Motivation for Factoring

Suppose you wish to prove some formula like:

\[ \text{Goal:} \quad (\alpha_1 \land \ldots \land \beta \land \ldots \land \alpha_k) \rightarrow \gamma \]

and you wish to rewrite \((\text{rev } (\text{rev } a))\) which occurs in \(\beta\). You wish to use the theorem:

\[ (\text{true-listp } x) \rightarrow (\text{rev } (\text{rev } x)) = x \]

What can you assume about \(a\)?
Factor $Goal$ into $Goal'$ so that $\beta$ is in the conclusion of $Goal'$. Then you can assume the hypothesis of $Goal'$.

By defining “factoring” we make it possible to prove $Goal$ by repeatedly rewriting anywhere in the formula without explicitly having to rearrange the formula as part of the proof.
Summaries of the Rules of Inference

To prove $\psi$ using:

- **Tautology**: find a tautology and instantiate it to get $\psi$. 

• Rewrite:
  - put $\psi$ into the form $\psi_h \rightarrow (\ldots \alpha' \ldots)$ where $\alpha'$ is the term you want to change,
  - pick some theorem in the form $\phi_h \rightarrow (\alpha = \beta)$ or $\phi_h \rightarrow (\alpha \leftrightarrow \beta),$
  - match $\alpha$ with $\alpha'$ so that $\alpha/\sigma = \alpha'$,
  - relieve the hypotheses by proving $\psi_h \rightarrow \phi_h/\sigma,$
  - make sure the equivalence ("=" or "\leftrightarrow") is ok,
  - replace $\alpha'$ with $\beta/\sigma.$
• Hypothesis:
  - pick a term $\alpha'$ in the conclusion of $\psi$ to rewrite with some hypothesis $(\alpha = \beta)$ or $(\alpha \leftrightarrow \beta)$,
  - make sure the equivalence (“=” or “$\leftrightarrow$”) is ok,
  - replace $\alpha$ with $\beta$. 

• **Cases:**
  
  – pick an exhaustive set of terms \((\phi'_1 \lor \ldots \lor \phi'_k)\)
  
  – for each, prove \(\phi'_i \rightarrow \psi\).
• Constant Expansion (Variant 1):
  – pick a list constant \( '(\alpha \ldots) \) in \( \psi \)
  – replace it by \( \text{cons } '(\alpha \ldots) \)
• Constant Expansion (Variant 2):
  – pick a non-0 natural $n$ in $\psi$
  – replace it by $(+ 1 \ n')$, where $n'$ is $n - 1$. 
• Computation:
  – pick a function call \((f \ c_1 \ldots \ c_n)\) in \(\psi\), where the \(c_i\) are all constants
  – run \(f\) on those inputs to get \(v\)
  – replace \((f \ c_1 \ldots \ c_n)\) by \('v')