Correspondence between names in the paper *Permutations in Coinductive Graph Representation* and names in Coq scripts

Celia Picard

February 2012

In this document you will find the list of correspondences between the definitions and lemmas given in the paper and the names used in the Coq scripts. Thoses names are given with the file where they are to be found and the corresponding line.

In the Paper			**In the scripts**		
Section	**Reference**	**Name**	**Name**	**File**	**Line**
1.1	Definition 1	*Fin*	`Fin`	Fin.v	32
1.1	Definition 2	*ilistn*	`ilistn`	Ilist.v	111
1.1	Definition 3	*ilist*	`ilist`	Ilist.v	118
1.1		*lg*	`lgti`	Ilist.v	125
1.1		*fct*	`fcti`	Ilist.v	127
1.1		*ilist* in pointwise bijection with lists (direction $\Rightarrow$)	`ilist2list_list2ilist_id`	Ilist.v	506
1.1		*ilist* in pointwise bijection with lists (direction $\Leftarrow$)	`list2ilist_ilist2list_id`	Ilist.v	459
1.1		*ilist2list*	`ilist2list`	Ilist.v	259
1.1		*list2ilist*	`list2ilist`	Ilist.v	360
1.1	Definition 4	*ilist_rel*	`ilist_rel`	Ilist.v	153
1.1	Definition 5	*Graph*	`Graph`	Graphs.v	32
1.1		*label*	`label`	Graphs.v	37
1.1		*sons*	`sons`	Graphs.v	38
1.1	Definition 6	*Geq*	`Geq`	Graphs.v	54
2.1		*remEl*	`extroduce`	Extroduce.v	24
2.1	(1)	Assertion (1) about *remEl*	`extroduce_lgti_S`	Extroduce.v	377
2.1	(2)	Assertion (2) about *remEl*	`extroduce_ok2'`	Extroduce.v	205
2.1	(3)	Assertion (3) about *remEl*	`extroduce_ok3'`	Extroduce.v	230
2.1		*weakFin*	`weakFin`	Extroduce.v	181
2.1		$\forall\ i, weakFin\ i =_{Fin} i$	`weakFin_ok`	Extroduce.v	185
2.1	Definition 7	*iperm*	`IlistPerm3`	IlistPerm.v	228
2.1	Definition 8	$iperm'$	`IlistPerm4`	IlistPerm.v	235
2.1		*lg* for *iperm*	`IlistPerm3_lgti`	IlistPerm.v	289
2.1	Theorem 1	$iperm' \Rightarrow iperm$	`IlistPerm4_IlistPerm3_eq`	IlistPerm.v	301
2.1	Theorem 1	$iperm \Rightarrow iperm'$	`IlistPerm3_IlistPerm4_eq`	IlistPerm.v	933
2.1	In proof of Thm. 1		`IlistPerm3_exists_rec`	IlistPerm.v	846

In the Paper			In the scripts		
Section	**Reference**	**Name**	**Name**	**File**	**Line**
2.1		*iperm* preserves reflexivity	`IlistPerm3_refl_refl`	IlistPerm.v	376
2.1		*iperm′* preserves reflexivity	`IlistPerm4_refl_refl`	IlistPerm.v	337
2.1		*iperm* preserves symmetry	`IlistPerm3_sym_sym`	IlistPerm.v	459
2.1		*iperm′* preserves transitivity	`IlistPerm4_trans_trans`	IlistPerm.v	512
2.1		*iperm* preserves equivalence	`IlistPerm3Rel`	IlistPerm.v	984
2.1		*iperm′* preserves equivalence	`IlistPerm4Rel`	IlistPerm.v	1008
2.1	Lemma 1	*iperm* decidable	`IlistPerm3_dec`	IlistPerm.v	1305
2.1	Lemma 1	*iperm′* decidable	`IlistPerm4_dec`	IlistPerm.v	1228
2.2	Definition 9	*skel_type*	`IlistPerm3Cert_list`	IlistPerm.v	1317
2.2	Definition 10	*iperm_skel*	`IlistPerm3Cert`	IlistPerm.v	1418
2.2	Lemma 2	*iperm* ⇔ *iperm_skel* (direction ⇒)	`IlistPerm3_IlistPerm3Cert`	IlistPerm.v	1437
2.2	Lemma 2	*iperm* ⇔ *iperm_skel* (direction ⇐)	`IlistPerm3Cert_IlistPerm3`	IlistPerm.v	1428
2.2		*iperm_skel* monotone in its base relation	`IlistPerm3Cert_mon`	IlistPerm.v	1450
2.2		*iperm* monotone in its base relation	`IlistPerm3_mon`	IlistPerm.v	1055
2.2	Lemma 3	Intersections in the relation argument of *iperm_skel*	`IlistPerm3Cert_inter`	IlistPerm.v	1475
2.3	Definition 11	*bij*	`Bijective`	IlistPerm.v	1791
2.3	Definition 12	*ipermb*	`IlistPerm7`	IlistPerm.v	1801
2.3		*ipermb* preserves equivalence	`IlistPerm7Rel`	IlistPerm.v	1836
2.3		*ipermb* is monotone in its base relation	`IlistPerm7_mon`	IlistPerm.v	1842
2.3	Lemma 4	*lg* for *ipermb*	`IlistPerm7_lgti`	IlistPerm.v	1851
2.4	Theorem 2	*iperm* ⇔ *ipermb* (direction ⇒)	`IlistPerm3_IlistPerm7`	IlistPerm.v	1865
2.4	Theorem 2	*iperm* ⇔ *ipermb* (direction ⇐)	`IlistPerm7_IlistPerm3`	IlistPerm.v	1877
2.4	Definition 13	*permcont*	`permut`	EquivContejean.v	29
2.4	Theorem 3	*permcont* ⇒ *iperm*	`equiv_list_permut_IlistPerm3`	EquivContejean.v	34
2.4	Theorem 4	*iperm* ⇒ *permcont*	`equiv_ilist_IlistPerm3_permut`	EquivContejean.v	91
2.5		*ipermcont* ⇒ *iperm*	`equiv_ilist_permut_IlistPerm3`	EquivContejean.v	133
3	Definition 16	*GPerm*	`GeqPerm0`	GPerm.v	310
3.2		*GPerm* (Impredicative)	`GeqPerm`	GPerm.v	315
3.2		*GPerm* (Mendler-style)	`GeqPerm1`	GPerm.v	321
3.2	Lemma 6	Coinduction principle for *GPerm*	`GeqPerm_coind`	GPerm.v	325
3.2	Lemma 7	Unfolding principle for *GPerm*	`GeqPerm_out`	GPerm.v	335
3.2	Lemma 8	Constructor for *GPerm*	`GeqPerm_intro`	GPerm.v	347
3.2		*GPerm* (impredicative) subrelation *GPerm* (Mendler-style)	`GeqPerm_GeqPerm1`	GPerm.v	389
3.2		*GPerm* (Mendler-style) subrelation *GPerm* (impredicative)	`GeqPerm1_GeqPerm`	GPerm.v	381
3.3	Lemma 5	*GPerm* preserves reflexivity	`GeqPerm_refl`	GPerm.v	660
3.3		*GPerm* preserves symmetry	`GeqPerm_sym`	GPerm.v	682
3.3		*GPerm* preserves transitivity	`GeqPerm_trans`	GPerm.v	708
3.3		*GPerm* preserves equivalence	`GeqPermRel`	GPerm.v	734
4.1	Definition 16	*iTree*	`TreeG`	GPerm.v	30
4.1		*labT*	`labelT`	GPerm.v	33
4.1		*sonsT*	`sonsT`	GPerm.v	34

