Protocol Technology

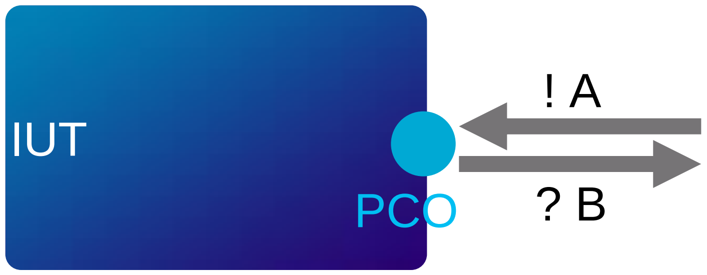
Testing

Gusztáv Adamis
BME TEMIT
2016

Protocol Testing

- Software Testing
  - White box testing
- Conformance Testing
  - IUT/SUT (Implementation/System Under Test conforms to specification)
  - Black box testing
    - Internal details not known/interested, only the communication
  - PCO (Point of Control and Observation)

Black Box Testing



Verdict:

pass,
fail,
inconclusive

- Black box testing
  - Implementation/System Under Test
  - Point of Control and Observation
- Not possible to test all the situations
  - Test Purposes

Test types

- Conformance testing
  - Function tests
  - System tests
  - Regression tests
    - when system changed – test if the 'rest' is not affected
- Interoperability testing
- Performance (Load) testing

Conformance Test Phases

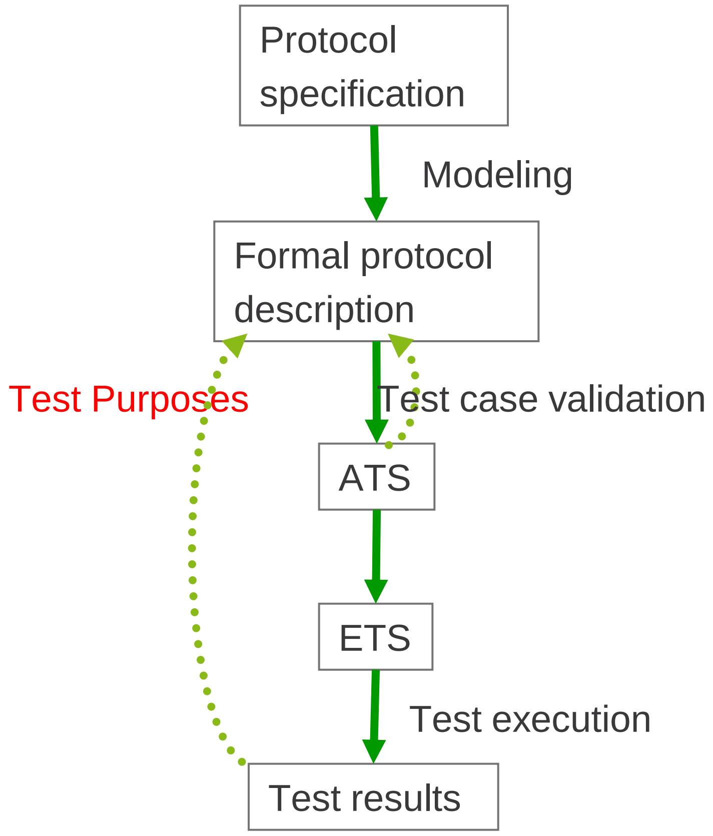
- Capability Test
  - Static analysis
    - if protocol options selected correctly
- Basic Interconnection Test
  - IUT able to communicate at all
- Behaviour Test
- Conformance Resolution Test
  - non standardised methods
  - multilayer tests
  - detect reasons of non-conform situations
    - inconclusive

ATS, ETS

- Abstract Test Suite
  - Contains tests for every protocol feature
  - Contains parameters (e.g. IP address)
  - High level communication
- Executable Test Suite
  - Contains selected test cases: only for the implemented test features
  - Parameters specified (e.g. IP address = 1.1.1.1.)
  - Low level communication
    - Encoding/Decoding messages to bit sequences that are transmitted to real network

Formal techniques in conformance tests

- Testing (black-box):
  - Check if Implementation Under Test (IUT) conforms to its specification
  - Experiments programmed into Test Cases
- Validation:
  - Ensure correctness of test cases of ATS

