CINA: Suppressing the Detection of Unstable Context Inconsistency

Chang Xu, Wang Xi, S.C. Cheung, Xiaoxing Ma, Chun Cao and Jian Lu

Abstract—Context-aware applications adapt their behavior based on contexts. Contexts can, however, be incorrect. A popular means to build dependable applications is to augment them with a set of constraints to govern the consistency of context values. These constraints are evaluated upon context changes to detect inconsistencies so that they can be timely handled. However, we observe that many context inconsistencies are unstable. They vanish by themselves and do not require handling. Such inconsistencies are detected due to misaligned sensor sampling or improper inconsistency detection scheduling. We call them *unstable context inconsistencies* (or STINs). STINs should be avoided to prevent unnecessary inconsistency handling and unstable behavioral adaptation to applications. In this article, we study STINs systematically, from examples to theoretical analysis, and present algorithms to suppress their detection. Our key insight is that only certain patterns of context changes can make a consistency constraint subject to the detection of STINs. We derive such patterns and proactively use them to suppress the detection of STINs. We implemented our idea and applied it to real-world applications. Experimental results confirmed its effectiveness in suppressing the detection of numerous STINs with negligible overhead, while preserving the detection of stable context inconsistencies that require inconsistency handling.

Index Terms—Constraint, context inconsistency, impact propagation, instability analysis, pervasive computing.

APPENDIX

In earlier discussions, our constraint instability analysis has derived the propagation function for universal formula. Here, we derive propagation functions for the other five formula types. We list them below:

1. $\mathcal{P}_\forall(p_i) := \{p_i\} \cup \{p_{FF}\}$.
2. $\mathcal{P}_\exists(p_i) := \{p_i\} \cup \{p_{TT}\}$.
3. $\mathcal{P}_{and}(p_i) := \{p_i\} \cup \{p_{FF}\}$.
4. $\mathcal{P}_{or}(p_i) := \{p_i\} \cup \{p_{TT}\}$.
5. $\mathcal{P}_{implies}(p_i) :=$
   (1) $\{\text{flip}(p_i)\} \cup \{p_{TT}\}$. // First sub-formula case
   (2) $\{p_i\} \cup \{p_{TT}\}$. // Second sub-formula case
6. $\mathcal{P}_{not}(p_i) := \{\text{flip}(p_i)\}$.

Propagation function derivation:

2. Existential formula:

Consider an existential formula g given by "$\exists\gamma\in C$ [f]", i.e., f is g's sub-formula. Then g's truth value is evaluated to f's satisfiability by any γ value in context C. There are three cases: (1) f is true for all γ values in C; (2) f is false for all γ values; (3) f is true only for some (but not all) γ values. According to existential formula's semantics, g is evaluated to true, false and true, respectively.

We consider four truth value changes in turn. Suppose that truth value change p_{TT} occurs to f's satisfiability by one of γ values. This applies only to Cases (1) and (3). Since p_{TT} does not change f's satisfiability, g's truth value would remain to be true. As a result, truth value change p_{TT} is mapped to p_{TT} as the enforcement by this existential formula. Similarly, truth value change p_{FF} is mapped to $\{p_{TT}, p_{FF}\}$. This is because p_{FF} applies only to Cases (2) and (3), and both cases have g's truth value remain unchanged. We then consider truth value change p_{TF}, which applies only to Cases (1) and (3). This indicates that f's truth value changes from true to false for one γ value. Then g's truth value can either change to false or remain to be true, depending on whether f becomes false for all γ values in C. As a result, truth value change p_{TF} is mapped to $\{p_{TT}, p_{TF}\}$. Finally, truth value change p_{FT} applies only to Cases (2) and (3). For both cases, since f holds for some γ values, g would be evaluated to true, i.e., leading to p_{FT} and p_{TT}, respectively, for the two cases. As a result, truth value change p_{FT} is mapped to $\{p_{TT}, p_{FT}\}$. Combining all these posibilities, an existential formula's propagation function is given as follows:

$\mathcal{P}_\exists(p_i) :=$
(1) $\{p_{TT}\}$, if $p_i == p_{TT}$;
(2) $\{p_{TT}, p_{TF}\}$, if $p_i == p_{TF}$;
(3) $\{p_{TT}, p_{FT}\}$, if $p_i == p_{FT}$;
(4) $\{p_{TT}, p_{FF}\}$, if $p_i == p_{FF}$.

