**Purpose**

This document describes test cases for the CTL Model Checker project.

**Scope**

This document contains a complete, cumulative set of test cases used in testing version 1.0 of the CTL Model Checker application.

**Result of Test Cases:** PASS 🗹 FAIL □ (Provide comments for failure)

**Comments:**

All test files are located at …\ModelCheckCTL\ModelCheckCTL.Test\Test Files

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test #** | |  |  |  | | --- | --- | --- | | **Actions** | **Expected** | **Attachment (Witness)** | | **Results** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | |  |  |  | | --- | --- | --- | | **This test demonstrates that CTL model return the correct result for a given model and state** | | | | 1. Run the application. | A command line input asking to enter the location for the CTL formula  model1 | T1.1 | | 1. Enter the State ID | A command line input asking to enter the state ID  s1 | T1.2 | | 1. Enter the CTL Formula | A command line input asking to enter the CTL formula  EGp and AGq | T1.3 | | Output of the entered CTL formula | EGp and AGq  Original Expression: EGp & AGq  Type SAT: And  Left Expression: EGp  Right Expression: AGq  -----------------------------------  Original Expression: EGp  Type SAT: EG  Left Expression: p  Right Expression:  -----------------------------------  Original Expression: !AF!p  Type SAT: Not  Left Expression: AF!p  Right Expression:  -----------------------------------  Original Expression: AF!p  Type SAT: AF  Left Expression: !p  Right Expression:  -----------------------------------  Original Expression: !p  Type SAT: Not  Left Expression: p  Right Expression:  -----------------------------------  Original Expression: p  Type SAT: Atomic  Left Expression: p  Right Expression:  -----------------------------------  Original Expression: AGq  Type SAT: AG  Left Expression:  Right Expression: q  -----------------------------------  Original Expression: !EF!  Type SAT: Not  Left Expression: EF!  Right Expression:  -----------------------------------  Original Expression: EF!  Type SAT: EF  Left Expression: !  Right Expression:  -----------------------------------  Original Expression: E(TU!)  Type SAT: EU  Left Expression: T  Right Expression: !  -----------------------------------  Original Expression: T  Type SAT: AllTrue  Left Expression: T  Right Expression:  -----------------------------------  Original Expression: !  Type SAT: Not  Left Expression:  Right Expression:  -----------------------------------  Original Expression:  Type SAT: Atomic  Left Expression:  Right Expression:  -----------------------------------  Property EGp and AGq does not hold in state s1! | T1.4 | | |  | | --- | | **Result** | | Pass 🗹Fail 🞎N/A 🞎 | | **Initials/Date** | | VL/06/29/2020 | |
| T2 | |  |  |  | | --- | --- | --- | | **This test demonstrates that given the proper model input kripke structure file, valid State ID and CTL formula, the system checks the formula for the given state ID and return the correct result as expected.**  **For example:**   * Property EGp and AGq does not hold in state s1!   **means that state s1 does NOT hold for formula EGp and AGq** | | | | 1. Open ModelCheckCTL.Test. | -- | -- | | 1. Print Model 1.txt | s1, s2, s3, s4;  t1 : s1 - s2,  t2 : s1 - s3,  t3 : s3 - s4,  t4 : s4 - s2,  t5 : s2 - s3;  s1 : p q,  s2 : q t r,  s3 : ,  s4 : t; | T2.2 | | 1. Print Model 1 – Test Formulas.txt | s1;EGp and AGq;False  s2;EGp and AGq;False  s3;EGp and AGq;False  s4;EGp and AGq;False  s1;(not AGp) or EFq;True  s2;(not AGp) or EFq;True  s3;(not AGp) or EFq;True  s4;(not AGp) or EFq;True  s1;EG(r->t);True  s2;EG(r->t);True  s3;EG(r->t);True  s4;EG(r->t);True  s1;AX(r->p);False  s2;AX(r->p);True  s3;AX(r->p);True  s4;AX(r->p);False  s1;AXq;False  s2;AXq;False  s3;AXq;False  s4;AXq;True  s1;EXq;True  s2;EXq;False  s3;EXq;False  s4;EXq;True  s1;not AXq;True  s2;not AXq;True  s3;not AXq;True  s4;not AXq;False  s1;not EXq;False  s2;not EXq;True  s3;not EXq;True  s4;not EXq;False  s1;A(pUq);True  s2;A(pUq);True  s3;A(pUq);False  s4;A(pUq);False  s1;E(pUq);True  s2;E(pUq);True  s3;E(pUq);False  s4;E(pUq);False  s1;AXq and A(pUq);False  s2;AXq and A(pUq);False  s3;AXq and A(pUq);False  s4;AXq and A(pUq);False  s1;AXq or A(pUq);True  s2;AXq or A(pUq);True  s3;AXq or A(pUq);False  s4;AXq or A(pUq);True  s1;EFr;True  s2;EFr;True  s3;EFr;True  s4;EFr;True  s1;AFr;True  s2;AFr;True  s3;AFr;True  s4;AFr;True  s1;EGt;False  s2;EGt;False  s3;EGt;False  s4;EGt;False  s1;AGq;False  s2;AGq;False  s3;AGq;False  s4;AGq;False  s1;AX((EFp)or(AFr));True  s2;AX((EFp)or(AFr));True  s3;AX((EFp)or(AFr));True  s4;AX((EFp)or(AFr));True  s1;EX((AFp)or(EFr));True  s2;EX((AFp)or(EFr));True  s3;EX((AFp)or(EFr));True  s4;EX((AFp)or(EFr));True  s1;A(pUA(qUr));False  s2;A(pUA(qUr));True  s3;A(pUA(qUr));False  s4;A(pUA(qUr));False  s1;E(A(qUr)Ut);False  s2;E(A(qUr)Ut);True  s3;E(A(qUr)Ut);False  s4;E(A(qUr)Ut);True  s1;AG(p->A(pU(not p and A(not pUq))));True  s2;AG(p->A(pU(not p and A(not pUq))));True  s3;AG(p->A(pU(not p and A(not pUq))));True  s4;AG(p->A(pU(not p and A(not pUq))));True | T2.3 | | 1. Print Model 4.txt | s1, s2, s3, s4, s5, s6, s7, s8, s9;  t1 : s1 - s2,  t2 : s2 - s3,  t3 : s3 - s4,  t4 : s5 - s4,  t5 : s2 - s5,  t6 : s4 - s6,  t7 : s8 - s2,  t8 : s1 - s6,  t9 : s6 - s7,  t10 : s7 - s8,  t11 : s9 - s8,  t12 : s6 - s9;  s1 : n1 n2 0,  s2 : t1 n2 1,  s3 : c1 n2 1,  s4 : c1 t2 1,  s5 : t1 t2 1,  s6 : n1 t2 2,  s7 : t1 t2 2,  s8 : t1 c2 2,  s9 : n1 c2 2; | T2.8 | | 1. Print Model 4 – Test Formulas.txt | s1;AG(t1 -> AF c1);True  s2;AG(t1 -> AF c1);True  s3;AG(t1 -> AF c1);True  s4;AG(t1 -> AF c1);True  s5;AG(t1 -> AF c1);True  s6;AG(t1 -> AF c1);True  s7;AG(t1 -> AF c1);True  s8;AG(t1 -> AF c1);True  s9;AG(t1 -> AF c1);True | T2.9 | | 1. Print Model Microwave.txt | s1, s2, s3, s4, s5, s6, s7;  t1 : s1 - s2,  t2 : s1 - s3,  t3 : s3 - s1,  t4 : s4 - s1,  t5 : s4 - s4,  t6 : s4 - s3,  t7 : s2 - s5,  t8 : s5 - s2,  t9 : s5 - s3,  t10 : s3 - s6,  t11 : s6 - s7,  t12 : s7 - s4;  s1 : ,  s2 : s e,  s3 : c,  s4 : c h,  s5 : s c e,  s6 : s c,  s7 : s c h; | T2.10 | | 1. Print Model Microwave – Test Formuals | s1;EFe and !AXe;True s2;EXh;False s3;AX((EFe)and(AFh));False s4;AX(!h->!e);True s5;!EF(c and h);False s6;EXs and EXc;True s7;AF(c or h);True |  | | 1. Run all test formulas with their expected results as described above. | * All formulas matched their expected results. * All test results are marked as passed | T2.16 | | |  | | --- | | **Result** | | Pass 🗹Fail 🞎N/A 🞎 | | **Initials/Date** | | VL/06/29/2020 | |