iterate and power

Johan G. F. Belinfante
2002 May 22

One readily discovers the connection between \texttt{iterate} and \texttt{power} by looking at a few examples:

\begin{verbatim}
Map[image[image[power[x], singleton[#]], y] &, NestList[succ, 0, 3]]
{y, image[x, y], image[x, image[x, y]], image[x, image[x, image[x, y]]]}

Map[image[iterate[x, y], singleton[#]] &, NestList[succ, 0, 3]]
{y, image[x, y], image[x, image[x, y]], image[x, image[x, image[x, y]]]}
\end{verbatim}

The obvious conjecture is that for any natural number \( n \) we have:

\begin{verbatim}
image[image[power[x], singleton[n]], y] == image[iterate[x, y], singleton[n]];
\end{verbatim}

We will actually show that something more general is true; we can replace \texttt{singleton[n]} by any class. This more general conjecture can be written as

\begin{verbatim}
image[image[power[x], z], y] == image[iterate[x, y], z];
\end{verbatim}

If this is the case, we can use \texttt{abstract} to eliminate the variable \( z \) altogether.

\begin{verbatim}
Map[abstract[z, #] &, image[image[power[x], z], y] == image[iterate[x, y], z]]
composite[SECOND, id[cart[y, V]], power[x]] == iterate[x, y]
\end{verbatim}

One can go further, and abstract \( y \) on the left side.

\begin{verbatim}
abstract[y, composite[SECOND, id[cart[y, V]], power[x]]]
inverse[rotate[composite[inverse[power[x]]], SWAP]]
\end{verbatim}

Thus, we could write \texttt{iterate[x,y]} as:
This may help to explain, for example, why \texttt{iterate}[x, y] preserves unions with respect to its second argument, something that had already been proved independently some time ago.

\begin{verbatim}
iterate[x, union[y, z]] == union[iterate[x, y], iterate[x, z]]
\end{verbatim}

\texttt{True}

\textbf{proof}

The proof of the conjecture is surprisingly short. We just use the uniqueness theorem for \texttt{iterate}. This is the whole thing:

\begin{verbatim}
SubstTest[implies, and[equal[image[w, singleton[0]], v],
equal[composite[w, id[omega]], iterate[u, v]],
{u -> y, v -> x, w -> composite[SECOND, id[cart[x, V]], power[y]]}]

equal[composite[SECOND, id[cart[x, V]], power[y]], iterate[y, x]] == True
\end{verbatim}

When this rule was discovered yesterday, I was perplexed as to how it should be oriented. One could of course use this to eliminate the concept of \texttt{iterate} in favor of \texttt{power}. But this would complicate the statement of many properties of \texttt{iterate}, and so we choose to keep both concepts.

\begin{verbatim}
composite[SECOND, id[cart[x, V]], power[y]] := iterate[y, x]
\end{verbatim}

\textbf{power[cross[x, y]]}

As a corollary, we obtain a formula relating \texttt{power[cross[Id,x]]} to \texttt{power[x]}:

\begin{verbatim}
composite[SECOND, id[cart[Id, V]], power[cross[Id, x]]]

power[x]
\end{verbatim}

We will later try to improve on this to get a formula for \texttt{power[cross[x,y]]}.

\textbf{back to the beginning}

The original conjecture can now be proved:

\begin{verbatim}
ImageComp[composite[SECOND, id[cart[y, V]]], power[x], z] // Reverse
image[image[power[x], z], y] == image[iterate[x, y], z]
\end{verbatim}

It is not entirely clear how to orient this equation. The following is tentative.

\begin{verbatim}
image[image[power[x, z, y], y] := image[iterate[x, y], z]
\end{verbatim}
In particular:

\[
\text{SubstTest}\left[\text{image, image[\text{power[x]}, z], y, z \rightarrow V}\right]
\]

\[
\text{image[range[\text{power[x]}], y]} \equiv \text{range[iterate[x, y]]}
\]

Since the quantity \text{range[\text{power[x]}]} can be rewritten in terms of transitive closure, we hold off adding this rule for now.