

Proving MCAPI Executions are Correct

Applying SMT Technology to Message Passing

Yu Huang

Verification & Validation Lab

Introduction

```
00 initialize(NODE_0, &v, &s);
01 e0 = create_endpoint(PORT_0, &s);

02 msg_rcv_i(e0, A, sizeof(A), &h1, &s);
03 wait(&h1, &size, &s, MCAPI_INF);
04 a = atoi(A);

05 msg_rcv_i(e0, B, sizeof(B), &h2, &s);
06 wait(&h2, &size, &s, MCAPI_INF);
07 b = atoi(B);

08 if(b > 0);
09 assert(a == 4);

10 finalize(&s);
```

Introduction

```
00 initialize(NODE_0, &v, &s);  
01 e0 = create_endpoint(PORT_0, &s);  
  
02 msg_rcv_i(e0, A, sizeof(A), &h1, &s); R0,2(* , a, &h1)  
03 wait(&h1, &size, &s, MCAPI_INF); w(&h1)  
04 a = atoi(A);  
  
05 msg_rcv_i(e0, B, sizeof(B), &h2, &s); R0,5(* , b, &h2)  
06 wait(&h2, &size, &s, MCAPI_INF); w(&h2)  
07 b = atoi(B);  
  
08 if(b > 0); assume(b>0)  
09 assert(a == 4); assert(a==4)  
  
10 finalize(&s);
```

Problem

Task 0

$R_{0,2}(*, a, \&h1)$

$w(\&h1)$

$R_{0,5}(*, b, \&h2)$

$w(\&h2)$

$\text{assume}(b > 0)$

$\text{assert}(a == 4)$

Task 1

$R_{1,3}(*, c, \&h3)$

$w(\&h3)$

$S_{1,5}(0, "1", \&h4)$

$w(\&h4)$

Task 2

$S_{2,4}(0, "4", \&h5)$

$w(\&h5)$

$S_{2,6}(1, "Go", \&h6)$

$w(\&h6)$

Given a concurrent program using message passing with assumes and asserts, how do you prove if a feasible violating execution exists?

Related Works

- Sharma et al. : Mcc - a runtime verification tool for mcapi user applications. (FMCAD '09)
- Wang et al. : Symbolic pruning of concurrent program executions. (FSE '09)
- Elwakil et al. : Debugging support tool for mcapi applications. (PADTAD '10)
- Vakkalanka et al. : Reduced execution semantics of mpi. (FM '09)

Solution

Task 0

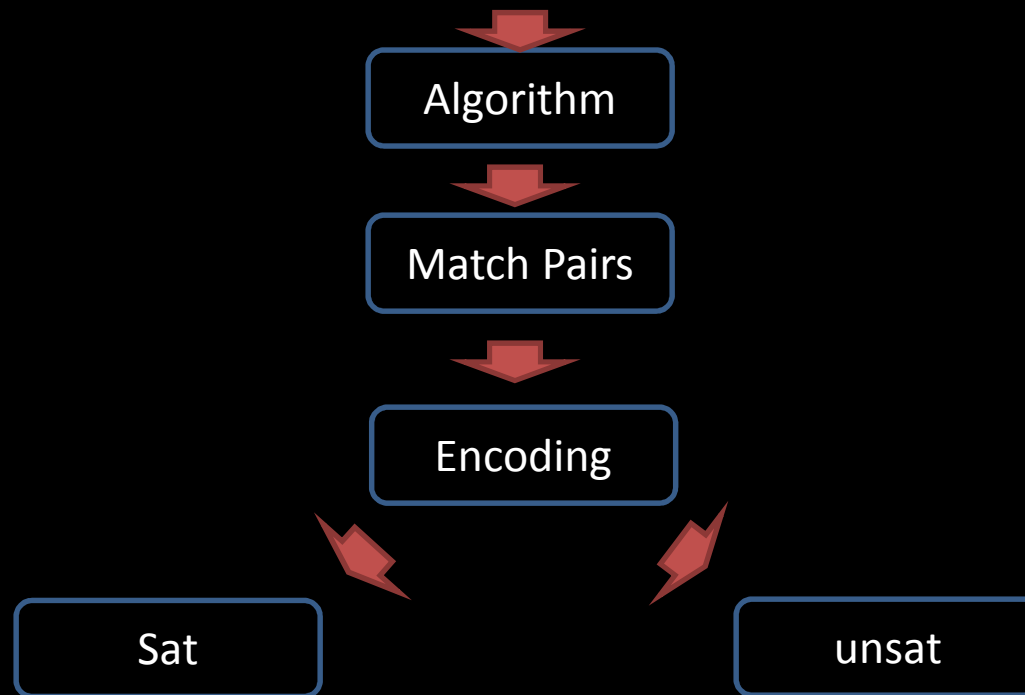
```
R0,2(* , a , &h1)  
w(&h1)  
R0,5(* , b , &h2)  
w(&h2)  
assume(b > 0)  
assert(a == 4)
```

