

THIS DOCUMENT IS THE ONLINE-ONLY APPENDIX TO:

A Game-Based Framework for CTL Counterexamples and 3-Valued Abstraction-Refinement

SHARON SHOHAM and ORNA GRUMBERG
Computer Science Department, Technion, Haifa, Israel

ACM Transactions on Computational Logic, Vol. V, No. N, January 2006, Pages 1–49.

A. PROOFS OF SECTION 2

Proof of Lemma 2.6

LEMMA. *Let B be a non-trivial strongly connected component (SCC)⁶ in a game-graph. Then the set of formulae that are associated with the nodes in B is exactly one of the sets $\text{exp}(\varphi)$, where $\varphi \in \{A(\varphi_1 U \varphi_2), E(\varphi_1 U \varphi_2), A(\varphi_1 V \varphi_2), E(\varphi_1 V \varphi_2)\}$.*

PROOF. By the rules of the game, which determine the edges in the game-graph, we have that the sons of a node $n = (s, \varphi)$ in the game-graph are either associated with strict subformulae of φ , or with expansions of it in case φ is an AU, AV, EU, EV formula. Therefore, a non-trivial SCC B , which contains cycles, must have at least one node n in it that is associated with an AU, AV, EU, EV formula (otherwise we have a loop where each formula is a strict subformula of the previous one, which is impossible). Consider the case where the node n has the formula $\varphi = A(\varphi_1 U \varphi_2)$. Other cases are similar. We prove that in this case the set of formulae that are associated with the nodes in B is exactly $\text{exp}(A(\varphi_1 U \varphi_2))$.

First, we prove that B cannot contain additional formulae. Let $n' = (s', \varphi')$ be a node in B , other than n . We show that $\varphi' \in \text{exp}(A(\varphi_1 U \varphi_2))$. By the rules of the game, we have that the descendants of a node $n'' = (s'', \varphi'')$ in the game-graph are associated with subformulae from $\text{sub}(\varphi'')$. Since n' lies on the same SCC as n , we have that n' is a descendent of n , and thus $\varphi' \in \text{sub}(\varphi)$. Since $\varphi = A(\varphi_1 U \varphi_2)$, we have that $\text{sub}(\varphi) = \text{exp}(A(\varphi_1 U \varphi_2)) \cup \text{sub}(\varphi_1) \cup \text{sub}(\varphi_2)$. It remains to show that $\varphi' \notin \text{sub}(\varphi_1) \cup \text{sub}(\varphi_2)$, thus it must be the case that $\varphi' \in \text{exp}(A(\varphi_1 U \varphi_2))$. Suppose the contrary, i.e. $\varphi' \in \text{sub}(\varphi_1) \cup \text{sub}(\varphi_2)$. This implies that $\text{sub}(\varphi') \subseteq \text{sub}(\varphi_1) \cup \text{sub}(\varphi_2)$. Since n is also a descendent of n' , we have that $\varphi \in \text{sub}(\varphi')$. i.e. $\varphi \in \text{sub}(\varphi_1) \cup \text{sub}(\varphi_2)$. Thus φ must obey one of the following.

- (1) $|\varphi| \leq |\varphi_1|$ or $|\varphi| \leq |\varphi_2|$, or:

⁶A non-trivial SCC contains at least one edge. Unless otherwise stated, it is not necessarily maximal.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2006 ACM 1529-3785/2006/0700-0001 \$5.00

(2) φ is of the form $(\varphi'_1 \vee \varphi'_2)$ or $(\varphi'_1 \wedge \varphi'_2)$ or $AX\varphi'$ (results from expansion).

Obviously, $\varphi = A(\varphi_1 U \varphi_2)$ obeys none of the above. Contradiction.

To complete the proof, it remains to show that B must contain *all* the formulae in $\text{exp}(A(\varphi_1 U \varphi_2))$. This is clear from the structure of the game-graph: if one of these formulae were missing, no loop could be formed in contradiction to the fact that B is a non-trivial SCC. \square

B. PROOFS OF SECTION 3

Throughout this section we set G to be a game-graph with a coloring function χ , and denote by C the subgraph of G computed by `ComputeCounter`.

Proof of Lemma 3.4

LEMMA. *Let G be a game-graph, χ its coloring function and A a subgraph of G with the following properties: (1) It contains an initial node n_0 , colored F by χ , (2) It is independent of G , and (3) It is minimal w.r.t. 1 and 2. Then, for every $n \in A$: $\chi(n) = F$.*

PROOF. Assume to the contrary that A contains at least one node that is colored T by χ . We show that removing all the nodes that are colored T from A will not affect 1 and 2. Thus, it will result in a strict subgraph of A that satisfies 1 and 2, in contradiction to the minimality of A (3).

Let $A' \subseteq A$ be the subgraph of G that results from removing all the nodes colored by T (and the corresponding edges) from A . Clearly, n_0 is not one of these nodes since $\chi(n_0) = F$ (by 1). Thus, A' contains n_0 and fulfills 1. It remains to show that it also satisfies 2, i.e. that it is independent of G .

We need to show that the partial coloring algorithm of G w.r.t. A' , given any initial coloring function, does not change the colors of all the nodes in A' , i.e. colors them by F . Let $\chi'_I : N \setminus A' \rightarrow \{T, F\}$ be such an initial coloring function and let $\bar{\chi}' : N \rightarrow \{T, F\}$ be the resulting partial coloring of G w.r.t. A' . We show that for every node $n' \in A'$: $\bar{\chi}'(n') = F$.

To do so, let us look at the initial coloring function $\chi''_I : N \setminus A' \rightarrow \{T, F\}$ that agrees with χ'_I on all the nodes in $N \setminus A$, but colors the nodes in $A \setminus A'$ by T . χ''_I differs from χ'_I only in (possibly) changing the colors of nodes in $A \setminus A'$ from T to F . We show that (a) if the partial coloring function $\bar{\chi}''$ of G w.r.t. A' based of χ''_I colors all the nodes of A' by F , then it implies that $\bar{\chi}'$ does the same, and that (b) $\bar{\chi}''$ indeed colors the nodes of A' by F .

