Euclid's theorem: there are infinitely many primes

Johan G. F. Belinfante
2005 April 21

```
In[1]:= SetDirectory["i:"]; << goedel68.21b; << tools.m

        :Package Title: goedel68.21b          2005 April 21 at 4:00 p.m.

        It is now:  2005 Apr 21 at 22:40

        Loading Simplification Rules

        TOOLS.M                       Revised 2005 April 16

        weightlimit = 40
```

summary

It is shown, by an argument rather similar to that used by Euclid, that there are infinitely many primes. The basic idea is to reason about the sum class **U[PRIMES]**. If it were a number, then all primes would be divisors of its factorial. Adding one to that factorial yields a number which of course also has a prime divisor, and this divisor must therefore divide two consecutive numbers. This is a contradiction because two consecutive numbers can only have **1** as a common divisor, and **1** is not a prime. From this one deduces that **U[PRIMES] = omega**, and from that it follows that there are infinitely many primes.

if U[PRIMES] is a number, then all primes divide its factorial

An application of the factorial divisibility theorem.

```
In[2]:= Map[not, SubstTest[and, implies[and[p1, p4], p5], implies[p1, p6],
          implies[and[p1, p2], p7], implies[and[p3, p6, p7], p8],
          implies[and[p3, p5, p7, p8], p9], not[implies[and[p1, p2, p3, p4], p9]],
          {p1 → member[y, x], p2 -> subclass[x, omega], p3 → member[U[x], omega],
           p4 → not[member[0, x]], p5 → not[equal[0, y]], p6 → subclass[y, U[x]],
           p7 → member[y, omega], p8 → not[member[U[x], y]],
           p9 -> member[pair[y, APPLY[FACTORIAL, U[x]]], DIV]}]]

Out[2]= or[member[0, x], member[pair[y, APPLY[FACTORIAL, U[x]]], DIV],
          not[member[y, x]], not[member[U[x], omega]], not[subclass[x, omega]]] == True

In[3]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Eliminating the variable **y** yields:

```
In[4]:= Map[equal[V, #] &, SubstTest[class, y, or[member[0, x], member[pair[y, u], v],
            not[member[y, x]], not[member[U[x], omega]], not[subclass[x, omega]]],
           {u -> APPLY[FACTORIAL, U[x]], v → DIV}]] // Reverse

Out[4]= or[member[0, x], not[member[U[x], omega]], not[subclass[x, omega]],
          subclass[x, image[inverse[DIV], set[APPLY[FACTORIAL, U[x]]]]]] == True

In[5]:= (% /. x → x_) /. Equal → SetDelayed
```

Corollary. If there were a largest prime, then all primes would be divisors of its factorial.

```
In[6]:= SubstTest[or, member[0, x], not[member[U[x], omega]], not[subclass[x, omega]],
          subclass[x, image[inverse[DIV], set[APPLY[FACTORIAL, U[x]]]]], x → PRIMES]

Out[6]= or[not[member[U[PRIMES], omega]], subclass[PRIMES,
           image[inverse[DIV], set[APPLY[FACTORIAL, U[PRIMES]]]]]] == True

In[7]:= or[not[member[U[PRIMES], omega]], subclass[PRIMES,
           image[inverse[DIV], set[APPLY[FACTORIAL, U[PRIMES]]]]]] := True
```

---

a lemma about consecutive multiples

The only number that can divide two consecutive numbers is **1 = set[0]**.

```
In[8]:= subclass[intersection[image[inverse[DIV], set[x]],
           image[inverse[DIV], set[succ[x]]]], set[set[0]]] // AssertTest

Out[8]= subclass[intersection[image[inverse[DIV], set[x]],
           image[inverse[DIV], set[succ[x]]]], set[set[0]]] == True
```

```
In[9]:= subclass[intersection[image[inverse[DIV], set[x_]],
          image[inverse[DIV], set[succ[x_]]], set[set[0]]] := True
```

Lemma.

```
In[10]:= equal[intersection[PRIMES, complement[set[set[0]]]], PRIMES]

Out[10]= True

In[11]:= intersection[PRIMES, complement[set[set[0]]]] := PRIMES
```

Lemma.

```
In[12]:= subclass[intersection[PRIMES, image[inverse[DIV], set[x]]], set[set[0]]] //
          AssertTest

Out[12]= subclass[intersection[PRIMES, image[inverse[DIV], set[x]]], set[set[0]]] ==
          or[equal[x, set[0]], not[member[x, omega]]]

In[13]:= subclass[intersection[PRIMES, image[inverse[DIV], set[x_]]], set[set[0]]] :=
          or[equal[x, set[0]], not[member[x, omega]]]
```

Theorem.

