unions of chains

Johan G. F. Belinfante
2007 June 23

\[\text{In [1]} := \text{SetDirectory["1"]}; \text{\textless\textless goedel94.21a; \textless\textless tools.m}\]

:Package Title: goedel94.21a 2007 June 21 at 7:55 p.m.

It is now: 2007 Jun 23 at 21:19

Loading Simplification Rules

TOOLS.M Revised 2007 June 10

weightlimit = 40

\section*{summary}

The class \texttt{Uchains[x]} is the class of all unions of chains \(a \subseteq b \subseteq \cdots\) in a class \(x\). The following new membership rule for the constructor \texttt{Uchains} has been added to the \texttt{GOEDEL} program.

\[\text{In [2]} := \text{Begin["Goedel\`Private`"]};\]

\[\text{In [3]} := \text{FirstMatch[class[w\_, member[x\_, HoldPattern[Uchains[y\_]]]]]}\]

\[\text{Out[3]} := \text{class[u\_, member[v\_, Uchains[x\_]]]} := \text{Module[\{w = Unique[\_], class[u, and[member[v, V], exists[w, and[subclass[w, x], subclass[cart[w, w], union[S, inverse[S]]], equal[v, U[w]]]]] \}]]\]

In this notebook rewrite rules for this new constructor are derived. Some of these rewrite rules replace existing rules about unions of chains that were formulated without this terminology. These old rules will be removed after \texttt{Uchains} has been normalized.

\section*{replacement rewrite rules}

In this section existing rewrite rules about unions of chains are rewritten using the new terminology. Normalization of \texttt{Uchains} will be delayed to take advantage of existing rules to derive their replacements. For example, since every singleton in \(x\) is a chain, each element \(w\) of \(x\) can be written as the union of the singleton chain \texttt{set[w]}. Consequently \(x\) is a subclass of \texttt{Uchains[x]}. The new rule expressing this fact can be quickly derived from the existing rule by using \texttt{AssertTest}.

\[\text{In [4]} := \text{subclass[x, Uchains[x]] // AssertTest}\]

\[\text{Out[4]} := \text{subclass[x, Uchains[x]] = True}\]

\[\text{In [5]} := \text{subclass[x\_, Uchains[x\_]] := True}\]
Perhaps the most important result about Uchains is that the union of an S-chain of x-cliques is an x-clique.

```
In[6]:= Uchains[cliques[x]] // Normality
Out[6]= Uchains[cliques[x]] = cliques[x]
```

```
In[7]:= Uchains[cliques[x_]] := cliques[x]
```

Theorem. (Monotonicity property.)

```
In[8]:= implies[subclass[x, y], subclass[Uchains[x], Uchains[y]]] // AssertTest
Out[8]= or[not[subclass[x, y]], subclass[Uchains[x], Uchains[y]]] = True
```

```
In[9]:= or[not[subclass[x_, y_], subclass[Uchains[x_], Uchains[y_]]]] := True
```

**the empty set**

The empty set is the union of the empty chain.

```
In[10]:= member[0, Uchains[x]] // AssertTest
Out[10]= member[0, Uchains[x]] = True
```

```
In[11]:= member[0, Uchains[x_]] := True
```

For the empty set, this is the only union of chains:

```
In[12]:= Uchains[0] // Normality
Out[12]= Uchains[0] = set[0]
```

```
In[13]:= Uchains[0] := set[0]
```

Corollary. The class Uchains[x] is never empty.

```
In[14]:= SubstTest[and, member[0, t], equal[0, t], t -> Uchains[x]] // Reverse
Out[14]= equal[0, Uchains[x]] = False
```

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

At the other extreme, one has:

```
In[16]:= Uchains[V] // Normality
Out[16]= Uchains[V] = V
```

```
In[17]:= Uchains[V] := V
```
examples

Theorem. The union of a chain of chains is a chain.

\[ \text{In [18]} := \text{SubstTest[Uchains, cliques[t], } t \rightarrow \text{union}[x, \text{inverse}[x]] \] // Reverse
\[ \text{Out [18]} = \text{Uchains[chains[x]]} = \text{chains[x]} \]

In [19] := Uchains[chains[x_]] := chains[x]

