Stuart Siroky

CS5392 Project – Model Checker

Test case description and results:

Test Case 1: Kripke Structure from Logic in Computer Science Ch 3.6 pg 226

Kripke Structure – testKripkeFile2.txt

CTL Formula – E[ not c2 U c1 ]

Starting State – s0

Results –

====================

Submitted Formula : E[ not c2 U c1 ]

====================

====================Results of labeling ====================

State : s3

Prepositions : true, t1, t2,

Satifies : not c2, E[ not c2 U c1 ],

State : s9

Prepositions : true, t1, t2,

Satifies : not c2,

State : s7

Prepositions : true, t1, c2,

Satifies : c2,

State : s6

Prepositions : true, n1, c2,

Satifies : c2,

State : s5

Prepositions : true, n1, t2,

Satifies : not c2,

State : s4

Prepositions : true, c1, t2,

Satifies : c1, not c2, E[ not c2 U c1 ],

State : s2

Prepositions : true, c1, n2,

Satifies : c1, not c2, E[ not c2 U c1 ],

State : s1

Prepositions : true, t1, n2,

Satifies : not c2, E[ not c2 U c1 ],

State : s0

Prepositions : true, n1, n2,

Satifies : not c2, E[ not c2 U c1 ],

====================

The formula: { E[ not c2 U c1 ] } \*\*SATIFIES\*\*

the kripke structure starting from state s0

====================

Test Case 2: Kripke Structure from modelCheckingAlg lecture

Kripke Structure – testKripkeFile6.txt

CTL Formula – AG(t1 implies AF c1)

Starting State – s0

Results –

====================

Submitted Formula : not E[ true U ( t1 and not AF c1 ) ]

====================

====================

The formula: { not E[ true U ( t1 and not AF c1 ) ] } \*\*SATIFIES\*\*

the kripke structure starting from state s0

====================

====================Results of labeling ====================

State : s3

Prepositions : true, t1, t2,

Satifies : AF c1, t1, true, not E[ true U ( t1 and not AF c1 ) ],

State : s9

Prepositions : true, t1, t2,

Satifies : AF c1, t1, true, not E[ true U ( t1 and not AF c1 ) ],

State : s7

Prepositions : true, t1, c2,

Satifies : AF c1, t1, true, not E[ true U ( t1 and not AF c1 ) ],

State : s6

Prepositions : true, n1, c2,

Satifies : AF c1, true, not E[ true U ( t1 and not AF c1 ) ],

State : s5

Prepositions : true, n1, t2,

Satifies : AF c1, true, not E[ true U ( t1 and not AF c1 ) ],

State : s4

Prepositions : true, c1, t2,

Satifies : c1, AF c1, true, not E[ true U ( t1 and not AF c1 ) ],

State : s2

Prepositions : true, c1, n2,

Satifies : c1, AF c1, true, not E[ true U ( t1 and not AF c1 ) ],

State : s1

Prepositions : true, t1, n2,

Satifies : AF c1, t1, true, not E[ true U ( t1 and not AF c1 ) ],

State : s0

Prepositions : true, n1, n2,

Satifies : AF c1, true, not E[ true U ( t1 and not AF c1 ) ],

====================

The formula: { not E[ true U ( t1 and not AF c1 ) ] } \*\*SATIFIES\*\*

the kripke structure starting from state s0

====================

Test Case 3: Kripke Structure from Project description test AF

Kripke Structure – testKripkeFile1.txt

CTL Formula – AF t

Starting State – s3

Results –

====================

Submitted Formula : AF t

====================

====================

Checking formula with starting state : s3

====================

====================Results of labeling ====================

State : s2

Prepositions : true, q, t, r,

Satifies : t, AF t,

State : s4

Prepositions : true, t,

Satifies : t, AF t,

State : s3

Prepositions : true,

Satifies : AF t,

====================

The formula: { AF t } \*\*SATIFIES\*\*

the kripke structure starting from state s3

====================

Test Case 4: Kripke Stucture from Project description test EX

Kripke Structure – testKripkeFile1.txt

CTL Formula – EX t

Starting State – s1

Results –

====================

Submitted Formula : EX t

====================

====================

Checking formula with starting state : s1

====================

====================Results of labeling ====================

State : s4

Prepositions : true, t,

Satifies : t, EX t,

State : s3

Prepositions : true,

Satifies :

