Capítulo 7

Conjuntos, funciones y relaciones

No busquéis el significado, buscad el uso.
L. WITTGENSTEIN

7.1 Conjuntos

7.1.1 Operaciones con conjuntos

Nota 7.1.1. La teoría elemental de conjuntos es `HOL/Set.thy`.

Nota 7.1.2. En un conjunto todos los elemento son del mismo tipo (por ejemplo, del tipo τ) y el conjunto tiene tipo (en el ejemplo, τ *set*).

Nota 7.1.3 (Reglas de la intersección)**.**

- $[\![c \in A; c \in B]\!] \Longrightarrow c \in A \cap B$ *(IntI)*
- $c \in A \cap B \Longrightarrow c \in A$ *(IntD1)*
- $c \in A \cap B \Longrightarrow c \in B$ *(IntD2)*

Nota 7.1.4. Propiedades del complementario:

- $(c \in -A) = (c \notin A)$ (*Compl_iff*)
- $-(A \cup B) = -A \cap -B$ (*Compl_Un*)

Nota 7.1.5. El conjunto **vacío** se representa por $\{\}$ y el **universal** por *UNIV*.

Nota 7.1.6. Propiedades de la **diferencia** y del complementario:

- $A \cap (B - A) = \{\}$ (*Diff_disjoint*)

- $A \cup - A = UNIV$ (*Compl_partition*)

Nota 7.1.7. Reglas de la relación de **subconjunto**:

- $(\bigwedge x.\ x \in A \Longrightarrow x \in B) \Longrightarrow A \subseteq B$ (*subsetI*)
- $[\![A \subseteq B; c \in A]\!] \Longrightarrow c \in B$ (*subsetD*)

Nota 7.1.8. Ejemplo trivial.

lemma $(A \cup B \subseteq C) = (A \subseteq C \wedge B \subseteq C)$
by *blast*

Nota 7.1.9. Otro ejemplo trivial.

lemma $(A \subseteq -B) = (B \subseteq -A)$
by *blast*

Nota 7.1.10. Principio de extensionalidad de conjuntos:

- $(\bigwedge x.\ (x \in A) = (x \in B)) \Longrightarrow A = B$ (*set_ext*)

Nota 7.1.11. Reglas de la **igualdad** de conjuntos:

- $[\![A \subseteq B; B \subseteq A]\!] \Longrightarrow A = B$ (*equalityI*)
- $[\![A = B; [\![A \subseteq B; B \subseteq A]\!] \Longrightarrow P]\!] \Longrightarrow P$ (*equalityE*)

Lema 7.1.12 (Analogía entre intersección y conjunción). *$x \in A \cap B$ syss $x \in A$ y $x \in B$.*

lemma $(x \in A \cap B) = (x \in A \wedge x \in B)$
by *simp*

Lema 7.1.13 (Analogía entre unión y disyunción). *$x \in A \cup B$ syss $x \in A$ ó $x \in B$.*

lemma $(x \in A \cup B) = (x \in A \vee x \in B)$
by *simp*

Lema 7.1.14 (Analogía entre subconjunto e implicación). *$A \subseteq B$ syss para todo x, si $x \in A$ entonces $x \in B$.*

lemma $(A \subseteq B) = (\forall\ x.\ x \in A \longrightarrow x \in B)$
by *auto*

Lema 7.1.15 (Analogía entre complementario y negación). *x pertenece al complementario de A syss x no pertenece a A.*

lemma $(x \in -A) = (x \notin A)$
by *simp*

7.1.2 Notación de conjuntos finitos

Nota 7.1.16. La teoría de conjuntos finitos es `HOL/Finite_Set.thy`.