(a) The coloring is monotonic in the sense that changing the color of a node from T to F in the initial coloring function of $N \setminus A'$ can only cause nodes in A' to change their colors from T to F as well and not the other way around: it cannot cause their colors to change from F to T . This monotonicity holds since the game-graph is based on a CTL formula in negation normal form, thus there are no \neg -nodes in it. Since χ''_I differs from χ'_I only in possibly coloring F some nodes (in $A \setminus A'$) that are colored T by χ'_I , then the monotonicity of the coloring ensures that if $\bar{\chi}''$ colors the nodes of A' by F , then so does $\bar{\chi}'$.

(b) It remains to show that $\bar{\chi}''$ indeed colors all the nodes in A' by F . To do so, we first use χ''_I to construct an initial coloring function $\chi_I : N \setminus A \rightarrow \{T, F\}$ that will result in a partial coloring $\bar{\chi}$ of G w.r.t. A . χ_I is defined such that for each

$n \in N \setminus A$: $\chi_I(n) = \chi_I''(n)$. Since A is independent of G (2), $\bar{\chi}$ does not change the color of the nodes in A . In particular, the nodes in A' remain colored by F and the nodes in $A \setminus A'$ remain colored T . Thus, in fact $\bar{\chi}$ colors all the nodes in $N \setminus A'$ as χ_I'' . This implies that each execution of the partial coloring algorithm w.r.t. A given χ_I is also a “legal” execution of it w.r.t. $A' \subseteq A$ given χ_I'' . Since the result of the partial coloring algorithm is unique, this means that $\bar{\chi}'$ (which results from χ_I'') colors all the nodes as $\bar{\chi}$, and in particular colors the nodes in A' by F .

From (a) and (b) we conclude that $\bar{\chi}'$ colors all the nodes of A' by F . \square

Proof of Lemma 3.7

LEMMA. *For each node $n \in C$, we have that $\chi(n) = F$.*

PROOF. This is shown by induction on the construction of C , when we rely on the property of χ , that an \vee -node is colored by F iff all its sons are colored by F and an \wedge -node is colored by F iff at least one of its sons is colored by F . This property is obviously correct when the coloring does not use a witness, but it is also true when a witness causes the coloring. \square

Proof of Lemma 3.8

LEMMA. (1) *If a node in C was colored due to a witness, then it belongs to a non-trivial SCC in C .*

(2) *Each non-trivial SCC in C contains at least one node that was colored due to a witness.*

PROOF. Consider the first part of the lemma. If the coloring of a node that appears in C was based on a witness, then this node resides on a non-trivial SCC that will be added to C . This results from the following properties. For an \vee -node all the sons are added to C and in particular the one(s) in the non-trivial SCC. For an \wedge -node that is colored by a witness, its cause is added to C , where the cause is a son within the non-trivial SCC that is also colored by a witness. Thus, a cycle is formed.

To prove the second part of the lemma, let us look at a non-trivial SCC in C . All the nodes in it are colored by F . Assume to the contrary that all of them were colored due to their sons (rules 2 and 3). Consider the first node on the SCC that was colored and denote it by n_1 . Since it is the first, it must be colored by F based on its sons *outside* the SCC. Yet, it obviously has sons within the (non-trivial) SCC too. Thus it must be an \wedge -node. The reason for this conclusion is that only \wedge -nodes can be colored F based on part of their sons only. Being an \wedge -node, n_1 has exactly one son in C , and by construction this son is its *cause*, which is outside the SCC by our assumption. This contradicts the fact that n_1 resides on the non-trivial SCC (and thus also has a son within the SCC). We conclude that at least one of the nodes in the SCC was colored due to a witness. \square

Proof of Corollary 3.9

COROLLARY. *Non-trivial SCCs in C are either AU-SCCs or EU-SCCs.*

PROOF. Lemma 3.8 tells us that if a non-trivial SCC appears in C then at least one of its nodes was colored by a witness. On the other hand, by Lemma 3.7 we

know that all the nodes in C are colored by F , and by the coloring algorithm we know that only nodes in AU or EU SCCs are colored by F due to a witness. Thus, the corollary is implied. \square

Proof of Lemma 3.10

LEMMA. *Non-trivial AU -SCCs in C are always simple cycles, rather than general SCCs.*

PROOF. It suffices to show that every node within a non-trivial AU -SCC in C has exactly one son within the SCC. Consider a non-trivial AU -SCC in C . We use a case analysis on the type of nodes in the SCC. By the construction of C , we have that \wedge -nodes in the SCC have a single son in C and in particular in the SCC. Apart from \wedge -nodes, such an SCC contains only \vee -nodes, that are not EX -nodes. This is because by Lemma 2.6, the game-graph, and C in particular, have the property that the set of formulae in a non-trivial AU -SCC is exactly $exp(A(\varphi_1 U \varphi_2))$, for some $A(\varphi_1 U \varphi_2) \in sub(\varphi)$. This property also implies that \vee -nodes in such an SCC are of the form $(s', \varphi_2 \vee (\varphi_1 \wedge AXA(\varphi_1 U \varphi_2)))$ and have two sons in the game-graph, one of which is with the subformula $\varphi_2 \notin exp(A(\varphi_1 U \varphi_2))$ and thus clearly does not belong to the SCC. Thus, such nodes also have at most one son within the SCC, and the claim is implied. \square

Proof of Theorem 3.11

THEOREM. *The subgraph C , computed by `ComputeCounter`, is independent of the game-graph G .*

For the proof of the theorem, we need the following technical lemma.