It can be shortened as:

$\mathcal{P}_\exists(p_i) := \{p_i\} \cup \{p_{TT}\}$.

This completes the derivation. □

3. "and" formula:

Consider an "and" formula g given by "(f_1) and (f_2)", i.e., f_1 and f_2 are g's first and second sub-formulae, respec-

• *C. Xu, W. Xi, X. Ma, C. Cao and J. Lu are with the State Key Laboratory for Novel Software Technology and the Department of Computer Science and Technology, Nanjing University, Nanjing (210023), Jiangsu, China. E-mails: changxu@nju.edu.cn, swangex@gmail.com, {xxm, caochun, lj}@nju.edu.cn.*
• *S.C. Cheung is with the Department of Computer Science and Engineering The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China. E-mail: scc@cse.ust.hk.*

tively. Then g's truth value is evaluated to f_1's and f_2's satisfiability. There are three cases: (1) both f_1 and f_2 are evaluated to true; (2) both f_1 and f_2 are evaluated to false; (3) f_1 is evaluated to true but f_2 is evaluated to false, or vice versa. According to "and" formula's semantics, g is evaluated to true, false and false, respectively.

We consider four truth value changes in turn. Suppose that truth value change p_{TT} occurs to f_1's or f_2's satisfiability. This applies only to Cases (1) and (3). Since p_{TT} changes neither f_1's nor f_2's satisfiability, g's truth value would remain unchanged. As a result, truth value change p_{TT} is mapped to {p_{TT}, p_{FF}} as the enforcement by this "and" formula (corresponding to the two cases, respectively). Similarly, truth value change p_{FF} is mapped to {p_{FF}}. This is because p_{FF} applies only to Cases (2) and (3) and both cases have g' truth value remain to be false. We then consider truth value change p_{TF}, which applies only to Cases (1) and (3). For both cases, since either f_1/f_2 or both fail(s) to hold, g would be evaluated to false, i.e., leading to p_{TF} and p_{FF}, respectively, for the two cases. As a result, truth value change p_{TF} is mapped to {p_{TF}, p_{FF}}. Finally, truth value change p_{FT} applies only to Cases (2) and (3). For Case (2), f_1's truth value changes from false to true while f_2's remains to be false, or vice versa. Thus g's truth value remains to be false. For Case (3), since both f_1 and f_2 are evaluated to true, g's truth value changes from false to true. As a result, truth value change p_{FT} is mapped to {p_{FT}, p_{FF}}. Combining all these posibilities, an "and" formula's propagation function is given as follows:

$\mathcal{P}_{\text{and}}(p_i)$:=
(1) {p_{TT}, p_{FF}}, if p_i == p_{TT};
(2) {p_{TF}, p_{FF}}, if p_i == p_{TF};
(3) {p_{FT}, p_{FF}}, if p_i == p_{FT};
(4) {p_{FF}}, if p_i == p_{FF}.

It can be shortened as:

$\mathcal{P}_{\text{and}}(p_i) := \{p_i\} \cup \{p_{FF}\}$.

This completes the derivation. □

4. "or" formula:

Consider an "or" formula g given by "(f_1) or (f_2)", i.e., f_1 and f_2 are g's first and second sub-formulae, respectively. Then g's truth value is evaluated to f_1's or f_2's satisfiability. There are three cases: (1) both f_1 and f_2 are evaluated to true; (2) both f_1 and f_2 are evaluated to false; (3) f_1 is evaluated to true but f_2 is evaluated to false, or vice versa. According to "or" formula's semantics, g is evaluated to true, false and true, respectively.

