

components^{11,40}

STM: l_member_type

STM: l_member_type2

STM: l_member_in_subtype

STM: l_member_in_subtype2

ABS: Namer($n; Id_list$) **Namer**

STM: Namer_wf

STM: Namer-subtype

ABS: namer-shift($n; namer$) **namer-shift**

STM: namer-shift_wf

ABS: namer-disjoint($n_1; n_2; nmr_1; nmr_2$) **namer-disjoint**

STM: namer-disjoint_wf

STM: namer-disjoint-shift

ABS: RealizerScheme{i:l}() **RealizerScheme**

STM: RealizerScheme_wf

ABS: scheme-plus($A; B$) **scheme-plus**

STM: scheme-plus_wf

ABS: $S \dashv\!\!-\!es.P(es)$ **scheme-realizes**

STM: scheme-realizes_wf

ABS: scheme-compatible($A; B$) **scheme-compatible**

STM: scheme-compatible_wf

STM: scheme-and-rule

STM: scheme-implies-rule

ABS: Interface($ds; da; A$) **interface**

STM: interface_wf

ABS: MaInterface(T) **ma-interface**

STM: ma-interface_wf
 STM: ma-interface-type-trivial
 STM: ma-interface-apply-type
 STM: ma-interface-da-type0
 STM: ma-interface-da-type1
 STM: ma-interface-da-type2
 ABS: ma-interface-locs(I) **ma-interface-locs**
 ABS: ma-interface-loc($I;i$) **ma-interface-loc**
 STM: ma-interface-locs_wf
 ABS: ma-interface-ds($I;i$) **ma-interface-ds**
 STM: ma-interface-ds_wf
 ABS: ma-interface-dom($I;i$) **ma-interface-dom**
 ABS: ma-interface-domb($I;i;k$) **ma-interface-domb**
 STM: ma-interface-domb_wf
 STM: ma-interface-dom_wf
 STM: assert-ma-interface-domb
 STM: ma-interface-dom-hasloc
 STM: ma-interface-loc_wf
 STM: assert-ma-interface-loc
 ABS: ma-interface-non-degenerate(I) **ma-interface-non-degenerate**
 STM: ma-interface-non-degenerate_wf
 ABS: ma-interface-info($I;i;k$) **ma-interface-info**
 STM: ma-interface-info_wf
 ABS: ma-interface-valtype($I;i;k$) **ma-interface-valtype**
 STM: ma-interface-valtype_wf
 ABS: ma-interface-code($I;i;k$) **ma-interface-code**
 STM: ma-interface-code_wf

ABS: $\text{ma-interface-msgs}(I;i;k)$ **ma-interface-msgs**
 STM: $\text{ma-interface-msgs_wf}$
 ABS: $\text{ma-interface-conds}(I;i)$ **ma-interface-conds**
 STM: $\text{ma-interface-conds_wf}$
 STM: $\text{ma-interface-conds_wf2}$
 STM: $\text{ma-interface-conds_wf3}$
 STM: $\text{ma-interface-conds-equals}$
 STM: $\text{l_member-subtype_rel}$
 STM: $\text{contravariance-general}$
 STM: $\text{contravariance-variant}$
 STM: $\text{ma-interface-da-type3}$
 ABS: $\text{Normal}(da)$ **normal-ma-da**
 STM: normal-ma-da_wf
 ABS: $\text{Normal}(A,I)$ **ma-interface-normal**
 STM: $\text{ma-interface-normal_wf}$
 STM: $\text{ma-interface-subtype}$
 ABS: $\text{ma-interface-consistent-at}(es;i;X)$ **ma-interface-consistent-at**
 STM: $\text{ma-interface-consistent-at_wf}$
 ABS: $\text{ma-interface-consistent}(es;X)$ **ma-interface-consistent**
 STM: $\text{ma-interface-consistent_wf}$
 ABS: $\text{ma-interface-consistent2}(es;I)$ **ma-interface-consistent2**
 STM: $\text{ma-interface-consistent2_wf}$
 STM: $\text{ma-interface-consistent-consistent2}$
 ABS: $\text{ma-trivial-interface}(i;k;V;f)$ **ma-trivial-interface**
 STM: $\text{ma-trivial-interface_wf}$
 STM: interface-subtype
 ABS: $\text{in-interface}(es;X;e)$ **in-interface**

STM: in-interface_wf
 ABS: ma-in-interface($es;X;e$) **ma-in-interface**
 STM: ma-in-interface_wf
 STM: ma-in-interface-loc
 STM: assert-ma-in-interface
 ABS: interface-val($es;X;e$) **interface-val**
 STM: interface-val_wf
 ABS: ma-interface-val($es;X;e$) **ma-interface-val**
 STM: ma-interface-val_wf
 ABS: Component($ds;da;A;B$) **component**
 STM: component_wf
 STM: component-subtype
 ABS: $[[X]]$ **abs-interface**
 STM: abs-interface_wf
 ABS: $[[X]]$ **ma-abs-interface**
 STM: ma-abs-interface_wf
 ABS: ma-interface-glued-p($es;A;I;l;tg$) **ma-interface-glued-p**
 STM: ma-interface-glued-p_wf
 STM: ma-abs-interface-loc
 STM: local-finite-interface-to-ma-interface
 ABS: ma-interface-compose($g;X$) **ma-interface-compose**
 STM: ma-interface-compose_wf
 STM: ma-in-interface-compose
 STM: ma-interface-consistent-at-compose
 STM: ma-interface-consistent-compose
 STM: ma-abs-interface-compose
 ABS: ma-interface-left(X) **ma-interface-left**

