finite-partial-functions^{0,22}

ABS: $a{:}A$ fp→ $B(a)$ **fpf**

STM: fpf_wf

STM: subtype-fpf

STM: subtype-fpf2

STM: subtype-fpf3

ABS: $x \in$ dom(f) **fpf-dom**

STM: fpf-dom_wf

ABS: fpf-domain(f) **fpf-domain**

STM: fpf-domain_wf

STM: member-fpf-domain

STM: fpf-trivial-subtype-set

STM: fpf-trivial-subtype-top

STM: fpf-dom_functionality

STM: fpf-dom_functionality2

STM: fpf-dom-type

STM: fpf-dom-type2

ABS: **fpf-empty**

STM: fpf-empty_wf

ABS: fpf-is-empty(f) **fpf-is-empty**

STM: fpf-is-empty_wf

STM: assert-fpf-is-empty

ABS: $f(x)$ **fpf-ap**

STM: fpf-ap_wf

STM: fpf-ap_functionality

ABS: $f(x)?z$ **fpf-cap**

STM: fpf-cap-wf-univ

STM: fpf-cap_wf

ABS: $z \mathrel{!=} f(x) \Rightarrow P(a;z)$ **fpf-val**

STM: fpf-val_wf

ABS: $f \subseteq g$ **fpf-sub**

STM: fpf-sub_wf

STM: sq_stable__fpf-sub

STM: fpf-empty-sub

STM: fpf-sub-functionality

STM: fpf-sub-functionality2

STM: fpf-sub_functionality

STM: fpf-sub_functionality2

STM: fpf-sub_transitivity

STM: fpf-sub_weakening

STM: subtype-fpf-cap

STM: subtype-fpf-cap-top

STM: fpf-cap-void-subtype

STM: subtype-fpf-cap-void

STM: fpf-cap_functionality

STM: fpf-cap-subtype_functionality

STM: fpf-cap_functionality_wrt_sub

STM: fpf-cap-subtype_functionality_wrt_sub

STM: fpf-cap-subtype_functionality_wrt_sub2

ABS: $f \parallel g$ **fpf-compatible**

STM: fpf-compatible_wf

STM: fpf-compatible-wf2

STM: fpf-sub-compatible-left

STM: fpf-sub-compatible-right

STM: subtype-fpf-cap5

STM: subtype-fpf-cap-void2

STM: subtype-fpf-cap-void-list

STM: fpf-cap-compatible

ABS: $f \oplus g$ **fpf-join**

STM: fpf-join_wf

STM: fpf-join-wf

STM: fpf-join-empty

STM: fpf-empty-join

STM: fpf-join-empty-sq

STM: fpf-join-idempotent

STM: fpf-join-assoc

STM: fpf-join-dom

STM: fpf-join-dom2

STM: fpf-join-dom-sq

STM: fpf-domain-join

STM: fpf-join-is-empty

STM: fpf-join-ap

STM: fpf-join-ap-left

STM: fpf-join-ap-sq

STM: fpf-join-cap-sq

STM: fpf-join-cap

STM: fpf-join-range

STM: fpf-sub-join-left

STM: fpf-sub-join-left2

STM: fpf-sub-join-right

STM: fpf-sub-join-right2

STM: fpf-sub-join

STM: fpf-join-sub

STM: fpf-join-sub2

STM: fpf-sub-join-symmetry

STM: fpf-sub-val

STM: fpf-sub-val2

STM: fpf-sub-val3

ABS: $L \vdash$fpf$\rightarrow v$ **fpf-const**

STM: fpf-const_wf

STM: fpf-const-dom

ABS: $x : v$ **fpf-single**

STM: fpf-single_wf

STM: fpf-single_wf2

STM: fpf-single-sub-reflexive

STM: fpf-cap-single1

STM: fpf-cap-single-join

STM: fpf-ap-single

STM: fpf-cap-single

STM: fpf-val-single1

ABS: $f x : v$ **fpf-add-single**

STM: fpf-add-single_wf

ABS: fpf-vals(eq;P;f) **fpf-vals**

STM: fpf-vals_wf

STM: member-fpf-vals

STM: member-fpf-vals2

STM: filter-fpf-vals

STM: fpf-vals-singleton

STM: fpf-vals-nil

ABS: $\forall x{\in}\mathrm{dom}(f).\ v{=}f(x) \Rightarrow P(x;v)$ **fpf-all**

STM: fpf-all_wf

ABS: fpf-map(a,v.f(a;v);x) **fpf-map**

STM: fpf-map_wf