Theorem. Power classes are closed under unions of chains.

\[ \text{In [20]} := \text{SubstTest[Uchains, cliques[t], } t \rightarrow \text{cart}[x, x] \] // Reverse
\[ \text{Out [20]} = \text{Uchains[P[x]]} = \text{P[x]} \]

In [21] := Uchains[P[x_]] := P[x]

Theorem. The class of all small functions is closed under unions of chains.

\[ \text{In [22]} := \text{SubstTest[Uchains, cliques[t], } t \rightarrow \text{composite}[\text{id}[\text{cart}[V, V]], \text{complement}[\text{cross}[\text{Id}, \text{Di}]]] \] // Reverse
\[ \text{Out [22]} = \text{Uchains[FUNS]} = \text{FUNS} \]

In [23] := Uchains[FUNS] := FUNS

A similar result holds for inverses of functions.

\[ \text{In [24]} := \text{SubstTest[Uchains, cliques[t], } t \rightarrow \text{composite}[\text{id}[\text{cart}[V, V]], \text{complement}[\text{cross}[\text{Di}, \text{Id}]]] \] // Reverse
\[ \text{Out [24]} = \text{Uchains[image[INVERSE, FUNS]]} = \text{image[INVERSE, FUNS]} \]


Theorem. The class of antisymmetric relations is closed under unions of chains.

\[ \text{In [26]} := \text{SubstTest[Uchains, cliques[t], } t \rightarrow \text{union}[\text{cart}[\text{Id}, V], \text{composite}[\text{Di, SWAP, id[Di]}]] \] // Reverse
\[ \text{Out [26]} = \text{Uchains[ANTISYM]} = \text{ANTISYM} \]

In [27] := Uchains[ANTISYM] := ANTISYM

Theorem. The class of constant functions is closed under unions of chains.

\[ \text{In [28]} := \text{SubstTest[Uchains, cliques[x], } x \rightarrow \text{composite}[\text{inverse[SECOND], SECOND}] \] // Reverse
\[ \text{Out [28]} = \text{Uchains[CONST]} = \text{CONST} \]

In [29] := Uchains[CONST] := CONST
intersections of classes closed under unions of chains

A class \( x \) is \textbf{closed under unions of chains} if \( \text{Uchains}[x] = x \). If \( x \) and \( y \) are closed under unions of chains, then so is their intersection.

\[
\text{In}[30]:= \text{implies}[
\text{and}[\text{equal}[x, \text{Uchains}[x]], \text{equal}[y, \text{Uchains}[y]]],
\text{equal}[\text{Uchains}[\text{intersection}[x, y]], \text{intersection}[x, y]]] \tag{AssertTest}
\]
\[
\text{Out}[30]= \text{or}[\text{equal}[\text{intersection}[x, y], \text{Uchains}[\text{intersection}[x, y]]],
\text{not}[\text{equal}[x, \text{Uchains}[x]]], \text{not}[\text{equal}[y, \text{Uchains}[y]]]] = \text{True}
\]

Theorem. The class of bijections is closed under unions of chains.

\[
\text{In}[32]:= \text{SubstTest}[
\text{implies}, \text{and}[\text{equal}[x, \text{Uchains}[x]], \text{equal}[y, \text{Uchains}[y]]],
\text{equal}[\text{Uchains}[\text{intersection}[x, y]], \text{intersection}[x, y]],
\{x \mapsto \text{FUNS}, y \mapsto \text{image}[\text{INVERSE}, \text{FUNS}]\}]
\]
\[
\text{Out}[32]= \text{True} = \text{equal}[\text{BIJ}, \text{Uchains}[\text{BIJ}]]
\]

\[
\text{In}[33]:= \text{Uchains}[\text{BIJ}] := \text{BIJ}
\]

\section*{normalization}

Theorem. (Normalization rule.)

\[
\text{In}[34]:= \text{Uchains}[x] \tag{Normality} \tag{Reverse}
\]
\[
\text{Out}[34]= \text{image}[\text{BIGCUP}, \text{intersection}[\text{chains}[S], \text{P}[x]]] = \text{Uchains}[x]
\]

\[
\text{In}[35]:= \text{image}[\text{BIGCUP}, \text{intersection}[\text{chains}[S], \text{P}[x_\_]]] := \text{Uchains}[x]
\]

