

```
-- The CXX compiler identification is GNU 12.2.0
-- The C compiler identification is GNU 12.2.0
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Success
-- Found Threads: TRUE
-- Configuring done
-- Generating done
-- Build files have been written to: /home/jube/Univ/theorie_des_langages/projet/corr/2023/sandbox-tests/build
[ 25%] Building CXX object CMakeFiles/testfa.dir/Automaton.cc.o
[ 50%] Building CXX object CMakeFiles/testfa.dir/testfa.cc.o
[ 75%] Building CXX object CMakeFiles/testfa.dir/googletest/googletest/src/gtest-all.cc.o
[100%] Linking CXX executable testfa
[100%] Built target testfa
==142799== Memcheck, a memory error detector
==142799== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==142799== Using Valgrind-3.19.0 and LibVEX; rerun with -h for copyright info
==142799== Command: ./testfa
==142799==
[=====] Running 79 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 79 tests from AutomatonTest
[ RUN      ] AutomatonTest.TestIsValid_TRUE
[       OK ] AutomatonTest.TestIsValid_TRUE (9 ms)
[ RUN      ] AutomatonTest.TestIsValid_FALSE
[       OK ] AutomatonTest.TestIsValid_FALSE (0 ms)
[ RUN      ] AutomatonTest.TestIsValid_TRUE_ASCII
[       OK ] AutomatonTest.TestIsValid_TRUE_ASCII (2 ms)
[ RUN      ] AutomatonTest.TestHasSymbol_TRUE
[       OK ] AutomatonTest.TestHasSymbol_TRUE (0 ms)
[ RUN      ] AutomatonTest.TestHasSymbol_FALSE_NoSymbolAdded
[       OK ] AutomatonTest.TestHasSymbol_FALSE_NoSymbolAdded (0 ms)
[ RUN      ] AutomatonTest.TestHasSymbol_FALSE_InvalidChar
[       OK ] AutomatonTest.TestHasSymbol_FALSE_InvalidChar (1 ms)
[ RUN      ] AutomatonTest.TestHasSymbol_FALSE
[       OK ] AutomatonTest.TestHasSymbol_FALSE (0 ms)
[ RUN      ] AutomatonTest.TestAddSymbol_TRUE
[       OK ] AutomatonTest.TestAddSymbol_TRUE (0 ms)
[ RUN      ] AutomatonTest.TestAddSymbol_FALSE_InvalidChar
[       OK ] AutomatonTest.TestAddSymbol_FALSE_InvalidChar (0 ms)
[ RUN      ] AutomatonTest.TestAddSymbol_FALSE_AlreadyPresent
[       OK ] AutomatonTest.TestAddSymbol_FALSE_AlreadyPresent (0 ms)
[ RUN      ] AutomatonTest.TestRemoveSymbol_TRUE
[       OK ] AutomatonTest.TestRemoveSymbol_TRUE (1 ms)
[ RUN      ] AutomatonTest.TestRemoveSymbol_FALSE_InvalidChar
[       OK ] AutomatonTest.TestRemoveSymbol_FALSE_InvalidChar (1 ms)
[ RUN      ] AutomatonTest.TestRemoveSymbol_FALSE_NotPresent
[       OK ] AutomatonTest.TestRemoveSymbol_FALSE_NotPresent (0 ms)
[ RUN      ] AutomatonTest.TestCountSymbols
[       OK ] AutomatonTest.TestCountSymbols (1 ms)
[ RUN      ] AutomatonTest.TestHasState_TRUE
[       OK ] AutomatonTest.TestHasState_TRUE (0 ms)
[ RUN      ] AutomatonTest.TestHasState_FALSE
```

```
[ OK ] AutomatonTest.TestHasState_FALSE (0 ms)
[ RUN ] AutomatonTest.TestAddState_TRUE
[ OK ] AutomatonTest.TestAddState_TRUE (0 ms)
[ RUN ] AutomatonTest.TestAddState_FALSE_AlreadyPresent
[ OK ] AutomatonTest.TestAddState_FALSE_AlreadyPresent (0 ms)
[ RUN ] AutomatonTest.TestRemoveState_TRUE
[ OK ] AutomatonTest.TestRemoveState_TRUE (1 ms)
[ RUN ] AutomatonTest.TestRemoveState_FALSE_NotPresent
[ OK ] AutomatonTest.TestRemoveState_FALSE_NotPresent (0 ms)
[ RUN ] AutomatonTest.TestRemoveState_TRUE_StateOnInitialState
[ OK ] AutomatonTest.TestRemoveState_TRUE_StateOnInitialState (0 ms)
[ RUN ] AutomatonTest.TestRemoveState_TRUE_StateOnFinalState
[ OK ] AutomatonTest.TestRemoveState_TRUE_StateOnFinalState (1 ms)
[ RUN ] AutomatonTest.TestRemoveState_TRUE_StateTransition
[ OK ] AutomatonTest.TestRemoveState_TRUE_StateTransition (5 ms)