Nota 7.1.17. Los conjuntos finitos se definen por inducción a partir de las siguientes reglas inductivas:

- El conjunto vacío es un conjunto finito.
  finite $\{\}$ (*emptyI*)
- Si se le añade un elemento a un conjunto finito se obtiene otro conjunto finito.
  finite $A \Longrightarrow$ *finite* (*insert a A*) (*insertI*)

Nota 7.1.18. En la notación matemática, las reglas anteriores se representan como sigue:

- El conjunto vacío es un conjunto finito.
  finite $\emptyset$ (*emptyI*)
- Si se le añade un elemento a un conjunto finito se obtiene otro conjunto finito.
  finite $A \Longrightarrow$ *finite* $(\{a\} \cup A)$ (*insertI*)

Ejemplo 7.1.19. Ejemplos de conjuntos finitos.

---

lemma
insert $2\ \{\} = \{2\} \wedge$
insert $3\ \{2\} = \{2,3\} \wedge$
insert $2\ \{2,3\} = \{2,3\} \wedge$
$\{2,3\} = \{3,2,3,2,2\}$
by *auto*

Nota 7.1.20. Los conjuntos finitos se representan con la notación conjuntista habitual: los elementos entre llaves y separados por comas.

Nota 7.1.21. Ejemplo trivial.

lemma $\{a,b\} \cup \{c,d\} = \{a,b,c,d\}$
by *blast*

Nota 7.1.22. Conjetura falsa.

lemma $\{a,b\} \cap \{b,c\} = \{b\}$
refute
oops

Nota 7.1.23. Conjetura corregida.

lemma $\{a,b\} \cap \{b,c\} = (\textit{if } a{=}c \textit{ then } \{a,b\} \textit{ else } \{b\})$
by *auto*

Nota 7.1.24 (Sumas y productos de conjuntos finitos)**.** Se pueden definir la suma y el producto de los elementos de un conjunto finito mediante las siguientes funciones:

- *setsum* tal que $(\textit{setsum } f\ A)$ es la suma de la aplicación de f a los elementos del conjunto finito A,
- *setprod* tal que $(\textit{setprod } f\ A)$ es producto de la aplicación de f a los elementos del conjunto finito A,
- $\sum$ tal que $\sum A$ es la suma de los elementos del conjunto finito A,
- $\prod$ tal que $\prod A$ es el producto de los elementos del conjunto finito A.

Definición 7.1.25 (Ejemplos de definiciones recursivas sobre conjuntos finitos)**.** *Sea A un conjunto finito de números naturales.*

- *sumaConj A es la suma de los elementos A.*
- *productoConj A es el producto de los elementos de A.*
- *sumaCuadradosconj A es la suma de los cuadrados de los elementos A.*

definition *sumaConj* :: *nat set* $\Rightarrow$ *nat* **where**
sumaConj S $\equiv \sum S$

definition *productoConj* :: *nat set* $\Rightarrow$ *nat* **where**
productoConj S $\equiv \prod S$

definition *sumaCuadradosConj* :: *nat set* $\Rightarrow$ *nat* **where**
sumaCuadradosConj S $\equiv$ *setsum* $(\lambda\ x.\ x{*}x)$ *S*

Nota 7.1.26. Para simplificar lo que sigue, declaramos las anteriores definiciones como reglas de simplificación.

declare *sumaConj-def* [*simp*]
declare *productoConj-def* [*simp*]
declare *sumaCuadradosConj-def* [*simp*]

Ejemplo 7.1.27. Ejemplos de evaluación de las anteriores definiciones recursivas.

lemma
$sumaConj\ \{1,2,3,4\} = 10\ \wedge$
$productoConj\ \{1,2,3\} = productoConj\ \{3,2\}\ \wedge$
$sumaCuadradosConj\ \{1,2,3,4\} = 30$
by *simp*

Nota 7.1.28 (Inducción sobre conjuntos finitos)**.** Para demostrar que todos los conjuntos finitos tienen una propiedad P basta probar que

1. El conjunto vacío tiene la propiedad P.
2. Si a un conjunto finito que tiene la propiedad P se le añade un nuevo elemento, el conjunto obtenido sigue teniendo la propiedad P.