LEMMA B.7. *Let n be a node in C that was colored due to a witness during the partial coloring of G with respect to C , given an initial coloring function χ_I . Suppose that all the nodes from C that were colored prior to n by the partial coloring algorithm were colored by F . Then n lies on a cycle in C .*

PROOF. Suppose n was colored due to a witness by the partial coloring algorithm of G . Let us look at the status of the game-graph at the phase of the partial coloring algorithm, where n was colored by a witness. Obviously, n has a son that is not yet colored at that time (otherwise n would be colored too, based on its sons). This son must be within the same set Q_i (all the sons outside Q_i are in “smaller” sets Q_j and are thus already colored). Let us show that at least one such uncolored son is in C . We use a case analysis of the type of n .

- If n is an \vee -node, then all of its sons are in C , and in particular the uncolored one, which concludes this case.
- If n is an \wedge -node, then it has exactly one son n' in C . If this son n' is not one of the uncolored ones, then by our assumption it is already colored by F , which would cause n to be already colored by F too (based on this son rather than on a witness) in contradiction.

In any case, we get that n has a son within its Q_i that is also in C and is not yet colored by the partial coloring algorithm. Such a son will be colored due to a witness along with n . The same arguments apply to this son and to its son, etc.

Since there is a finite number of nodes in Q_i , we get a cycle of such nodes, which are all in C . \square

We now return to the proof of Theorem 3.11.

PROOF OF THEOREM 3.11. We need to show that in any partial coloring function of G with respect to C , all the nodes of C are colored as they were originally colored by χ , i.e. by F . Thus, for each node $n = (s, \varphi') \in C$ we prove that n is colored F by the partial coloring algorithm with respect to C , regardless of the initial coloring. The proof is by induction on the execution of the partial coloring algorithm. We disregard nodes in $N \setminus C$, for which there is nothing to prove.

Base case: $n \in C$ is colored by rule 1, as a terminal node (leaf). Since $n \in C$, we have that it was originally colored F by χ , which means φ' is either $l \notin L(s)$ or false. Thus n is again colored F by the partial coloring algorithm.

Induction step: We refer to nodes that are colored based on rules 2, 3 and 4 of the (partial) coloring algorithm.

— $n \in C$ is colored by rule 2, as an \vee -node.

Suppose to the contrary that n is colored T by the partial algorithm. This means that it has at least one son n' that is already colored by T . However, since $n \in C$ is an \vee -node, then by the construction of the annotated counterexample C , all of its sons are in C , and in particular $n' \in C$. Thus the induction hypothesis, applied to n' which was already colored by our assumption, assures us that n' is colored by F . This contradicts our assumption, i.e. n must be colored by F .

— $n \in C$ is colored by rule 3, as an \wedge -nodes.

Suppose to the contrary that n is colored by T . This means that all of its sons are already colored by T . However, since $n \in C$, then by the construction of the annotated counterexample, n has exactly one son n' in C . Thus, the induction hypothesis, applied to n' which was already colored by our assumption, assures us that n' is colored by F . This contradicts our assumption that all the sons of n , and in particular n' , are colored by T , i.e. n must be colored by F .

— $n \in C$ is colored by rule 4, due to a witness.

If n is colored due to the existence of an AU or EU witness, then by the description of the algorithm, it is colored by F , as required. It remains to show that the witness cannot be AV or EV , which would have caused n to be colored by T . Suppose to the contrary that n is colored due to a witness of the form AV or EV by the partial coloring algorithm. The induction hypothesis provides the information that is needed in order to use Lemma B.7. According to Lemma B.7, we get that n lies on a cycle of nodes from C , which forms a non-trivial SCC. However, by Corollary 3.9, non-trivial SCCs in C are either AU -SCCs or EU -SCCs, which contradicts the assumption that n is colored due to an AV or EV witness (By Lemma 2.6, such a witness cannot exist in an AU or EU SCC). Thus the witness that caused the coloring of n cannot be one of the above.

\square

Proof of Theorem 3.12

THEOREM. C is minimal w.r.t. the property of being independent of the game-graph G .

PROOF. It suffices to show that any node and edge that will be removed from C will result in a subgraph C' that is not independent of G . That is, the partial coloring of G with respect to C' may change its coloring. This property is guaranteed because of the following. If the son of an \wedge -node n (or the edge that connects them) is removed, then there is no longer a son for this node in C' , thus there exists an initial coloring function (input for the partial coloring algorithm) that colors all the sons of n by T , which will cause n to become colored T by the partial coloring algorithm. If a son of an \vee -node n is removed, then this son can be colored by T by the initial coloring function (input of the partial coloring algorithm), thus its parent n will also be colored T by the partial coloring algorithm. \square

C. PROOFS OF SECTION 4

Proof of Lemma 4.5

LEMMA. Let n be a node that is uncolored at the beginning of phase 2 in its set Q_i . Then n lies on a non-trivial SCC that is a subgraph of Q_i and all nodes of the SCC are uncolored at the beginning of phase 2.

We conclude that if a set Q_i has uncolored nodes at the beginning of phase 2 then Q_i is a non-trivial may-MSCC.

PROOF. Consider an uncolored node in Q_i , denoted n . n has outgoing edges only to nodes in smaller sets Q_j 's, which are already colored, or to nodes in the same set Q_i (by the choice of the order \leq). Since n is uncolored, we have that it has an outgoing edge to an uncolored node n' , otherwise it could be colored in phase 1. This node n' can only be in the same set Q_i (the others are already colored). Thus, each uncolored node in Q_i has a son within Q_i that is not colored. Since Q_i is finite, this results in a non-trivial SCC, whose nodes are all within Q_i . \square

Proof of Lemma 4.6

LEMMA. Let Q_i be the set that is handled by the coloring algorithm at its i th iteration, and let n be a node in Q_i that is uncolored at the beginning of phase 2b. Then

- (1) If Q_i has an EU or AV witness, then n lies on a non-trivial may-SCC that is a subgraph of Q_i and all nodes of the may-SCC are uncolored at the beginning of phase 2b.
- (2) If Q_i has an AU or EV witness, then n lies on a non-trivial must-SCC that is a subgraph of Q_i and all nodes of the must-SCC are uncolored at the beginning of phase 2b.