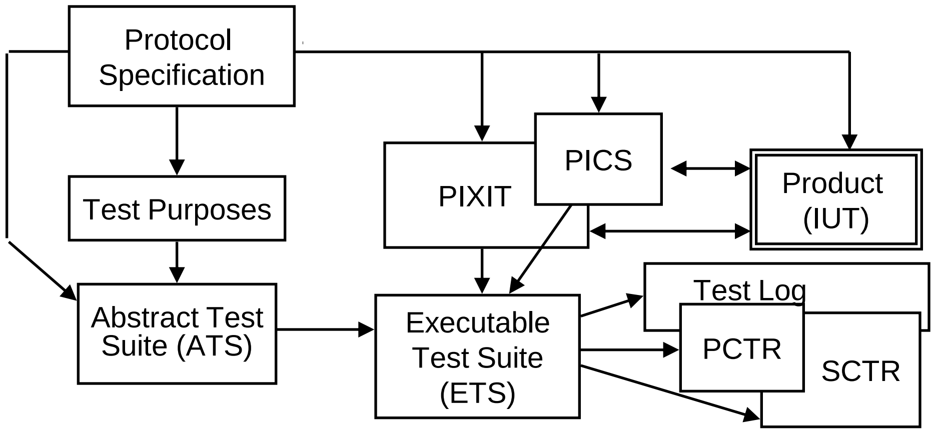


Test results

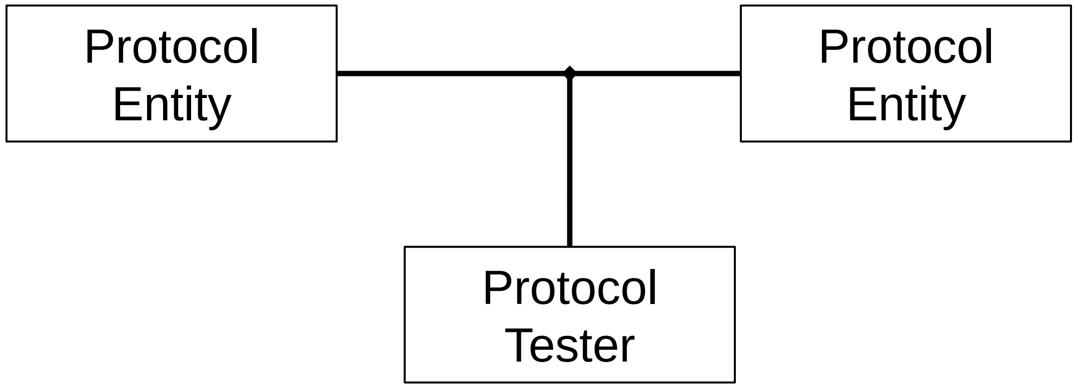
- Test outcome
  - foreseen
  - unforeseen – test case errors
- Verdict
  - pass
  - fail
  - inconclusive
- Test log
- Requirements on test outcomes
  - repeatable
  - comparable
  - auditable

Conformance Test Documents

- PICS: Protocol Implementation Conformance Statement
- PIXIT: Protocol Implementation eXtra Information on Testing
- PCTR/SCTR: Protocol/System Conformance Test Report

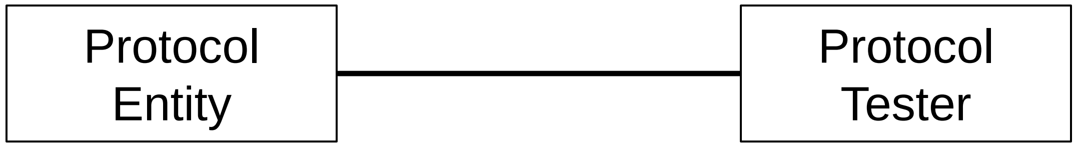


Passive Tester



- Only observes
  - waits for error
    - no guarantee to happen
- Protocol Analyzer

Active Tester



- Active
  - can send messages
- Valid testing
- Provocative testing
  - Invalid
    - Sends syntactically incorrect messages
  - Improper
    - Sends syntactically correct messages, but at wrong time
- Test cases are generated before testing starts