Task 1

```
R1,3(* , c , &h3)  
w(&h3)  
S1,5(0 , "1" , &h4)  
w(&h4)
```

Task 2

```
S2,4(0 , "4" , &h5)  
w(&h5)  
S2,6(1 , "Go" , &h6)  
w(&h6)
```



SMT Model

defs

constraints

match

SMT Model

defs

All definitions of the send,
receive operations and the free
variables

constraints

match

SMT Model

defs

All definitions of the send, receive operations and the free variables

constraints

All constraint clauses, including the assert clauses and the clauses built by the HB function

match

SMT Model

defs

All definitions of the send, receive operations and the free variables

constraints

All constraint clauses, including the assert clauses and the clauses built by the HB function


match


The clauses built by the MATCH function on a set of match pairs, and the clauses built by NE function

Execution Traces

Trace 1		Trace 2	
Task	Command	Task	Command
2	$S_{2,4}(0, "4", \&h5)$	2	$S_{2,4}(0, "4", \&h5)$
2	$w(\&h5)$	2	$w(\&h5)$
0	$R_{0,2}(2, a, \&h1)$	2	$S_{2,6}(1, "Go", \&h6)$
0	$w(\&h1)$	2	$w(\&h6)$
2	$S_{2,6}(1, "Go", \&h6)$	1	$R_{1,3}(2, c, \&h3)$
2	$w(\&h6)$	1	$w(\&h3)$
1	$R_{1,3}(2, c, \&h3)$	1	$S_{1,5}(0, "1", \&h4)$
1	$w(\&h3)$	1	$w(\&h4)$
1	$S_{1,5}(0, "1", \&h4)$	0	$R_{0,2}(2, a, \&h1)$
1	$w(\&h4)$	0	$w(\&h1)$
0	$R_{0,5}(1, b, \&h2)$	0	$R_{0,5}(1, b, \&h2)$
0	$w(\&h2)$	0	$w(\&h2)$
0	$assume(b > 0)$	0	$assume(b > 0)$
0	$assert(a == 4)$	0	$assert(a == 4)$

Execution Traces

Trace 1	
Task	Command
2	$S_{2,4}(0, "4", \&h5)$
2	$w(\&h5)$
0	$R_{0,2}(2, a, \&h1)$
0	$w(\&h1)$
2	$S_{2,6}(1, "Go", \&h6)$
2	$w(\&h6)$
1	$R_{1,3}(2, c, \&h3)$
1	$w(\&h3)$
1	$S_{1,5}(0, "1", \&h4)$
1	$w(\&h4)$
0	$R_{0,5}(1, b, \&h2)$
0	$w(\&h2)$
0	$\text{assume}(b > 0)$
0	$\text{assert}(a == 4)$ 

Trace 2	
Task	Command
2	$S_{2,4}(0, "4", \&h5)$
2	$w(\&h5)$
2	$S_{2,6}(1, "Go", \&h6)$
2	$w(\&h6)$
1	$R_{1,3}(2, c, \&h3)$
1	$w(\&h3)$
1	$S_{1,5}(0, "1", \&h4)$
1	$w(\&h4)$
0	$R_{0,2}(2, a, \&h1)$
0	$w(\&h1)$
0	$R_{0,5}(1, b, \&h2)$
0	$w(\&h2)$
0	$\text{assume}(b > 0)$
0	$\text{assert}(a == 4)$ 

SMT Encoding for Our Example – Step 1

Task 0	Task 1	Task 2
$R_{0,2}(*, a, \&h1)$	$R_{1,3}(*, c, \&h3)$	$S_{2,4}(0, "4", \&h5)$
$W(\&h1)$	$W(\&h3)$	$W(\&h5)$
$R_{0,5}(*, b, \&h2)$	$S_{1,5}(0, "1", \&h4)$	$S_{2,6}(1, "Go", \&h6)$
$W(\&h2)$	$W(\&h4)$	$W(\&h6)$
$\text{assume}(b > 0)$		
$\text{assert}(a == 4)$		

