power sets with 2 elements

Johan G. F. Belinfante
2005 January 9

summary

The only power sets with exactly 2 elements are the power sets of singletons.

a needed lemma and a better one

Among other things, the following fact will be needed:

Rather than adding this special rule, a more general rule could be added, justified by the following fact:

This can be made into a conditional rewrite rule.

Now the special rule is no longer needed:
power sets of singletons have exactly two elements

This is half of the theorem: if $x$ is singleton, then $P[x]$ has two elements.

To establish the converse, the idea is to capitalize on this rewrite rule that is already available:

The following is a start toward proving the converse:

covering lemma

The number of elements in a finite set is one more than that of the set obtained by removing one of its elements. Applying this idea to the case of a set with two elements yields:
The finiteness hypothesis can be removed, since any set with 2 elements is finite.

The following lemma rewrites the condition that $P[x]$ has exactly one element besides the empty set.

If a power set $P[x]$ has exactly 2 elements, then $\text{diff}[P[x], \text{singleton}[0]] = \text{singleton}[x]$. It follows that if $P[x]$ has two elements, then $x$ is a singleton.
In[19]:= Map[not, SubstTest[and, implies[p1, p2],
     implies[p1, not[p3]], implies[p2, or[p3, p4]], not[implies[p1, p4]],
     {p1 -> member[P[x], image[PAIRSET, Di]],
      p2 -> equal[intersection[complement[singleton[0]], P[x]], singleton[x]],
      p3 -> equal[0, x], p4 -> member[x, range[SINGLETON]]}]]

Out[19]= or[member[x, range[SINGLETON]], not[member[P[x], image[PAIRSET, Di]]]] == True

In[20]:= (\% /. x_ -> \_). Equal \rightarrow SetDelayed

This is a logical equivalence, which is made into a rewrite rule.

In[21]:= equiv[member[P[x], image[PAIRSET, Di]], member[x, range[SINGLETON]]]

Out[21]= True

In[22]:= member[P[x_], image[PAIRSET, Di]] := member[x, range[SINGLETON]]

Corollary

In[23]:= SubstTest[member, P[x], union[u, v],
     {u -> range[SINGLETON], v -> image[PAIRSET, Di]}]

Out[23]= member[P[x], range[PAIRSET]] == or[equal[0, x], member[x, range[SINGLETON]]]

In[24]:= member[P[x_], range[PAIRSET]] := or[equal[0, x], member[x, range[SINGLETON]]]

variable free formulations

In this section, variable-free formulations are derived.

In[25]:= image[inverse[POWER], image[PAIRSET, Di]] // Normality

Out[25]= image[inverse[POWER], image[PAIRSET, Di]] == range[SINGLETON]

In[26]:= image[inverse[POWER], image[PAIRSET, Di]] := range[SINGLETON]

In[27]:= ImageComp[POWER, inverse[POWER], image[PAIRSET, Di]]

Out[27]= intersection[image[PAIRSET, Di], range[POWER]] ==
            image[POWER, range[SINGLETON]]

In[28]:= intersection[image[PAIRSET, Di], range[POWER]] :=
            image[POWER, range[SINGLETON]]

In[29]:= SubstTest[image, inverse[POWER], union[x, y],
     {x -> range[SINGLETON], y -> image[PAIRSET, Di]}]

Out[29]= image[inverse[POWER], range[PAIRSET]] == union[range[SINGLETON], singleton[0]]
The idea in this section is to obtain a variable-free version of this rewrite rule for power sets of singletons.

Using `reify` did not work directly, but the following result was found by carefully examining the `reify` formulas:

This formula characterizes the power sets of singletons as those sets that can be written as the union of `singleton[0]` and the singleton of a singleton.