event-structures^{11,40}

ABS: first(e) **first**

STM: first_wf

ABS: pred(e) **pred**

STM: pred_wf

ABS: ecase1(e;$info$;$i.f(i)$;l,$e'.g(l;e')$) **ecase1**

STM: ecase1_wf

ABS: loc(e) **loc**

STM: loc_wf

ABS: rcv?(e) **rcv?**

STM: rcv?_wf

ABS: sender(e) **sender**

STM: sender_wf

ABS: link(e) **link**

STM: link_wf

ABS: pred!(e;e') **pred!**

STM: pred!_wf

STM: pred-total

ABS: $e < e'$ **cless**

STM: cless_wf

STM: cless-eq-loc

ABS: sends-bound(p;e;l) **sends-bound**

STM: sends-bound_wf

STM: sends-bound-property

STM: strong-sends-bound-property

STM: pred-first-lemma

ABS: eventlist($pred?$;e) **eventlist**

STM: eventlist_wf

STM: member_eventlist

STM: l_before_eventlist

STM: l_before_eventlist_iff

ABS: rcv-from-on(dE;dL;$info$;e;l;r) **rcv-from-on**

STM: rcv-from-on_wf

STM: assert-rcv-from-on

ABS: receives(dE;dL;$pred?$;$info$;p;e;l) **receives**

STM: receives_wf

STM: member_receives

ABS: index(dE;dL;$pred?$;$info$;p;r) **index**

STM: index_wf

STM: index-property1

ABS: kind(e) **kind**

STM: kind_wf

ABS: rtag($info$;e) **rtag**

STM: rtag_wf

STM: rcv?-kind

STM: rcv?-link

ABS: EOrderAxioms(E; $pred?$; $info$) **EOrderAxioms**

STM: EOrderAxioms_wf

ABS: EState(T) **EState**

STM: EState_wf

ABS: s+r **es-shift**

STM: es-shift_wf

ABS: when-after(e;$info$;$pred?$;$init$;$Trans$;val;$time$) **when-after**

STM: when-after_wf

ABS: state_when(e) **state_when**

STM: state_when_wf

ABS: state_after(e) **state_after**

STM: state_after_wf

ABS: val-axiom($E;V;M;\mathit{info};\mathit{pred?};\mathit{init};\mathit{Trans};\mathit{Choose};\mathit{Send};\mathit{val};\mathit{time}$) **val-axiom**

STM: val-axiom_wf

ABS: rmsg($\mathit{info};\mathit{val};e$) **rmsg**

STM: rmsg_wf

ABS: sends($dE;dL;\mathit{pred?};\mathit{info};\mathit{val};p;e;l$) **sends**

STM: sends_wf

STM: better-sends_wf

ABS: SESAxioms{i:l}($E;T;\mathit{pred?};\mathit{info};\mathit{when};\mathit{after};\mathit{time}$) **SESAxioms**

STM: SESAxioms_wf

ABS: eventtype($k;\mathit{loc};V;M;e$) **eventtype**

STM: eventtype_wf

ABS: ESAxioms{i:l}($E;T;M;\mathit{loc};\mathit{kind};\mathit{val};\mathit{when};\mathit{after};\mathit{time};\mathit{sends};\mathit{sender};\mathit{index};\mathit{first};\mathit{pred};\mathit{causl}$)

ESAxioms

STM: ESAxioms_wf

STM: SES-implies-ES

ABS: constant_function($f;A;B$) **constant_function**

STM: constant_function_wf

ABS: ES **event_system**

STM: event_system_wf

ABS: E **es-E**

STM: es-E_wf

ABS: es-eq(es) **es-eq**

STM: es-eq_wf

ABS: es-pred?(es) **es-pred?**

STM: es-pred?_wf

ABS: es_info(es) **es_info**

STM: es_info_wf

ABS: loc(e) **es-loc**

STM: es-loc_wf

ABS: kind(e) **es-kind**

STM: es-kind_wf

ABS: es-oaxioms(es) **es-oaxioms**

STM: es-oaxioms_wf

ABS: es-T(es) **es-T**

STM: es-T_wf

ABS: es-V(es) **es-V**

STM: es-V_wf

ABS: es-M(es) **es-M**

STM: es-M_wf

ABS: discrete(i;x) **es-isconst**

STM: es-isconst_wf

ABS: Msg **es-Msg**

STM: es-Msg_wf

ABS: (Msg on l) **es-Msgl**

STM: es-Msgl_wf

ABS: isrcv(e) **es-isrcv**

STM: es-isrcv_wf

ABS: tag(e) **es-tag**

STM: es-tag_wf

ABS: lnk(e) **es-lnk**

STM: es-lnk_wf

ABS: act(e) **es-act**

STM: es-act_wf

ABS: rcvtype(e) **es-rcvtype**

STM: es-rcvtype_wf

ABS: acttype(e) **es-acttype**

STM: es-acttype_wf

ABS: kindtype(i;k) **es-kindtype**

STM: es-kindtype_wf

ABS: valtype(e) **es-valtype**

STM: es-valtype_wf

ABS: es_vartype(es;i;x) **es_vartype**

STM: es_vartype_wf

ABS: vartype(i;x) **es-vartype**

STM: es-vartype_wf

ABS: es_state(es;i) **es_state**

STM: es_state_wf

STM: es-shift_wf2

ABS: state@i **es-state**

ABS: state@$i|xs$ **es-partial-state**

STM: es-partial-state_wf

STM: es-state_wf

STM: es-state-subtype-partial-state

ABS: $s.x$ **es-state-ap**

STM: es-state-ap_wf

ABS: es_init(es) **es_init**

STM: es_init_wf

ABS: x initially@i **es-initially**

STM: es-initially_wf

ABS: initial state @i **es-init-state**

STM: es-init-state_wf

ABS: s(now) **es-read-state**

STM: es-read-state_wf

ABS: es-Trans(es) **es-Trans**

ABS: es_val(es) **es_val**

STM: es_val_wf

STM: es-Trans_wf

ABS: es-Send(es) **es-Send**

STM: es-Send_wf

ABS: es-Choose(es) **es-Choose**

STM: es-Choose_wf

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

STM: es-first_wf

STM: can-apply-pred?