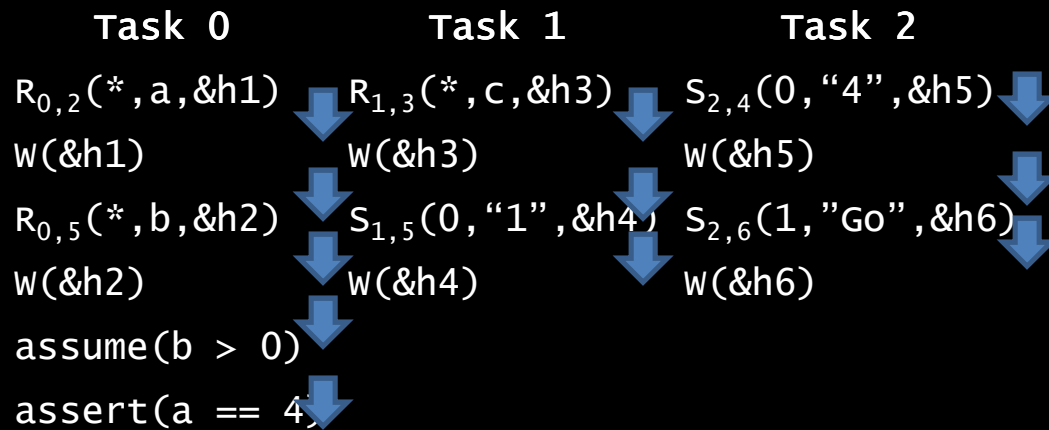
Match Pairs

$(R_{0,2}, S_{2,4})$
 $(R_{0,5}, S_{1,5})$
 $(R_{1,3}, S_{2,7})$

Definitions

```
(define R0,2.event :: int)
(define W(&h1).event :: int)
(define R0,5.event :: int)
(define W(&h2).event :: int)
(define R1,3.event :: int)
(define W(&h3).event :: int)
(define S1,5.event :: int)
(define W(&h4).event :: int)
(define S2,4.event :: int)
(define W(&h5).event :: int)
(define S2,6.event :: int)
(define W(&h6).event :: int)
(define assume.event :: int)
(define assert.event :: int)
```

SMT Encoding for Our Example – Step 2



Match Pairs

$(R_{0,2}, S_{2,4})$
 $(R_{0,5}, S_{1,5})$
 $(R_{1,3}, S_{2,7})$

Constraints

00 (HB $R_{0,2} \cdot \text{event}$ $W(\&h1) \cdot \text{event}$)
01 (HB $W(\&h1) \cdot \text{event}$ $R_{0,5} \cdot \text{event}$)
02 (HB $R_{0,5} \cdot \text{event}$ $W(\&h2) \cdot \text{event}$)
03 (HB $W(\&h2) \cdot \text{event}$ $\text{assume} \cdot \text{event}$)
04 (HB $\text{assume} \cdot \text{event}$ $\text{assert} \cdot \text{event}$)
05 (HB $R_{1,3} \cdot \text{event}$ $W(\&h3) \cdot \text{event}$)
06 (HB $W(\&h3) \cdot \text{event}$ $S_{1,5} \cdot \text{event}$)
07 (HB $S_{1,5} \cdot \text{event}$ $W(\&h4) \cdot \text{event}$)
08 (HB $S_{2,4} \cdot \text{event}$ $W(\&h5) \cdot \text{event}$)
09 (HB $W(\&h5) \cdot \text{event}$ $S_{2,6} \cdot \text{event}$)
10 (HB $S_{2,6} \cdot \text{event}$ $W(\&h6) \cdot \text{event}$)

SMT Encoding for Our Example – Step 3

Task 0	Task 1	Task 2
$R_{0,2}(*, a, \&h1)$	$R_{1,3}(*, c, \&h3)$	$S_{2,4}(0, "4", \&h5)$
$w(\&h1)$	$w(\&h3)$	$w(\&h5)$
$R_{0,5}(*, b, \&h2)$	$S_{1,5}(0, "1", \&h4)$	$S_{2,6}(1, "Go", \&h6)$
$w(\&h2)$	$w(\&h4)$	$w(\&h6)$
$\text{assume}(b > 0)$		
$\text{assert}(a == 4)$		

Match Pairs

$(R_{0,2}, S_{2,4})$
 $(R_{0,5}, S_{1,5})$
 $(R_{1,3}, S_{2,7})$

Constraints

```
00 (HB  $R_{0,2}$ .event  $w(\&h1)$ .event)
01 (HB  $w(\&h1)$ .event  $R_{0,5}$ .event)
02 (HB  $R_{0,5}$ .event  $w(\&h2)$ .event)
03 (HB  $w(\&h2)$ .event  $\text{assume.event}$ )
04 (HB  $\text{assume.event}$   $\text{assert.event}$ )
05 (HB  $R_{1,3}$ .event  $w(\&h3)$ .event)
06 (HB  $w(\&h3)$ .event  $S_{1,5}$ .event)
07 (HB  $S_{1,5}$ .event  $w(\&h4)$ .event)
08 (HB  $S_{2,4}$ .event  $w(\&h5)$ .event)
09 (HB  $w(\&h5)$ .event  $S_{2,6}$ .event)
10 (HB  $S_{2,6}$ .event  $w(\&h6)$ .event)
11 ( $\text{assert}(> b\ 0)$ )
12 ( $\text{assert}(\text{not}(= a\ 4))$ )
```