```
In[14]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]],
          subclass[u, w], {u → PRIMES, v -> image[inverse[DIV], set[x]],
           w → union[set[set[0]], complement[image[inverse[DIV], set[succ[x]]]]]}]

Out[14]= or[equal[0, x], not[member[x, omega]],
           not[subclass[PRIMES, image[inverse[DIV], set[x]]]]] == True

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

Euclid's theorem

The theorem of the preceding section produces a conclusion contrary to the one deduced in the section before it.

```
In[16]:= SubstTest[implies,
          and[member[x, omega], subclass[PRIMES, image[inverse[DIV], set[x]]]],
          equal[0, x], x -> APPLY[FACTORIAL, U[PRIMES]]]

Out[16]= or[not[member[U[PRIMES], omega]], not[subclass[PRIMES,
             image[inverse[DIV], set[APPLY[FACTORIAL, U[PRIMES]]]]]]] == True

In[17]:= % /. Equal → SetDelayed
```

Theorem. The sum class of **PRIMES** is not a natural number.

```
In[18]:= Map[not, SubstTest[and, implies[p1, p2],
           implies[p1, not[p2]], {p1 -> member[U[PRIMES], omega],
            p2 -> subclass[PRIMES, image[inverse[DIV],
               set[APPLY[FACTORIAL, U[PRIMES]]]]]}]] // Reverse

Out[18]= member[U[PRIMES], omega] == False

In[19]:= member[U[PRIMES], omega] := False
```

Corollary. The sum class of **PRIMES** is the set of all natural numbers.

```
In[20]:= SubstTest[implies, subclass[x, omega],
          or[equal[U[x], omega], member[U[x], omega], equal[0, x]], x → PRIMES]

Out[20]= equal[omega, U[PRIMES]] == True

In[21]:= U[PRIMES] := omega
```

Lemma.

```
In[22]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]],
          subclass[u, w], {u → PRIMES, v → omega, w → FINITE}]

Out[22]= subclass[PRIMES, FINITE] == True

In[23]:= subclass[PRIMES, FINITE] := True
```

Corollary. There are infinitely many primes.

```
In[24]:= SubstTest[member, U[x], FINITE, x → PRIMES] // Reverse

Out[24]= member[PRIMES, FINITE] == False

In[25]:= member[PRIMES, FINITE] := False
```

---

a rewrite rule for ub[DIV, PRIMES]

In this section it is shown that the only common multiple of the set of all primes is zero, using an argument very similar to that used in the proof of Euclid's theorem. The following lemma is quite general:

```
In[26]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
          {u → ub[x, y], v → image[x, y], w → range[x]}]

Out[26]= or[equal[0, y], subclass[ub[x, y], range[x]]] == True
```

```
In[27]:= or[equal[0, y_], subclass[ub[x_, y_], range[x_]]] := True
```

In particular, this lemma has the following application:

```
In[28]:= SubstTest[implies, not[empty[y]],
          subclass[ub[x, y], range[x]], {x → DIV, y → PRIMES}]

Out[28]= subclass[ub[DIV, PRIMES], omega] == True

In[29]:= % /. Equal → SetDelayed
```

Lemma.

```
In[30]:= equal[intersection[omega, ub[DIV, PRIMES]], ub[DIV, PRIMES]]

Out[30]= True

In[31]:= intersection[omega, ub[DIV, PRIMES]] := ub[DIV, PRIMES]
```

The main work in the present derivation consists of eliminating the variable **x** from the theorem proved in the section on consecutive multiples.

```
In[32]:= Map[equal[V, #] &, SubstTest[class, x, or[equal[0, x],
             not[member[x, w]], not[subclass[u, image[inverse[v], set[x]]]]],
            {u → PRIMES, v → DIV, w → omega}]] // Reverse

Out[32]= subclass[ub[DIV, PRIMES], set[0]] == True

In[33]:= % /. Equal → SetDelayed
```

Since the reverse inclusion also holds, this can be strengthened to an equation and made into a rewrite rule.

```
In[34]:= SubstTest[and, subclass[u, v],
          subclass[v, u], {u -> ub[DIV, PRIMES], v -> set[0]}]

Out[34]= True == equal[set[0], ub[DIV, PRIMES]]

In[35]:= ub[DIV, PRIMES] := set[0]
```