En forma de regla

$$(\textit{finite\_induct}) \qquad \frac{\textit{finite } F \qquad P\ \emptyset \qquad \bigwedge x\ F.\ \dfrac{\textit{finite } F \qquad x \notin F \qquad P\ F}{P\ (\{x\} \cup F)}}{P\ F}$$

Lema 7.1.29 (Ejemplo de inducción sobre conjuntos finitos)**.** *Sea S un conjunto finito de números naturales. Entonces todos los elementos de S son menores o iguales que la suma de los elementos de S.*

Demostración automática:

lemma $\textit{finite } S \Longrightarrow \forall\, x{\in}S.\ x \leq sumaConj\ S$
by (*induct rule*: *finite-induct*) *auto*

Demostración estructurada:

lemma *sumaConj-acota*: $\textit{finite } S \Longrightarrow \forall\, x{\in}S.\ x \leq sumaConj\ S$
proof (*induct rule*: *finite-induct*)
 show $\forall\, x{\in}\{\}.\ x \leq sumaConj\ \{\}$ **by** *simp*
next
 fix x **and** F
 assume *fF*: *finite F*
  and *xF*: $x \notin F$

```
   and HI: ∀ x∈F. x ≤ sumaConj F
 show ∀y∈insert x F. y ≤ sumaConj (insert x F)
 proof
  fix y
  assume y ∈ insert x F
  show y ≤ sumaConj (insert x F)
  proof (cases y = x)
   assume y = x
   hence y ≤ x + (sumaConj F) by simp
   also have ... = sumaConj (insert x F) using fF xF by simp
   finally show ?thesis .
  next
   assume y ≠ x
   hence y ∈ F using ‹y ∈ insert x F› by simp
   hence y ≤ sumaConj F using HI by blast
   also have ... ≤ x + (sumaConj F) by simp
   also have ... = sumaConj (insert x F) using fF xF by simp
   finally show ?thesis .
  qed
 qed
qed
```

7.1.3 Definiciones por comprensión

Nota 7.1.30. El conjunto de los elementos que cumple la propiedad P se representa por $\{x.\ P\}$.

Nota 7.1.31. Reglas de comprensión (relación entre colección y pertenencia):

- $(a \in \{x.\ P\ x\}) = P\ a$ *(mem_Collect_eq)*
- $\{x.\ x \in A\} = A$ *(Collect_mem_eq)*

Nota 7.1.32. Dos ejemplos triviales.

```
lemma {x. P x ∨ x ∈ A} = {x. P x} ∪ A
by blast
```

```
lemma {x. P x ⟶ Q x} = -{x. P x} ∪ {x. Q x}
by blast
```

Nota 7.1.33. Ejemplo con la sintaxis general de comprehensión.

lemma
$\{p{*}q \mid p\ q.\ p{\in}prime \wedge q{\in}prime\} =$
$\{z.\ \exists\, p\ q.\ z = p{*}q \wedge p{\in}prime \wedge q{\in}prime\}$
by *blast*

Nota 7.1.34. En HOL, la notación conjuntista es azúcar sintáctica:

- $x{\in}A$ es equivalente a $A(x)$.
- $\{x.\ P\}$ es equivalente a $\lambda x.\ P$.

Definición 7.1.35 (Ejemplo de definición por comprensión)**.** *El conjunto de los pares es el de los números n para los que existe un m tal que* $n = 2 * m$*.*

definition *Pares* :: *nat set* **where**
$Pares \equiv \{\ n.\ \exists\ m.\ n = 2{*}m\ \}$

Ejemplo 7.1.36. Los números 2 y 34 son pares.

lemma
$2 \in Pares\ \wedge$
$34 \in Pares$
by (*simp add*: *Pares-def*)

Definición 7.1.37. *El conjunto de los impares es el de los números n para los que existe un m tal que* $n = 2 * m + 1$*.*

definition *Impares* :: *nat set* **where**
$Impares \equiv \{\ n.\ \exists\ m.\ n = 2{*}m + 1\ \}$