Hybrid Tester

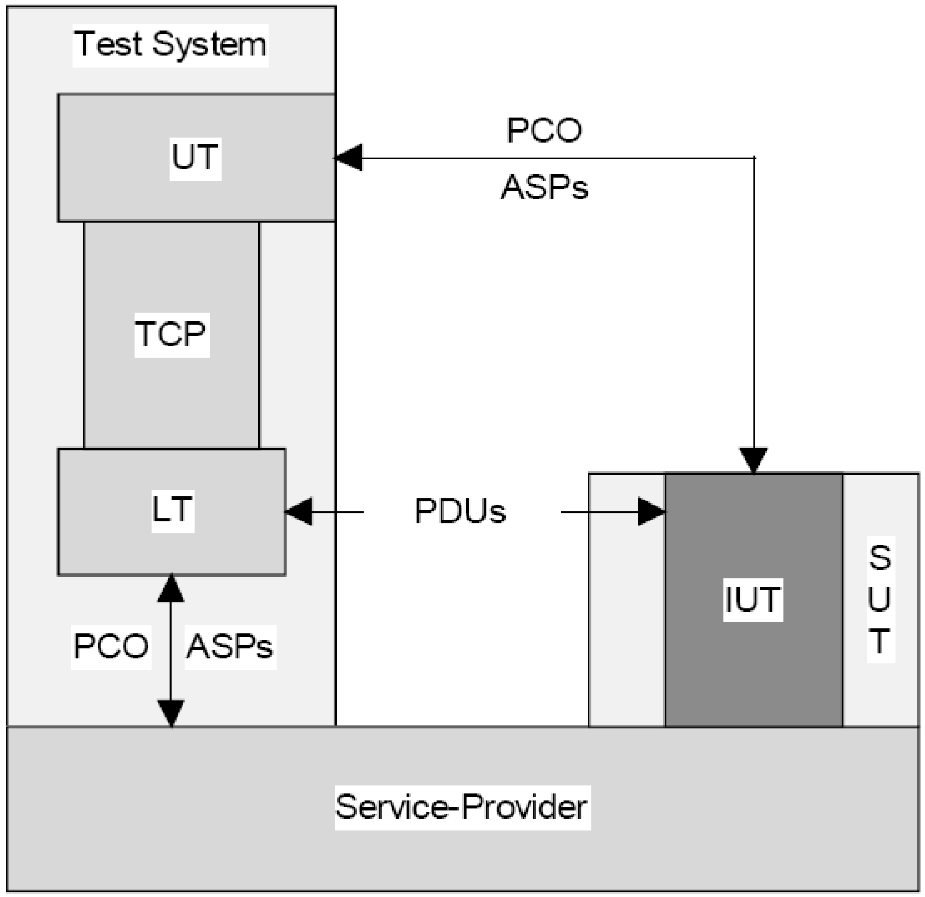
- Test cases are generated during the execution from the protocol specification
  - „On-the-fly” testing
  - Depending on the reaction of IUT
    - No guarantee to reach all the states