PROOF. We first show that such a node n lies on an uncolored non-trivial SCC.

Any node whose sons are all colored already in phase 1 or phase 2a, also gets colored in these phases. This is shown by a simple case analysis of the rules of the coloring algorithm. In phase 1 all the cases where all the sons are colored are handled. As for phase 2a, the reasoning is similar, with the exception that some of

the cases where all the sons of a node are colored by *definite* colors (T or F) are *not* handled. Yet, such cases are not possible in phase 2a: if all the sons of a node are colored by definite colors in this phase, this means that they were already colored in phase 1 (since in phase 2a nodes get colored by $?$ only), which would make their parent already colored as well.

Thus, each node n' that is uncolored at the beginning of phase 2b has an uncolored son n'' . Since all the sons of n' are either in the same set Q_i or in smaller sets which are already colored, the uncolored son n'' is definitely within Q_i as well. Moreover, since Q_i is finite, this results in a non-trivial SCC, which is a subgraph of Q_i and is uncolored at the beginning of phase 2b.

It remains to refer to the type of the resulting SCC (*must* versus *may*). For the case of an EU or an AV witness, the claim is already implied, since every SCC is a *may*-SCC. We now consider the case of an AU or an EV witness. Only AX and EX nodes have outgoing progress edges, which affect the type of the SCC. Thus, it suffices to show that for AX -nodes (in a set with an AU witness) and EX -nodes (in a set with an EV witness), at least one of the uncolored sons is a *must*-son. This is also shown by a case analysis of the rules of the coloring algorithm: if all the *must*-sons of such a node n' were colored, then n' would already be colored either in phase 1 (if at least one of them is colored F for AX or T for EX), or in phase 2a (otherwise). Hence, in the case of an AU or an EV witness, an uncolored non-trivial *must*-SCC is formed. \square

Proof of Theorem 4.3

THEOREM. *All the nodes in the game-graph get colored by the 3-valued coloring algorithm.*

PROOF. It suffices to prove that whenever phase 2 of the coloring is reached (with the existence of uncolored nodes), the set Q_i that is handled by the algorithm indeed has exactly one witness. This results from Lemma 4.5 that guarantees that Q_i is a non-trivial MSCC, as well as Lemma 2.6, which guarantees that a non-trivial *may*-MSCC has exactly one witness. Thus, all the remaining uncolored nodes in Q_i are colored according to this witness. As a conclusion, we get that no node is left uncolored. \square

Proof of Theorem 4.4

THEOREM. *Let $G_{M \times \varphi}$ be a 3-valued game-graph, colored by the above 3-valued coloring algorithm with χ being the coloring function, and let n be a node in the game-graph, then:*

- (1) $\chi(n) = T$ iff \exists loise has a winning strategy starting at n .
- (2) $\chi(n) = F$ iff \forall belard has a winning strategy starting at n .
- (3) $\chi(n) = ?$ iff none of the players has a winning strategy starting at n .

PROOF. The proof is by induction on the computation of the coloring algorithm. It suffices to prove the implication from each result of the coloring of n to the existence (or non-existence) of winning strategies.

Base case: n is a terminal node. On the one hand, by the coloring algorithm, n is colored according to the player that wins the game in such a configuration. On the other hand, this also determines the existence of a winning strategy, since each play that starts from such a configuration also ends in it. Thus, the claim is implied.

Induction step: We refer to the different phases of the coloring algorithm.

Phase 1: n is colored in phase 1, due to the coloring of its sons. We use a case analysis of the rules of phase 1, excluding the rule dealing with a terminal node, which was handled in the base case.

Consider the case where n is an AX -node, colored by the AX rule. The first move in each play that starts from the configuration n is done by \forall belard.

- (1) If n is colored T then by the description of the algorithm all its may-sons are colored T , which means \exists loise has a winning strategy from each one of them (by the induction hypothesis). Thus, no matter which move \forall belard makes (this is his turn), she can win, i.e. she has a winning strategy from n , which is the union of the winning strategies of all the sons of n .
- (2) If n is colored F then by the description of the algorithm it has a must-son n' that is colored F , which means \forall belard has a winning strategy from n' (by the induction hypothesis). Thus, \forall belard has a winning strategy from n , which is to choose n' as the first move and continue by the guaranteed winning strategy from n' . Note, that the choice of n' provides a false-consistent play, since it is a must-son of n .
- (3) If n is colored by $?$ then by the description of the algorithm, it has a may-son n' that is colored by $?$ or F , and all its must-sons are colored by T or $?$. The existence of n' assures us that \exists loise does not have a winning strategy for a game that starts from n , since \forall belard (which makes the move in such a configuration) can always choose as his first move the configuration n' , for which \exists loise does not have a winning strategy by the induction hypothesis. In addition, the information about the must sons, assures us that \forall belard does not have a winning strategy from n , since for the play to be false-consistent \forall belard has to proceed to one of the must-sons, but from them, by the induction hypothesis, he does not have a winning strategy.

If n is an \wedge -node, other than AX -node, then the same proof holds, where instead of may or must edges we use auxiliary edges.

If n is an EX -node, or another \vee -node, then the proof is symmetric to the proof for AX or \wedge -nodes respectively, where the coloring rules are opposite and so are the roles of the players.

Phase 2: n is colored during phase 2, due to a witness in a Q_i that is a non-trivial may-MSCC (by Lemma 4.5). We use a case analysis of the witness of Q_i .

Consider the case where the witness is $A(\varphi_1 U \varphi_2)$ or $E(\varphi_1 U \varphi_2)$. Then n is colored by either $?$ (in phase 2a) or F (in phase 2b).

