

## message-automata<sup>0,22</sup>

ABS:  $\text{Msg}(da)$  **ma-Msg**

STM: ma-Msg\_wf

ABS:  $\text{MsgA}$  **msga**

STM: msga\_wf

ABS:  $\text{ds}(M)$  **ma\_ds**

STM: ma\_ds\_wf

ABS:  $\text{da}(M)$  **ma\_da**

STM: ma\_da\_wf

ABS:  $M.\text{ds}(x)$  **ma-ds**

STM: ma-ds\_wf

ABS:  $M.\text{da}(a)$  **ma-da**

ABS:  $a$  declared in  $M$  **ma-decla**

STM: ma-decla\_wf

ABS:  $\text{rcv}(l, tg)$  declared in  $M$  **ma-declm**

STM: ma-declm\_wf

STM: ma-da\_wf

ABS:  $M.\text{din}(l, tg)$  **ma-din**

STM: ma-din\_wf

ABS:  $M.\text{dout}(l, tg)$  **ma-dout**

STM: ma-dout\_wf

ABS:  $M.\text{init}(x, v)$  **ma-init**

STM: ma-init\_wf

ABS:  $M.\text{init}(x)?v$  **ma-init-val**

STM: ma-init-val\_wf

ABS:  $M.\text{state}$  **ma-st**

STM: ma-st\_wf  
 ABS:  $M.\text{Msg}$  **ma-msg**  
 STM: ma-msg\_wf  
 ABS:  $M.V(k)$  **ma-v**  
 STM: ma-v\_wf  
 ABS: unsolvable  $M.\text{pre}(a,s)$  **ma-npre**  
 STM: ma-npre\_wf  
 ABS:  $M.\text{pre}(a,s,v)$  **ma-pre**  
 STM: ma-pre\_wf  
 ABS:  $a$  in  $\text{dom}(M.\text{pre})$  **ma-has-pre**  
 STM: ma-has-pre\_wf  
 STM: decidable\_ma-has-pre  
 ABS:  $M.\text{ef}(k,x,s,v,w)$  **ma-ef**  
 STM: ma-ef\_wf  
 ABS:  $M.\text{ef}(k,x,s,v)?w$  **ma-ef-val**  
 STM: ma-ef-val\_wf  
 ABS:  $M.\text{send}(k;l;s;v;ms;i)$  **ma-send**  
 STM: ma-send\_wf  
 ABS:  $M.\text{sends}(k,s,v)$  **ma-send-val**  
 STM: ma-send-val\_wf  
 ABS:  $M$  sends on link  $l$  **ma-sends-on**  
 STM: ma-sends-on\_wf  
 ABS:  $M.\text{dout2}(l;tg)$  **ma-dout2**  
 STM: ma-dout2\_wf  
 ABS:  $M.\text{frame}(k \text{ affects } x)$  **ma-frame**  
 STM: ma-frame\_wf  
 ABS:  $M.\text{sframe}(k \text{ sends } \langle l,tg \rangle)$  **ma-sframe**

STM: ma-sframe\_wf

ABS:  $M.aframe(k \text{ affects } x)$  **ma-afame**

STM: ma-afame\_wf

ABS:  $M.bframe(k \text{ sends on } l)$  **ma-bframe**

STM: ma-bframe\_wf

ABS:  $M.rframe(k \text{ reads } x)$  **ma-rframe**

STM: ma-rframe\_wf

ABS:  $M:k$  may not read  $x$  **ma-no-read**

STM: ma-no-read\_wf

STM: assert-ma-no-read

ABS:  $(s_1 \equiv s_2 \text{ mod } x)$  **ma-x-equiv**

STM: ma-x-equiv\_wf

ABS:  $M.rframe(A.pre \ p \text{ for } a)$  **ma-rframe-pre**

ABS:  $M.rframe(A.effect \ f \text{ of } k \text{ on } y)$  **ma-rframe-ef**

ABS:  $M.rframe(A.sends \ tfl \text{ of } k \text{ on } l)$  **ma-rframe-send**

ABS:  $M_1 \subseteq M_2$  **ma-sub**

STM: ma-sub\_wf

STM: ma-sub\_weakening

ABS: mk-ma( $ds;$   
 $da;$   
 $init;$   
 $pre;$   
 $ef;$   
 $send;$   
 $frame;$   
 $sframe;$   
 $afame;$   
 $bframe;$   
 $rframe)$

