unnecessary^{0,22}

STM: w-kindtype_wf

ABS: R-pre-init(i;ds;$init$;a;T;P) **R-pre-init**

STM: R-pre-init_wf

STM: R-pre-init-feasible

ABS: R-pre-init1(i;x;A;x_0;a;T;P) **R-pre-init1**

STM: R-pre-init1_wf

STM: R-pre-init1-feasible

ABS: MsgFrom(i) **w-Msg-from**

STM: w-Msg-from_wf

STM: better-w-m-wf

ABS: (x initially i) **w-initially**

STM: w-initially_wf

ABS: first(e) **w-first**

STM: w-first-aux

STM: w-first_wf

STM: assert-w-first

STM: w-loc-time

STM: better-w-sends-wf

STM: w-causl-time-iff

STM: decl-rename-cap

ABS: action-rename($rainv$;$rtinv$;a) **action-rename**

STM: action-rename_wf

ABS: world-rename(rx;ra;rt;$rainv$;$rtinv$;w) **world-rename**

STM: world-rename_wf

STM: es-kindtype-w-valtype

ABS: d(e;e') **w-d**

STM: w-d_wf

STM: w-d-properties

ABS: Kind(da) **ma-kind**

STM: ma-kind_wf

ABS: MsgA(ds;da) **msga-body**

STM: msga-body_wf

STM: msga-factor

ABS: Shape(M) **ma-shape**

STM: ma-shape_wf

ABS: ma-body(M) **ma-body**

STM: ma-body_wf

STM: msga-subtype

ABS: M.X **ma-X**

STM: ma-X_wf

ABS: M.A **ma-A**

STM: ma-A_wf

ABS: x declared in M **ma-declx**

STM: ma-declx_wf

STM: decidable__ma-declx

STM: not-ma-declx-implies

ABS: k declared in M **ma-declk**

STM: ma-declk_wf

STM: not-ma-declk-implies

STM: decidable__ma-declk

STM: decidable__ma-declm

ABS: da-outlink-f(da;k) **da-outlink-f**

STM: da-outlink-f_wf

ABS: da-outlinks(da;i) **da-outlinks**

STM: da-outlinks_wf

STM: da-outlinks-empty

STM: da-outlinks-single

STM: da-outlinks-join

ABS: ma-outlinks(M;i) **ma-outlinks**

STM: ma-outlinks_wf

STM: din-not-declared

ABS: tag tg always has type T in M **ma-tag-type**

STM: ma-tag-type_wf

ABS: M.kind **ma-k**

STM: ma-k_wf

ABS: ma-has-effect(M;k) **ma-has-effect**

STM: ma-has-effect_wf

STM: ma-no-effect

STM: decidable__ma-has-effect

ABS: ma-has-sends(M;k) **ma-has-sends**

STM: ma-has-sends_wf

STM: decidable__ma-has-sends

ABS: sends-on-pair(s;l;tg) **sends-on-pair**

STM: sends-on-pair_wf

STM: assert-sends-on-pair

STM: ma-no-sends

STM: ma-dout2-subtype

STM: ma-rframe-pre_wf

STM: ma-rframe-ef_wf

STM: ma-rframe-send_wf

STM: ma-sub_transitivity

ABS: ma-is-empty(M) **ma-is-empty**

STM: ma-is-empty_wf

STM: assert-ma-is-empty

STM: ma-empty-is-empty

STM: ma-empty-sub

STM: ma-is-empty-sub

STM: ma-empty-tag-type

STM: ma-compatible-decls_wf

STM: ma-join-empty

STM: ma-empty-join

STM: ma-comp-decls-join

STM: ma-join-assoc

STM: ma-compatible_weakening

STM: ma-join-sub

ABS: with declarations ds:dsda:da **ma-single-decls**

STM: ma-single-decls_wf

ABS: ma-single-effect0(x;A;k;T;f) **ma-single-effect0**

STM: ma-single-effect0_wf

ABS: ma-single-effect1(x;A;y;B;k;T;f) **ma-single-effect1**

STM: ma-single-effect1_wf

ABS: ma-single-sends0(B;T;a;l;tg;f) **ma-single-sends0**

STM: ma-single-sends0_wf

ABS: a(v) sends [tg, $f(x$, v)] on link l **ma-single-sends1**

ABS: ma-single-pre1(x;A;a;T;y,$v.P(y$;v)) **ma-single-pre1**

STM: ma-single-pre1_wf

ABS: precondition a: True **ma-single-pre-true**

STM: ma-single-pre-true_wf

ABS: with ds: ds init: $init$action a:T precondition a(v) is P **ma-single-pre-init**