Before referring to each phase separately, we first prove that either way, \exists loise does not have a winning strategy for the game that starts from n . Let B be the *maximal* subgraph of $G_{M \times \varphi}$ that is reachable from n through uncolored nodes by the time phase 2 (coloring by witness) starts. Obviously, B is a subgraph of Q_i (because by the choice of \leq , only nodes in Q_i or smaller sets are reachable from

n and the ones in smaller sets are already colored). Since it is a subgraph of Q_i , then by Lemma 2.6, its formulae are from $\text{exp}(A(\varphi_1 U \varphi_2))$ or $\text{exp}(E(\varphi_1 U \varphi_2))$.

Suppose to the contrary that \exists loise has a winning strategy for the game that starts from n . Then \exists loise manages to “force” the play to exit B to a configuration for which she has a winning strategy. This is because if the play stays within B , then the play is infinite (there are no terminal nodes in $\text{exp}(A(\varphi_1 U \varphi_2))$ or $\text{exp}(E(\varphi_1 U \varphi_2))$) and the witness is $A(\varphi_1 U \varphi_2)$ or $E(\varphi_1 U \varphi_2)$, thus \exists loise loses.

To contradict the above assumption, let us first show that it is not possible that \forall belard is “forced” to exit B in his moves. Any node in B is uncolored at the beginning of phase 2 and thus lies on an uncolored non-trivial SCC (by Lemma 4.5). Hence, in particular, it has an uncolored son, which is also in B (by the maximality of B). Thus, for every node in B there exists a consecutive node in B that \forall belard can choose in his moves.

To get a contradiction we now show that \exists loise cannot exit B to a configuration for which she has a winning strategy. The only configurations in which \exists loise makes a move are \vee -nodes, with subformulae of the form $\varphi' \vee \varphi''$ or $EX\varphi'$. Thus, if \exists loise manages to exit B , she does it in such a configuration. We consider each possibility separately, and show that each of them leads to a contradiction.

$\varphi' \vee \varphi''$: This means that there exists a configuration $(s', \varphi' \vee \varphi'')$ in B for which, without loss of generality, \exists loise chooses $n' = (s', \varphi')$, which is outside B , as her next move. This node, n' , is already colored at the beginning of phase 1 (otherwise it would be in B as well). In addition, by our assumption \exists loise has a winning strategy from n' , thus by the induction hypothesis it is colored by T . However, by the coloring algorithm this means that the configuration $(s', \varphi' \vee \varphi'')$ that is *in* B could already be colored by T as well in phase 1, in contradiction to the property that B is uncolored.

$EX\varphi'$: If \exists loise exits B in a configuration of the form $(s', EX\varphi')$, then similar arguments apply, with the difference that in this case \exists loise chooses a *must*-son $n'' = (s'', \varphi')$ outside B as her next move. It has to be a *must*-son since this move is a part of her winning strategy, and as such it has to ensure that the implied plays are true-consistent. By the same arguments, this configuration (node) n'' is already colored. In addition, \exists loise has a winning strategy from n'' , thus by the induction hypothesis it is already colored by T . Again, this results in contradiction, since by the coloring algorithm this would cause its parent that is *in* B to be colored by T as well in phase 1.

We now show that if n is colored by $?$ (in phase 2a), then none of the players has a winning strategy from it, and if it is colored by F (in phase 2b), then \forall belard has a winning strategy.

Phase 2a: Consider the case where n is colored in phase 2a. i.e., it is colored by $?$. In this case, it remains to show that \forall belard does not have a winning strategy for the game that starts from n (we already know this about \exists loise). The proof is by induction on the computation of phase 2a, where the main idea is that a node n is colored by $?$ only when the F option is overruled.

—If n is an AX -node (or an \wedge -node), this means that all its *must*-sons are colored by $?$ or T , such that indeed \forall belard has no winning strategy from this node: he can either choose a *may*-son, which will make the play not

false-consistent, or he can choose a must-son, for which he has no winning strategy, by the induction hypothesis.

- If n is an EX -node (or an \vee -node), this means that it has a may-son n' that is colored by $?$ or T , such that indeed \forall belard has no winning strategy from this node: in such a configuration \exists loise makes the first move, thus she can choose n' for which \forall belard has no winning strategy by the induction hypothesis.

Phase 2b: Consider the case where n is colored in phase 2b. i.e., it is colored by F . We need to show that \forall belard has a winning strategy for the game that starts from n .

The proof here is similar to the proof that was used to show that \exists loise does not have a winning strategy from any node that is left uncolored at the beginning of phase 2, where here we claim that the winning strategy of \forall belard is basically to stay within the subgraph B . The main difference is that here we choose B to be the maximal uncolored subgraph of $G_{M \times \varphi}$ that is reachable from n through uncolored nodes by the time phase 2b (rather than phase 2) of the coloring algorithm starts. Furthermore, in the case of a witness of the form $A(\varphi_1 U \varphi_2)$ we restrict the nodes in B to nodes that are reachable from n through must-edges or auxiliary edges (but *not* may-edges). In addition, we base our arguments on Lemma 4.6 rather than on Lemma 4.5 and clearly we adapt our arguments to match the relevant rules of the coloring algorithm⁷.

If the witness is of the form $A(\varphi_1 V \varphi_2)$ or $E(\varphi_1 V \varphi_2)$, then the proof is symmetric, where the coloring rules are opposite and so are the roles of the players.

□

D. PROOFS OF SECTION 5

We start by presenting an important observation.

OBSERVATION D.1. *Each node $n_c \in C_C$, that is processed by the concretization algorithm `ConcretizeCounter`, has exactly one matching node n_a in C_A . This is the node that is paired to n_c when it is added to *new*. The nodes n_c and n_a share the same formula. Furthermore, let s_c, s_a be the states of n_c, n_a respectively, then $s_c \in \gamma(s_a)$.*

Proof of Lemma 5.1

LEMMA. *The algorithm `ConcretizeCounter` does not fail. That is, given an abstract annotated counterexample, it always succeeds to produce a result.*

PROOF. We show that whenever a state needs to be chosen or found by the algorithm (lines 1, 11 and 18), then a suitable state exists. This results from the properties of the mixed simulation between M_A and M_C , induced by γ , and of the abstract annotated counterexample, as follows.

