rank of a well-founded relation

Johan G. F. Belinfante
2012 April 2

```
In[1]:= SetDirectory["l:"]; << goedel.12mar30a

:Package Title: goedel.12mar30a                    2012 March 30 at 2:15 p.m.

Loading takes about seventeen minutes, half that time due to builtin pauses.

It is now:  2012 Apr 2 at 10:51

Loading Simplification Rules

TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

weightlimit = 40

Loading completed.

It is now:  2012 Apr 2 at 11:8
```

summary

The theory of well-founded recursion is applied to the study of the rank function associated with a small well-founded relation. If **w** is a small wellfounded relation, the associated rank function is **rec[composite[TC ∘ IMAGE[SECOND] ∘ SECOND], inverse[w]]**. The compound wrapper **wf[setpart[x]]** is used in this notebook to obtain simpler rewrite rules.

the recursion equation

From the general theory of well-founded recursion one has the following recursion equation for the rank function.

Theorem. A recursion equation for the rank function.

```
In[2]:= SubstTest[or,
          equal[composite[TC, IMAGE[rec[composite[TC, IMAGE[SECOND], SECOND], y]], VERTSECT[y]],
           rec[composite[TC, IMAGE[SECOND], SECOND], y]], not[equal[V, domain[VERTSECT[y]]]],
          not[WELLFOUNDED[inverse[y]]], y → inverse[wf[setpart[x]]]] // Reverse

Out[2]= equal[composite[TC,
           IMAGE[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]],
           VERTSECT[inverse[wf[setpart[x]]]]],
          rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]] == True
```

```
In[3]:= composite[TC,
          IMAGE[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]],
          VERTSECT[inverse[wf[setpart[x_]]]]] :=
         rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]
```

Theorem. The only solution of the recursion equation is the rank function.

```
In[4]:= SubstTest[or, equal[w, rec[composite[TC, IMAGE[SECOND], SECOND], y]],
          not[equal[V, domain[VERTSECT[y]]]],
          not[equal[w, composite[TC, IMAGE[w], VERTSECT[y]]]],
          not[WELLFOUNDED[inverse[y]]], y → inverse[wf[setpart[x]]]] // Reverse

Out[4]= or[equal[w, rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]],
          not[equal[w, composite[TC, IMAGE[w], VERTSECT[inverse[wf[setpart[x]]]]]]]] == True

In[5]:= or[equal[w_, rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]],
          not[equal[composite[TC, IMAGE[w_], VERTSECT[inverse[wf[setpart[x_]]]]], w_]]] := True
```

properties of the rank function

Most of the theorems about rank functions follow from this recursion equation, but since this has already been done in a more general context, the general theory will be used ro derive most theorems, instead of going back to this recursion equation

Theorem. The rank function is a function.

```
In[6]:= SubstTest[or, FUNCTION[rec[composite[TC, IMAGE[SECOND], SECOND], y]],
          not[equal[V, domain[VERTSECT[y]]]], not[WELLFOUNDED[inverse[y]]],
          y → inverse[wf[setpart[x]]]] // Reverse

Out[6]= FUNCTION[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]] == True

In[7]:= FUNCTION[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]] := True
```

Theorem. The rank function is total.

```
In[8]:= SubstTest[or, equal[V, domain[rec[composite[TC, IMAGE[SECOND], SECOND], y]]],
          not[equal[V, domain[VERTSECT[y]]]], not[WELLFOUNDED[inverse[y]]],
          y → inverse[wf[setpart[x]]]] // Reverse

Out[8]= equal[V, domain[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]]] ==
         True

In[9]:= domain[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]] := V
```

intertwining property of rank functions

The following lemma was used to derive an intertwining property of the rank function.

```
In[11]:= composite[inverse[E], IMAGE[funpart[w]], VERTSECT[y]]

Out[11]= composite[funpart[w], thinpart[y]]
```

Theorem. The **funpart** wrapper can be removed:

```
In[13]:= SubstTest[implies, equal[t, funpart[w]],
           equal[composite[inverse[E], IMAGE[t], VERTSECT[x]],
            composite[t, thinpart[x]]], t → w] // Reverse

Out[13]= or[equal[composite[w, thinpart[x]], composite[inverse[E], IMAGE[w], VERTSECT[x]]],
           not[FUNCTION[w]]] == True

In[16]:= or[equal[composite[w_, thinpart[x_]], composite[inverse[E], IMAGE[w_], VERTSECT[x_]]],
           not[FUNCTION[w_]]] := True
```

Theorem. Intertwining property of rank functions.

```
In[23]:= SubstTest[implies, and[FUNCTION[w],
            equal[V, domain[w]], equal[w, composite[TC, IMAGE[w], VERTSECT[y]]]],
           subclass[composite[w, y], composite[inverse[E], w]],
           {w -> rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
            y → inverse[wf[setpart[x]]]}] // Reverse

Out[23]= subclass[composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
            inverse[wf[setpart[x]]]], composite[inverse[E],
            rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]]] == True

In[25]:= subclass[
           composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]],
            inverse[wf[setpart[x_]]]], composite[inverse[E],
            rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]]] := True
```

That is, the rank function **w** satisfies the intertwining inclusion **w ∘ inverse[wf[setpart[x]]] ⊂ inverse[E] ∘ w**.

