State Machine Correctness
The AGREE state machine specification is correct
Context
{AGREE specification}
Definition
[Correct State Machine]: State machine correctness is defined by the decomposition within the argument structure.
Justification
The complexity of the concept of correctness cannot be easily defined without fully enumerating the contents of the argument.
J
Assumption
A correct specification does not need to consider if the specification is satisfied by the implementation.
A
Rodes
We are looking at the AADL model admitting that in some cases the model is the spec, and yet in some cases it is a refinement of another spec. This means some times we may want to validate the model against reviews, and in other cases we validate against other models/documents.
Rodes
General note: any use of a lemma to produce a "witness" should require the user to examine and approve the counterexample. Chris has discussed the possibility of visualizing counterexamples in simulink as a possible "tool" for this purpose.
Strategy
Argue correctness of each AADL component's internal AGREE state machine representation
{Component} State Machine Correctness
{Component} AGREE state machine specification is correct.
Context
{component AGREE state machine}
Rodes
TODO: include a placeholder or note for the kind of validation that has been done with derek manually.
Definition
[Correct]:
TBD
Use of the correct infrastructure for all components?
Goal
composition of states (but does not include full reachability, gets into system level properties)
Goal
system level reachability?
Strategy
Argue correctness of each characteristic of a finite state machine
Context
{AGREE states set}
Context
{AGREE transitions set}
Context
{AGREE start state}
Context
{AGREE terminal states set}
Assumption
State machine inputs and outputs are assumed to be correct (addressed as general correctness properties of the model and not specifically with respect to the state machine). Inputs and outputs are indirectly assessed with respect to correctness of transition and state behavior.
A
Rodes
Note: in general additional lemmas can be used to aid manual inspection: e.g., an inspection that the state machine as correct transition connections can have lemmas to assert start and end states on a transition. Another example: probing behavior of states with lemmas.
Correct States
The component's AGREE states are correct.
Strategy
Argue over the characteristics of correct AGREE states
Rodes
TODO: how do I address the possibility of screwing up the state infrastructure?
Correct State Behavior
All states in the state set for the component defined in the AGREE model have the correct state behavior.
Strategy
Argue correct behavior for each individual state.
Context
{set of AGREE states}
{State} Behavior Correctness
{State} behavior is correct
Context
{state}
Strategy
Argue correct state behavior by {misc. model validation} and mitigation of design faults
Definition
[Design Fault]: An error in the design undermining assurance properties and/or functionality as is intended within the design specification.
Justification
This strategy is cross-cutting and potentially overlapping. Validation involves reviews with domain experts and analysis of available documentation and artifacts. Fault mitigation examines the model from a separate angle, looking for possible ways in which the model may have been incorrectly specified. Ideally no model is validated that has residual faults, but these faults capture best practices, and expert defined standards of developing "bug free" specifications.
J
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Design Fault Mitigation
All faults contributing to the correct specification of state behavior are mitigated.
Strategy
Argue absence of individual identified design faults contributing to incorrect state behavior specification.
Context
{identified design faults}
Rodes
A work in progress. Towards an AGREE development standard.
Abstraction Correctness
All abstractions within the state are correct.
Definition
[State abstractions]: {TBD}
Rodes
A work in progress.
Definition
[Correct State Abstractions]: {TBD}
Rodes
A work in progress.
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Preservation of variables
Rodes
A work in progress.
State Identifier Uniqueness (Model Correctness)
All AGREE state identifiers are unique.
Definition
[Unique AGREE State Identifiers]: identifiers used to represent different states do not share the same identifier name or underlying value.
Justification
It is possible to specify state identifiers in AGREE that have the same underlying value. As a result, the model may look correct in that different identifiers are used, but be incorrect since their underlying values are not unique. Failure to have unique state identifiers will result in a failure to correctly specify a state transition.
J
Evidence Scheme
{AGREE Lemma idiom}
Complete and Necessary States
The state set defined in the AGREE model is complete and contains only necessary states.
Strategy
Argue state completeness and necessity separately
Complete
The state set defined in the AGREE model is complete
Strategy
Argue state set completeness by {misc. model validation}
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Necessary
The state set defined in the AGREE model contains only necessary states
Strategy
Argue necessity of the state set by {misc. model validation} and demonstration no other states are possible in the model.
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Exclusive States
No states, other than those defined in the AGREE state set, are possible in the AGREE model (the component's states as defined within the state set are the only possible state transitions).
Evidence Scheme
{AGREE Lemma idiom}
Correct Transitions
The component's AGREE transitions are correct
Assumption
Assume correct AGREE states
A
Strategy
Argue over characteristics of correct AGREE state transitions
Assumption
Unreachable states and transitions, and non-exclusive states and transitions is indicative that the state machine is not correctly specified in the design.