ABS: pred(e) **es-pred**

STM: es-pred_wf

STM: do-apply-pred?

ABS: es-pred!(es;e;e') **es-pred!**

STM: es-pred!_wf

STM: es-loc-pred

STM: es-loc-pred-plus

STM: es-pred!-wellfounded

ABS: val(e) **es-val**

STM: es-val_wf

ABS: time(e) **es-time**

STM: es-time_wf

ABS: es_time(es) **es_time**

STM: es_time_wf

ABS: es_state_when(es;e) **es_state_when**

STM: es_state_when_wf

ABS: (timed)state after e **es_state_after**

STM: es_state_after_wf

ABS: (state when e) **es-state-when**

ABS: state after e **es-state-after**

ABS: x when e **es-when**

STM: es-when_wf

STM: es-state-when_wf

ABS: (x after e) **es-after**

STM: es-after_wf

STM: es-state-after_wf

ABS: sends(l;e) **es-sends**

STM: es-sends_wf

ABS: sender(e) **es-sender**

STM: es-sender_wf

ABS: index(e) **es-index**

STM: es-index_wf

ABS: ($e < e'$) **es-causl**

STM: es-causl_wf

ABS: (e <loc e') **es-locl**

STM: es-locl_wf

ABS: e $\leq$loc e' **es-le**

STM: es-le_wf

ABS: Trans(i) **es-trans**

STM: es-trans_wf

ABS: Send(i) **es-send**

STM: es-send_wf

ABS: Choose(i) **es-choose**

STM: es-choose_wf

STM: es-axioms

STM: es-locl-wellfnd

STM: es-discrete-const

STM: es-isconst-property

STM: es-locl-antireflexive

STM: es-locl_irreflexivity

STM: es-le-loc

STM: es-locl-iff

STM: es-dst-lnk

ABS: mtag(m) **es-mtag**

STM: es-mtag_wf

ABS: $s_1 \equiv s_2$ mod x@i **es-x-equiv**

STM: es-x-equiv_wf

ABS: es-independent(es;i;k;x) **es-independent**

STM: es-independent_wf

STM: mlnk_wf2

ABS: sends(l,tg,e) **es-tg-sends**

STM: es-tg-sends_wf

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

STM: decl-state_wf

STM: decl-state-eta

ABS: $\forall e@i.\ P(e)$ **alle-at**

STM: alle-at_wf

STM: es-rcv-kind

STM: es-kind-rcv

ABS: $\exists e'{\geq}e.P(e')$ **existse-ge**

STM: existse-ge_wf

ABS: $@i$ state ds **es-state-type**

STM: es-state-type_wf

STM: es-state-type-implies

ABS: DeclaredType(ds;x) **decl-type**

STM: decl-type_wf

ABS: $@i(x{:}T)$ **es-dtype**

STM: es-dtype_wf

ABS: $@i$ x initially $v{:}T$ **init-p**

STM: init-p_wf

ABS: $@i$ continuous x initially $v{:}T$ **init_p**

STM: init_p_wf

STM: init_p-discrete

ABS: $@i$ only events in L change $x : T$ **frame-p**

STM: frame-p_wf

ABS: only events in L send on l with tg **sframe-p**

STM: sframe-p_wf

ABS: $@i$: k affects only L **aframe-p**

STM: aframe-p_wf

ABS: @i:k sends only on links in L **bframe-p**

STM: bframe-p_wf

ABS: @i:k sends only on links in L **kind-send-frame**

STM: kind-send-frame_wf

ABS: @i: only members of L read x **rframe-p**

STM: rframe-p_wf

ABS: @i events of kind k change x to f State(ds) (val:T) **effect-p**

STM: effect-p_wf

ABS: @i events of kind k change continuous x to f State(ds) (val:T) **effect_p**

STM: effect_p_wf

STM: effect_p-discrete

ABS: rcvs from e on $l = L$ **es-rcv-from**

STM: es-rcv-from_wf

ABS: loc-ordered(es;L) **loc-ordered**

STM: loc-ordered_wf

STM: loc-ordered-equality

ABS: es-receives(es;e;l) **es-receives**

STM: es-receives_wf

STM: member-es-receives

STM: loc-ordered-es-receives

STM: es-rcv-from-equal-receives

STM: es-rcv-from-member-index

STM: es-rcv-from-implies

ABS: sends-msgs(s;v;tg_f) **sends-msgs**

STM: sends-msgs_wf

ABS: sends k(v:T) on l:tagged(g,State(ds),v):dt **sends-p**

STM: sends-p_wf

ABS: (state after e)+t **es-state-after-elapsed**

STM: es-state-after-elapsed_wf

STM: discrete-after-elapsed

ABS: (initial state @ i)+t **es-init-elapsed**

STM: es-init-elapsed_wf

STM: discrete-init-elapsed

ABS: es-kind-index(es;k;e) **es-kind-index**

STM: es-kind-index_wf

ABS: @i Precondition for a:Outcome(p) is P:State(ds)$\rightarrow$ $\mathbb{B}$ **pre-p**

STM: pre-p_wf

STM: es-time-order