1. Line 1: for each initial abstract state s_{0a} , there exists $s_{0c} \in S_{0C}$ such that $s_{0c} \in \gamma(s_{0a})$. Thus, $\gamma(s_{0a}) \cap S_{0C} \neq \emptyset$, and line 1 is well defined.

⁷The full proof can be found in [Shoham 2003].

For the next cases, let (n_c, n_a) be the pair handled by the algorithm, where $n_c = (s_c, \varphi')$, $n_a = (s_a, \varphi')$ and $s_c \in \gamma(s_a)$ (by Observation D.1).

2. Line 11: for an *AX*-node n_a the abstract annotated counterexample C_A contains a *must-son* $n'_a = (s_a, \varphi_1)$. Thus, $s_a \xrightarrow{\text{must}} s'_a$. By the properties of γ , this ensures that for each concrete state in $\gamma(s_a)$, and in particular for the state s_c of n_c , there exists $s'_c \in \gamma(s'_a)$ such that $s_c \rightarrow s'_c$. Thus $\{s'_c \in S_C : s_c \rightarrow s'_c\} \cap \gamma(s'_a) \neq \emptyset$, and line 11 never fails.
3. Line 18: if $s_c \in \gamma(s_a)$, then for each s'_c such that $s_c \rightarrow s'_c$, there exists an abstract state s'_a such that $s'_c \in \gamma(s'_a)$. Thus line 18 never fails.

□

Proof of Lemma 5.2

LEMMA. *Let C_C be the result of **ConcretizeCounter**. C_C is a subgraph of the concrete game-graph G_C for M_C and φ , containing an initial node.*

PROOF. This results from the following properties.

- C_C is initialized by a node that consists of a concrete initial state and the original formula φ (which also appeared in the abstract initial node). Thus, it is an initial node of G_C .
- It can be shown by induction that every node $n'_c \in C_C$ is a legal son of each of its parents (in particular, this means that it is a legal node of G_C). The correctness in terms of the formulae in the nodes of C_C is “inherited” from C_A , because the formula in a concrete node n'_c is taken from its matching abstract node n'_a . As for the states, the proof is by a simple case analysis. Let n_c be a parent of n'_c in C_C . If n_c is an *AX* or an *EX* node (lines 9 and 14), then the algorithm makes sure that there exists a real transition in the model from the state of n_c to the state of its son n'_c (lines 11 and 15). Otherwise, the same state is used both in n_c and in n'_c (lines 22 and 26).

□

Proof of Lemma 5.3

LEMMA. *Let n_c be a node in the subgraph C_C of the concrete game-graph $G_C = (N_C, E_C)$, computed by **ConcretizeCounter**. Then given any initial coloring function of $N_C \setminus C_C$, n_c is colored by F in the (concrete) partial coloring of G_C w.r.t. C_C .*

PROOF. Given any initial coloring function that colors $N_C \setminus C_C$, the proof is by induction on the computation of the *concrete* partial coloring algorithm w.r.t. C_C (see Section 2.1.2). We consider nodes in C_C only, and show that they are colored F by the partial coloring algorithm. The rest are colored by the initial coloring function, however their colors are not relevant to the proof. For each node $n_c \in C_C$ we denote its *matching* abstract node by n_a (see Observation D.1).

Base case: $n_c \in C_C$ is colored by rule 1, as a terminal node. Its matching abstract node $n_a = (s_a, \varphi')$ has the same formula, thus it is also a terminal node. Further-

more, since $n_a \in C_A$, we have that it was colored by F . Thus, the formula of n_a and therefore of n_c can be either false or l where $\neg l \in L_A(s_a)$. In the first case, n_c is obviously colored by F . In the second case, by Observation D.1, we know that the (concrete) state s_c of n_c belongs to $\gamma(s_a)$. In addition, $\neg l \in L_A(s_a)$ means that $\neg l$ labels *all* the concrete states in $\gamma(s_a)$ and in particular $\neg l \in L_C(s_c)$. Hence, n_c is colored by F as well.

Induction step: We refer to nodes that are colored based on rules 2, 3 and 4 of the concrete (partial) coloring algorithm.

— $n_c \in C_C$ is colored by rule 2, as an \vee -node.

Assume to the contrary that n_c is colored by T . This means that it has a son $n'_c \in G_C$ that is already colored by T . By the description of the concretization algorithm, we get that all of the sons of n_c from G_C appear in C_C (lines 15-16, 25-26). This is true in particular for n'_c . Therefore, by the induction hypothesis, if n'_c is already colored, then it must be colored by F , in contradiction.

— $n_c \in C_C$ is colored by rule 3, as an \wedge -node.

Assume to the contrary that n_c is colored by T . This means that all of its sons in G_C are already colored by T . By the concretization algorithm, one of its sons, denoted by n'_c , is in C_C . However, by our assumption, n'_c is already colored by T , which contradicts the induction hypothesis.

— $n_c \in C_C$ is colored by rule 4, due to a witness.

It suffices to show that the witness cannot be AV or EV . Similarly to Lemma B.7, it can be shown that for n_c to be colored by a witness, it must reside on a loop of nodes that matches an abstract loop that passes through its matching node n_a in C_A . By a generalization of Lemma 3.8 to the abstract case, if n_a resides on a loop in C_A , then at least one of the nodes n'_a on the loop must have been colored by a witness. Furthermore, since n'_a lies on a loop that contains n_a , then by Lemma 2.6 which holds for the abstract case as well, their formulae result from the same witness. Furthermore, n_c has the same formula as n_a , thus its formula also results from the same witness. Now, suppose to the contrary that the witness that caused the coloring of n_c is AV or EV . Then so is the witness that caused the coloring of n'_a , which means n'_a is colored by T , in contradiction to its being in C_A .