State : s2

Prepositions : true, q, t, r,

Satifies : t,

State : s1

Prepositions : true, p, q,

Satifies : EX t,

====================

The formula: { EX t } \*\*SATIFIES\*\*

the kripke structure starting from state s1

====================

Test Case 5: Kripke Structure from Project description test EU

Kripke Structure – testKripkeFile1.txt

CTL Formula – E[ p U q ]

Starting State – s1

Results –

====================

Submitted Formula : E[ p U q ]

====================

====================

Checking formula with starting state : s1

====================

====================Results of labeling ====================

State : s4

Prepositions : true, t,

Satifies :

State : s3

Prepositions : true,

Satifies :

State : s2

Prepositions : true, q, t, r,

Satifies : q, E[ p U q ],

State : s1

Prepositions : true, p, q,

Satifies : q, p, E[ p U q ],

====================

The formula: { E[ p U q ] } \*\*SATIFIES\*\*

the kripke structure starting from state s1

====================

Test Case 6: Fail for testing AF

Kripke Structure – testKripkeFile3.txt

CTL Formula – AF t

Starting State – s1

Results –

====================

Submitted Formula : AF t

====================

====================

Checking formula with starting state : s1

====================

====================Results of labeling ====================

State : s4

Prepositions : true, t,

Satifies : t, AF t,

State : s3

Prepositions : true,

Satifies : AF t,

State : s2

Prepositions : true, q, t, r, s,

Satifies : t, AF t,

State : s6

Prepositions : true, p, r,

Satifies :

State : s5

Prepositions : true, p,

Satifies :

State : s1

Prepositions : true, p, q,

Satifies :

====================

The formula: { AF t } \*\*FAILS\*\* to satifies

the kripke structure starting from state s1

====================

Test Case 7: Fail for EX

Kripke Structure – testKripikeFile1.txt

CTL Formula – EX (r and t)

Starting State – s2

Results –

====================

Submitted Formula : EX ( r and t )

====================

====================

Checking formula with starting state : s2

====================

====================Results of labeling ====================

State : s4

Prepositions : true, t,

Satifies : t, EX ( r and t ),

State : s3

Prepositions : true,

Satifies :

State : s2

Prepositions : true, q, t, r,

Satifies : t, r, ( r and t ),

====================

The formula: { EX ( r and t ) } \*\*FAILS\*\* to satifies

the kripke structure starting from state s2

====================

Test Case 8: Fail for EU

Kripke Structure – testKripkeFile1.txt

CTL Formula – E[ t U p]

Starting State – s3

Results –

====================

Submitted Formula : E[ p U r ]

====================

====================

Checking formula with starting state : s3

====================

====================Results of labeling ====================

State : s2

Prepositions : true, q, t, r,

Satifies : r, E[ p U r ],

State : s4

Prepositions : true, t,

Satifies :

State : s3

Prepositions : true,

Satifies :

====================

The formula: { E[ p U r ] } \*\*FAILS\*\* to satifies

the kripke structure starting from state s3

====================

Test Case 9: Pass for EU

Kripke Structure – testKripkeFile1.txt

CTL Formula – E[ t U p]

Starting State – s2

Results –

====================

Submitted Formula : E[ t U p ]

====================

====================

Checking formula with starting state : s2

====================

====================Results of labeling ====================

State : s4

Prepositions : true, t,

Satifies :

State : s3

Prepositions : true,

Satifies :

State : s2

Prepositions : true, q, t, r,

Satifies : r, E[ p U r ],

====================

The formula: { E[ p U r ] } \*\*SATIFIES\*\*

the kripke structure starting from state s2

====================

Test Case 10: Fail EU with a not

Kripke Structure – testKripkeFile1.txt

CTL Formula – E[ t U not p]

Starting State – s1

Results –

====================

Submitted Formula : E[ t U not p ]

====================

====================

Checking formula with starting state : s1

====================

====================Results of labeling ====================

State : s4

Prepositions : true, t,

Satifies : not p, t, E[ t U not p ],

State : s3

Prepositions : true,

Satifies : not p, E[ t U not p ],

State : s2

Prepositions : true, q, t, r,

Satifies : not p, t, E[ t U not p ],

State : s1

Prepositions : true, p, q,

Satifies : p,

====================

The formula: { E[ t U not p ] } \*\*FAILS\*\* to satifies

the kripke structure starting from state s1

====================

Test Case 11: PASS EU with a not

Kripke Structure – testKripkeFile1.txt

CTL Formula – E[ t U not p]

