J Bolton Set 4

Logic for Applications

Section 2, ch4: 1, 2, 4
Ch5: 5, 6

Section 2, chapter 4

1.

First structure
The domain A is *N*
P (x) = x >= 0
f (x,y) = x+y
∀x. P(f(x,c)) = True
Second structure would be the same, except that the domain is *Z*, and clearly the formula is not true because we can get negative numbers as sums with integers.

2.

Even though it goes against definition 2.1, I will assume that 'p' and 'q' are meant to be predicate symbols and 'f' a function symbol, otherwise the formula doesn't make sense to me.
Domain is *N*
p(x) = x < x+1
q(x) = x > 0
f(x) = x+1

4.

Domain is Q
$0^A = 2$
$1^A = ½$
Let < be the predicate less than
Let + be multiplication (we will call it f instead to avoid confusion)
Thus:

f(x, 1) < x	this will always be true (I.e. x*1/2 < x)
f(x, x) < x	not always true since f(0, 0) > 0 (I.e. 2*2>2)

Section 2, chapter 5

5.

a(c,c,c).
a(X,Y,Z) :- s(A,X), s(B,Y), s(C,Z), a(A,B,C) .

6.

The program clause consisting of a fact is always satisfiable because we can give an assignment of T to the only literal. (5.2i)
Rules can be shown as an implication: $l_1 \wedge l_2 \wedge ... l_n \rightarrow l_g$. Thus, the assignment that makes $l_1 \wedge l_2 \wedge ... l_n$ true will satisfy a rule clause, and so it is satisfiable.

7.

[Note: not really sure what I'm doing here.]
A fact is always satisfiable with an assignment of true. A rule can always be satisfied by providing either an assignment of true or false to the body of the rule. If the body is false, then the goal does not matter and the rule is trivially true. If the body is true, then the goal must also have an assignment of true. “The assignment that makes every propositional letter true satisfies every program clause” (p. 68)