Test Arrangements

- Upper Tester
- Lower Tester

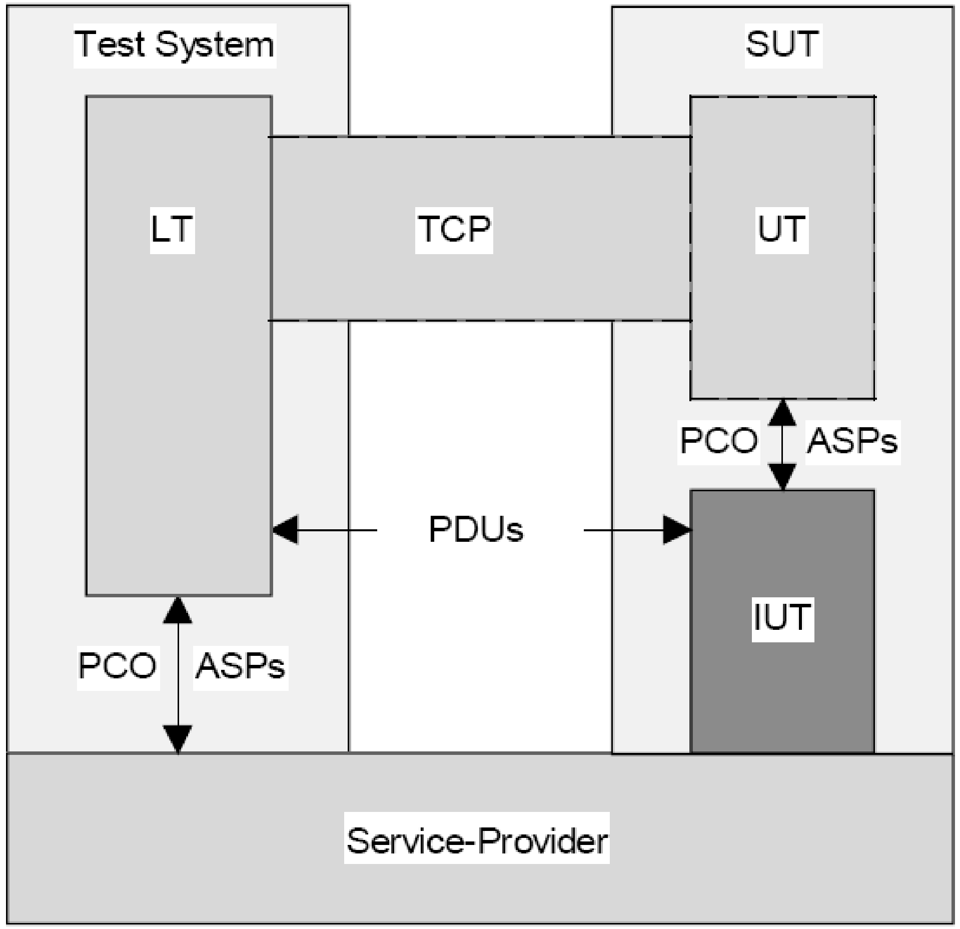
- Local Test Method
- Distributed Test Method
- Coordinated Test Method
- Remote Test Method

