Recall: Last time we proved Problem 1.2, that every formula of $\mathcal{L}_P$ has the same number of left parentheses as it has right parentheses, by induction on the length of the formula (as a string of symbols).

Alternate proof of ~~#~~1.2 using induction on the number of connectives in the formula α:

Let n be the number of connective symbols (including repetitions) in the formula α of $\mathcal{L}_P$.

Base Step: ($n=0$) If α has no connective symbols, then α is an atomic formula and so has no parenthes (#left parentheses $= 0 =$ #right parentheses).

✓

<u>Inductive Hypothesis:</u> $(n=k)$ If α has $\leq k$ connective symbols, then α has as many left parentheses as right parentheses.

<u>Inductive Step:</u> $(n=k+1)$ Suppose α has $k+1$ connectives.

Then α has at least one connective so it had to have been built from shorter formulas using negation or implication.

case (a): α is $(\neg\beta)$ for some formula β with $k=(k+1)-1$ connectives.

By the inductive hypothesis, β has just as many left as right parentheses, say p. Then α has $p+1$ left and $p+1$ right parentheses, so it has an equal number of each.

case (b): α is $(\gamma \to \delta)$ for some formulas γ & δ with $(k+1)-1 = k$ connectives between them.

Since each of γ & δ has $\leq k$ connectives, the I.H. tells us that each has the same number of left as right parentheses, say p for γ and q for δ. Then α has $p+q+1$ left parentheses and

$p+q+1$ right parentheses,

so it has the same number of each.

$\therefore$ Every formula has as many left as right parentheses. //

Some conventions used to make formulas more easily read & written by humans:

1) We use the (informal) connectives and ($\wedge$), or ($\vee$), and if and only if ($\Leftrightarrow$) as shorthand for certain "official" constructions using $\to$ & $\neg$:

$$(\alpha \wedge \beta) \text{ is short for } (\neg(\alpha \to (\neg\beta)))$$
$$(\alpha \vee \beta) \text{ — '' — } ((\neg\alpha) \to \beta)$$
$$(\alpha \Leftrightarrow \beta) \text{ — '' — } ((\alpha \to \beta) \wedge (\beta \to \alpha))$$

This will give $\wedge, \vee, \Leftrightarrow$ the appropriate truth tables (in Ch-2).

2) We usually drop the outside-most parentheses in a formula and write $\alpha \to \beta$ for $(\alpha \to \beta)$ and $\neg\alpha$ for $(\neg\alpha)$.

3) We will use precedence to drop ~~and~~ more parentheses on occasion, with the precedence order being $\neg, \wedge, \vee, \to, \Leftrightarrow$
so <u>eg</u> $\neg\alpha \to \beta$ means $((\neg\alpha) \to \beta)$, <u>not</u> $(\neg(\alpha \to \beta))$.

4) Group repetitions of $\rightarrow, \vee, \wedge, \leftrightarrow$ to the right when parentheses are missing

<u>eg</u> $\alpha \rightarrow \beta \rightarrow \gamma$

means $(\alpha \rightarrow (\beta \rightarrow \gamma))$

<u>not</u>, $((\alpha \rightarrow \beta) \rightarrow \gamma)$.

<u>Def'n</u>: Informally, the collection of subformulas of a formula α (in its official version), is all the formulas that occur in constructing α from atomic formulas, including α itself.

<u>eg</u> $(A_0 \rightarrow (A_1 \rightarrow (\neg A_0)))$ has the following subformulas:

$(A_0 \rightarrow (A_1 \rightarrow (\neg A_0)))$, A_0, $(A_1 \rightarrow (\neg A_0))$, A_1, $(\neg A_0)$, ~~A_0~~

Exercise: How can you make this informal definition precise?

For the Computer Science folks:

Backus-Naur style grammar for $\mathcal{L}_P$:

```
<atomic-formulas> ::= "A_0" | "A_1" | "A_2" | ...
<formula> ::= <atomic-formula> | "(" "¬" <formula> ")"
            | "(" <formula> "->" <formula> ")"
```