ABS: fpf-accum(z,a,v.f(z;a;v);y;x) **fpf-accum**

STM: fpf-accum_wf

ABS: rename(r;f) **fpf-rename**

STM: fpf-rename_wf

STM: fpf-rename-dom

STM: fpf-rename-dom2

STM: fpf-rename-ap

STM: fpf-rename-ap2

STM: fpf-rename-cap

STM: fpf-rename-cap2

STM: fpf-rename-cap3

ABS: fpf-inv-rename(r;$rinv$;f) **fpf-inv-rename**

STM: fpf-inv-rename_wf

ABS: g o f **fpf-compose**

STM: fpf-compose_wf

ABS: compose-fpf(a;b;f) **compose-fpf**

STM: compose-fpf_wf

STM: compose-fpf-dom

STM: fpf-sub-reflexive

ABS: mkfpf(a;b) **mkfpf**

STM: mkfpf_wf

STM: fpf-join-compatible-left

STM: fpf-join-compatible-right

STM: fpf-compatible-self

STM: fpf-compatible-join

STM: fpf-compatible-join-iff

STM: fpf-compatible-symmetry

STM: fpf-disjoint-compatible

STM: fpf-compatible-join2

STM: fpf-compatible-singles

STM: fpf-compatible-singles-trivial

STM: fpf-single-dom

STM: fpf-single-dom-sq

STM: fpf-compatible-single

STM: fpf-compatible-single-iff

STM: fpf-compatible-single2

STM: fpf-compatible-singles-iff

STM: fpf-compatible-join-cap

STM: fpf-ap-equal

STM: fpf-join-dom-decl

STM: fpf-join-dom-da

STM: fpf-cap-join-subtype

STM: fpf-cap-join-subtype2

STM: fpf-all-empty

STM: fpf-all-single

STM: fpf-all-single-decl

STM: fpf-all-join-decl

ABS: non-void(d) **non-void-decl**

STM: non-void-decl_wf

STM: non-void-decl-join

STM: non-void-decl-single

ABS: AtomFree(d) **atom-free-decl**

STM: atom-free-decl_wf

STM: atom-free-decl-join

STM: atom-free-decl-single

STM: fpf-empty-compatible-right

STM: fpf-empty-compatible-left

STM: fpf-compatible-triple

ABS: fpf-dom-list(f) **fpf-dom-list**

STM: fpf-dom-list_wf

STM: member-fpf-dom

ABS: lnk-decl(l;dt) **lnk-decl**

STM: lnk-decl_wf

STM: lnk-decl-cap

STM: lnk-decl-dom

STM: lnk-decl-dom-not

STM: lnk-decl-dom2

STM: lnk-decl-cap2

STM: lnk-decl-ap

STM: lnk-decl-dom-implies

STM: lnk-decl-compatible-single

STM: lnk-decl-compatible-single2

STM: lnk-decls-compatible

STM: l_disjoint-fpf-dom

STM: l_disjoint-fpf-join-dom

ABS: fpf(L) **pairs-fpf**

STM: pairs-fpf_wf

STM: pairs-fpf_property

STM: no_repeats-pairs-fpf

STM: atom-free-fpf

ABS: fpf-normalize(eq;g) **fpf-normalize**

STM: fpf-normalize_wf

STM: fpf-normalize-dom

STM: fpf-normalize-ap

ABS: Valtype(da;k) **ma-valtype**

STM: ma-valtype_wf

ABS: Msgtype(da;k) **ma-msgtype**

STM: ma-msgtype_wf

ABS: State(ds) **ma-state**

STM: ma-state_wf

STM: ma-valtype-subtype

STM: ma-state-subtype

STM: ma-state-subtype2

ABS: dt(l;da) **es-dt**

STM: es-dt_wf

STM: es-dt-dom

STM: es-dt-ap

STM: es-dt-cap

ABS: Normal(T) **normal-type**

STM: normal-type_wf

STM: normal-top

ABS: Normal(ds) **normal-ds**

STM: normal-ds_wf

STM: implies-normal-ds

STM: normal-ds-single

STM: normal-ds-join

ABS: Normal(da) **normal-da**

STM: normal-da_wf

STM: normal-da-single

STM: normal-da-join

STM: normal-valtype

STM: normal-cap-void

STM: normal-es-dt