existence of prime divisors

Johan G. F. Belinfante
2005 February 22

```
In[1]:= SetDirectory["i:"]; << goedel66.22a; << tools.m

        :Package Title: goedel66.22a          2005 February 22 at 3:40 p.m.

        It is now:  2005 Feb 22 at 16:49

        Loading Simplification Rules

        TOOLS.M                      Revised 2005 February 22

        weightlimit = 40
```

summary

Every natural number other than 1 has a prime divisor. The proof uses the second form of induction.

strategy

The class of natural numbers that have prime divisors is:

```
In[2]:= class[y, exists[x, and[member[x, PRIMES], member[pair[x, y], DIV]]]]

Out[2]= image[DIV, PRIMES]
```

The idea is to apply induction to the following set, for which a temporary abbreviation will be made to save some writing.

```
In[3]:= temp = union[image[DIV, PRIMES], set[set[0]]]

Out[3]= union[image[DIV, PRIMES], set[set[0]]]
```

some lemmas

It is immediate that 1 has no prime divisors.

```
In[4]:= member[set[0], image[DIV, PRIMES]] // AssertTest

Out[4]= member[set[0], image[DIV, PRIMES]] == False

In[5]:= % /. Equal → SetDelayed
```

Lemma.

```
In[6]:= equal[intersection[omega, PRIMES], PRIMES]

Out[6]= True

In[7]:= intersection[omega, PRIMES] := PRIMES
```

Lemma.

```
In[8]:= Map[not, SubstTest[implies, member[x, y],
          not[equal[0, y]], {x → succ[set[0]], y → PRIMES}]]

Out[8]= equal[0, PRIMES] == False

In[9]:= equal[0, PRIMES] := False
```

Lemma.

```
In[10]:= member[0, image[DIV, PRIMES]] // AssertTest

Out[10]= member[0, image[DIV, PRIMES]] == True

In[11]:= % /. Equal → SetDelayed
```

Lemma.

```
In[12]:= SubstTest[implies, subclass[u, v],
          subclass[image[u, w], image[v, w]], {u → id[omega], v → DIV, w → PRIMES}]

Out[12]= subclass[PRIMES, image[DIV, PRIMES]] == True

In[13]:= % /. Equal → SetDelayed
```

Temporary lemma.

```
In[14]:= equiv[or[not[member[x, omega]],
            subclass[image[inverse[DIV], set[x]], set[x, set[0]]]],
          subclass[image[inverse[DIV], set[x]], set[x, set[0]]]]

Out[14]= True

In[15]:= or[not[member[x_, omega]],
            subclass[image[inverse[DIV], set[x_]], set[x_, set[0]]]] :=
          subclass[image[inverse[DIV], set[x]], set[x, set[0]]]
```

Theorem.

```
In[16]:= Map[or[not[member[x, omega]], #] &, SubstTest[and, subclass[u, v],
           subclass[v, u], {u -> set[x, set[0]], v -> image[inverse[DIV], set[x]]}]]

Out[16]= subclass[image[inverse[DIV], set[x]], set[x, set[0]]] ==
          or[equal[x, set[0]], member[x, PRIMES], not[member[x, omega]]]

In[17]:= subclass[image[inverse[DIV], set[x_]], set[x_, set[0]]] :=
          or[equal[x, set[0]], member[x, PRIMES], not[member[x, omega]]]
```

---

derivation

Lemma.

```
In[18]:= Map[not, SubstTest[and, implies[and[q3, q1], or[r1, r2]],
           implies[and[r1, p1, q2], r3], implies[and[q3, r3], p4],
           not[implies[and[p1, q1, q2, q3, not[r2]], p4]],
           {p1 -> subclass[x, union[image[DIV, PRIMES], set[set[0]]]],
            q1 → not[equal[x, y]], q2 → not[equal[y, set[0]]],
            q3 → member[pair[y, x], DIV], r1 → member[y, x], r2 → equal[0, x],
            r3 → member[y, image[DIV, PRIMES]], p4 → member[x, image[DIV, PRIMES]]}]]

Out[18]= or[equal[0, x], equal[x, y], equal[y, set[0]],
           member[x, image[DIV, PRIMES]], not[member[pair[y, x], DIV]],
           not[subclass[x, union[image[DIV, PRIMES], set[set[0]]]]]] == True

In[19]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

The variable **y** is eliminated:

```
In[20]:= Map[equal[V, #] &,
           SubstTest[class, y, or[equal[0, x], equal[x, y], equal[y, set[0]],
             member[x, v], not[member[pair[y, x], DIV]], not[subclass[x, u]]],
            {u -> union[image[DIV, PRIMES], set[set[0]]],
             v → image[DIV, PRIMES]}]] // Reverse

Out[20]= or[equal[x, set[0]], member[x, PRIMES],
           member[x, image[DIV, PRIMES]], not[member[x, omega]],
           not[subclass[x, union[image[DIV, PRIMES], set[set[0]]]]]] == True

In[21]:= (% /. x → x_) /. Equal → SetDelayed
```

This can be cleaned up a bit:

```
In[22]:= Map[not, SubstTest[and,
           implies[and[p1, p2], or[p3, p4, p5, p6]], implies[p3, p6], implies[p5, p6],
           not[implies[and[p1, p2], or[p4, p6]]], {p1 → member[x, omega],
            p2 → subclass[x, temp], p3 → equal[0, x], p4 → equal[x, set[0]],
            p5 → member[x, PRIMES], p6 → member[x, image[DIV, PRIMES]]}]]

Out[22]= or[equal[x, set[0]], member[x, image[DIV, PRIMES]], not[member[x, omega]],
           not[subclass[x, union[image[DIV, PRIMES], set[set[0]]]]]] == True

In[23]:= (% /. x → x_) /. Equal → SetDelayed
```

Restatement.

```
In[24]:= implies[and[member[x, omega], subclass[x, temp]], member[x, temp]]

Out[24]= True
```

The variable **x** is now eliminated.

```
In[25]:= Map[equal[V, #] &, SubstTest[class, x, implies[
             and[member[x, omega], subclass[x, t]], member[x, t]], t → temp]] // Reverse

Out[25]= subclass[intersection[omega, P[union[image[DIV, PRIMES], set[set[0]]]]],
           union[image[DIV, PRIMES], set[set[0]]]] == True

In[26]:= % /. Equal → SetDelayed
```

The second form of induction is now applied.

```
In[27]:= SubstTest[implies, subclass[intersection[omega, P[t]], t],
          subclass[omega, t], t → temp]

Out[27]= subclass[omega, union[image[DIV, PRIMES], set[set[0]]]] == True

In[28]:= % /. Equal → SetDelayed
```

This can be strengthened to an equation, yielding the main theorem. It says that any number other than 1 has a prime divisor.

```
In[29]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u → dif[omega, set[set[0]]], v -> image[DIV, PRIMES]}]

Out[29]= True == equal[image[DIV, PRIMES], intersection[omega, complement[set[set[0]]]]]

In[30]:= image[DIV, PRIMES] := intersection[omega, complement[set[set[0]]]]
```