Theorem. A monotonicity property of the rank function.

```
In[43]:= SubstTest[implies, subclass[u, v], subclass[composite[u, w], composite[v, w]],
           {u -> composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
              inverse[wf[setpart[x]]]], v -> composite[inverse[E],
              rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]], w → inverse[
              rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]]}] // Reverse

Out[43]= subclass[composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
            wf[setpart[x]], inverse[
             rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]]], E] == True

In[44]:= subclass[composite[rec[composite[TC, IMAGE[SECOND], SECOND],
             inverse[wf[setpart[x_]]]], wf[setpart[x_]], inverse[
             rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]]], E] := True
```

Restatement.

```
In[45]:= subclass[P[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]],
          monotone[wf[setpart[x]], E]]

Out[45]= True
```

Because the rank function is total, a stronger inclusion can be derived.

Theorem. Another monotonicity property.

```
In[54]:= SubstTest[implies, and[subclass[u, cart[V, V]], FUNCTION[w],
            subclass[composite[w, u, inverse[w]], v], equal[domain[w], V]],
          subclass[u, composite[inverse[w], v, w]], {u → wf[setpart[x]], v → E,
            w -> rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]}] // Reverse

Out[54]= subclass[wf[setpart[x]], composite[
            inverse[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]],
            E, rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]]] == True

In[56]:= subclass[wf[setpart[x_]], composite[
            inverse[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]],
            E, rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]]] := True
```

---

rank functions have ordinal values

Theorem. The values of the rank function are ordinals.

```
In[26]:= SubstTest[or, not[equal[V, domain[VERTSECT[y]]]], not[WELLFOUNDED[inverse[y]]],
          subclass[range[rec[composite[TC, IMAGE[SECOND], SECOND], y]], OMEGA],
          y → inverse[wf[setpart[x]]]] // Reverse

Out[26]= subclass[range[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]],
          OMEGA] == True

In[27]:= subclass[range[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]],
          OMEGA] := True
```

---

another monotonicity property

Theorem.

```
In[75]:= SubstTest[implies, subclass[u, v], subclass[monotone[x, u], monotone[x, v]],
          {u → composite[id[OMEGA], E], v → S}] // Reverse

Out[75]= subclass[monotone[x, composite[id[OMEGA], E]], monotone[x, S]] == True

In[76]:= subclass[monotone[x_, composite[id[OMEGA], E]], monotone[x_, S]] := True
```

Theorem.

```
In[80]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u → P[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]],
            v -> monotone[wf[setpart[x]], composite[id[OMEGA], E]],
            w -> monotone[wf[setpart[x]], S]}] // Reverse

Out[80]= subclass[composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
           wf[setpart[x]], inverse[
            rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]]], S] == True

In[81]:= subclass[composite[rec[composite[TC, IMAGE[SECOND], SECOND],
            inverse[wf[setpart[x_]]]], wf[setpart[x_]], inverse[
            rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]]], S] := True
```

---

the trivial part of the rank function

Theorem. The rank function **w** is trivial on the complement of the range of the well-founded relation.

```
In[28]:= SubstTest[or, equal[complement[domain[y]],
            image[inverse[rec[composite[TC, IMAGE[SECOND], SECOND], y]], set[0]]],
           not[equal[V, domain[VERTSECT[y]]]], not[WELLFOUNDED[inverse[y]]],
           y → inverse[wf[setpart[x]]]] // Reverse

Out[28]= equal[complement[range[wf[setpart[x]]]],
           image[inverse[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]],
            set[0]]] == True

In[29]:= image[inverse[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]]],
           set[0]] := complement[range[wf[setpart[x]]]]
```

Corollary.

```
In[30]:= Map[subclass[#, set[0]] &,
           ImageComp[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
            inverse[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]],
            set[0]]] // Reverse

Out[30]= subclass[image[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
            complement[range[wf[setpart[x]]]]], set[0]] == True

In[31]:= subclass[image[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]],
            complement[range[wf[setpart[x_]]]]], set[0]] := True
```

Corollary. The restriction of the rank function to the complement of the range of the well-founded relation is a constant with value **0**.

```
In[32]:= (equal[composite[w, id[t]], cart[t, set[0]]] // AssertTest) /.
          {w → rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
           t -> complement[range[wf[setpart[x]]]]}

Out[32]= equal[cart[complement[range[wf[setpart[x]]]], set[0]],
           composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
            id[complement[range[wf[setpart[x]]]]]]] == True

In[33]:= composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]],
           id[complement[range[wf[setpart[x_]]]]]] :=
          cart[complement[range[wf[setpart[x]]]], set[0]]
```

Corollary. The rank function can be recovered from its restriction to the range of the well-founded relation.

```
In[34]:= Map[equal[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]], #] &,
           SubstTest[union, composite[w, id[t]], composite[w, id[complement[t]]],
            {w → rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
             t → range[wf[setpart[x]]]}]] // Reverse

Out[34]= equal[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
           union[cart[complement[range[wf[setpart[x]]]], set[0]],
            composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
             id[range[wf[setpart[x]]]]]]] == True

In[36]:= union[cart[complement[range[wf[setpart[x_]]]], set[0]],
           composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]],
            id[range[wf[setpart[x_]]]]]] :=
          rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]]
```

Theorem. A mapping property for the nontrivial restriction of the rank function.

```
In[38]:= member[composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
            id[range[wf[setpart[x]]]]], map[range[wf[setpart[x]]], OMEGA]] // AssertTest

Out[38]= member[composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x]]]],
            id[range[wf[setpart[x]]]]], map[range[wf[setpart[x]]], OMEGA]] == True

In[39]:= member[composite[rec[composite[TC, IMAGE[SECOND], SECOND], inverse[wf[setpart[x_]]]],
            id[range[wf[setpart[x_]]]]], map[range[wf[setpart[x_]]], OMEGA]] := True
```