Local Test Method



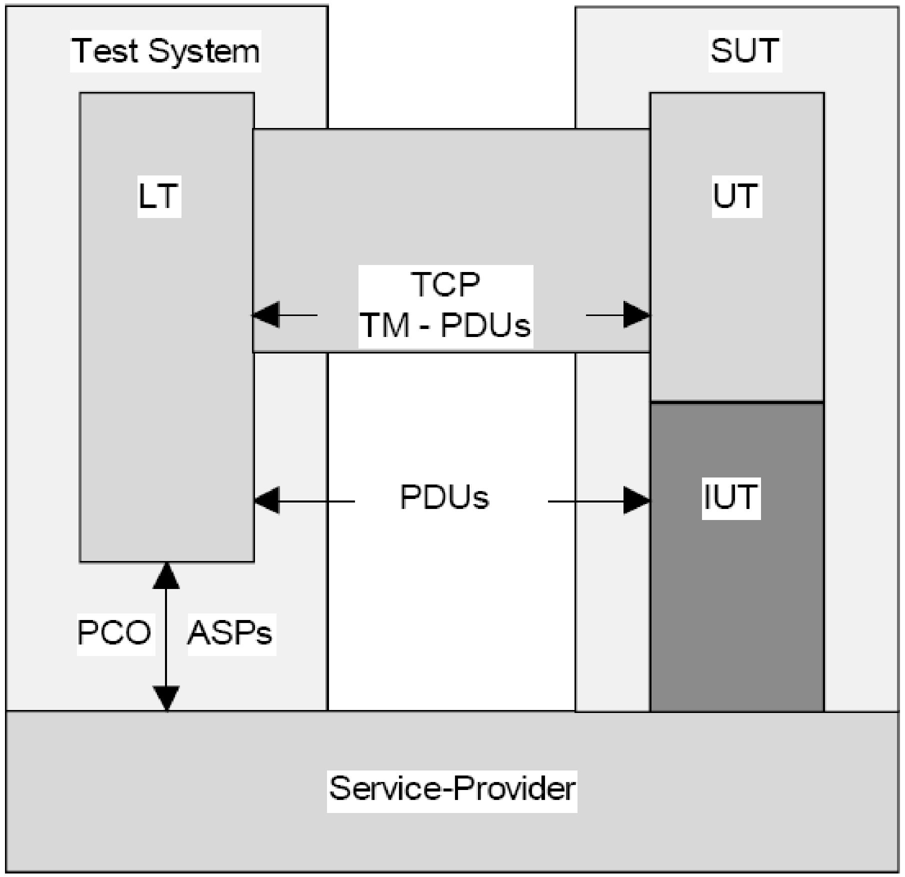
a) The Local test methods

Distributed Test Method



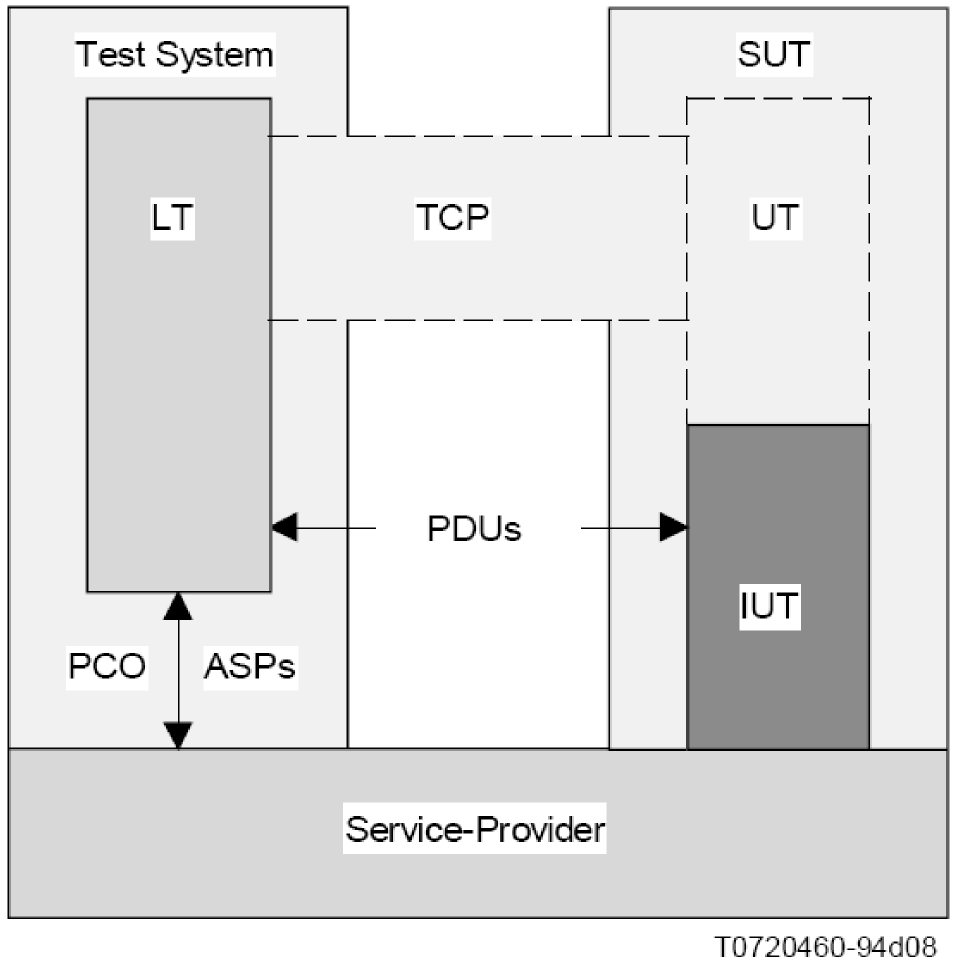
b) The Distributed test methods

Coordinated Test Method



c) The Coordinated test methods

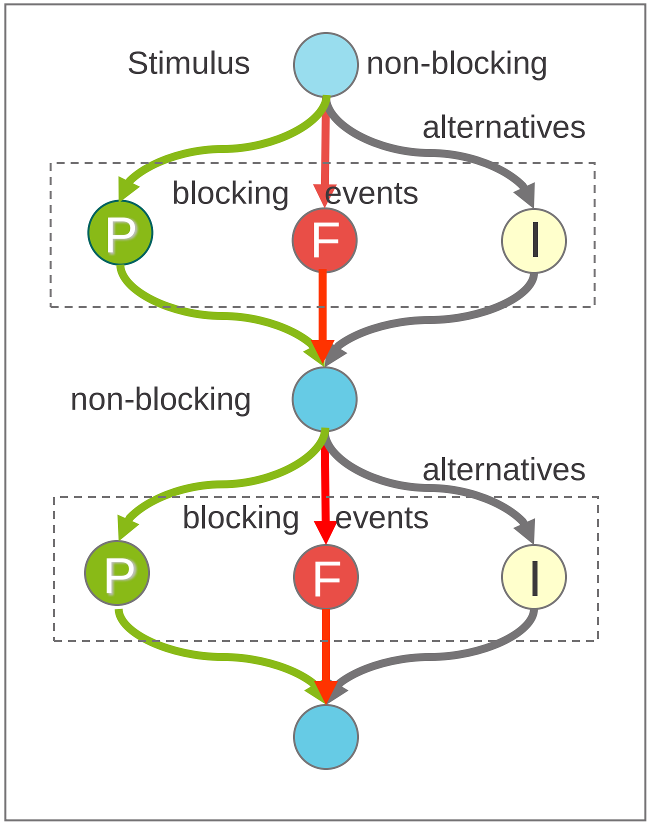
Remote Test Method



d) The Remote test methods

Test cases in black-box test

- Implementation of Test Purpose
  - TP defines an experiment
- Focus on a single requirement
- Returns verdict (pass, fail, inconclusive)
- Typically a sequence of action-observation-verdict update:
  - Action (stimulus): non-blocking (e.g. transmit PDU, start timer)
  - Observation (event): takes care of multiple alternative events (e.g. expected PDU, unexpected PDU, timeout)

