next prime

Johan G. F. Belinfante
2005 August 8

```
In[1]:= SetDirectory["i:"]; << goedel72.08a; << tools.m

        :Package Title: goedel72.08a       2005 August 8 at 11:25 a.m.

        It is now:  2005 Aug 8 at 14:6

        Loading Simplification Rules

        TOOLS.M                      Revised 2005 August 2

        weightlimit = 40
```

summary

The smallest prime that contains a given finite set of numbers x is **hull[PRIMES, x]**. In particular, if x is a prime, then the next prime is **hull[PRIMES, succ[x]]**. Because the **GOEDEL** program already knows the first few primes, this can be illustrated explicitly.

HULL[PRIMES]

Lemma.

```
In[2]:= SubstTest[implies, subclass[u, v],
         subclass[image[u, w], image[v, w]], {u -> composite[inverse[E], id[omega]],
          v -> composite[id[omega], inverse[S], id[omega]], w → PRIMES}]

Out[2]= subclass[omega, image[inverse[S], PRIMES]] == True

In[3]:= % /. Equal → SetDelayed
```

Lemma.

```
In[4]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]],
         subclass[u, w], {u → PRIMES, v → omega, w → image[inverse[S], omega]}]

Out[4]= subclass[PRIMES, image[inverse[S], omega]] == True

In[5]:= % /. Equal → SetDelayed
```

Theorem:

```
In[6]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u → image[inverse[S], PRIMES], v → image[inverse[S], omega]}]

Out[6]= True == equal[image[inverse[S], omega], image[inverse[S], PRIMES]]

In[7]:= image[inverse[S], PRIMES] := image[inverse[S], omega]
```

The domain of the function **HULL[PRIMES]** is the set of all finite sets of natural numbers, and the range is the set of all primes.

```
In[8]:= domain[HULL[PRIMES]] == intersection[FINITE, P[omega]]

Out[8]= True

In[9]:= range[HULL[PRIMES]]

Out[9]= PRIMES
```

Lemma.

```
In[10]:= SubstTest[member, x, intersection[y, z], {y → FINITE, z → P[omega]}]

Out[10]= member[x, image[inverse[S], omega]] ==
          and[member[x, FINITE], subclass[x, omega]]

In[11]:= member[x_, image[inverse[S], omega]] :=
          and[member[x, FINITE], subclass[x, omega]]
```

Corollary.

```
In[12]:= SubstTest[member, hull[w, x], range[HULL[w]], w → PRIMES]

Out[12]= member[hull[PRIMES, x], PRIMES] == and[member[x, FINITE], subclass[x, omega]]

In[13]:= member[hull[PRIMES, x_], PRIMES] := and[member[x, FINITE], subclass[x, omega]]
```

Further corollaries: **hull[PRIMES, x]** cannot be **0** or **1**.

```
In[14]:= Map[not, SubstTest[implies, or[member[y, PRIMES], equal[y, V]],
            not[equal[y, 0]], y → hull[PRIMES, x]]]

Out[14]= equal[0, hull[PRIMES, x]] == False

In[15]:= equal[0, hull[PRIMES, x_]] := False
```

```
In[16]:= Map[not, SubstTest[implies, or[member[y, PRIMES], equal[y, V]],
           not[equal[y, set[0]]], y → hull[PRIMES, x]]]

Out[16]= equal[hull[PRIMES, x], set[0]] == False

In[17]:= equal[hull[PRIMES, x_], set[0]] := False
```

---

corollary needed later for the use of nat wrappers

Lemma.

```
In[18]:= SubstTest[implies, and[member[u, v], subclass[v, w]],
          member[u, w], {u -> hull[PRIMES, x], v → PRIMES, w -> omega}]

Out[18]= or[member[hull[PRIMES, x], omega],
           not[member[x, FINITE]], not[subclass[x, omega]]] == True

In[19]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[20]:= SubstTest[implies, member[y, omega], member[y, V], y → hull[PRIMES, x]]

Out[20]= or[and[member[x, FINITE], subclass[x, omega]],
           not[member[hull[PRIMES, x], omega]]] == True

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

Theorem.