In the Paper			In the scripts		
Section	**Reference**	**Name**	**Name**	**File**	**Line**
4.1	Definition 17	*TPerm*	`TeqPerm`	GPerm.v	159
4.1		*TPerm* preserves equivalence	`TeqPermRel`	GPerm.v	223
4.1	Definition 18	*G2iT*	`Graph2TreeG`	GPerm.v	118
4.1		*imap*	`imap`	Ilist.v	139
4.1	Definition 19	$\equiv$	`TeqPermn`	GPerm.v	398
4.1	Lemma 9	First assertion	`TeqPermn_0`	GPerm.v	421
4.1	Lemma 9	Second assertion Dir. $\Rightarrow$	`TeqPermn_Sn_back`	GPerm.v	449
4.1	Lemma 9	Second assertion Dir. $\Leftarrow$	`TeqPermn_Sn`	GPerm.v	436
4.1		$\equiv$ preserves decidability	`TeqPermn_dec`	GPerm.v	491
4.1	Definition 20	*GTPerm*	`TeqPerm_gene`	GPerm.v	519
4.1		*GTPerm* preserves equivalence	`TeqPerm_geneRel`	GPerm.v	541
4.2	Theorem 5 Dir. $\Rightarrow$	*GPerm* $\Rightarrow$ *GTPerm*	`GeqPerm_TeqPerm`	GPerm.v	561
4.2	Theorem 5 Dir. $\Leftarrow$	*GTPerm* $\Rightarrow$ *GPerm*	`TeqPerm_GeqPerm`	GPerm.v	633
4.2	Lemma 10	*GTPerm* $\Rightarrow$ *iperm*	`TeqPerm_gene_IlistPerm3`	GPerm.v	595
4.2	Lemma 11	Antitone on $\equiv$	`TeqPermn_antitone`	GPerm.v	482
4.2	Axiom 1	Infinite Pigeonhole Principle (IPP)	`IPPIlistPerm3Cert`	GPerm.v	586
4.2		Justification of IPP with Double Negation Elimination	`IPPJustification`	IPPJustification.v	546
4.2		Justification of IPP with excluded middle	`IPPJustification'`	IPPJustification.v	561
5.1		"Old" notion of strict inclusion *GinG*	`Graph_in_Graph`	Graphs.v	162
5.1	Definition 21	$GinG^*$	`Graph_in_Graph_Gene`	Graphs.v	225
5.1	Definition 22	*GinGP*	`Graph_in_Graph_Perm`	GPerm.v	743
5.1		*GinGP* preserves transitivity	`Graph_in_Graph_Perm_trans`	GPerm.v	745
5.1		*GinGP* compatible with *GPerm* in its first argument	`Graph_in_Graph_PermPM`	GPerm.v	774
5.1		*GinGP* compatible with *GPerm* in its second arguments	`Graph_in_Graph_PermM2`	GPerm.v	793
5.2		$GinG^*$ preserves reflexivity	`Graph_in_Graph_Gene_refl`	Graphs.v	242
5.2	Definition 23	*GeqPerm*	`GPPerm`	GPerm.v	807
5.2		*GeqPerm* preserves equivalence	`GPPermRel`	GPerm.v	831
5.2		Graphs of Figure 1 are equivalent	`GPPerm_g012_g021`	GPerm.v	916
5.2		Graphs of Figure 2 are equivalent	`GPPerm_G01_G10`	GPerm.v	844
5.2		Graphs of Figure 4 are not equivalent	`not_GPPerm_g3_g4`	GPerm.v	943