Starting State – s2

Results –

====================

Submitted Formula : E[ t U not p ]

====================

====================

Checking formula with starting state : s2

====================

====================Results of labeling ====================

State : s4

Prepositions : true, t,

Satifies : not p, t, E[ t U not p ],

State : s3

Prepositions : true,

Satifies : not p, E[ t U not p ],

State : s2

Prepositions : true, q, t, r,

Satifies : not p, t, E[ t U not p ],

====================

The formula: { E[ t U not p ] } \*\*SATIFIES\*\*

the kripke structure starting from state s2

====================

Syntax Test Case 1: Unbalanced Paretheses

CTLFormula – AF( p and not (q or r )

Results – message output : Unbalanced Parentheses.

Syntax Test Case 2: Unmatched until

CTLFormula – E[ x y]

Results – message output: CTL Until not formed correctly. Ie A[p U q]

Syntax Test Case 3: No CTL operator

CTLFormula – p and q

Results – message output: No CTL quantifiers found ie AG().

Syntax Test Case 4: LTL operator

CTLFormula – AG( p and F q )

Results – Operator without A or E qualifers.

Conversion Test Case 1: Convert false & true

CTLFormula – AF( true and false)

Results – AF ( true and not true )

Conversion Test Case 2: Convert not ( no conversion )

CTLFormula – AF (not p)

Results – AF not p

Conversion Test Case 3: Convert or

CTLFormula – AF ( p or q )

Results – AF not ( not p and not q )

Conversion Test Case 4: Convert and ( no conversion )

CTLFormula – AF ( p and q )

Results – AF (p and q)

Conversion Test Case 5: Convert implies

CTLFormula – AF ( p implies q )

Results – AF not ( p and not q )

Conversion Test Case 6: Convert AG

CTLFormula – AG p

Results – not E[ true U not p]

Conversion Test Case 7: Convert AX

CTLFormula – AX p

Results – not EX not p

Conversion Test Case 8: Convert AF (no conversion)

CTLFormula – AF p

Results – AF p

Conversion Test Case 9: Convert AU

CTLFormula – A[ p U q]

Results – not ( E[ not q U not ( p and q ) ] and AF q )

Conversion Test Case 10: Convert EG

CTLFormula – EG p

Results – not AF not p

Conversion Test Case 11: Convert EX ( no conversion )

CTLFormula – EX p

Results – EX p

Conversion Test Case 12: Convert EF

CTLFormula – EF p

Results – E[ true U p]

Conversion Test Case 13: Convert EU ( no conversion )

CTLFormula – E[ p U q]

Results – E[ p U q ]

Conversion Test Case 14: Convert AF p and q

Results – Malformed CTL Formula, not all element begin with CTL.

File Test Case 1: Not correct Deliminers in Kripke Structure file.

Kripke Structure – testKripkeFileERR1.txt

Results – States p and null dont exist and can't be added to transition list

File Test Case 2: Not correct Deliminers in Kripke Structure file.

Kripke Structure – testKripkeFileERR2.txt

Results – State s3 doesnt exist and can't be added to transition list

File Test Case 3: Not correct Deliminers in Kripke Structure file.

Kripke Structure – testKripkeFileERR3.txt

Results – States p and null dont exist and can't be added to transition list

File Test Case 2: Transition to non existing state

Kripke Structure – testKripkeFileERR4.txt

Results – State s8 doesnt exist and can't be added to transition list

Run Test Case 1: Select a starting state that does not exist in Kripke Structure.

Kripke Structure – testKripkeFile1.txt

CTLFormula – EF p

Starting State – s5

Results – No state s5 is avaliable in Kripke structure

Run Test Case 2: Select starting state without loading kripke structure.

CTLFormula – EX p

Starting State – s5

Results – Please load a valid kripke stucture.

Run Test Case 3: Select starting state without entering a CTL formula.

KripkeStructure – testKripkeFile1.txt

Starting State – s1

Results – Please load a valid CTL formula.

Run Test Case 4: Select a starting state without entering a CTL formula or KripkeStructure.

Starting State – s5

Results – Please load a valid kripke stucture.

Please load a valid CTL formula.