STM: ma-single-pre-init_wf

ABS: ma-single-pre-init1(x;T;c;a; T';y,v.P(y;v)) **ma-single-pre-init1**

STM: ma-single-pre-init1_wf

STM: ma-state-atom-free

STM: ma-da-atom-free

STM: ma-dout-atom-free

ABS: $\oplus(L)$ **ma-join-list**

STM: ma-join-list-property

STM: ma-join-list_wf

STM: ma-sub-join-list

STM: ma-join-list-feasible

STM: ma-join-list-compat

STM: ma-join-list-compat2

STM: ma-join-declm

STM: ma-join-list-declm

STM: ma-join-list-declm2

STM: ma-join-list-din

STM: ma-join-list-dout

ABS: MsgAForm **msg-form**

STM: msg-form_wf

STM: msga-sub-msg-form

STM: ma-outlinks-wf2

STM: msg-form-join

STM: msg-form-join-list

STM: ma-is-empty_wf_join

STM: ma-join-list-is-empty

STM: assert-ma-join-list-is-empty

STM: ma-outlinks-join

STM: ma-outlinks-join-list

STM: sub-join-list-din

STM: ma-join-list-tag-type

ABS: ma-rename(rx;ra;rt;M) **ma-rename**

STM: ma-rename_wf

STM: msga-at-sub-left

STM: msga-at-sub-right

STM: ma-single-init-feasible

STM: ma-single-frame-feasible

STM: ma-single-sframe-feasible

STM: ma-single-pre-init1-feasible

STM: ma-single-pre-init-feasible

STM: ma-single-effect-feasible

STM: ma-single-effect0-feasible

STM: ma-single-pre-true-feasible

STM: ma-single-pre-feasible

STM: ma-single-pre1-feasible

STM: ma-single-sends-feasible

ABS: Chooser(dec) **ma-chooser**

STM: ma-chooser_wf

ABS: Trans(M) **ma-trans**

STM: ma-trans_wf

ABS: Sends(M) **ma-sends**

STM: ma-sends_wf

STM: ma-atom-free

STM: atom-free-ma-state

STM: atom-free-ma-decider

STM: atom-free-ma-shape

STM: atom-free-ma-msg-from

STM: ma-init-val-inherence

STM: ma-ef-val-inherence

STM: ma-send-val-inherence

ABS: Inlnk(i) **d-I**

STM: d-I_wf

ABS: Outlnk(i) **d-O**

STM: d-O_wf

ABS: d-empty() **d-empty**

STM: d-empty_wf

STM: d-sub-null

STM: d-sub_transitivity

ABS: @i: with declarations ds:ds da:da **d-single-decls**

STM: d-single-decls_wf

ABS: @i (with ds: ds init: $init$ action a:T precondition a(v) is P s v) **d-single-pre-init**

