

AUFGABE 7: ABSTRAKTE INTERPRETATION

In dieser Aufgabe werden Sie zunächst einen Sensorstub sowie einen Ringpuffer zur Zwischenspeicherung der Messergebnisse implementieren. Im Anschluss sollen die Abwesenheit von Laufzeitfehlern in beiden Komponenten mittels Astrée nachgewiesen werden. Ferner wird ebenfalls mittels Astrée im Rahmen der erweiterten Übungen die numerische Stabilität eines digitalen α - β -Filter bewiesen.

Grundlegende Übungen

1. Einführende Frage Was ist der Unterschied zwischen den Astrée-Konzepten der *Known Facts* und der *Assertions*? Wann sollte welches dieser beiden Konzepte zum Einsatz kommen?

Antwort:

Sensor

2. Sensor-Stub Implementieren Sie einen Stub für einen Sensor. Dieser soll bei jedem Aufruf einen simulierten Messwert vom Datentyp `float` liefern. Sorgen Sie dabei dafür, dass für Astrée klar ist, dass es von den auf den Übungsfolien angegebenen einschränkenden Annahmen bezüglich des konkreten Messwerts ausgehen darf.

≡ volatile

Wenn Sie möchten, können Sie hier die in der Datei `sensor.dat` enthaltenen Messwerte verwenden um Ihre Implementierung später testen zu können. Welche Teile Ihrer Implementierung sollte Astrée sehen? Wieso?

≡ fscanf(3p)

Antwort:

Ringpuffer

Moderne Mikrocontroller stellen häufig eine DMA-Einheit zur Verfügung, die Peripheriedaten *asynchron* zum Prozessor beschaffen kann. Deswegen sollen in dieser

Aufgabe die von der Sensorhardware gelieferten Daten in Schüben verarbeitet werden.

3. Implementierung Implementieren Sie hierzu einen *Ringpuffer*, der diese Daten-schübe aufnimmt. Machen Sie sich zunächst mit der vorgegebenen Schnittstelle des Ringpuffers in den Dateien `include/bndbuf.h` und `src/bndbuf.c` vertraut. Vervollständigen Sie nun die Implementierung und testen Sie ihre Lösung indem Sie den Ringpuffer in `src/main.c` integrieren.

☞ `src/bndbuf.c`

4. Laufzeitfehler Weisen Sie nun die Abwesenheit von Laufzeitfehlern wie etwa Speicherzugriffsfehler in Ihrer Implementierung mittels Astrée nach. Verwenden Sie für den Nachweis ausschließlich Assertions. Diese Teilaufgabe gilt als erfolgreich bearbeitet, wenn Astrée in dem von Ihnen implementierten Modul keine Fehler mehr meldet.

5. Korrektheit Versuchen Sie nun mittels Astrée auch die funktionale Korrektheit ihrer Implementierung zu beweisen. Betrachten Sie dabei mindestens die folgenden Eigenschaften Ihrer Implementierung:

- Die Füllstandsvariable `count` verlässt nie die Pufferkapazität.
- Die Füllstandsvariable `count` entspricht der Differenz (unter Berücksichtigung der Umlaufsemantik an den Puffergrenzen) zwischen Schreib- und Lesezeiger.
- Falls der Puffer nicht vollständig belegt ist, speichert `bnd_put` das Element tatsächlich im Puffer ab.

Hinweis: Nicht alle dieser Eigenschaften lassen sich tatsächlich durch Astrée nachweisen. Wieso? Betrachten Sie dazu den Analysezustand an verschiedenen Stellen Ihrer Implementierung. Welche Zusammenhänge müssten Sie ausdrücken, können diese aber nicht durch Annotationen formulieren? Wie sind die relevanten Informationen im Analysezustand modelliert?

☞ `__ASTREE_analysis_log`

☞ `__ASTREE_log_vars`

Antwort:

Erweiterte Übung

Filter

6. Filterstabilität In der Datei `src/ab_filter.c` findet sich eine Implementierung des in der Übung vorgestellten α - β -Filter für den Datentyp `float`. Machen Sie sich zunächst mit der Implementierung vertraut. Die Filterverarbeitung wird in der Datei `src/main.c` durch die `#define EXTENDED`-Direktive aktiviert. Entfernen Sie dazu den Kommentar vor der Direktive.

Welche Nachbedingung muss für die Filterinitialisierung gelten, damit das Filter stabil ist?

Antwort:

Aus den Erfahrungen mit dem I4Copter ist bekannt, dass sich die Filterparameter in den folgenden Bereichen bewegen:

Abtastintervall $T \in (0 \dots 1]$

Prozessvarianz $\sigma_w^2 \in [0.5 \dots 30.0]$

Rauschvarianz $\sigma_v^2 \in [10^{-3} \dots 10^{-1}]$

Wert $y[\kappa] \in [-10 \dots 10]$

Vergewissern Sie sich durch händische Intervallrechnung, inwiefern sich aus diesen Parametern durch die in der Übung beschriebene und hier umgesetzte Filterinitialisierung stabile Werte für α und β ergeben.

Antwort:

Welcher Wertebereich ergibt sich daraus für den Parameter λ ?

Antwort:

Annotieren Sie die obigen Wertebereiche für die Filterparameter sowie die Nach-

bedingungen für stabile Filterparameter α und β in der Filterinitialisierung für Astrée.

Weisen sie nun die Filterstabilität mittels Astrée nach. Wo müssen Sie die Analyse durch zusätzliches Wissen unterstützen? Geben Sie sich an den jeweiligen Stellen den Analysezustand relevanter Variablen aus.

Antwort:

__ASTREE_known_

*

__ASTREE_log_vars

7. Filterschritt Die Funktion `ab_process` führt einen einzelnen Filterschritt aus und wählt dabei zwischen Varianten für den eingeschwungenen Zustand und die Filtererholung aus.

Betrachten Sie zunächst die Implementierung für den eingeschwungenen Zustand. Was kann in jedem Filterschritt schiefgehen?

Antwort:

Wie können Sie erkennen, dass das Filter in einen unbrauchbaren Zustand übergeht?

Antwort:

Treffen Sie bei der Implementierung Maßnahmen, die das Filter in kontrollierter Art und Weise wieder in einen brauchbaren Zustand überführen!

Betrachten Sie dazu zunächst das Filter für den eingeschwungenen Zustand und fangen Sie nun die Fälle ab, in denen das Filter im laufenden Betrieb in einen unbrauchbaren Zustand übergehen würde. Finden Sie die relevanten Stellen, bspw. mittels Astrée und schalten Sie in diesem Fall auf das ebenfalls implementierte Erholungsverfahren um. Auch während der Erholungsphase müssen die Filterparameter α_n und β_n sich im stabilen Bereich befinden. Weisen Sie auch dies mit Hilfe von Astrée nach.

ab_step

ab_step_transient

Hinweis: Achten Sie darauf, dass Sie die Korrektheit ihrer Implementierung nicht nur für bestimmte Filterparametersätze nachweisen, sondern für den gesamten Definitionsbereich! Diese Teilaufgabe gilt als erfolgreich bearbeitet, wenn alle Vor- und Nachbedingungen annotiert sind und Astrée in den relevanten Funktionen keine Fehler mehr meldet.

Hinweise

- Bearbeitung: Gruppe mit je zwei bis drei Teilnehmern.
- Abgabefrist: 06.07.2018
- Fragen bitte an i4ezs@lists.cs.fau.de