Lema 7.1.38 (Ejemplo con las reglas de intersección y comprensión)**.** *El conjunto de los pares es disjunto con el de los impares.*

lemma $x \notin (Pares \cap Impares)$
proof
fix x **assume** S: $x \in (Pares \cap Impares)$
hence $x \in Pares$ **by** (*rule IntD1*)
hence $\exists\ m.\ x = 2 * m$ **by** (*simp only*: *Pares-def mem-Collect-eq*)
then obtain p **where** p: $x = 2 * p$ **..**

from S **have** $x \in Impares$ **by** (*rule IntD2*)

hence $\exists\ m.\ x = 2 * m + 1$ **by** (*simp only*: *Impares-def mem-Collect-eq*)
then obtain q **where** q: $x = 2 * q + 1$ **..**

from p **and** q **show** *False* **by** *arith*
qed

7.1.4 Cuantificadores acotados

Nota 7.1.39. Reglas de **cuantificador universal acotado** ("bounded"):

- $(\bigwedge x.\ x \in A \Longrightarrow P\ x) \Longrightarrow \forall\, x{\in}A.\ P\ x$ (*ballI*)
- $[\![\forall\, x{\in}A.\ P\ x;\ x \in A]\!] \Longrightarrow P\ x$ (*bspec*)

Nota 7.1.40. Reglas de **cuantificador existencial acotado** ("bounded"):

- $[\![P\ x;\ x \in A]\!] \Longrightarrow \exists\, x{\in}A.\ P\ x$ (*bexI*)
- $[\![\exists\, x{\in}A.\ P\ x;\ \bigwedge x.\ [\![x \in A;\ P\ x]\!] \Longrightarrow Q]\!] \Longrightarrow Q$ (*bexE*)

Nota 7.1.41. Reglas de la **unión indexada**:

- $(b \in (\bigcup x{\in}A.\ B\ x)) = (\exists\, x{\in}A.\ b \in B\ x)$ (*UN_iff*)
- $[\![a \in A;\ b \in B\ a]\!] \Longrightarrow b \in (\bigcup x{\in}A.\ B\ x)$ (*UN_I*)
- $[\![b \in (\bigcup x{\in}A.\ B\ x);\ \bigwedge x.\ [\![x \in A;\ b \in B\ x]\!] \Longrightarrow R]\!] \Longrightarrow R$ (*UN_E*)

Nota 7.1.42. Reglas de la **unión de una familia**:

- $\bigcup S = (\bigcup x{\in}S.\ x)$ (*Union_def*)
- $(A \in \bigcup C) = (\exists\, X{\in}C.\ A \in X)$ (*Union_iff*)

Nota 7.1.43. Reglas de la **intersección indexada**:

- $(b \in (\bigcap x{\in}A.\ B\ x)) = (\forall\, x{\in}A.\ b \in B\ x)$ (*INT_iff*)
- $(\bigwedge x.\ x \in A \Longrightarrow b \in B\ x) \Longrightarrow b \in (\bigcap x{\in}A.\ B\ x)$ (*INT_I*)
- $[\![b \in (\bigcap x{\in}A.\ B\ x);\ b \in B\ a \Longrightarrow R;\ a \notin A \Longrightarrow R]\!] \Longrightarrow R$ (*INT_E*)

Nota 7.1.44. Reglas de la **intersección de una familia**:

- $\bigcap S = (\bigcap x{\in}S.\ x)$ (*Inter_def*)
- $(A \in \bigcap C) = (\forall\, X{\in}C.\ A \in X)$ (*Inter_iff*)

Nota 7.1.45. Abreviaturas:

- *Collect P* es lo mismo que $\{x.\ P\}$.
- *All P* es lo mismo que $\forall\, x.\ P\ x$.
- *Ex P* es lo mismo que $\exists\, x.\ P\ x$.
- *Ball P* es lo mismo que $\forall\, x{\in}A.\ P\ x$.
- *Bex P* es lo mismo que $\exists\, x{\in}A.\ P\ x$.