SMT Encoding for Our Example – Step 4

Task 0	Task 1	Task 2
$R_{0,2}(*, a, \&h1)$	$R_{1,3}(*, c, \&h3)$	$S_{2,4}(0, "4", \&h5)$
$w(\&h1)$	$w(\&h3)$	$w(\&h5)$
$R_{0,5}(*, b, \&h2)$	$S_{1,5}(0, "1", \&h4)$	$S_{2,6}(1, "Go", \&h6)$
$w(\&h2)$	$w(\&h4)$	$w(\&h6)$
$\text{assume}(b > 0)$		
$\text{assert}(a == 4)$		

Match Pairs

$(R_{0,2}, S_{2,4})$
 $(R_{0,5}, S_{1,5})$
 $(R_{1,3}, S_{2,7})$

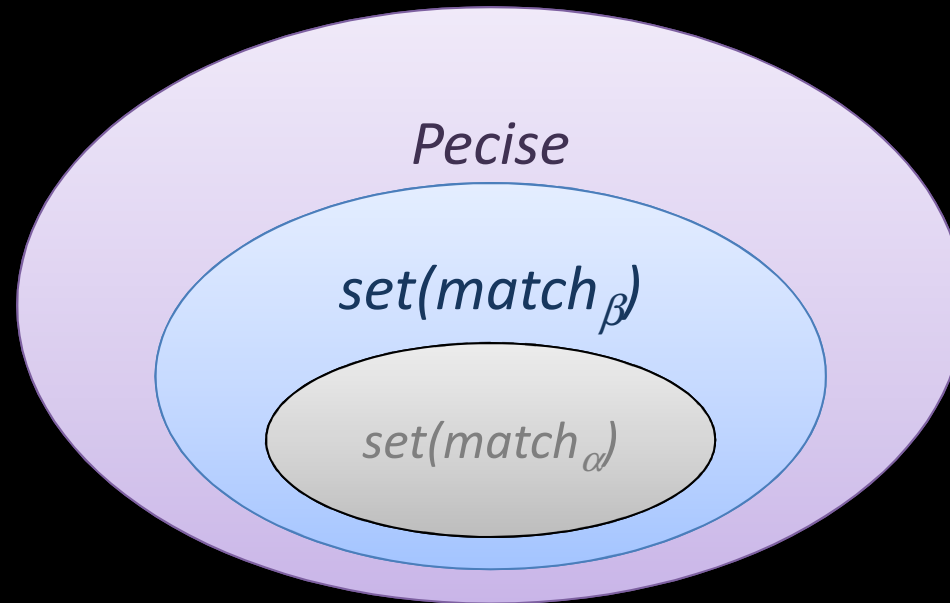
Constraints

```
00 (HB  $R_{0,2} \cdot \text{event}$   $w(\&h1) \cdot \text{event}$ )
01 (HB  $w(\&h1) \cdot \text{event}$   $R_{0,5} \cdot \text{event}$ )
02 (HB  $R_{0,5} \cdot \text{event}$   $w(\&h2) \cdot \text{event}$ )
03 (HB  $w(\&h2) \cdot \text{event}$   $\text{assume} \cdot \text{event}$ )
04 (HB  $\text{assume} \cdot \text{event}$   $\text{assert} \cdot \text{event}$ )
05 (HB  $R_{1,3} \cdot \text{event}$   $w(\&h3) \cdot \text{event}$ )
06 (HB  $w(\&h3) \cdot \text{event}$   $S_{1,5} \cdot \text{event}$ )
07 (HB  $S_{1,5} \cdot \text{event}$   $w(\&h4) \cdot \text{event}$ )
08 (HB  $S_{2,4} \cdot \text{event}$   $w(\&h5) \cdot \text{event}$ )
09 (HB  $w(\&h5) \cdot \text{event}$   $S_{2,6} \cdot \text{event}$ )
10 (HB  $S_{2,6} \cdot \text{event}$   $w(\&h6) \cdot \text{event}$ )
11 ( $\text{assert}(> b\ 0)$ )
12 ( $\text{assert}(\text{not}(= a\ 4))$ )
```

