

Disclaimer

- This is an example exam for the course “Software Engineering” in summer term 2015.
- As such, it has no implications on the real exam whatsoever.
- The real exam may, among others,
 - be typeset in landscape,
 - have completely different tasks for certain areas,
 - put completely different emphasis (like all COCOMO, nothing else),
 - follow a completely different grading scheme,
 - use different colours,
 - have a different number of total points,
 - pose tasks in a completely different style,
 - be shorter or longer (based on results of still to be conducted experiments),
 - have more difficult or easier tasks,
 - be free of typos or awkward wording,
 - have solvable tasks,
 - etc.
- That said, this example exam is a proper example in the sense that it *could* serve as a valid course assessment for the course “Software Engineering” (it is definitely not much too easy or much too short).
- Note: the blue correction directions are not part of the example exam. If they appear, they probably have accidentally not been switched off. ;-)

Haftungsausschluss

- Das vorliegende Dokument ist ein Beispiel für eine Klausur zur Vorlesung “Softwaretechnik” aus dem Sommersemester 2015.
- Als solches bestehen keinerlei Implikationen bezüglich der am Klausurtermin tatsächlich gestellten Klausur.
- Die eigentliche Klausur könnte, unter anderem,
 - im Querformat gesetzt sein,
 - vollständig andere Aufgaben für bestimmte Bereiche enthalten,
 - vollständig andere Schwerpunkte setzen (zum Beispiel nur COCOMO und sonst nichts),
 - ein vollständig anderes Bewertungsschema anwenden,
 - andere Farben verwenden,
 - eine andere maximale Punktzahl haben,
 - Aufgaben auf eine vollständig andere Art und Weise stellen,
 - kürzer oder länger sein (abhängig von noch durchzuführenden Experimenten),
 - schwerere oder leichtere Teilaufgaben enthalten,
 - keine Rechtschreibfehler und keine sperrigen Formulierungen enthalten,
 - lösbare Aufgaben enthalten,
 - etc.
- Abgesehen davon ist diese Beispielklausur ein gültiges Beispiel für eine Klausur in dem Sinne, daß sie als valider Leistungsnachweis für die Vorlesung “Softwaretechnik” verwendet werden *könnte* (sie ist definitiv nicht viel zu leicht oder viel zu kurz).
- Achtung: die blauen Korrekturhinweise sind nicht Teil der Beispielklausur. Wenn diese sichtbar sind, wurden sie bestimmt unabsichtlich nicht deaktiviert. ;-)

1 Procedure and Process Models

Vorgehens- und Prozessmodelle

Problem 1.1 (Procedure Models)

Aufgabe 1.1 (Vorgehensmodelle)

- Write down the procedure underlying the spiral model. (2)

Geben Sie die Schritte der dem Spiralmodell zugrundeliegende Prozedur an.

- Why is it called spiral model? Where is the spiral? Explain! (1)

Warum wird das Spiralmodell "Spiralmodell" genannt? Wo ist die Spirale? Erläutern Sie!

Correction:

- 0.5 points for each item of the procedure
- 1 point for correct answer (reduce 0.5 if too imprecise)

Problem 1.2 (V-Modell XT)

Aufgabe 1.2 (V-Modell XT)

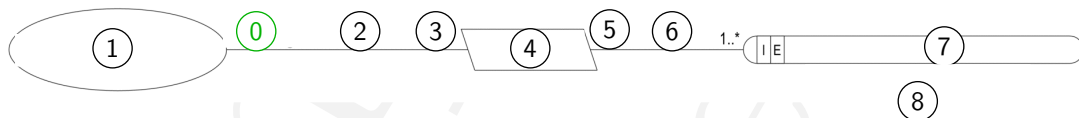
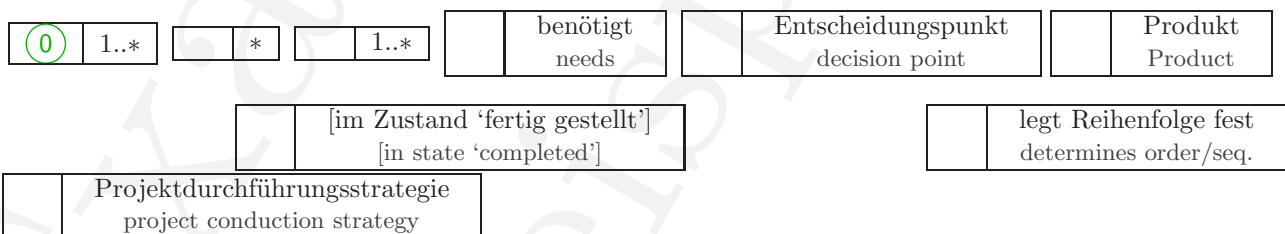