7.1.5 Conjuntos finitos y cardinalidad

Nota 7.1.46. El número de elementos de un conjunto finito A es el cardinal de A y se representa por *card A*.

Ejemplo 7.1.47. Ejemplos de cardinales de conjuntos finitos.

lemma
 $card\ \{\} = 0 \wedge$
 $card\ \{4\} = 1 \wedge$
 $card\ \{4,1\} = 2 \wedge$
 $x \neq y \Longrightarrow card\ \{x,y\} = 2$
by *simp*

Nota 7.1.48. Propiedades de cardinales:

- Cardinal de la unión de conjuntos finitos:
  $[\![finite\ A; finite\ B]\!] \Longrightarrow card\ A + card\ B = card\ (A \cup B) + card\ (A \cap B)$ (*card_Un_Int*)
- Cardinal del conjunto potencia:
  $finite\ A \Longrightarrow card\ (Pow\ A) = 2\ \hat{}\ card\ A$ (*card_Pow*)

7.2 Funciones

La teoría de funciones es *HOL/Fun.thy*.

7.2.1 Nociones básicas de funciones

Nota 7.2.1. Principio de **extensionalidad para funciones**:

- $(\bigwedge x.\, f\, x = g\, x) \Longrightarrow f = g$ $\quad (ext)$

Nota 7.2.2. Actualización de funciones

- $(f(x := y))\; z = (if\; z = x\; then\; y\; else\; f\; z)$ $\quad (fun\_upd\_apply)$
- $f(x := y, x := z) = f(x := z)$ $\quad (fun\_upd\_upd)$

Nota 7.2.3. Función identidad

- $id \equiv \lambda x.\, x$ $\quad (id\_def)$

Nota 7.2.4. Composición de funciones:

- $f \circ g = (\lambda x.\, f\;(g\, x))$ $\quad (o\_def)$

Nota 7.2.5. Asociatividad de la composición:

- $f \circ (g \circ h) = (f \circ g) \circ h$ $\quad (o\_assoc)$

7.2.2 Funciones inyectivas, suprayectivas y biyectivas

Nota 7.2.6. Función **inyectiva** sobre A:

- $inj\text{-}on\; f\; A \equiv \forall\, x{\in}A.\; \forall\, y{\in}A.\; f\, x = f\, y \longrightarrow x = y$ $\quad (inj\_on\_def)$

Nota 7.2.7. $inj\; f$ es una abreviatura de $inj\text{-}on\; f\; UNIV$.

Nota 7.2.8. Función **suprayectiva**:

- $surj\; f \equiv \forall\, y.\; \exists\, x.\; y = f\, x$ $\quad (surj\_def)$

Nota 7.2.9. Función **biyectiva**:

- $bij\; f \equiv inj\; f \wedge surj\; f$ $\quad (bij\_def)$

Nota 7.2.10. Propiedades de las **funciones inversas**:

- $inj\; f \Longrightarrow inv\; f\; (f\, x) = x$ $\quad (inv\_f\_f)$
- $surj\; f \Longrightarrow f\; (inv\; f\; y) = y$ $\quad (surj\_f\_inv\_f)$

- $bij\,f \Longrightarrow inv\ (inv\,f) = f$ (*inv_inv_eq*)

Nota 7.2.11. Igualdad de funciones (por extensionalidad):

- $(f = g) = (\forall x.\, f\,x = g\,x)$ (*expand_fun_eq*)

Lema 7.2.12. *Una función inyectiva puede cancelarse en el lado izquierdo de la composición de funciones.*

lemma
 assumes $inj\,f$
 shows $(f \circ g = f \circ h) = (g = h)$
proof
 assume $f \circ g = f \circ h$
 thus $g = h$ **using** ⟨$inj\,f$⟩ **by** (*simp add:expand-fun-eq inj-on-def*)
next
 assume $g = h$
 thus $f \circ g = f \circ h$ **by** *auto*
qed