[ RUN ] AutomatonTest.TestCountStates
[ OK ] AutomatonTest.TestCountStates (1 ms)
[ RUN ] AutomatonTest.TestSetAndGetStateInitial
[ OK ] AutomatonTest.TestSetAndGetStateInitial (1 ms)
[ RUN ] AutomatonTest.TestSetAndGetStateFinal
[ OK ] AutomatonTest.TestSetAndGetStateFinal (1 ms)
[ RUN ] AutomatonTest.TestAddTransition_TRUE
[ OK ] AutomatonTest.TestAddTransition_TRUE (1 ms)
[ RUN ] AutomatonTest.TestAddTransition_FALSE_AlreadyPresent
[ OK ] AutomatonTest.TestAddTransition_FALSE_AlreadyPresent (1 ms)
[ RUN ] AutomatonTest.TestAddTransition_FALSE_StateNotPresent
[ OK ] AutomatonTest.TestAddTransition_FALSE_StateNotPresent (1 ms)
[ RUN ] AutomatonTest.TestAddTransition_FALSE_SymbolNotPresent
[ OK ] AutomatonTest.TestAddTransition_FALSE_SymbolNotPresent (0 ms)
[ RUN ] AutomatonTest.TestAddTransition_TRUE_SameState
[ OK ] AutomatonTest.TestAddTransition_TRUE_SameState (0 ms)
[ RUN ] AutomatonTest.TestAddTransition_TRUE_EPSILON
[ OK ] AutomatonTest.TestAddTransition_TRUE_EPSILON (0 ms)
[ RUN ] AutomatonTest.TestRemoveTransition_TRUE
[ OK ] AutomatonTest.TestRemoveTransition_TRUE (2 ms)
[ RUN ] AutomatonTest.TestRemoveTransition_FALSE_NotPresent
[ OK ] AutomatonTest.TestRemoveTransition_FALSE_NotPresent (2 ms)
[ RUN ] AutomatonTest.TestRemoveTransition_TRUE_EpsilonTransition
testfa: /home/jube/Univ/theorie_des_langages/projet/corr/2023/sandbox-tests/Automaton.cc:437:
bool fa::Automaton::hasEpsilonTransition() const: Assertion 'isValid()' failed.
==142799==
==142799== Process terminating with default action of signal 6 (SIGABRT): dumping core
==142799== at 0x4C19D3C: __pthread_kill_implementation (pthread_kill.c:44)
==142799== by 0x4BCAF31: raise (raise.c:26)
==142799== by 0x4BB5471: abort (abort.c:79)
==142799== by 0x4BB5394: __assert_fail_base.cold (assert.c:92)
==142799== by 0x4BC3E31: __assert_fail (assert.c:101)
==142799== by 0x1259FE: fa::Automaton::hasEpsilonTransition() const (Automaton.cc:437)
==142799== by 0x13BAC2: fa::AutomatonTest_TestRemoveTransition_TRUE_EpsilonTransition_Test::
TestBody() (testfa.cc:282)
==142799== by 0x172F7E: HandleSehExceptionsInMethodIfSupported<testing::Test, void> (gtest.
cc:2631)
==142799== by 0x172F7E: void testing::internal::HandleExceptionsInMethodIfSupported<testing
::Test, void>(testing::Test*, void (testing::Test::*)(), char const*) (gtest.cc:2667)
==142799== by 0x1651B5: Run (gtest.cc:2706)
==142799== by 0x1651B5: testing::Test::Run() (gtest.cc:2696)
==142799== by 0x1653B4: testing::TestInfo::Run() [clone .part.0] (gtest.cc:2885)
==142799== by 0x16558A: Run (gtest.cc:2858)
==142799== by 0x16558A: testing::TestSuite::Run() [clone .part.0] (gtest.cc:3039)
==142799== by 0x1682BC: Run (gtest.cc:3018)
==142799== by 0x1682BC: testing::internal::UnitTestImpl::RunAllTests() (gtest.cc:5897)
==142799==
==142799== HEAP SUMMARY:
```

```
==142799==      in use at exit: 111,035 bytes in 333 blocks
==142799==    total heap usage: 1,230 allocs, 897 frees, 216,753 bytes allocated
==142799==
==142799== LEAK SUMMARY:
==142799==    definitely lost: 0 bytes in 0 blocks
==142799==    indirectly lost: 0 bytes in 0 blocks
==142799==    possibly lost: 0 bytes in 0 blocks
==142799==    still reachable: 111,035 bytes in 333 blocks
==142799==    suppressed: 0 bytes in 0 blocks
==142799== Rerun with --leak-check=full to see details of leaked memory
==142799==
==142799== For lists of detected and suppressed errors, rerun with: -s
==142799== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
timeout: la commande surveillÃ©e a renvoyÃ© un core dump
Aborted
```