Match

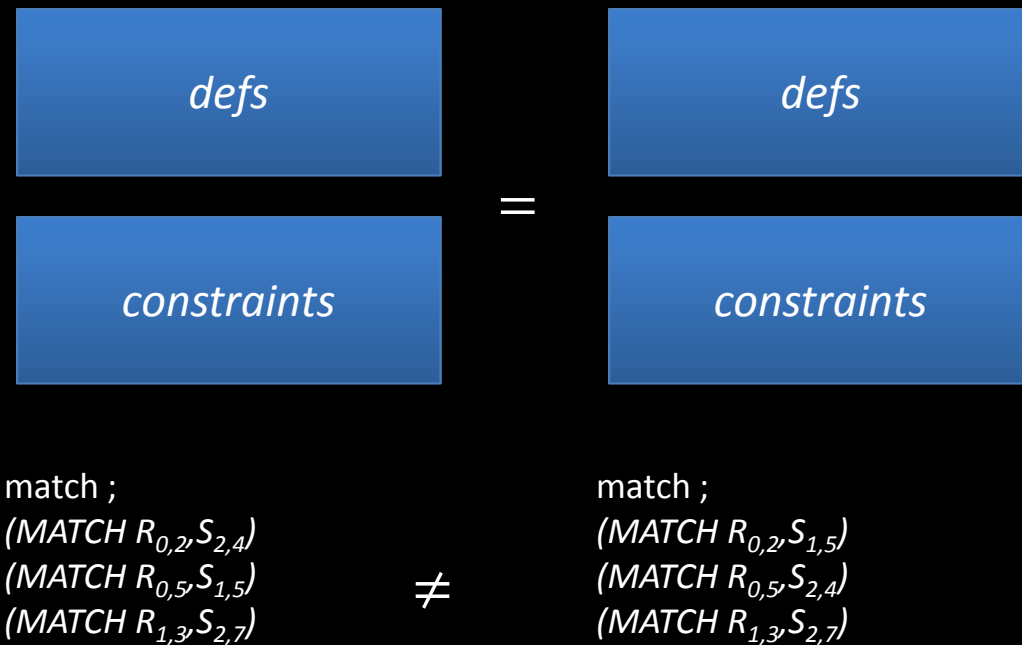
```
13 (MATCH  $R_{0,2}, S_{2,4}$ )
14 (MATCH  $R_{0,5}, S_{1,5}$ )
15 (MATCH  $R_{1,3}, S_{2,7}$ )
```


Theorem 1

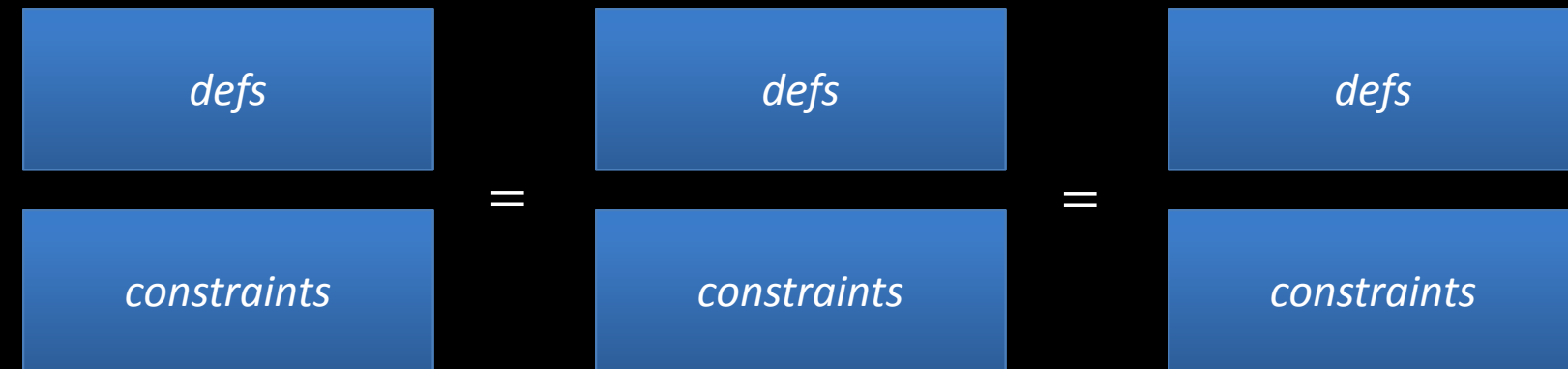


$$Ans(smt_{\alpha}) \leq Ans(smt_{\beta}) \leq Ans(Precise)$$

Theorem 1 – Proof Sketch



Theorem 1 – Proof Sketch



match ;
 (or (MATCH $R_{0,2} S_{2,4}$)
 (MATCH $R_{0,2} S_{1,5}$))
 (or (MATCH $R_{0,5} S_{2,4}$)
 (MATCH $R_{0,5} S_{1,5}$))
 (MATCH $R_{1,3} S_{2,7}$)
 (NE $R_{0,2} R_{0,5}$)

≠

match ;
 (MATCH $R_{0,2} S_{2,4}$)
 (MATCH $R_{0,5} S_{1,5}$)
 (MATCH $R_{1,3} S_{2,7}$)