STM: d-single-pre-init_wf

STM: d-feasible-types

STM: d-feasible-types2

STM: d-decl-atom-free

STM: equal-d-world-states

ABS: local-atom(A;dec;a) **local-atom**

STM: local-atom_wf

ABS: onlnk(l;mss) **d-onlnk**

STM: d-onlnk_wf

ABS: d-rename(rx;ra;rt;D) **d-rename**

STM: d-rename_wf

ABS: interface-check(D;l;tg;T) **interface-check**

STM: interface-check_wf

STM: interface-check-tag-type

STM: finite-support-feasible

STM: finite-support-interface-feasible

STM: atom-free-d-msg-from

STM: ma-dout-sub

STM: realizes-monotone-wrt-sub

ABS: Decl **s-decl**

STM: s-decl_wf

ABS: **s-decl-null**

STM: s-decl-null_wf

ABS: $d(a)$ **s-declared**

STM: s-declared_wf

STM: m-at_wf

ABS: InDecl(i) **in-decl**

STM: in-decl_wf

ABS: OutDecl(i) **out-decl**

STM: out-decl_wf

ABS: $d(p)$ **s-in-declared**

ABS: $d(p)$ **s-out-declared**

STM: s-out-declared_wf

STM: s-in-declared_wf

ABS: Valtype(k;da;din) **s-valtype**

STM: s-valtype_wf

ABS: s-state(ds) **s-state**

STM: s-state_wf

STM: m-sys-compatible-symmetry

STM: m-at-compatible

STM: m-at-distinct-compatible

STM: m-at-self-compatible

STM: m-sys-compatible-join

STM: interface-compatible-at-same

STM: m-sys-sub-join-left

STM: m-sys-sub-join-right

STM: dsys-join-sub

ABS: $\oplus(L)$ **m-sys-join-list**

STM: m-sys-join-list-property

STM: dsys-join-list-property

STM: m-sys-join-list_wf

STM: m-sys-join-list_wf2

STM: m-sys-sub-join-list

STM: m-sys-sub-join-map

STM: m-sys-join-list-property2

STM: join-list-compatible

STM: join-list-compatible2

STM: interface-compatible-join-list

STM: interface-compatible-join-list2

STM: feasible-join-list

STM: ma-single-pre1_wf2

STM: ma-single-effect0_wf2

ABS: s-dsys(M) **s-dsys**

STM: s-dsys_wf

ABS: DsysNull **dsys-null**

STM: dsys-null_wf

STM: s-at-sub-s-dsys

STM: s-at-sub-s-at

STM: s-dsys-sub-s-dsys

STM: dsys-single-sub-dsys

ABS: Rcv(l;tg) **Rcv**

STM: Rcv_wf

STM: s-pre-rule

STM: s-pre-rule0

STM: s-pre-rule1

STM: s-effect-rule

STM: s-effect-rule0

STM: effect-rule1

STM: s-sends-rule

STM: ma-single-sends1_wf

STM: s-sends-rule1

STM: s-unconditional-send1-rule

STM: conditional-send1-rule

STM: sends-rule0

STM: unconditional-send-rule0

STM: s-init-rule

STM: s-frame-rule

STM: frame-rule0

STM: frame-rule1

STM: frame-rule2

STM: frame-rule3

STM: s-sframe-rule

STM: better-sframe-rule

STM: ma-component-types

ABS: effect-type(ds;ds';da) **effect-type**

STM: effect-type_wf

STM: effect-type-subtype

STM: proper-at-join

STM: m-sys-feasible

STM: msys-at-compatible-right

STM: msys-at-compatible-left

STM: msys-at-at

STM: ma-single-pre-init-ma-single-pre-compatible

STM: ma-single-pre-ma-single-pre-init-compatible

STM: ma-single-pre-init-ma-single-sends-compatible

STM: ma-single-sends-ma-single-pre-init-compatible

STM: ma-single-pre-init-ma-single-effect-compatible

STM: ma-single-effect-ma-single-pre-init-compatible

STM: ma-single-pre-init-ma-single-sframe-compatible

STM: ma-single-sframe-ma-single-pre-init-compatible

STM: ma-single-pre-init-ma-single-frame-compatible

STM: ma-single-frame-ma-single-pre-init-compatible

STM: ma-single-pre-init-ma-single-init-compatible

STM: ma-single-init-ma-single-pre-init-compatible

STM: ma-single-pre-ma-single-sends-compatible

STM: ma-single-sends-ma-single-pre-compatible

STM: ma-single-pre-ma-single-effect-compatible

STM: ma-single-effect-ma-single-pre-compatible

STM: ma-single-pre-ma-single-sframe-compatible

STM: ma-single-sframe-ma-single-pre-compatible

STM: ma-single-pre-ma-single-frame-compatible

STM: ma-single-frame-ma-single-pre-compatible

STM: ma-single-pre-ma-single-init-compatible

STM: ma-single-init-ma-single-pre-compatible

STM: ma-single-sends-ma-single-effect-compatible

STM: ma-single-effect-ma-single-sends-compatible

STM: ma-single-sends-ma-single-sframe-compatible

STM: ma-single-sframe-ma-single-sends-compatible

STM: ma-single-sends-ma-single-frame-compatible

STM: ma-single-frame-ma-single-sends-compatible

STM: ma-single-sends-ma-single-init-compatible

STM: ma-single-init-ma-single-sends-compatible

STM: ma-single-effect-ma-single-sframe-compatible

STM: ma-single-sframe-ma-single-effect-compatible

STM: ma-single-effect-ma-single-frame-compatible

STM: ma-single-frame-ma-single-effect-compatible

STM: ma-single-effect-ma-single-init-compatible

STM: ma-single-init-ma-single-effect-compatible

STM: ma-single-sframe-ma-single-frame-compatible

STM: ma-single-frame-ma-single-sframe-compatible

STM: ma-single-sframe-ma-single-init-compatible

STM: ma-single-init-ma-single-sframe-compatible

STM: ma-single-frame-ma-single-init-compatible

STM: ma-single-init-ma-single-frame-compatible

STM: ma-single-pre-init-ma-single-pre-init-compatible

STM: ma-single-pre-ma-single-pre-compatible

STM: ma-single-sends-ma-single-sends-compatible

STM: ma-single-effect-ma-single-effect-compatible

STM: ma-single-sframe-ma-single-sframe-compatible

STM: ma-single-frame-ma-single-frame-compatible

STM: ma-single-init-ma-single-init-compatible

STM: m-sys-at-sub-lemma1

STM: d-sub-lemma1

STM: ma-sub-join-mapl

STM: R-Dsys-Rall2

STM: R-Dsys-Rall

STM: R-Dsys-Rall-init

STM: R-Dsys-sub-Rall

STM: R-sub-feasible-Dsys

STM: R-state-property

STM: R-da-property

STM: R-ds-property

STM: R-Feasible-interface

ABS: R-init(R;i) **R-init**

STM: R-init_wf

STM: R-Feasible-action