CS 512, Spring 2014

Assignment 3

Shan Sikdar

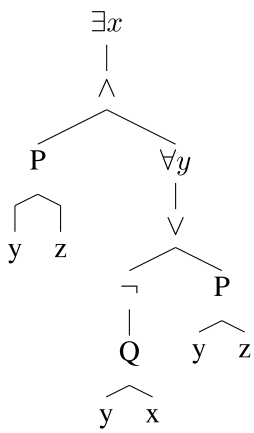
Due Monday Feburary 17th

1 Problem 1

Excercise 2.1.4 parts (a),(b),(c),(d),(e),(g). Skip part f which is ambiguous.
(a) Everybody has a mother.
$\forall x\ \ \exists y\ \ M(y, x)$
(b) Everbody has a father and a mother.
$\forall x\ \ \exists y\ \ \exists z\ \ M(y, x) \wedge F(z, x)$
(c) Whoever has a mother has a father.
$\forall x\ (\exists y M(y, x) \rightarrow \exists z\ \ F(z, x))$
(d) Ed is a grandfather.
$\exists x \exists y(M(y, x) \vee F(y, x)) \wedge F(Ed, y)$
(e) All fathers are parents.
$\forall x\ \ (\exists y F(x, y) \rightarrow \exists y(F(x, y) \vee M(x, y)))$
(g) No uncle is an aunt.
$\forall x\ (\exists y(F(x, y) \vee H(x, y) \vee B(x, y))\ \ \rightarrow\ \ (\ \neg(M(x, y) \vee S(x, y)))).$

2 Problem 2

Excercise 2.24 (a),(b),(c) and (d) only.
Let ϕ be $\exists x(P(y,z) \wedge (\forall y(\neg Q(y,x) \vee P(y,z))))$
(a)



(b) Identify all bound and free variable leaves in ϕ
All the leaf nodes with z are free. the leaf of y on the left side of the tree is free. The two y leafs on the right side of the tree are bound.

(c) Is there are variables in ϕ which has a bound and fee occurances?
Yes the variable y.
(d)
(i) Compute $\phi[w/x], \phi[w/y], \phi[f(x)/y]$ and $\phi[g(y,z)/z]$
$\phi[w/x]$ is ϕ since all occurances of x in ϕ are bound.
$\phi[w/y]$ is $\exists x(P(w,z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ (replace the sole free occurance of y with w)
$\phi[f(x)/y]$ is $\exists x(P(f(x),z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ (replace the sole free occurance of y with f(x))
$\phi[g(y,z)/z]$ is $\exists x(P(y,g(y,z)) \wedge \forall y(\neg Q(y,x) \vee P(y,g(y,z))))$ since we replace all free occurances of z with g(y,z)
(ii) Since there are no free occurances of x in ϕ you can say all of them are free for x in ϕ
(iii) Only w and g(y,z). You can't use f(x) is not free for y because there is an $\exists x$ quantifier above it.

3

Let ϕ be a propostional WFF in NNF and let $CNF(\phi) =< l, \Delta >$. Then ϕ is satisfiable iff $l \cup \Delta$ is satisfiable.
Proof: induction on the number of times the 4 cases of CNF() are used.
Base Case:
(i)If $\phi = p$ then $CNF(\phi) =< p, \emptyset >$. Then looking at $p \cup \emptyset$, it will be satisfiable if and only if p is satisfiable.
(ii) If $\phi = \neg p$. $CNF(\phi) =< \neg p, \emptyset >$ Then looking at $\neg p \cup \emptyset$, it will be satisfiable if and only if $\neg p$ is satisfiable.
(iii) If $\phi = p \wedge q$ for some propsitonal atoms p and q. From the application of the formula on the handout we get
For some propsitonal atom l: $CNF(\phi) = \langle l \cup \{\neg l \vee p, \neg l \vee q, \neg p \vee \neg q \vee l\}\rangle$.
Let everything inside the brace be denoted by Δ. Now look at $\Delta \cup \{l\}$. If ϕ is satsifiable, $\Delta \cup \{l\}$ will be satisfiable since the atoms must either be true or false. If either are true the $\vee$ statements will $\neg l \vee p$ or $\neg l \vee q$ will valuate to True. If both are false then $\neg p \vee \neg q \vee l$ will valuate to true.
Now look at $\Delta \cup \{l\}$ and show that implies ϕ is satisfiable. Assume that ϕ was not satisfiable, then delta would end up being not satisfiable based on the recursive definition o CNF and using similar logic as above. Similary the propostinal atom l would also not be satisifiable
(iv) $\phi = p \vee q$ this case is similar to (iii) except istead of worrying about both p and q being true because of the $\wedge$ we now only need one of them to get satisfibility.
Assume true for k, show for k+1:
Well by the recursive definiton of CNF, a wff that uses the number of cases k+1 times, will need to use it at least k times in the process. Then those k times will hold by induction hypothesis and reduces the k+1 case to the cases above. Therefore the k+1 case holds.

