make a k input function invertible is just to choose k samples from input and each input picked once. This is equivalent to a permute without replacement of k input.

$$
\begin{aligned}
\therefore \quad & \text{\#invertible n-dimensional boolean vector function} \\
=& \text{\#invertible function with } 2^n \text{ inputs} \\
=& \text{\#permutation without replacement of } 2^n \text{ inputs} \\
=& (2^n)!
\end{aligned}
$$

To enumerate all possible ways to mapping k input function with k output is equivalent to find distinct choices of sampling from inputs. This is achieved by permuting with replacement.

$$
\begin{aligned}
\therefore \quad & \text{\#n-dimensional boolean vector function} \\
=& \text{\#function with } 2^n \text{ inputs mapped to } 2^n \text{ with replacement} \\
=& \text{\#permutation with replacement of } 2^n \text{ inputs} \\
=& (2^n)^{2^n}
\end{aligned}
$$

With all discussed above,

$$
\begin{aligned}
& \text{fraction of distinct n-dimensional invertible Boolean vector functions} \\
=& \frac{\text{\#n-dimensional invertible Boolean vector functions}}{\text{\#n-dimensional Boolean vector functions}} \\
=& \frac{(2^n)!}{(2^n)^{2^n}}
\end{aligned}
$$