A
Context
{correct transition characteristics}
Rodes
Work in progress
Definition
[Transition reachable]: a transition is reachable there exists a state machine execution path where the transition is taken.
Definition
[State reachability]: a given state is reachable if a state machine execution path exists by which the state is entered.
Correct Transition Behavior
All transitions in the transition set for the component defined in the AGREE model have the correct transition behavior.
Definition
[Correct Transition Behavior]: The source and destination states are correct and the transition conditions are correct.
Strategy
Argue transition correctness for each individual transition
Context
{set of AGREE transitions}
{Transition} Behavior Correctness
{Transition} behavior is correct
Context
{transition}
Strategy
Argue over the characteristics of correct transition behavior.
Definition
[Correct Transition Behavior]: The source and destination states are correct and the transition conditions are correct.
Intended Source and Destination
The source and destination states for the transition are the intended states.
Strategy
Argue the destination state is correct and the source state are intended separately
Destination Correctness
The transition destination state is the intended destination.
Strategy
Argue destination state correctness by {misc. model validation} and mitigation of design faults
Definition
[Design Fault]: An error in the design undermining assurance properties and/or functionality as is intended within the design specification.
Justification
This strategy is cross-cutting and potentially overlapping. Validation involves reviews with domain experts and analysis of available documentation and artifacts. Fault mitigation examines the model from a separate angle, looking for possible ways in which the model may have been incorrectly specified. Ideally no model is validated that has residual faults, but these faults capture best practices, and expert defined standards of developing "bug free" specifications.
J
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Design Fault Mitigation
All faults contributing to incorrectly specifying the intended state definition are mitigated.
Strategy
Argue absence of individual identified design faults contributing to incorrect destination state specification.
Context
{identified design faults}
Rodes
A work in progress. Towards an AGREE development standard.
State Identifier Uniqueness (Model Correctness)
All AGREE state identifiers are unique.
Definition
[Unique AGREE State Identifiers]: identifiers used to represent different states do not share the same identifier name or underlying value.
Justification
It is possible to specify state identifiers in AGREE that have the same underlying value. As a result, the model may look correct in that different identifiers are used, but be incorrect since their underlying values are not unique. Failure to have unique state identifiers will result in a failure to correctly specify a state transition.
J
Evidence Scheme
{AGREE Lemma idiom}
Source Correctness (Appropriate Transition)
The transition is an intended transition out of the source state.
Strategy
Argue source state correctness by {misc. model validation}
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Transition Condition Correctness
The conditions under which the transition is triggered are intended.
Strategy
Argue trandition conditions are intended by {misc. model validation} and mitigation of design faults
Definition
[Design Fault]: An error in the design undermining assurance properties and/or functionality as is intended within the design specification.
Justification
This strategy is cross-cutting and potentially overlapping. Validation involves reviews with domain experts and analysis of available documentation and artifacts. Fault mitigation examines the model from a separate angle, looking for possible ways in which the model may have been incorrectly specified. Ideally no model is validated that has residual faults, but these faults capture best practices, and expert defined standards of developing "bug free" specifications.
J
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Design Hazard Mitigation
All faults contributing to incorrectly specifying a transition condition are mitigated.
Strategy
Argue absence of individual identified design faults contributing to incorrect transition condition specification.
Context
{identified design faults}
Rodes
A work in progress. Towards an AGREE development standard.
Consistent Transition Edge
The AGREE transition uses the prescribed transition edge.
Definition
[Transition Edge]: Transitions may occur in the same time step in which a transition condition occurs, or the transition occurs on the next time step after a transition condition occurs. This difference is somewhat analogous to rising edge and falling edge signals.
Context
{prescribed transition edge}
Justification
Failure to understand and consistently apply transition edges has been observed to lead to incorrect specs.
J
Evidence Scheme
{AGREE Lemma idiom}
Correct Transition Edge Idioms
The AGREE transition correctly models the transition for the prescribed transition edge.
Context
{prescribed transition edge}
Definition
[Transition Edge]: Transitions may occur in the same time step in which a transition condition occurs, or the transition occurs on the next time step after a transition condition occurs. This difference is somewhat analogous to rising edge and falling edge signals.
Strategy
{tbd}
Rodes
Argue over idioms generated based on experience for appropriately modeling state machines with the given transition edge.
Rodes
Work in progress
TBD
Non-transition conditions accounted for.
TBD
{transition edge specific claims}
Abstraction Correctness
All abstractions within the transition are correct.
Definition
[State abstractions]: {TBD}
Rodes
A work in progress.
Definition
[Correct State Abstractions]: {TBD}
Rodes
A work in progress.