□

Proof of Lemma 5.4

LEMMA. *The subgraph C_C , produced by ConcreteCounter, is minimal w.r.t. the property of being independent of the concrete game-graph G_C .*

PROOF. similar to the proof of Theorem 3.12. □

E. PROOFS OF SECTION 6

Proof of Lemma 6.2

LEMMA. *The algorithm FailureSearch terminates.*

PROOF. As long as the current node n is not a failure node, the algorithm continues to $cont(n)$. This is a node that was colored ? *prior* to n . Note, that by the definition of a failure node, there always exists such a node if n is not a failure node. Thus, each recursive call is applied on a node that was colored ? earlier. Hence, the number of recursive calls is bounded by the running time of the coloring algorithm, which is finite. \square

Proof of Lemma 6.3

LEMMA. *A failure node, and in particular the one returned by the algorithm FailureSearch, is a node colored by ?, which is one of the following.*

- (1) *A terminal node of the form (s_a, l) where $l \in Lit$.*
- (2) *An AX-node (EX-node) that has a may-son colored by F (T).*
- (3) *An AX-node (EX-node) that was colored during phase 2a based on an AU (EV) witness, and has a may-son colored by ?.*

PROOF. We first show that a failure node cannot be an \wedge -node or an \vee -node other than an AX-node or an EX-node. By its definition, a failure node n is colored by ? and none of its sons were colored by ? at the time it got colored. A case analysis of the coloring algorithm shows that this can never happen if n is an \wedge -node or an \vee -node other than an AX-node or an EX-node. For example, if n gets colored ? in phase 2a as an \wedge -node (which is not an AX-node) in a set with an AU or EU witness, then by the description of the coloring algorithm, both its sons are already colored by T or ?. It is not possible that both of them were colored by T , since this would mean they were already colored this way in phase 1, thus n would already be colored in this phase as well. Thus, n has at least one son that is already colored ? when n gets colored. We omit the other cases⁸. We conclude that a failure node can only be a terminal node, an AX-node or an EX-node. We consider each of these possibilities separately and show that they fulfill one of the cases of the lemma.

- If n is a terminal node, then it must be of the form (s_a, l) since terminal nodes of the form $(s_a, true)$, $(s_a, false)$ are colored by definite colors rather than ?. Thus it fulfills case 1.
- If n is an AX-node, then we have two possibilities. If n has a may-son colored by F , then case 2 applies and we are done. Otherwise, we show that case 3 applies. We first show that n was colored during phase 2a based on an AU witness. This is proved by elimination. (a) n could not be colored during phase 1 or during phase 2a based on an AV or an EV witness, since for this to happen, n had to have a may-son n' that was already colored by F or ? when n got colored. Yet, the F option contradicts our assumption that n does not have a may-son colored by F (at the end of the coloring and thus also when n got colored), and the ? option contradicts the property that n is a failure node. (b) n could not be colored during phase 2b, since this would make it colored by a definite color rather than ?. (c) n cannot belong to a set Q_i with an EU witness since it is an

⁸The full proof can be found in [Shoham 2003].

AX-node (by Lemma 2.6). The only remaining possibility is that n was colored during phase 2a based on an *AU* witness.

It remains to show that n has at least one may-son that is colored by ?. Note, that we claim that this son is colored by ? at the *end* of the coloring, but *not* at the time n got colored (otherwise n would not be a failure node). Since n does not have a may-son colored by *F*, then all of its may-sons are clearly colored by *T* or ?. The case where all of them are colored by *T* would make n colored by *T* as well, in contradiction to its being a failure node. Thus, this case is impossible and n has at least one may-son that is colored by ?. We conclude that case 3 holds.

—If n is an *EX*-node the analysis is similar to the analysis of an *AX*-node.

□

Proof of Theorem 6.5

THEOREM. *For finite concrete models, iterating the abstraction-refinement process is guaranteed to terminate with a definite answer.*

PROOF. Since refinement is done by splitting the abstract states, then applying the refinement on the abstract model results in a refined abstract model with the following properties. Every state s_r in the refined model has some *super-state* s_u in the unrefined model, in the sense that the set of concrete states that s_r represents is a subset of those represented by s_u . Furthermore, since our refinement splits at least one state, it ensures that at least one of the refined states represents strictly less concrete states than its super-state. Thus, the number of iterations in the abstraction-refinement process is bounded by the number of concrete states and is guaranteed to end when the state space is finite. □

F. PROOFS OF SECTION 7

Proof of Theorem 7.2

THEOREM. *Let M_C be a concrete model, and let M_A be an abstract KMTS, such that $M_C \preceq M_A$, with a concretization function γ . Let G_A be the game-graph for M_A and some CTL formula φ , constructed and colored by the incremental algorithm. Then, for each $n_a = (s_a, \varphi_1) \in G_A$:*

- (1) *If n_a is colored *T*, then $\forall s_c \in \gamma(s_a) : [(M_C, s_c) \models \varphi_1] = tt$.*
- (2) *If n_a is colored *F*, then $\forall s_c \in \gamma(s_a) : [(M_C, s_c) \models \varphi_1] = ff$.*

In order to prove Theorem 7.2, we wish to provide variations on Theorems 4.7 and 2.16 that will hold in the case of incremental model checking. Theorem 7.2 will then easily follow. We first need to prepare some background. We note that the incremental coloring algorithm can be seen as a partial coloring algorithm (see Definition 3.1) that uses χ_D as an initial coloring function to determine how to color the nodes in D . Therefore, in order to relate its results to the semantics, a new definition of the 3-valued semantics should be used, which also allows an initial evaluation function. That is, it allows the values of some subformulae in some states to be given by an initial function. The formal definition follows.

Definition F.2. Let M be a KMTS, and let $f : S \times \text{CTL} \rightarrow \{\text{tt}, \text{ff}, \perp\}$ be a partial function that assigns a truth value to (certain) CTL formulae in (certain) states of M . We call f an *initial evaluation function* for M . The *partial 3-valued semantics* of a CTL formula φ in a state s of M w.r.t. f , denoted $[(M, s) \stackrel{3}{\models} \varphi]_f$, is defined as follows.