We consider four truth value changes in turn. Suppose that truth value change p_{TT} occurs to f_1's or f_2's satisfiability. This applies only to Cases (1) and (3). Since p_{TT} changes neither f_1's nor f_2's satisfiability, g's truth value would remain to be true. As a result, truth value change p_{TT} is mapped to p_{TT} as the enforcement by this "or" formula. Similarly, truth value change p_{FF} is mapped to {p_{TT}, p_{FF}}. This is because p_{FF} applies only to Cases (2) and (3), and both cases have g's truth value remain unchanged. We then consider truth value change p_{TF}, which applies only to Cases (1) and (3). For Case (1), f_1's truth value changes from true to false while f_2's remains to be true, or vice versa. Thus g's truth value remains to be true. For Case (3), since both f_1 and f_2 are evaluated to false, g's truth value changes from true to false. As a result, truth value change p_{TF} is mapped to {p_{TT}, p_{TF}}. Finally, truth value change p_{FT} applies only to Cases (2) and (3). For both cases, since either f_1/f_2 or both hold(s), g would be evaluated to true, i.e., leading to p_{FT} and p_{TT}, respectively, for the two cases. As a result, truth value change p_{FT} is mapped to {p_{TT}, p_{FT}}. Combining all these posibilities, an existential formula's propagation function is given as follows:

$\mathcal{P}_{\text{or}}(p_i)$:=
(1) {p_{TT}}, if p_i == p_{TT};
(2) {p_{TT}, p_{TF}}, if p_i == p_{TF};
(3) {p_{TT}, p_{FT}}, if p_i == p_{FT};
(4) {p_{TT}, p_{FF}}, if p_i == p_{FF}.

It can be shortened as:

$\mathcal{P}_{\text{or}}(p_i) := \{p_i\} \cup \{p_{TT}\}$.

This completes the derivation. □

5. "implies" formula:

Consider an "implies" formula g given by "(f_1) implies (f_2)", i.e., f_1 and f_2 are g's first and second sub-formulae, respectively. Then g's truth value is evaluated to f_2's or $\neg f_1$'s satisfiability. There are four cases: (1) both f_1 and f_2 are evaluated to true; (2) both f_1 and f_2 are evaluated to false; (3) f_1 is evaluated to true and f_2 is evaluated to false; (4) f_1 is evaluated to false and f_2 is evaluated to true. According to "implies" formula's semantics, g is evaluated to true, true, false and true, respectively.

We first discuss truth value changes applied to the first sub-formula f_1. Suppose that truth value change p_{TT} occurs to f_1's satisfiability. This applies only to Cases (1) and (3). Since p_{TT} does not change f_1's satisfiability, g's truth value would remain unchanged. As a result, truth value change p_{TT} is mapped to {p_{TT}, p_{FF}} as the enforcement by this "implies" formula (corresponding to the two cases, respectively). Similarly, truth value change p_{FF} is mapped to {p_{TT}}. This is because p_{FF} applies only to Cases (2) and (4) and both cases have g' truth value remain to be true. We then consider truth value change p_{TF}, which applies only to Cases (1) and (3). For both cases, since f_1 fails to hold, g would be evaluated to true, i.e., leading to p_{TT} and p_{FT}, respectively, for the two cases. As a result, truth value change p_{TF} is mapped to {p_{TT}, p_{FT}}. Finally, truth value change p_{FT} applies only to Cases (2) and (4). For Case (2), f_1's truth value changes from false to true while f_2's remains to be false. Thus g's truth value changes from true to false. For Case (4), since both f_1 and f_2 are evaluated to true, g's truth value remains to be true. As a result, truth value change p_{FT} is mapped to {p_{TT}, p_{TF}}. Combining all these posibilities, one part of an "implies" formula's propagation function is given as follows:

$\mathcal{P}_{\text{implies}}(p_i)$:= // First sub-formula case
(1) {p_{TT}, p_{FF}}, if p_i == p_{TT};
(2) {p_{TT}, p_{FT}}, if p_i == p_{TF};
(3) {p_{TT}, p_{TF}}, if p_i == p_{FT};
(4) {p_{TT}}, if p_i == p_{FF}.

