Solutions to Assignment #1

1. Consider the set S of strings s of propositional symbols for which $N(s) = (\#\text{ of })\text{'s} - \#\text{ of } \to)$ in s is ≥ 0. We show that S is closed under the three conditions in the definition of formula:

① $N(p) = 0$ where p is any prop. var. so $p \in S$.

② $N(\neg\varphi) = N(\varphi) \geq 0$ for any $\varphi \in S$.

③ $N((\varphi \square \psi)) = N(\varphi) + N(\psi) + 1 > 0$
if $\square$ is either $\vee$ or $\wedge$

$$N((\varphi \to \psi)) = N(\varphi) + N(\psi) \geq 0.$$

In either case, if $\varphi, \psi \in S$ then $\varphi \square \psi \in S$ for any binary connective.

So we conclude $N(\varphi) \geq 0$ for any formula φ.

2. $\varphi_1 := (\neg((p \wedge \neg q) \vee (q \to r)) \to s)$

$\varphi_2 := \neg((p \wedge \neg q) \vee (q \to r))$ $\quad$ s

$\varphi_3 := ((p \wedge \neg q) \vee (q \to r))$

$(p \wedge \neg q)$ $\quad$ $(q \to r)$

p $\quad$ $\neg q$ $\quad$ q $\quad$ r

q

So the subformulas are $p, q, r, \neg q, (p \wedge \neg q)$, s, $(q \to r)$, φ_3, φ_2 and φ_1.

3. We need to show the set of accepted strings satisfies:

1. every prop. var is accepted.
2. if s is accepted then $\neg s$ is accepted and
3. if s, t are accepted then $(s \square t)$ is accepted for any binary connective $\square$.

By induction on formulas, if we do these 3 things then every formula is accepted.

1 is just (b) of the algorithm.
2 is (d).

So we need to verify 3. We need to prove one small lemma that was alluded to in class:

Lemma: If φ is an ~~formula~~ accepted string then the bracket count of any binary connective is > 0.

Pf/ We do this by induction on ~~formulas~~ the length of the string ~~as well~~.
For propositional variables ~~and truths~~ there are no binary connectives. For $\neg\varphi$, the bracket count for a binary connective doesn't change.
If we are looking at $(\varphi \square \psi)$ then the bracket count of a binary connective in φ or ψ increases by 1 so remains > 0. For $\square$, the bracket count is $1 > 0$. $\blacksquare$

Now back to 3: Suppose we have $(s \square t)$

for accepted strings s, t and binary connective $\square$.

By our Lemma, the bracket count of any binary connective in s or t is > 0. So the bracket count of that binary connective in $(s \square t)$ is > 1. Since s is accepted, the bracket count of s is allowable so the bracket count of $\square$ is 1. Putting this together then when the algorithm sees $(s \square t)$, the first binary connective with bracket count 1 is $\square$ and s and t are accepted so it accepts $(s \square t)$.