STM: ma-interface-left_wf
 ABS: ma-interface-right(X) **ma-interface-right**
 STM: ma-interface-right_wf
 ABS: ma-interface-inl(X) **ma-interface-inl**
 STM: ma-interface-inl_wf
 ABS: ma-interface-inr(X) **ma-interface-inr**
 STM: ma-interface-inr_wf
 STM: ma-abs-interface-left
 STM: ma-abs-interface-right
 ABS: interface-left(X) **interface-left**
 STM: interface-left_wf
 ABS: interface-right(X) **interface-right**
 STM: interface-right_wf
 ABS: interface-inl(X) **interface-inl**
 STM: interface-inl_wf
 ABS: interface-inr(X) **interface-inr**
 STM: interface-inr_wf
 ABS: interface-union($X;Y$) **interface-union**
 STM: interface-union_wf
 STM: interface-union-dom
 STM: interface-union-ap
 ABS: mux-component($Ca;Cb$) **mux-component**
 STM: mux-component_wf
 STM: abs-interface-left
 STM: abs-interface-right
 STM: union-interface-right
 STM: union-interface-left

ABS: $C \vdash_{es, in, out}. P(es; in; out)$ **component-realizes**
 STM: component-realizes_wf
 ABS: component-compatible($ds; da; T_1; T_2; C_1; C_2$) **component-compatible**
 STM: component-compatible_wf
 ABS: component-output-disjoint $\{i:l\}(ds; da; T_1; T_2; C_1; C_2)$ **component-output-disjoint**
 STM: component-output-disjoint_wf
 STM: mux-component-property
 ABS: scheme-constant(R) **scheme-constant**
 STM: scheme-constant_wf
 ABS: scheme-none() **scheme-none**
 STM: scheme-none_wf
 ABS: interface-compose($f; X$) **interface-compose**
 STM: interface-compose_wf
 STM: abs-interface-compose
 STM: is-interface-compose
 STM: interface-compose_val
 ABS: trivial-component(f) **trivial-component**
 STM: trivial-component_wf
 STM: trivial-component-property
 ABS: gluable($I; l; tg$) **gluable**
 STM: gluable_wf
 ABS: gluable2($A; I; l; tg$) **gluable2**
 STM: gluable2_wf
 STM: ma-interface-kinds-aux0
 STM: ma-interface-kinds-aux1
 STM: ma-interface-kinds-aux2
 ABS: ma-interface-kinds(I) **ma-interface-kinds**

STM: ma-interface-kinds_wf
 STM: ma-interface-kinds-property
 ABS: ma-interface-tags(I) **ma-interface-tags**
 STM: ma-interface-tags_wf
 STM: ma-interface-tags-property
 STM: ma-interface-tags-property2
 STM: link-trivia
 STM: l_member-trivia
 STM: map_wf2
 STM: map_wf3
 ABS: $[[I|i]]$ **ma-interface-triggers**
 STM: ma-interface-triggers_wf
 STM: ma-interface-triggers-val
 STM: ma-interface-triggers-loc
 STM: ma-interface-triggers-kind
 STM: ma-interface-triggers-can-apply
 STM: es-is-interface-ma-interface-triggers
 STM: ma-interface-triggers-do-apply
 STM: ma-interface-triggers-glued
 ABS: $[[I|\forall]]$ **ma-interface-triggers-list**
 STM: ma-interface-triggers-list_wf
 STM: ma-interface-triggers-list-disjoint
 STM: ma-interface-triggers-list-p-first
 ABS: trigger-send($A; ds; x; cond; l; tg$) **trigger-send**
 STM: trigger-send_wf
 ABS: triggersGlue($A; l; tg; ds; conds$) **triggersGlue**
 STM: triggersGlue_wf

STM: triggersGlue_feasible
STM: triggersGlue_feasible2
STM: triggersGlue_has-loc
STM: triggersGlue_compatible
STM: triggersGlue_compatible2
STM: sender-glues-triggers-realizable
STM: triggers-glued-realizable
ABS: interfaceGlue($A; I; l; tg; nmr$) **interfaceGlue**
STM: interfaceGlue_helper0
STM: interfaceGlue_helper0.5
STM: interfaceGlue_helper0.6
STM: interfaceGlue_helper
STM: interfaceGlue_helper2
STM: interfaceGlue_wf
STM: interfaceGlue_feasible
STM: ma-interface-glued-p-realizable