Figure 1: Elements of V-Modell XT.
Abbildung 1: Elemente des V-Modell XT.

Assign the following eight labels to their correct place in Figure 1 (similar to the example "0" in green).¹
Weise die folgenden acht Beschriftungen ihrem korrekten Platz in Abbildung 1 zu (analog dem Beispiel "0" in grün).



(2)

Correction:

- 0.25 points per correct label
- putting two labels A and B at one place counts as the one label "A B", which is wrong

¹In the lecture, we considered figures from the German V-Modell XT document with ad-hoc translations to English.

2 Software Metrics

Software Metriken

Problem 2.1 (Lines of Code)

Aufgabe 2.1 (Anzahl Codezeilen)

```
1 public int convert(char[] str) throws Exception {
2     if (str.length > 6)
3         throw new Exception("Length_exceeded");
4
5     int number = 0;
6     int digit;
7     int i = 0;
8
9     // none of the early submitters found this bug! :-)
10    //
11    if (str[0] == '-')
12        i = 1;
13
14    while (i < str.length){
15        digit = str[i] - '0';
16        if (digit < 0 || digit > 9)
17            throw new Exception("Invalid_character");
18        number = number * 10 + digit;
19        i = i + 1;
20    }
21
22    if (str[0] == '-')
23        number = -number;
24
25    if (number > 32767 || number < -32768)
26        throw new Exception("Range_exceeded");
27
28    return number;
29 }
```

Figure 2: Function `convert`.

Abbildung 2: Funktion `convert`.

Consider the function `convert` shown in Figure 2.

- (i) Which lines (line numbers) contribute to the program size of `convert`?
- (ii) What is the net program size of `convert`?
Which lines *do not* contribute to the net program size of `convert`?
- (iii) What is the code size of `convert`?
Which lines *do not* contribute to the code size of `convert`?

Abbildung 2 zeigt die Funktion `convert`.

- (i) Welche Zeilen(nummern) werden für die Programmgröße von `convert` gezählt?
- (ii) Was ist die Netto-Programmgröße von `convert`?
Welche Zeilen(nummern) werden für die Netto-Programmgröße von `convert` *nicht* gezählt?
- (iii) Was ist die Codegröße von `convert`?
Welche Zeilen(nummern) werden für die Codegröße von `convert` *nicht* gezählt?

(5)

Correction:

- $\frac{2}{29}$ points per correct line (2.0)
- 1 point for correct answer, $\frac{1}{6}$ points per correct line (2.0)
- 0.5 point for correct answer, $\frac{1}{16}$ points per correct line (1.0)

3 Decision Tables

Entscheidungstabellen

T : registration		r_1	r_2	r_3	r_4
c_1	already registered	×	-	-	×
c_2	$n < n_{\max}$	*	×	-	*
c_3	delayed payments	*	-	-	×
a_1	message: “already registered”	×	-	-	-
a_2	message: “fully booked”	-	-	×	-
a_3	message: “payment reminder”	-	-	-	×
a_4	register applicant	-	×	-	-

Figure 3: Decision table.

Abbildung 3: Entscheidungstabelle.

Problem 3.1 (Analysing Decision Tables)

Aufgabe 3.1 (Analyse von Entscheidungstabellen)

Consider the decision table for a registration procedure as shown in Figure 3. In some cases, if applicants receive a message anyway, that message should be combined with a payment reminder if there are delayed payments. The maximum number of registrations is n_{\max} , the current number of registrations n .