4

To make it look neater im using * instead of x for multiplication.

(a) write a first order WFF $\rho_1(x)$ which defines the set $\{0\}$

$\rho_1(x) := \forall y(x + y = y)$

(b) write a first order WFF $\rho_2(x)$ which defines the set $\{1\}$

$\rho_2(x) := \forall y(x * y = y)$

(c) first order WFF $\rho_3(x, y)$ which defines $\{< m, n > |n = m + 1 \in \mathbb{N}\}$

$\rho_3(x) := \exists c \ \ (\rho_2(c) \wedge (x = y + c))$

(d) first order WFF $\rho_3(x, y)$ which defines $\{< m, n > |m < n \in \mathbb{N}\}$

$\rho_4(x, y) := \exists z \ \ (\neg(\rho_1(z)) \wedge (x + z = y))$

5

(a) Formalize the PHP as a satisfibility problem. Specifically write a propostional WFF ϕ whoose satisfiability means the PHP holds.
Let P_{ij} indicate that pigeon i is in hole j, where$1 \leq i \leq n$ and $1 \leq j \leq n$.

If there is an empty hole, since the number of holes is less than number of pigeons, there must a hole that has more than one pigeon. If the number of holes and pigeons was 3 for example we could express an empty hole as:
$(\neg P_{11} \wedge \neg P_{21} \wedge \neg P_{31}) \vee (\neg P_{12} \wedge \neg P_{22} \wedge \neg P_{32}) \vee (\neg P_{13} \wedge \neg P_{23} \wedge \neg P_{33})$
and so on.
Generalizing to the n case we can write this as:
$\bigvee_j^{n-1}(\bigwedge_i^n \neg P_{ij})$ Where I use to $\bigvee$ and $\bigwedge$ as a shorthand for first ranging through all pigeons and then holes. Now if there are two birds in one hole, then the pidgeon principle must also be true. Expressed in terms of the example condition from before:
$(P_{11} \wedge P_{21}) \vee (P_{11} \wedge P_{31}) \vee (P_{21} \wedge P_{31})$... and so on.
More Generally we have: $\bigvee_j \bigvee_i (P_{i_1 j} \wedge P_{i_2 j})$ where $i_1 \neq i_2$ (First range over all holes and then for each hole through each possible pair of pigeon combinations.)

Combining these two facts togther we have:
$\bigvee_j^{n-1}(\bigwedge_i^n \neg P_{ij}) \vee \bigvee_j \bigvee_i (P_{i_1 j} \wedge P_{i_2 j})$
Using the fact that $\phi \to \psi \equiv \neg\phi \vee \psi$:
$\bigwedge(\bigvee P_{ij}) \to \bigvee_j \bigvee_i (P_{i_1 j} \wedge P_{i_2 j})$

Choosing the left to ρ_1 and the right expression to be ρ_2 we have:
$\rho := \rho_1 \to \rho_2$
And so we have a propositonal wff for the pigeon hole principle.

(b) Implmentation. I wrote two different implmentation cases in Isabelle, one for the n = 3 and one for n =4. In my experience the n = 3 case pretty much ran almost instantly. For the n = 4 case it turned out to run a little slower. To handle the n = 10 case I would try and optomize the wff so that it can backtrack less.

6

(1) 2,
Using the DPLL transitonal rules and procedure, there are three different Models which make ρ_1 true:
$M_1 = \{p, q, r, s\}$
$M_2 = \{p, \neg q, r, s\}$
$M_3 = \{p, q, \neg r, s\}$
The last two will make ρ_1 true so they will terminate without any more iterations. Only the one in the hint takes 2 iterations.
So the maximum number is 2.
(2) Take the original ρ_0 and add wffs that contain contradictions. This will force the SMT to do 3 more iterations. So let
$\psi_0 = x + y \geq 0 \wedge (x = z \rightarrow z + y = -1) \wedge z > 3t \wedge (g = 1 \wedge g > 1) \wedge (h = 2 \wedge h > 2) \wedge (j = 3 \wedge j > 3)$