It can be shortened as:

$\mathcal{P}_{\text{implies}}(p_i) := \{\text{flip}(p_i)\} \cup \{p_{TT}\}$. // First sub-formula case

We then discuss truth value changes applied to the second sub-formula f_2. Suppose that truth value change p_{TT} occurs to f_2's satisfiability. This applies only to Cases (1) and (4). Since p_{TT} does not change f_2's satisfiability, g's truth value would remain to be true. As a result, truth value change p_{TT} is mapped to $\{p_{TT}\}$ as the enforcement by this "implies" formula. Similarly, truth value change p_{FF} is mapped to $\{p_{TT}, p_{FF}\}$. This is because p_{FF} applies only to Cases (2) and (3) and both cases have g' truth value remain unchanged. We then consider truth value change p_{TF}, which applies only to Cases (1) and (4). For Case (1), f_2's truth value changes from true to false while f_1's remains to be true. Thus g's truth value changes from true to false. For Case (4), since both f_1 and f_2 are evaluated to false, g's truth value remains to be true. As a result, truth value change p_{TF} is mapped to $\{p_{TT}, p_{TF}\}$. Finally, truth value change p_{FT} applies only to Cases (2) and (3). For both cases, since f_2 holds, g would be evaluated to true, i.e., leading to p_{TT} and p_{FT}, respectively, for the two cases. As a result, truth value change p_{TF} is mapped to $\{p_{TT}, p_{FT}\}$. Combining all these posibilities, the other part of an "implies" formula's propagation function is given as follows:

$\mathcal{P}_{\text{implies}}(p_i)$:= // Second sub-formula case
(1) $\{p_{TT}\}$, if $p_i == p_{TT}$;
(2) $\{p_{TT}, p_{TF}\}$, if $p_i == p_{TF}$;
(3) $\{p_{TT}, p_{FT}\}$, if $p_i == p_{FT}$;
(4) $\{p_{TT}, p_{FF}\}$, if $p_i == p_{FF}$.

It can be shortened as:

$\mathcal{P}_{\text{implies}}(p_i) := \{p_i\} \cup \{p_{TT}\}$. // Second sub-formula case

Combining these two parts, an "implies" formula's propagation function is given as follows:

$\mathcal{P}_{\text{implies}}(p_i)$:=
(1) $\{\text{flip}(p_i)\} \cup \{p_{TT}\}$. // First sub-formula case
(2) $\{p_i\} \cup \{p_{TT}\}$. // Second sub-formula case

This completes the derivation. □

6. "not" formula:

Consider a "not" formula g given by "not (f)", i.e., f is g's sub-formula. Then g's truth value is evaluated to ¬ f's satisfiability. There are two cases: (1) f is evaluated to true; (2) f is evaluated to false. According to "not" formula's semantics, g is evaluated to false and true, respectively.

We consider four truth value changes in turn. Suppose that truth value change p_{TT} occurs to f's satisfiability. This applies only to Case (1). Since p_{TT} does not change f's satisfiability, g's truth value would remain to be false. As a result, truth value change p_{TT} is mapped to $\{p_{FF}\}$ as the enforcement by this "not" formula. Similarly, truth value change p_{FF} is mapped to $\{p_{TT}\}$. This is because p_{FF} applies only to Case (2), which has g' truth value remain to be true. We then consider truth value change p_{TF}, which applies only to Case (1). Since f fails to hold, g's truth value changes from false to true, i.e., leading to p_{FT}. As a result, truth value change p_{TF} is mapped to $\{p_{FT}\}$. Finally, truth value change p_{FT} applies only to Case (2). Since f holds, g's truth value changes from true to false, i.e., leading to p_{TF}. As a result, truth value change p_{FT} is mapped to $\{p_{TF}\}$. Combining all these posibilities, an "not" formula's propagation function is given as follows:

$\mathcal{P}_{\text{not}}(p_i)$:=
(1) $\{p_{FF}\}$, if $p_i == p_{TT}$;
(2) $\{p_{FT}\}$, if $p_i == p_{TF}$;
(3) $\{p_{TF}\}$, if $p_i == p_{FT}$;
(4) $\{p_{TT}\}$, if $p_i == p_{FF}$.

It can be shortened as:

$\mathcal{P}_{\text{not}}(p_i) := \text{flip}(p_i)$.

This completes the derivation. □

The above five derived propagation functions are exactly what we list for the five formula types earlier. This completes all derivations.