**mk-ma**

STM: mk-ma\_wf  
 ABS: **ma-empty**  
 STM: ma-empty\_wf  
 ABS:  $M_1 \parallel \text{decl } M_2$  **ma-compatible-decls**  
 ABS:  $M_1 \oplus M_2$  **ma-join**  
 STM: ma-join\_wf  
 STM: ma-join-sends-on  
 STM: ma-sub-join-left  
 ABS:  $M_1 \parallel M_2$  **ma-compatible**  
 STM: ma-compatible\_wf  
 STM: ma-compatible-symmetry  
 STM: ma-compatible-self  
 STM: ma-sub-join-right  
 STM: ma-empty-compatible-left  
 STM: ma-empty-compatible-right  
 ABS:  $x : t$  initially  $x = v$  **ma-single-init**  
 STM: ma-single-init\_wf  
 ABS: only members of  $L$  affect  $x : t$  **ma-single-frame**  
 STM: ma-single-frame\_wf  
 ABS: only  $L$  sends on  $(l$  with  $tg)$  **ma-single-sframe**  
 STM: ma-single-sframe\_wf  
 ABS:  $k$  affects only members of  $L$  **ma-single-iframe**  
 STM: ma-single-iframe\_wf  
 ABS:  $k$  sends only on links in  $L$  **ma-single-bframe**  
 STM: ma-single-bframe\_wf  
 ABS: only members of  $L$  read  $x$  **ma-single-rframe**  
 STM: ma-single-rframe\_wf

ABS: with declarations  $ds:dsda:daeffect$  of  $k(v)$  is  $x := f$  s  $v$  **ma-single-effect**  
 STM: ma-single-effect\_wf  
 ABS: with declarations  $ds:dsda:dak(v)$  sends  $f$  s  $v$  on link  $l$  **ma-single-sends**  
 STM: ma-single-sends\_wf  
 ABS: (with  $ds: ds$  action  $a:T$  precondition  $a(v)$  is  $P$  s  $v$ ) **ma-single-pre**  
 STM: ma-single-pre\_wf  
 ABS:  $ma\text{-frame}\text{-compat}(A;B)$  **ma-frame-compat**  
 STM: ma-frame-compat\_wf  
 STM: ma-join-frame-compat  
 STM: ma-join-frame-compat2  
 ABS:  $ma\text{-frame}\text{-compatible}(A;B)$  **ma-frame-compatible**  
 STM: ma-frame-compatible\_wf  
 ABS:  $Feasible(M)$  **ma-feasible**  
 STM: ma-feasible\_wf  
 STM: ma-feasible-ma-no-read  
 STM: ma-feasible-rframe-ef  
 STM: ma-feasible-rframe-send  
 STM: ma-empty-feasible  
 STM: ma-frame-compatible\_symmetry  
 STM: ma-empty-frame-compatible-left  
 STM: ma-empty-frame-compatible-right  
 STM: ma-join-compatible  
 ABS:  $A \Vdash B$  **ma-compat**  
 STM: ma-compat\_wf  
 STM: ma-compat-symmetry  
 STM: ma-compat\_weakening  
 STM: ma-empty-compat-left

STM: ma-empty-compat-right

STM: ma-join-feasible

STM: ma-compat-join

STM: ma-compat-join2

STM: ma-send-send-val

STM: ma-send-val-nil

STM: ma-send-val-nil2

STM: ma-feasible-bframe

STM: w-withlnk\_wf\_ma

STM: atom-free-ma-ds

STM: atom-free-ma-da

STM: atom-free-ma-dout