

fifo^{11,40}

ABS: $es \equiv es' \text{ mod } es, e.P(es;e)$ **es-p-equiv**

STM: es-p-equiv_wf

STM: es-p-equiv_weakening

STM: es-p-equiv_inversion

STM: es-p-equiv_transitivity

ABS: $\text{sub-es-pred}(es;dom;e)$ **sub-es-pred**

STM: sub-es-pred_wf

ABS: $\text{sub-es-sender}(es;dom;e)$ **sub-es-sender**

STM: sub-es-sender_wf

ABS: $a.f(a)$ is $c \leq$ preserving on $e.P(e)$ **causale-order-preserving**

STM: causale-order-preserving_wf

ABS: $a.f(a)$ is $c <$ preserving on $e.P(e)$ **causal-order-preserving**

STM: causal-order-preserving_wf

COM: explanation of antecedent functions

ABS: $Q \leftarrow -f-- P$ **antecedent-function**

STM: antecedent-function_wf

STM: antecedent-function_functionality_wrt_iff

ABS: $Q \leftarrow\leftarrow f-- P$ **antecedent-surjection**

STM: antecedent-surjection_wf

STM: antecedent-surjection_functionality_wrt_iff

ABS: $(e.P(e) \rightarrow a.f(a) \rightarrow e'.Q(e'))$ with R **causal-bijection**

STM: causal-bijection_wf

STM: causal-bijection_functionality_wrt_iff

STM: causal-bijection-interleaving

ABS: $(e.P(e) \rightarrow\rightarrow e'.Q(e'))$ with R **causally-related**

STM: causally-related_wf
 ABS: $e.P(e) \rightarrow_{\text{op}} e'.Q(e')$ with R **causally-op-related**
 STM: causally-op-related_wf
 ABS: causal-predecessor($es;p$) **causal-predecessor**
 STM: causal-predecessor_wf
 STM: causal-pred-wellfounded
 STM: causal-pred-from-relation
 ABS: es-r-pred{i:l}($es;d;R$) **es-r-pred**
 STM: es-r-pred_wf
 STM: es-r-pred-property
 ABS: es-p-immediate-pred($es;P$) **es-p-immediate-pred**
 STM: es-p-immediate-pred_wf
 STM: decidable_es-p-immediate-pred
 STM: es-p-immediate-pred-wellfounded
 ABS: es-r-immediate-pred($es;R;e';e$) **es-r-immediate-pred**
 STM: es-r-immediate-pred_wf
 STM: decidable_es-r-immediate-pred
 STM: not-r-immediate-pred
 STM: es-r-immediate-pred-exists
 ABS: causal-weak-predecessor($es;p$) **causal-weak-predecessor**
 STM: causal-weak-predecessor_wf
 ABS: $e p < e'$ **es-p-locl**
 STM: es-p-locl_wf
 ABS: $e p \leq e'$ **es-p-le**
 STM: es-p-le_wf
 STM: es-p-locl_transitivity
 STM: es-p-le_transitivity

STM: es-p-le-weakening
 STM: es-p-le-weakening_eq
 STM: es-p-locl.transitivity1
 STM: es-p-locl.transitivity2
 STM: es-p-locl-test
 STM: es-causl_weakening_p-locl
 STM: es-causle_weakening_p-le
 STM: es-causl-p-locl-test
 ABS: same-thread($es;p;e;e'$) **same-thread**
 STM: same-thread_wf
 STM: same-thread-do-apply
 STM: same-thread_weakening
 STM: same-thread_inversion
 STM: same-thread.transitivity
 STM: decidable__same-thread
 STM: thread-p-ordered
 STM: thread-ordered
 ABS: es-height($es;e$) **es-height**
 STM: es-height_wf
 STM: es-height-causl
 STM: single-threaded-relation
 STM: single-threaded-relation3
 ABS: single-thread-generator $\{i:l\}(es;P;R)$ **single-thread-generator**
 STM: single-thread-generator_wf
 STM: cond_rel_equivalent
 STM: cond_equiv_to_causl
 STM: generated-thread-properties

STM: generated-thread-binrel-properties

STM: generated-thread-binrel-properties2

STM: generated-thread-properties2

STM: generated-thread-properties3

STM: generated-thread-no-joins

ABS: forclients C sends FIFO
from j to i via $(S[j,i],codes)$
receives at i via $(R[i],decodes)$