≠

match ;
 (MATCH $R_{0,2} S_{1,5}$)
 (MATCH $R_{0,5} S_{2,4}$)
 (MATCH $R_{1,3} S_{2,7}$)

Theorem 1 – Proof Sketch

defs

constraints

match ;
(or (MATCH $R_{0,2} S_{2,4}$)
(MATCH $R_{0,2} S_{1,5}$))
(or (MATCH $R_{0,5} S_{2,4}$)
(MATCH $R_{0,5} S_{1,5}$))
(MATCH $R_{1,3} S_{2,7}$)
(NE $R_{0,2} R_{0,5}$)

SAT

defs

constraints

match ;
(MATCH $R_{0,2} S_{2,4}$)
(MATCH $R_{0,5} S_{1,5}$)
(MATCH $R_{1,3} S_{2,7}$)

UNSAT

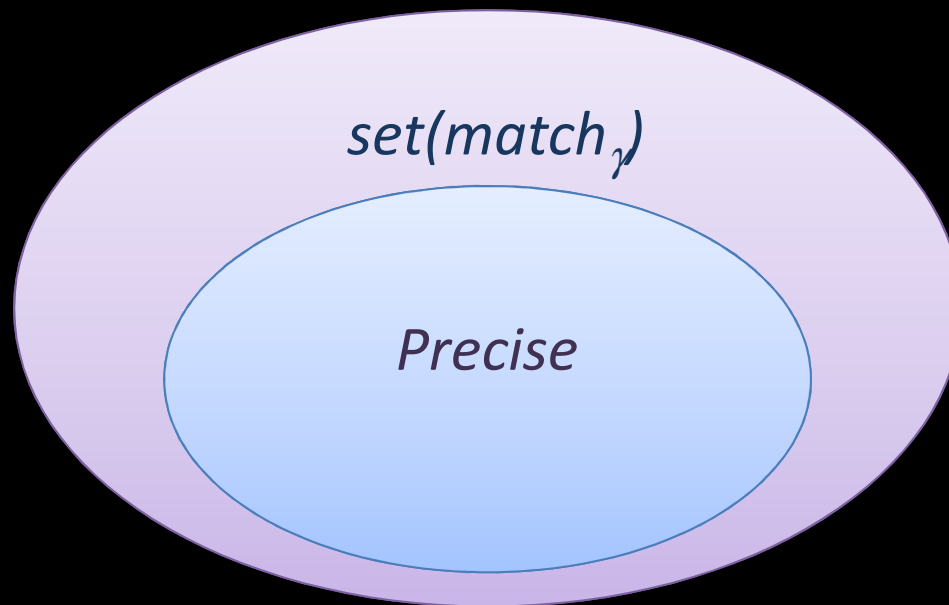
defs

constraints

match ;
(MATCH $R_{0,2} S_{1,5}$)
(MATCH $R_{0,5} S_{2,4}$)
(MATCH $R_{1,3} S_{2,7}$)

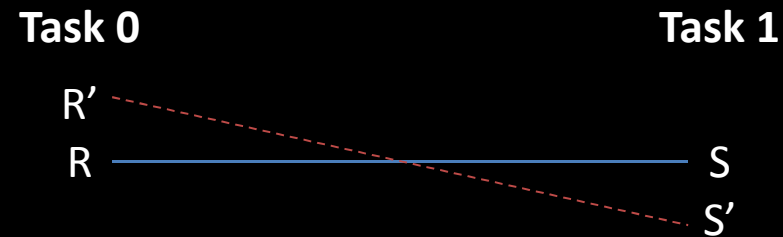
SAT

Theorem 2



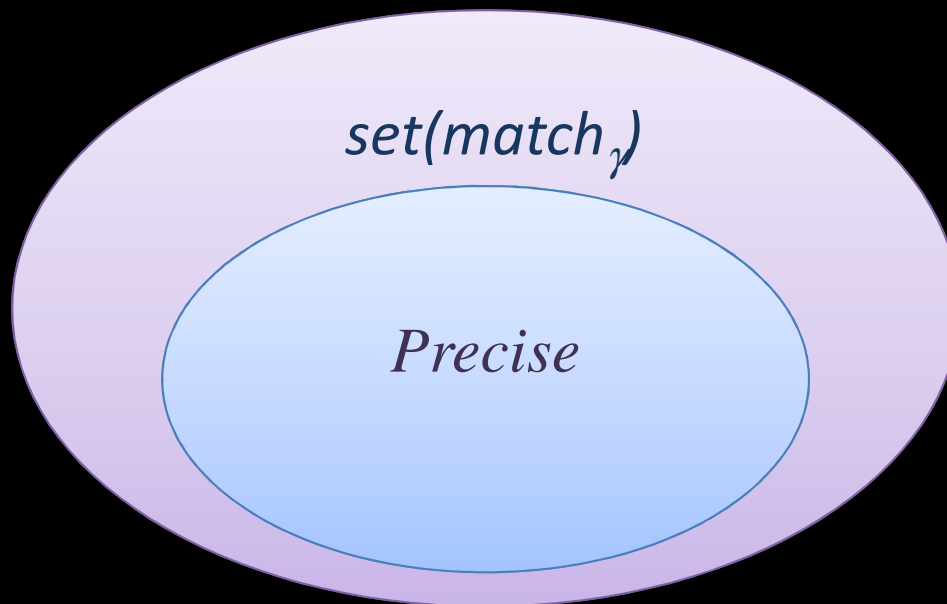
$$Ans(smt_\gamma) = SAT \rightarrow Ans(Precise) = SAT$$