Una demostración más detallada es la siguiente

lemma
 assumes $inj\,f$
 shows $(f \circ g = f \circ h) = (g = h)$
proof
 assume $f \circ g = f \circ h$
 show $g = h$
 proof
  fix x
  have $(f \circ g)(x) = (f \circ h)(x)$ **using** ⟨$f \circ g = f \circ h$⟩ **by** *simp*
  hence $f(g(x)) = f(h(x))$ **by** *simp*
  thus $g(x) = h(x)$ **using** ⟨$inj\,f$⟩ **by** (*simp add:inj-on-def*)
 qed
next
 assume $g = h$
 show $f \circ g = f \circ h$
 proof
  fix x
  have $(f \circ g)\ x = f(g(x))$ **by** *simp*
  also have $\ldots = f(h(x))$ **using** ⟨$g = h$⟩ **by** *simp*
  also have $\ldots = (f \circ h)\ x$ **by** *simp*
  finally show $(f \circ g)\ x = (f \circ h)\ x$ **by** *simp*

qed
qed

Una demostración más automática es la siguiente

lemma
assumes *inj f*
shows $(f \circ g = f \circ h) = (g = h)$
by (*metis Un-UNIV-left assms id-o inj-iff inj-on-Un o-assoc*)

El desarrollo de la demostración automática es la siguiente

lemma
assumes *inj f*
shows $(f \circ g = f \circ h) = (g = h)$
proof (*neg-clausify*)
assume *0*: $(f \circ g \neq f \circ h) \vee (g \neq h)$
assume *1*: $(g = h) \vee (f \circ g = f \circ h)$
have *2*: $\bigwedge X1.$ *inj-on f X1*
by (*metis assms inj-on-Un Un-UNIV-left*)
have *3*: $(inv\, f \circ (f \circ g) = h) \vee h = g \vee \neg\, inj\, f$
by (*metis 1 o-assoc inj-iff id-o*)
have *4*: $(inv\, f \circ (f \circ g) = h) \vee h = g$ **by** (*metis 2 3*)
have *5*: $h = g \vee \neg\, inj\, f$ **by** (*metis id-o o-assoc inj-iff 4*)
have *6*: $h = g$ **by** (*metis 5 2*)
have *7*: $h \neq g$ **by** (*metis 6 0*)
show *False* **by** (*metis 6 7*)
qed

Función imagen

Nota 7.2.13. Imagen de un conjunto mediante una función:

- $f\ `\ A = \{y.\ (\exists x \in A.\ y = f\,x)$ (*image_def*)

Nota 7.2.14. Propiedades de la imagen:

- $(f \circ g)`r = f`g`r$ (*image_compose*)
- $f`(A \cup B) = f`A \cup f`B$ (*image_Un*)
- $inj\, f \Longrightarrow f`(A \cap B) = f`A \cap f`B$ (*image_Int*)

Nota 7.2.15. Ejemplos de demostraciones triviales de propiedades de la imagen.

lemma $f`A \cup g`A = (\bigcup x{\in}A.\ \{f\,x,\,g\,x\})$
by *auto*

lemma $f`\{(x,y).\ P\ x\ y\} = \{f(x,y)\ |\ x\ y.\ P\ x\ y\}$
by *auto*

Nota 7.2.16. El **rango** de una función (*range f*) es la imagen del universo ($f`UNIV$).

Nota 7.2.17. Imagen inversa de un conjunto:

- $f\ -`\ B \equiv \{x.\ f\,x : B\}$ (*vimage_def*)

Nota 7.2.18. Propiedad de la imagen inversa de un conjunto:

- $f\ -`\ (-A) = -(f\ -`\ A)$ (*vimage_Compl*)

7.3 Relaciones

7.3.1 Relaciones básicas

Nota 7.3.1. La teoría de relaciones es *HOL/Relation.thy*.

Nota 7.3.2. Las relaciones son conjuntos de pares.