fifo

STM: fifo_wf

ABS: FIFO **FIFO**

STM: FIFO_wf

ABS: links2Fifo-impl($es;l_1;l_2;A;tg$) **links2Fifo-impl**

STM: links2Fifo-impl_wf

ABS: $ff.C$ **fifoC**

STM: fifoC_wf

ABS: $ff.T$ **fifoT**

STM: fifoT_wf

ABS: $ff.S$ **fifoS**

STM: fifoS_wf

ABS: $ff.R$ **fifoR**

STM: fifoR_wf

ABS: $ff.Codes$ **fifoCodes**

STM: fifoCodes_wf

ABS: $ff.Decodes$ **fifoDecodes**

STM: fifoDecodes_wf

ABS: $ff.Sender$ **fifoSender**

STM: fifoSender_wf
 STM: fifoSender-antecedent
 STM: fifoSender-causes
 STM: fifoSender-preserves-order
 STM: fifoSender-one-one
 STM: fifoSender-reverse-order
 STM: fifoSender-encoding
 STM: fifo-FIFO
 ABS: $[e: i \dashrightarrow p \rightarrow j]$ **snd-it**
 ABS: $[e: i \leftarrow p \dashrightarrow j]$ **rcv-it**
 STM: rcv-it_wf
 STM: rcv-it-loc
 STM: snd-it_wf
 STM: snd-it-loc
 STM: decidable__snd-it
 STM: decidable__rcv-it
 STM: snd-it-of-rcv-it
 STM: fifoReceiver-exists
 ABS: ff .Receiver **fifoReceiver**
 STM: fifoReceiver_wf
 STM: fifoReceiver-properties
 STM: fifoReceiver-caused
 STM: rcv-it-of-snd-it
 STM: compose-fifo-send

 ABS: for clients C sends FIFO
 from j to i via $(S[j,i], codes)$
 receives at i via $(R[i], decodes)$ requests $Req[j]$ are acknowledged by $Ack[j,i]$

fifo+

STM: fifo+_wf

ABS: fifo+property(*es;codes;decodes;C;S;R;T;Req;Ack;i;f*) **fifo+property**

STM: fifo+property_wf

STM: fifo+rewrite

ABS: F2F+-decls **F2F+-decls**

STM: F2F+-decls_wf

ABS: is_req **f2f+Req**

STM: f2f+Req_wf

ABS: is_ack **f2f+Ack**

STM: f2f+Ack_wf

ABS: awaiting **f2f+Wait**

STM: f2f+Wait_wf

ABS: owes_ack **f2f+Owes**

STM: f2f+Owes_wf

ABS: req_dcdr **f2f+ReqDcdr**

ABS: ack_dcdr **f2f+AckDcdr**

ABS: R_dcdr **f2f+RDcdr**

ABS: S_dcdr **f2f+SDcdr**

STM: f2f+SDcdr_wf

STM: f2f+RDcdr_wf

STM: f2f+AckDcdr_wf

STM: f2f+ReqDcdr_wf

STM: f2f+-property

STM: loc-of-req-sender

STM: loc-of-ack-sender

STM: loc-of-req-receiver
 STM: loc-of-ack-receiver
 ABS: plus-ify $\{i:l\}(es;ff;is_req;is_ack;awaiting;owes_ack)$ **plus-ify**
 STM: plus-ify_wf
 STM: owes_ack-2-ff
 STM: awaiting-2-ff
 STM: owes_ack-2-tt
 STM: awaiting-2-tt
 STM: acks-between-reqs
 STM: reqs-between-acks
 STM: bool-tt-or-ff
 STM: the-first-event
 ABS: first-event $\{i:l\}(es;e)$ **first-event**
 STM: first-event_wf
 STM: first-event-property
 STM: init-p-implies2
 STM: req-received-before-ack
 STM: req-sent-before-ack
 STM: max-of-intset
 STM: max-of-eventset
 ABS: f2f+-pred(e',e) **f2f+-pred**
 STM: f2f+-pred_wf
 STM: f2f+-pred-field
 STM: f2f+-pred-dom
 STM: f2f+-pred-rng
 STM: f2f+-pred-sub-causal
 STM: decidable_f2f+-pred

STM: f2f+-pred-alternates
STM: req-pred-ack
STM: ack-pred-req
STM: f2f+-pred-closed
STM: req-pred-ack2
STM: f2f+-pred-no-forks
STM: ack-has-f2f+-pred
STM: f2f+-pred-unique-min
STM: f2f+-pred-at-most-one-min
STM: f2f+-pred-generates-thread
ABS: f2f+-p+ **f2f+-p+**
STM: f2f+-p+_wf
STM: f2f+-p+-sub-causal
STM: f2f+-pred-is-immediate
STM: f2f+-p+-field
STM: f2f+-p+-equiv-causal
STM: f2f+-p+-total
STM: f2f+-req-exists
STM: f2f+-pred-preserves-order
STM: plus-ify-makes-FIFO+
ABS: f2f+-event(*e*) **f2f+-event**
STM: f2f+-event_wf
ABS: crossed-pair{i:l}(*es;ff;is_req;is_ack;sndr;rcvr;r;a*) **crossed-pair**
STM: crossed-pair_wf
STM: combine-causal-bijections
STM: combine-causally-related

ABS: switch between fifo+ send $S_1(j;i;e)$
request $Req_1(i;e)$
acknowledge $Ack_1(j;i;e)$ and
send $S_2(j;i;e)$ request $Req_2(i;e)$ acknowledge $Ack_2(j;i;e)$

fifo+switch

STM: fifo+switch_wf

STM: combine-antecedent-surjections

STM: causal-order-preserving-interleaving

ABS: [$S?$ codes₁ : codes₂] **union-codes**

STM: union-codes_wf

STM: union-codes-property

ABS: [$R ?$ decodes₁ : decodes₂] **union-decodes**

STM: union-decodes_wf

STM: union-decodes-property

STM: union-decodes-exists

STM: combine-fifo+

ABS: abs-R **abs-R**

STM: abs-R_wf

ABS: abs-S **abs-S**

STM: abs-S_wf

ABS: abs-fifo{i:l}(C;T) **abs-fifo**

STM: abs-fifo_wf