- (1) If (s, φ) is in the domain of f , then $[(M, s) \stackrel{3}{\models} \varphi]_f = f(s, \varphi)$.
- (2) Otherwise, the definition of $[(M, s) \stackrel{3}{\models} \varphi]_f$ is given by Definition 2.15, except that whenever the definition recursively depends on the value of $[(M, s') \stackrel{3}{\models} \varphi']$ for some state s' and subformula φ' of φ , we now use $[(M, s') \stackrel{3}{\models} \varphi']_f$.

We can induce an initial evaluation function from the initial coloring function of the partial coloring algorithm, used in the incremental algorithm, in the following way.

Definition F.3. Consider the j th iteration of abstraction-refinement. Denote the abstract KMTS used in this iteration by M_A . Let D be the set of nodes remembered from previous iterations and χ_D their coloring. The *induced* initial evaluation function f for M_A is defined as follows. The domain of f consists of all pairs (s_a, φ_1) , where s_a is a state of M_A , that are sub-nodes of nodes in D . For such a node (s_a, φ_1) , that is a sub-node of $n_d \in D$:

$$f(s_a, \varphi_1) = \begin{cases} \text{tt} & \text{if } \chi_D(n_d) = T \\ \text{ff} & \text{if } \chi_D(n_d) = F \end{cases}$$

Definition F.3 assures that whenever the color of $n_d \in D$ is set to be T or F by the initial coloring function χ_D , then for any sub-node (s_a, φ_1) of n_d , the induced initial evaluation function for M_A determines the value of φ_1 in s_a to be the matching value. It can now be shown that a variation of Theorem 4.7 about the correspondence between the coloring and the 3-valued semantics holds when referring to the incremental coloring algorithm, also seen as a partial coloring algorithm, and to the new semantics with the induced initial evaluation function.

THEOREM F.4. *Let $G_A = (N_A, E_A)$ be the (pruned) game-graph, constructed for a KMTS M_A and a CTL formula φ , in the j th iteration of abstraction-refinement. Let $\chi_D : D \rightarrow \{T, F\}$ be the coloring of the nodes remembered from previous iterations, and let f be the induced initial evaluation function for M_A . Then, in the incremental coloring algorithm of G_A , for each $n_a = (s_a, \varphi_1) \in N_A \setminus D$:*

- (1) $[(M_A, s_a) \stackrel{3}{\models} \varphi_1]_f = \text{tt}$ iff $n_a = (s_a, \varphi_1)$ is colored by T .
- (2) $[(M_A, s_a) \stackrel{3}{\models} \varphi_1]_f = \text{ff}$ iff $n_a = (s_a, \varphi_1)$ is colored by F .
- (3) $[(M_A, s_a) \stackrel{3}{\models} \varphi_1]_f = \perp$ iff $n_a = (s_a, \varphi_1)$ is colored by $?$.

Since we are now using a new semantics, the next step is to show that Theorem 2.16, that relates the 3-valued semantics to the concrete semantics, still holds for the new semantics. This is true provided that the initial evaluation function fulfills a *consistency* requirement.

Definition F.5. Let M_C be a concrete model, and let M_A be an abstract KMTS, such that $M_C \preceq M_A$, with a concretization function γ . An initial evaluation

function f for M_A is *consistent w.r.t.* M_C if for every state s_a of M_A and every CTL formula φ we have that if $f(s_a, \varphi) \in \{\text{tt}, \text{ff}\}$, then $\forall s_c \in \gamma(s_a): [(M_C, s_c) \models \varphi] = f(s_a, \varphi)$.

That is, consistency holds if whenever a value $val \in \{\text{tt}, \text{ff}\}$ is given for φ in s_a by the initial evaluation function, then it is also the case that for every $s_c \in \gamma(s_a)$, $[(M_C, s_c) \models \varphi] = val$. We now have the following.

THEOREM F.6. *Let M_C be a concrete model, and let M_A be an abstract KMTS, such that $M_C \preceq M_A$, with a concretization function γ . Let f be an initial evaluation function for M_A . If f is consistent w.r.t. M_C , then for every CTL formula φ , and every $s_c \in \gamma(s_a)$, where s_c is a state of M_C and s_a is a state of M_A , we have that:*

- (1) $[(M_A, s_a) \stackrel{\exists}{\models} \varphi]_f = \text{tt}$ implies that $[(M_C, s_c) \models \varphi] = \text{tt}$.
- (2) $[(M_A, s_a) \stackrel{\exists}{\models} \varphi]_f = \text{ff}$ implies that $[(M_C, s_c) \models \varphi] = \text{ff}$.

We now return to the proof of Theorem 7.2.

PROOF OF THEOREM 7.2. We only sketch the proof. The proof is by natural induction on the number of iterations of abstraction-refinement, in which the incremental algorithm is used. In the base case the incremental algorithm is identical to the regular one, thus the claim holds by Theorems 4.7 and 2.16.

As for the induction step, it suffices to show that the initial evaluation function for the semantics that is induced by χ_D in the incremental algorithm fulfills the consistency requirement. This means that Theorem F.6 holds for it, and together with Theorem F.4 that holds for the incremental coloring, it implies the theorem. The main claim that needs to be shown is as follows. Let $n_a = (s_a, \varphi_1)$ be a sub-node of a node $n_d = (s_d, \varphi_1)$ that was remembered from a previous iteration (i.e., n_d was colored by a definite color). Then the color of n_d will induce a consistent initial evaluation for the semantics of φ_1 in s_a . This is true because based on the induction hypothesis, the value of φ_1 in every concrete state represented by s_d matches the (definite) color of n_d . Since n_a is a sub-node of n_d , then s_a represents a subset of the concrete states represented by s_d , and the latter also holds for the concrete states represented by s_a . Therefore, the initial coloring function induces a consistent initial evaluation for the semantics of φ in s_a . \square