Theorem.

\[
\text{In}[36]:= \text{SubstTest}[U, \text{image}[\text{BIGCUP}, t], t \mapsto \text{intersection}[\text{chains}[S], \text{P}[x]]] \tag{Reverse}
\]
\[
\text{Out}[36]= U[\text{Uchains}[x]] = U[x]
\]

\[
\text{In}[37]:= U[\text{Uchains}[x_\_]] := U[x]
\]

Theorem. Sethood property.

\[
\text{In}[38]:= \text{SubstTest}[\text{member}, U[t], V, t \mapsto \text{Uchains}[x]]
\]
\[
\text{Out}[38]= \text{member}[\text{Uchains}[x], V] = \text{member}[x, V]
\]

\[
\text{In}[39]:= \text{member}[\text{Uchains}[x_\_], V] := \text{member}[x, V]
\]
**Uclosure and Uchains**

The formula for Uchains[x] resembles the definition of Uclosure[x]. Many, but not all, properties of Uclosure carry over to Uchains.

```
In[42]:= image[BIGCUP, P[x]]
Out[42]= Uclosure[x]
```

Theorem.

```
In[43]:= SubstTest subclass, image[BIGCUP, intersection[u, v]],
        image[BIGCUP, v], {u → intersection[chains[S], P[x]], v → P[x]}} // Reverse
Out[43]= subclass[Uchains[x], Uclosure[x]] = True
In[44]:= subclass[Uchains[x_], Uclosure[x_]] := True
```

Lemma.

```
In[45]:= SubstTest subclass, Uchains[t], Uclosure[t], t → Uclosure[x] // Reverse
Out[45]= subclass[Uchains[Uclosure[x]], Uclosure[x]] = True
In[46]:= (% /. x → x_) /. Equal → SetDelayed
```

```
In[47]:= SubstTest and, subclass[u, v], subclass[v, u],
        {u → Uchains[Uclosure[x]], v → Uclosure[x]}}
Out[47]= equal[Uchains[Uclosure[x]], Uclosure[x]] = True
```

```
In[48]:= Uchains[Uclosure[x_]] := Uclosure[x]
```

Lemma.

```
In[49]:= SubstTest implies, subclass[x, t],
        subclass[Uclosure[x], Uclosure[t]], t → Uchains[x] // Reverse
Out[49]= subclass[Uclosure[x], Uclosure[Uchains[x]]] = True
In[50]:= (% /. x → x_) /. Equal → SetDelayed
```
The formulas connecting Uchains and Uclosure can be used to show that many familiar classes are closed under unions of chains. For example:
Theorem. The class of full sets is closed under arbitrary unions, but it is closed under unions of chains.

Theorem. The class of reflexive relations is closed under arbitrary unions, and therefore also under unions of chains.

Theorem. The class of symmetric relations is closed under arbitrary unions, and therefore also under unions of chains.

Theorem. A similar result holds for the class of \( x \)-subvariant sets.
Corollary. The intersection of $\text{invar}[x]$ and $\text{subvar}[x]$ is closed under arbitrary unions, and hence under unions of chains.

Corollary. $\text{transvar}[x, y]$ is closed under arbitrary unions, and hence under unions of chains.

Corollary. $\text{subcommutant}[x]$ is closed under arbitrary unions, and hence under unions of chains.

Corollary. $\text{commutant}[x]$ is closed under arbitrary unions, and hence under unions of chains.

**UNOPS**

The class of unary operations is not closed under arbitrary unions, but it is closed under unions of chains.
adjointing 0

Lemma.

\[ \text{In [87]} := \text{composite}[	ext{BIGCUP}, \text{id}[	ext{chains}[	ext{S}] ], \text{ADJOIN}[	ext{set}[0]]] \] // RelnNormality

\[ \text{Out [87]} = \text{composite}[	ext{BIGCUP}, \text{id}[	ext{chains}[	ext{S}] ], \text{ADJOIN}[	ext{set}[0]]] = \text{composite}[	ext{BIGCUP}, \text{id}[	ext{chains}[	ext{S}] ]]
\]

\[ \text{In [88]} := \text{composite}[	ext{BIGCUP}, \text{id}[	ext{chains}[	ext{S}] ], \text{ADJOIN}[	ext{set}[0]]] := \text{composite}[	ext{BIGCUP}, \text{id}[	ext{chains}[	ext{S}] ]]
\]