```
In[22]:= equiv[member[hull[PRIMES, x], omega],
          and[member[x, FINITE], subclass[x, omega]]]

Out[22]= True

In[23]:= member[hull[PRIMES, x_], omega] := and[member[x, FINITE], subclass[x, omega]]
```

---

the first two primes

The least prime is 2:

```
In[24]:= hull[PRIMES, 0]

Out[24]= succ[set[0]]
```

Every nonempty set of primes has a least member:

```
In[25]:= Map[not, SubstTest[and, implies[p1, p3], implies[and[p2, p3], p4],
           implies[and[p1, p4], p5], not[implies[and[p1, p2], p5]],
           {p1 → subclass[x, PRIMES], p2 → not[empty[x]], p3 → subclass[x, omega],
            p4 → member[A[x], x], p5 → member[A[x], PRIMES]}]]

Out[25]= or[equal[0, x], member[A[x], PRIMES], not[subclass[x, PRIMES]]] == True

In[26]:= or[equal[0, x_], member[A[x_], PRIMES], not[subclass[x_, PRIMES]]] := True
```

Lemma.

```
In[27]:= SubstTest[implies, member[x, y], subclass[A[y], x], {x → succ[succ[set[0]]],
           y -> intersection[PRIMES, image[S, set[succ[succ[set[0]]]]]]}]

Out[27]= subclass[hull[PRIMES, succ[succ[set[0]]]], succ[succ[set[0]]]] == True

In[28]:= % /. Equal → SetDelayed
```

Theorem: the next prime after 2 is 3.

```
In[29]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> hull[PRIMES, succ[succ[set[0]]]], v -> succ[succ[set[0]]]}]

Out[29]= True == equal[hull[PRIMES, succ[succ[set[0]]]], succ[succ[set[0]]]]

In[30]:= hull[PRIMES, succ[succ[set[0]]]] := succ[succ[set[0]]]
```

---

5 is the next prime after 3

An upper bound is easy:

```
In[31]:= SubstTest[implies, member[x, y],
          subclass[A[y], x], {x → succ[succ[succ[succ[set[0]]]]],
           y -> intersection[PRIMES, image[S, set[succ[succ[succ[set[0]]]]]]]}]

Out[31]= subclass[hull[PRIMES, succ[succ[succ[set[0]]]]],
           succ[succ[succ[succ[set[0]]]]]] == True

In[32]:= % /. Equal → SetDelayed
```

It was already shown that **hull[PRIMES, x]** can never be **0** or **1**. One can rule out **2** and **3** as well for the present case.

```
In[33]:= Map[not, SubstTest[subclass, nat[x], nat[y], {x -> succ[succ[succ[set[0]]]],
             y -> hull[PRIMES, succ[succ[succ[set[0]]]]]}]] // Reverse

Out[33]= or[equal[hull[PRIMES, succ[succ[succ[set[0]]]]], succ[set[0]]],
           equal[hull[PRIMES, succ[succ[succ[set[0]]]]], succ[succ[set[0]]]]] == False

In[34]:= % /. Equal → SetDelayed
```

Four is not a prime:

```
In[35]:= Map[not, SubstTest[implies, and[equal[u, v], member[v, w]],
           member[u, w], {u -> succ[succ[succ[set[0]]]],
            v -> hull[PRIMES, succ[succ[succ[set[0]]]]], w → PRIMES}]]

Out[35]= equal[hull[PRIMES, succ[succ[succ[set[0]]]]],
           succ[succ[succ[set[0]]]]] == False

In[36]:= % /. Equal → SetDelayed
```

Corollary

```
In[37]:= SubstTest[subclass, nat[x], nat[y], {x -> succ[succ[succ[succ[set[0]]]]],
           y -> hull[PRIMES, succ[succ[succ[set[0]]]]]}]

Out[37]= member[succ[succ[succ[set[0]]]],
           hull[PRIMES, succ[succ[succ[set[0]]]]]] == True

In[38]:= % /. Equal → SetDelayed
```

By a process of elimination one has arrived at the fact that the next prime after 3 is 5.

```
In[39]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> hull[PRIMES, succ[succ[succ[set[0]]]]],
           v -> succ[succ[succ[succ[set[0]]]]]}]

Out[39]= True == equal[hull[PRIMES, succ[succ[succ[set[0]]]]],
           succ[succ[succ[succ[set[0]]]]]]

In[40]:= hull[PRIMES, succ[succ[succ[set[0]]]]] := succ[succ[succ[succ[set[0]]]]]
```