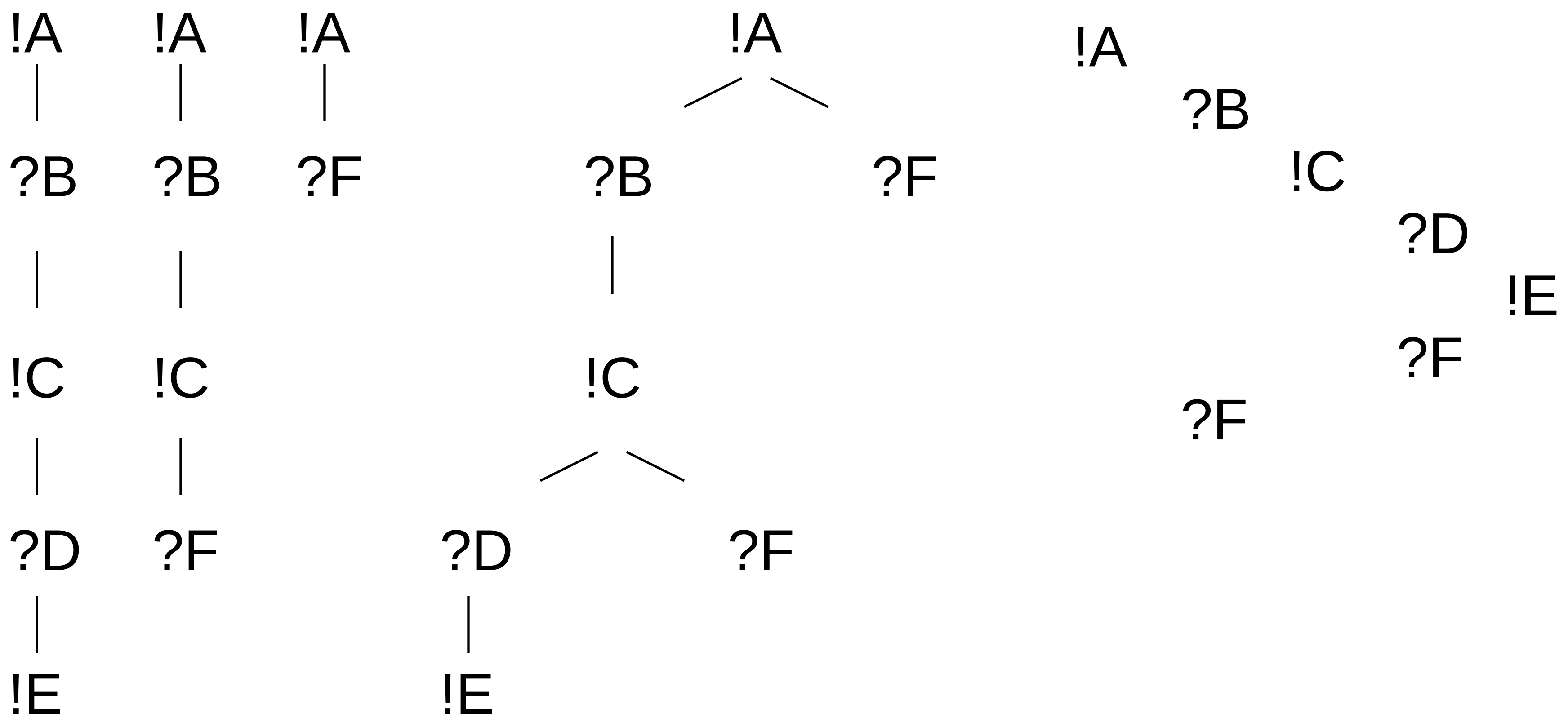


Test Tree

Possible event sequences

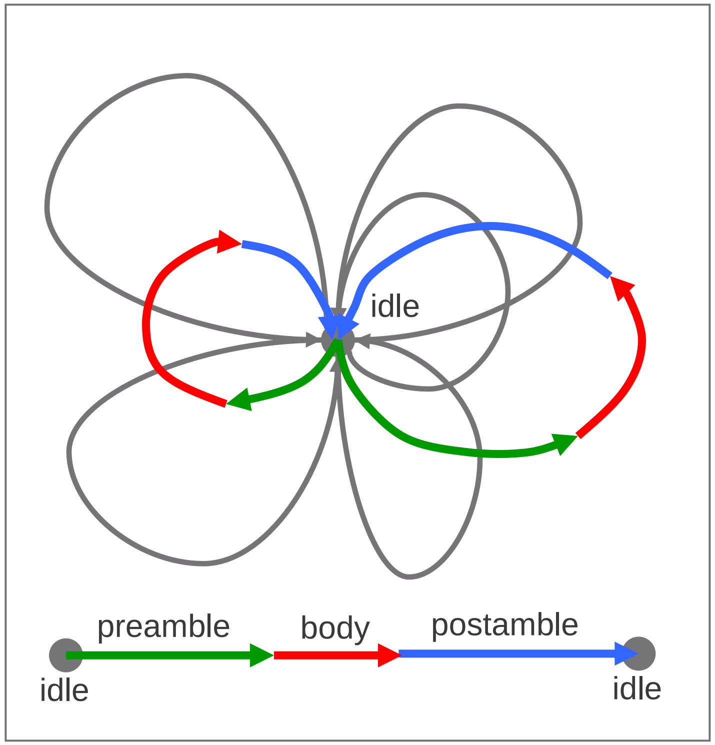
Behaviour tree

Alternatives



Independence and structure of abstract test cases

- *Abstract test cases* should contain
  - preamble*:* sequence of test events to drive IUT into *initial testing state* from the *starting stable testing state*
  - test body*:* sequence of test events to achieve the *test purpose*
  - postamble: sequence of test events which drive IUT into a *finishing stable testing state*
- Preamble/postamble may be absent
- *Starting stable testing state* and *finishing stable testing state* are the idle state in TTCN-3



Requirements on test suites

- All test cases in an ATS must be *sound*
  - *Sound* test case results pass verdict if IUT is correct (practically impossible with finite number of test cases)
  - *Exhaustive* test case gives fail verdict if IUT behaves incorrectly
  - *Complete* test case is both sound and exhaustive
- Must not terminate with none or error verdict

Phases of black-box (functional) testing

- Test purpose definition
  - Formally or informally
- TTCN-3 Abstract Test Suite (ATS)
  - design or generation
- Executable Test Suite (ETS) implementation
  - using the Means of Testing (MoT)
- Test execution against the Implementation Under Test (IUT)
  - with MoT
- Analysis of test results
  - verdicts, logs (validation)

Abstract Test Suite design

- Manual design:
    - Identify *test purposes* from protocol specification based on the test requirements
    - Implement *abstract test cases* from *test purposes* using a standardized test notation (TTCN-3)
- Automatic design:
    - Generate *test purposes* and *abstract test cases* directly from formal protocol specification in e.g. UML, SDL, ASN.1
    - Requires formal protocol specification
    - Computer Aided Test Generation (CATG) is an open problem
    - Model Based Testing