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Complete and Necessary Transitions
The transition set defined in the AGREE model is complete and contains only necessary transitions.
Strategy
Argue transition completeness and necessity separately
Complete
The AGREE transition set is complete (no transitions are missing)
Strategy
Argue complete AGREE transitions by {misc. model validation}
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Necessary
The AGREE transition set only contains necessary transitions (no unnecessary transitions)
Strategy
Argue necessity of state transitions by {misc. model validation} and demonstration no other transitions are possible in the model.
{Misc. Model Validation}
{Various validation of the model with domain experts demonstrating the model captures the intended state machine}
Example Validation
AADL model consistency; Natural language model consistency; consistency/reduction with other models/specifications; traceability to and satisfaction of requirements; simulation of the model (model tests)
Exclusive Transition Set
No other transitions other than those in the AGREE transition set are possible within the AGREE mode (the component's transitions as defined within the transition set are the only possible state transitions).
Justification
{this is here because it is possible to have additional states and we considered the analogous problem for transitions}
J
Context
{TBD: example}
Evidence Scheme
{AGREE Lemma idiom}
State Reachability
All states in the set of AGREE states for the component are reachable {within the specified time bound}.
Definition
[State reachability]: a given state is reachable if a state machine execution path exists by which the state is entered.
Justification
Unreachable states may indicate an over-engineered architecture or an error in architecture design (i.e., an incorrectly specified finite state machine).
J
Assumption
There are no legitimately defined unreachable states
A
Strategy
Argue reachability of each state individually and completeness of the set of states.
{State} Reachable
There exists a trace for which state {State} is reachable.
Assumption
Reachability is only considered at this level with respect to the component in isolation. This means we assume all inputs (as constrained by any component assumptions) are possible. In the larger system, states may not be reachable (i.e., an input may not be globally possible under system constraints).
A
Context
{state}
Evidence Scheme
{AGREE state reachability idiom}
Context
{idiom documentation}
Evidence
{Agree lemma witness for the state}
Reachable State Transitions
All transitions in the set of AGREE transitions for the component are reachable {within the specified time bound}.
Definition
[Transition reachable]: a transition is reachable there exists a state machine execution path where the transition is taken.
Justification
Unexecutable transition may indicate an over-engineered architecture or an error in architecture design (i.e., an incorrectly specified finite state machine).
J
Assumption
There are no legitimately unreachable transitions.
A
Strategy
Argue reachability of each state individually and completeness of the set.
{Transition} Reachable
There exists a trace for which the {transition} is reachable.
Context
{transition}
Evidence Scheme
{AGREE transition reachability idiom}
Context
{idiom documentation}
Evidence
{Agree lemma witness for transition reachability}
Correct Start State
The component's AGREE state machine is initialized at the correct start state.
Assumption
The set of transitions and states are correct
A
Strategy
Argue by verification that {start state} is the start state in the spec. and that {start state} is the intended state machine start state.
Justification
Correctness of the start state is determined by the spec actually initializing the state machine to the start state, and assessment that the given initialized start state is the intended start state. Further assessment of the correctness of the start state behavior is assessed in the state correctness claim.
J
Start State Validation
The component's start state is the intended start state.
Context
{intended start state}
Evidence Scheme
Manual inspection/review of the AGREE model compared to system specification documents and system requirements with system developers.
Context
{System specification documents (must include any informal documentation, other documents of state machine models, and the AADL model}
Evidence
{inspection report and sign off}
Start State Model Verification
The component's AGREE state machine model starts in {start state}.
Evidence Scheme
{AGREE lemma for start states }
Context
{idiom documentation}
Evidence
{AGREE lemma proof}
Correct Terminal States
The component's AGREE state machine terminates in the correct terminal states
Definition
[Terminates]: {termination definition}
Assumption
The set of transitions and states are correct
A
Strategy
Argue correct terminal states by validation that the terminal states are intended and the model only terminates in the defined terminal states.
Justification
Terminal state correctness is determined by determining the state machine can only terminate in the specified states and those states are complete. Further assessment of the correctness of the behavior of the states is assessed under the state correctness claim.
J
Terminal States Validation
The set of the component's terminal states is complete and necessary
Definition
[Complete]: All terminal states that should be present are present
Definition
[Necessary]: Only terminal states that should be present are present.
Evidence Scheme
Manual inspection/review of the AGREE model compared to system specification documents and system requirements with system developers.
Context
{System specification documents (must include any informal documentation, other documents of state machine models, and the AADL model}
Evidence
{inspection report and sign off}
Terminal State Model Verification
The component's AGREE state machine model only terminates in the defined terminal states.
Evidence Scheme
{AGREE lemma for terminal states }
Context
{idiom documentation}
Evidence
{AGREE lemma proof}