Lemma. (Temporary rule.)

\[ \text{In [89]} := \text{ImageComp}[	ext{composite}[	ext{BIGCUP}, \text{id}[	ext{chains}[	ext{S}]]], \text{ADJOIN}[	ext{set}[0]], \text{P}[	ext{x}]] \] // Reverse

\[ \text{Out [89]} = \text{image}[	ext{BIGCUP}, \text{intersection}[	ext{chains}[	ext{S}]], \text{complement}[	ext{P}[	ext{complement}[	ext{set}[0]]]], \text{P}[	ext{union}[	ext{x}, \text{set}[0]]]]} = \text{Uchains}[	ext{x}]
\]

\[ \text{In [90]} := \text{image}[	ext{BIGCUP}, \text{intersection}[	ext{chains}[	ext{S}]], \text{complement}[	ext{P}[	ext{complement}[	ext{set}[0]]]], \text{P}[	ext{union}[	ext{x}_{\text{\_}}, \text{set}[0]]]]} := \text{Uchains}[	ext{x}]
\]

Theorem. Adjoining 0 does not change the closure under unions of chains.

\[ \text{In [91]} := \text{ImageComp}[	ext{composite}[	ext{BIGCUP}, \text{id}[	ext{chains}[	ext{S}]]], \text{ADJOIN}[	ext{set}[0]], \text{P}[	ext{union}[	ext{x}, \text{set}[0]]]]
\]

\[ \text{Out [91]} = \text{Uchains}[	ext{union}[	ext{x}, \text{set}[0]]] = \text{Uchains}[	ext{x}]
\]

\[ \text{In [92]} := \text{Uchains}[	ext{union}[	ext{x}_{\text{\_}}, \text{set}[0]]] := \text{Uchains}[	ext{x}]
\]

Corollary. One can also delete 0 instead of adding 0.

\[ \text{In [93]} := \text{ImageComp}[	ext{composite}[	ext{BIGCUP}, \text{id}[	ext{chains}[	ext{S}]]], \text{ADJOIN}[	ext{set}[0]], \text{P}[	ext{dif}[	ext{x}, \text{set}[0]]]]
\]

\[ \text{Out [93]} = \text{Uchains}[	ext{intersection}[	ext{x}, \text{complement}[	ext{set}[0]]]] = \text{Uchains}[	ext{x}]
\]

\[ \text{In [94]} := \text{Uchains}[	ext{intersection}[	ext{x}_{\text{\_}}, \text{complement}[	ext{set}[0]]]] := \text{Uchains}[	ext{x}]
\]

Corollary. Unions of chains of superset say.

\[ \text{In [95]} := \text{SubstTest}[	ext{Uchains}, \text{Uclosure}[	ext{t}], \text{t} \rightarrow \text{image}[	ext{S}, \text{set}[	ext{x}]]] \] // Reverse

\[ \text{Out [95]} = \text{Uchains}[	ext{image}[	ext{S}, \text{set}[	ext{x}]]] = \text{union}[	ext{image}[	ext{S}, \text{set}[	ext{x}]], \text{set}[0]]
\]

\[ \text{In [96]} := \text{Uchains}[	ext{image}[	ext{S}, \text{set}[	ext{x}_{\text{\_}}]]] := \text{union}[	ext{image}[	ext{S}, \text{set}[	ext{x}]], \text{set}[0]]
\]
**reify rule for Uchains[x]**

Theorem.

\[\text{In}[97]:= \text{SubstTest[reify, x, image[BIGCUP, intersection[t, P[f[x]]]], t \rightarrow chains[S]] // Reverse}\]

\[\text{Out}[97]= \text{reify[x, Uchains[f[x]]] = composite[BIGCUP, id[chains[S]], inverse[LB[reify[x, f[x]]]]]}\]

\[\text{In}[98]:= \text{reify[x_, Uchains[y_]] := composite[BIGCUP, id[chains[S]], inverse[LB[reify[x, y]]]]}\]