Nota 7.3.3. Relación identidad:

- $Id \equiv \{p.\ EX\ x.\ p = (x,x)\}$ (*Id_def*)

Nota 7.3.4. Composición de relaciones:

- $r\ O\ s \equiv \{(x,z).\ EX\ y.\ (x,y) \in r\ \&\ (y,z) \in s\}$ (*rel_comp_def*)

Nota 7.3.5. Propiedades:

- $R\ O\ Id = R$ (*R_O_Id*)
- $[\![r' \subseteq r;\ s' \subseteq s]\!] \Longrightarrow (r'\ O\ s') \subseteq (r\ O\ s)$ (*rel_comp_mono*)

Nota 7.3.6. Imagen inversa de una relación:

- $((a,b) \in r^{-1}) = ((b,a) \in r)$ (*converse_iff*)

Nota 7.3.7. Propiedad de la imagen inversa de una relación:

- $(r\ O\ s)^{-1} = s^{-1}\ O\ r^{-1}$ (*converse_rel_comp*)

Nota 7.3.8. Imagen de un conjunto mediante una relación:

- $(b \in r``A) = (\exists\, x{:}A.\ (x, b) \in r)$ (*Image_iff*)

Nota 7.3.9. Dominio de una relación:

- $(a \in Domain\ r) = (\exists\, y.\ (a, y) \in r)$ (*Domain_iff*)

Nota 7.3.10. Rango de una relación:

- $(a \in Range\ r) = (\exists\, y.\ (y, a) \in r)$ (*Range_iff*)

7.3.2 Clausura reflexiva y transitiva

Nota 7.3.11. La teoría de la clausura reflexiva y transitiva de una relación es *HOL/Transitive-Closure.thy*.

Nota 7.3.12. Potencias de relaciones:

- $R\ \hat{}\hat{}\ 0 = Id$
- $R\ \hat{}\hat{}\ (Suc\ n) = (R\ \hat{}\hat{}\ n)\ O\ R$

Nota 7.3.13. La **clausura reflexiva y transitiva** de la relación r es la menor solución de la ecuación:

- $r^* = Id \cup (r^*\ O\ r)$ (*rtrancl_unfold*)

Nota 7.3.14. Propiedades básicas de la clausura reflexiva y transitiva:

- $(a, a) \in r^*$ (*rtrancl_refl*)
- $p \in r \Longrightarrow p \in r^*$ (*r_into_rtrancl*)
- $[\![(a, b) \in r^*; (b, c) \in r^*]\!] \Longrightarrow (a, c) \in r^*$ (*rtrancl_trans*)

Nota 7.3.15. Inducción sobre la clausura reflexiva y transitiva

- $$\frac{(a, b) \in r^* \qquad P\ a \qquad \bigwedge y\ z.\ \dfrac{(a, y) \in r^* \qquad (y, z) \in r \qquad P\ y}{P\ z}}{P\ b}$$ (*rtrancl_induct*)

Nota 7.3.16. Idempotencia de la clausura reflexiva y transitiva:

- $(r^*)^* = r^*$ (*rtrancl_idemp*)

Nota 7.3.17. Reglas de introducción de la **clausura transitiva**:

- $p \in r \Longrightarrow p \in r^+$ (*r_into_trancl'*)
- $[\![(a, b) \in r^+; (b, c) \in r^+]\!] \Longrightarrow (a, c) \in r^+$ (*trancl_trans*)

Nota 7.3.18. Ejemplo de propiedad:

- $(r^{-1})^+ = (r^+)^{-1}$ (*trancl_converse*)

7.3.3 Una demostración elemental

Nota 7.3.19. El teorema que se desea demostrar es que la clausura reflexiva y transitiva conmuta con la inversa (*rtrancl-converse*). Para demostrarlo introducimos dos lemas auxiliares: *rtrancl-converseD* y *rtrancl-converseI*.