Theorem 2 – Proof Sketch



Match R and S in the generated encoding will make the encoding unsat

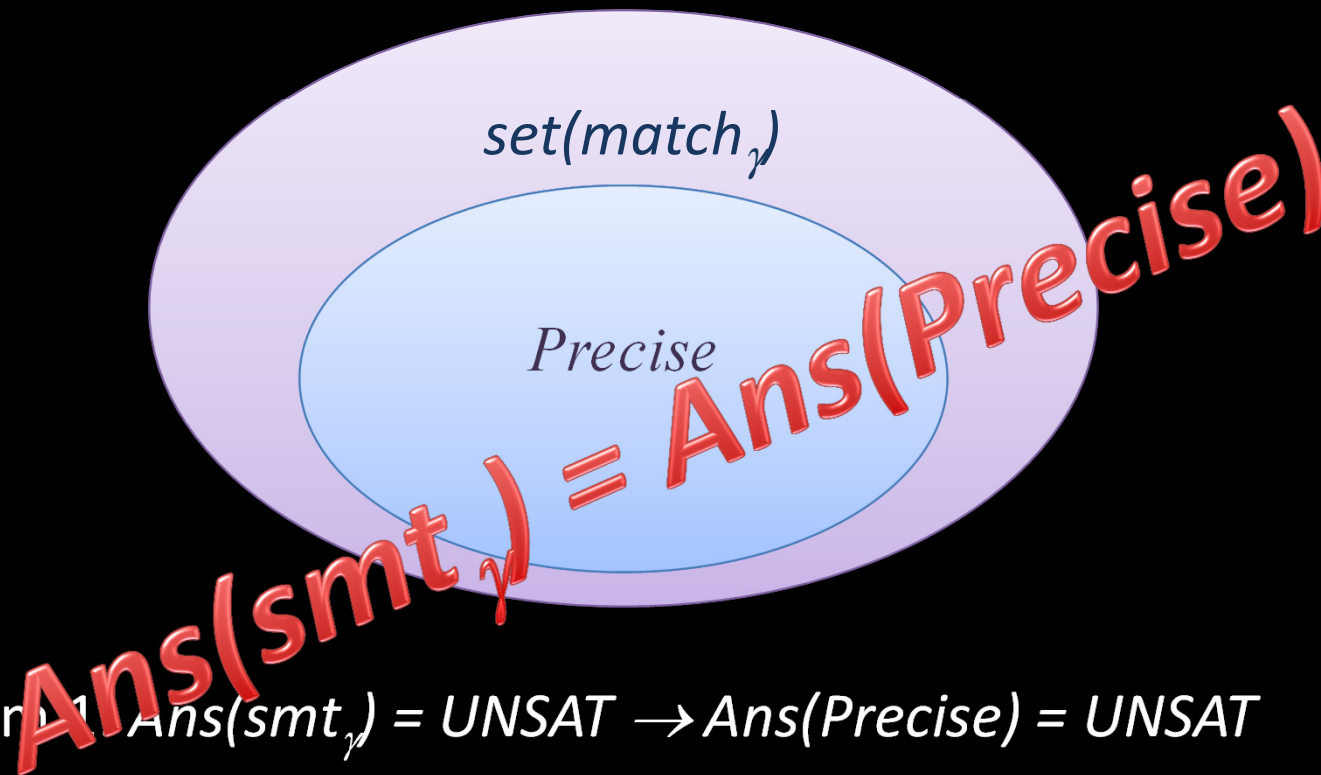
Theorem 1 & 2



Theorem 1: $Ans(smt_\gamma) = UNSAT \rightarrow Ans(Precise) = UNSAT$

Theorem 2: $Ans(smt_\gamma) = SAT \rightarrow Ans(Precise) = SAT$

Theorem 1 & 2



Theorem 1: $Ans(smt_\gamma) = UNSAT \rightarrow Ans(Precise) = UNSAT$

Theorem 2: $Ans(smt_\gamma) = SAT \rightarrow Ans(Precise) = SAT$

Generating Match Pairs

Task 0

$R_{0,1}(*, a, \&h1)$

$R_{0,2}(*, b, \&h2)$

$S_{0,3}(1, "1", \&h3)$

$R_{0,4}(*, c, \&h4)$

Task 1

$S_{1,1}(0, "2", \&h5)$

$R_{1,2}(*, d, \&h6)$

$S_{1,3}(0, "3", \&h7)$

Task 2

$S_{2,1}(0, "4", \&h8)$

Generating Match Pairs

Task 0

$R_{0,1}(*, a, \&h1)$

$R_{0,2}(*, b, \&h2)$

$S_{0,3}(1, "1", \&h3)$

$R_{0,4}(*, c, \&h4)$

Task 1

$S_{1,1}(0, "2", \&h5)$

$R_{1,2}(*, d, \&h6)$

$S_{1,3}(0, "3", \&h7)$

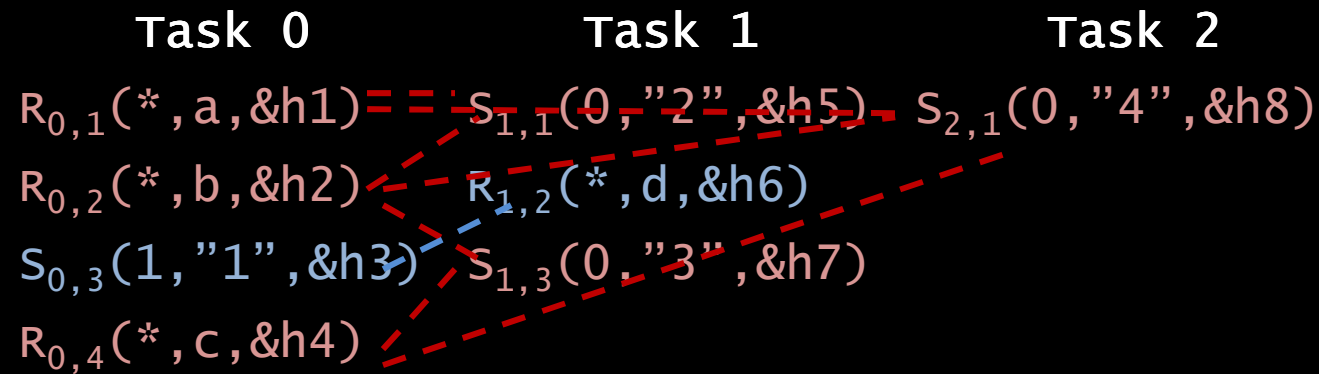
Task 2

$S_{2,1}(0, "4", \&h8)$

Receive List: $(0 \rightarrow ((0, R_{0,1}), (1, R_{0,2}), (2, R_{0,4})))$
 $(1 \rightarrow ((0, R_{1,2})))$

Send List: $(0 \rightarrow ((1 \rightarrow ((0, S_{1,1}), (1, S_{1,3})), (2 \rightarrow ((0, S_{2,1}))))))$
 $(1 \rightarrow ((0 \rightarrow ((0, S_{0,3}))))))$

Generating Match Pairs



Rule 1: $ep_s = ep_r$