lemma *rtrancl-converseD*: $(x,y) \in (r^{-1})^* \Longrightarrow (y,x) \in r^*$
proof (*induct rule*:*rtrancl-induct*)
 show $(x,x) \in r^*$ **by** (*rule rtrancl-refl*)
next
 fix *y z*
 assume $(x,y) \in (r^{-1})^*$ **and** $(y,z) \in r^{-1}$ **and** $(y,x) \in r^*$
 show $(z,x) \in r^*$
 proof (*rule rtrancl-trans*)
  show $(z,y) \in r^*$ **using** $\langle(y,z) \in r^{-1}\rangle$ **by** *simp*
 next
  show $(y,x) \in r^*$ **using** $\langle(y,x) \in r^*\rangle$ **by** *simp*
 qed
qed

lemma *rtrancl-converseI*: $(y,x) \in r^* \Longrightarrow (x,y) \in (r^{-1})^*$
proof (*induct rule*:*rtrancl-induct*)
 show $(y,y) \in (r^{-1})^*$ **by** (*rule rtrancl-refl*)
next
 fix *u z*
 assume $(y,u) \in r^*$ **and** $(u,z) \in r$ **and** $(u,y) \in (r^{-1})^*$
 show $(z,y) \in (r^{-1})^*$
 proof (*rule rtrancl-trans*)
  show $(z,u) \in (r^{-1})^*$ **using** $\langle(u,z) \in r\rangle$ **by** *auto*

next
show $(u,y) \in (r^{-1})^*$ **using** $\langle (u,y) \in (r^{-1})^* \rangle$ **by** *simp*
qed
qed

theorem *rtrancl-converse*: $(r^{-1})^* = (r^*)^{-1}$
proof
show $(r^{-1})^* \subseteq (r^*)^{-1}$ **by** (*auto simp add*:*rtrancl-converseD*)
next
show $(r^*)^{-1} \subseteq (r^{-1})^*$ **by** (*auto simp add*:*rtrancl-converseI*)
qed

Nota 7.3.20. Puede demostrarse de manera más corta como sigue:

theorem $(r^{-1})^* = (r^*)^{-1}$
by (*auto intro*: *rtrancl-converseI dest*: *rtrancl-converseD*)

7.4 Relaciones bien fundamentadas e inducción

Nota 7.4.1. La teoría de las relaciones bien fundamentadas es *HOL/Wellfounded-Relations.thy*.

Nota 7.4.2. La relación–objeto *less-than* es el orden de los naturales que es bien fundamentada:

- $((x,y) \in \textit{less-than}) = (x < y)$ (*less_than_iff*)
- *wf less-than* (*wf_less_than*)

Nota 7.4.3. Notas sobre **medidas**:

- **Imagen inversa** de una relación mediante una función:
  $\textit{inv-image}\ r\ f \equiv \{(x,y).\ (f\,x, f\,y) \in r$ (*inv-image-def*)
- Conservación de la buena fundamentación:
  $\textit{wf}\ r \Longrightarrow \textit{wf}\ (\textit{inv-image}\ r\ f)$ (*wf-inv-image*)
- Definición de la **medida**:
  $\textit{measure} \equiv \textit{inv-image less-than}$ (*measure-def*)
- Buena fundamentación de la medida:
  $\textit{wf}\ (\textit{measure}\ f)$ (*wf-measure*)

Nota 7.4.4. Notas sobre el **producto lexicográfico**:

- Definición del producto lexicográfico (*lex-prod-def*):
  $ra <\!*lex*\!> rb \equiv \{((a,b),(a',b')).\ (a,a') \in ra \lor (a = a' \land (b,b') \in rb)\}$

- Conservación de la buena fundamentación:
  $[\![wf\ ra; wf\ rb]\!] \Longrightarrow wf\ \ (ra <\!*lex*\!> rb)$ (*wf-lex-prod*)

Nota 7.4.5. El orden de multiconjuntos está en la teoría *HOL/Library/Multiset.thy*.

Nota 7.4.6. Inducción sobre relaciones bien fundamentadas:

- $$\frac{wf\ r \qquad \bigwedge x.\ \dfrac{\forall y.\ (y,x) \in r \longrightarrow P\ y}{P\ x}}{P\ a}$$ (*wf-induct*)