Rule 2: $l_r \geq l_s$

Rule 3: $l_r \leq l_s + (n_s(*, dst) - n_s(src, dst))$

Experimental Results

	Program Order	Matches	Assume&Assert	Extra
Our Encoding	11	4	2	0
Elwakil's Encoding	22	13	3	8

Test Programs		Our Encoding		Elwakil's Encoding	
Name	#Msg	Property	#Clauses	Property	#Clauses
EP	3	sat	17	unsat	47
Small1	2	unsat	8	unsat	33
Small2	1	unsat	4	unsat	18
small3	3	Sat	11	unsat	44

Experimental Results

Test Programs		Our Encoding		Elwakil's Encoding	
Name	#Msg	Time(ms)	Memory(MB)	Time(ms)	Memory(MB)
Leader	30	120	25.473	—	—
5ites	15	72	18.519	392	37.218
6ites	18	72	18.918	636	48.703
7ites	21	80	19.375	940	62.718
8ites	24	96	20.168	—	—

Conclusions

- An SMT encoding of an MCAPI program execution that uses match pairs;
- The first encoding that correctly captures the non-deterministic behavior of an MCAPI program execution under infinite-buffer semantics allowed in the MCAPI specification;
- Such an encoding can be generated by only giving an execution trace and an over-approximated set of match pairs;
- An algorithm with $O(N^2)$ time complexity that over-approximates the true set of match pairs;
- The experiment shows that our encoding reduces 70% of that of the existing work, and runs average eight times faster and uses two times less memory.

Q & A?