Betrachten Sie die Entscheidungstabelle für eine Registrierungsprozedur in Abbildung 3. In einigen Fällen, in denen Antragsteller eh eine Nachricht bekommen, soll diese Nachricht mit einer Zahlungserinnerung kombiniert werden, falls Zahlungen ausstehen. Maximal können n_{\max} Registrierungen vorgenommen werden, n bezeichnet die aktuelle Anzahl an Registrierungen.

- (i) Is T as given in Figure 3 complete?

Prove your claim by giving a counter-example or a proof of completeness, respectively.

Ist T in Abbildung 3 vollständig?

Beweisen Sie Ihre Behauptung durch ein Gegenbeispiel bzw. einen Beweis der Vollständigkeit.

(2)

Correction:

- $\frac{2}{N}$ points for the correct answer, and for each of the N parts of the argumentation (example or proof)

- (ii) Is T as given in Figure 3 deterministic?

Prove your claim by giving a counter-example or a proof of completeness, respectively.

Ist T in Abbildung 3 deterministisch?

Beweisen Sie Ihre Behauptung durch ein Gegenbeispiel bzw. einen Beweis der Vollständigkeit.

(2)

Correction:

- $\frac{2}{N}$ points for the correct answer, and for each of the N parts of the argumentation (example or proof)

- (iii) The customer in addition stated that payment reminders should be combined with “other messages with a positive connotation”, i.e. being already registered or having been successfully registered, but not with other messages. Is this policy ensured by T in the collecting semantics? Explain!

Der Kunde hat zusätzlich geäußert, daß Zahlungserinnerungen mit “anderen Nachrichten mit positiver Konnotation”, also bereits registriert sein oder erfolgreich registriert worden sein, kombiniert werden sollen, aber nicht mit anderen Nachrichten. Wird dieser Grundsatz von T in der Sammelsemantik garantiert? Erläutern Sie!

(1)

- max. 1 point for correct answer and explanation

Problem 3.2 (Writing Decision Tables)

Aufgabe 3.2 (Entscheidungstabellen Erstellen)

Gathering requirements for the software of a new heating controller yielded the following statements by the customer:

- If the room temperature is below target temperature (condition c_1), shift to the next higher heating level (action a_1).
- If the room temperature is above the target temperature (condition c_2), shift to the next lower heating level (action a_2).
- When the “power heating” button is pushed (condition c_3), and the room temperature is not above the target temperature, shift to the next higher heating level.
- If room is neither above nor below the target temperature, and if “power heating” is not pressed, do not change heating level.

Eine erste Anforderungsanalyse für das Programm eines neuen Steuerungsgeräts einer Heizungsanlage hat die folgenden Aussagen des Kunden ergeben:

- Wenn die Raumtemperatur unterhalb der Zieltemperatur liegt (Bedingung c_1), schalte in die nächsthöhere Heizstufe (Aktion a_1).
- Wenn die Raumtemperatur oberhalb der Zieltemperatur liegt (Bedingung c_2), schalte in die nächstniedrigere Heizstufe (Aktion a_2).
- Wenn der “schnell aufheizen”-Knopf gedrückt wird (Bedingung c_3) und die Raumtemperatur nicht oberhalb der Zieltemperatur liegt, schalte in die nächsthöhere Heizstufe.
- Wenn die Raumtemperatur weder oberhalb noch unterhalb der Zieltemperatur liegt und der “schnell aufheizen”-Knopf nicht gedrückt ist, ändere die Heizstufe nicht.

- (i) Provide a decision table which formalises these requirements.

Formalisieren Sie die Anforderungen mit Hilfe einer Entscheidungstabelle.

(4)

Correction:

- one point per correct rule

- (ii) Analyse the requirements for completeness by checking the resulting decision table for completeness.

If the decision table is complete, provide a proof, if not, propose an environment assumption in the form of a conflict axiom (which is consistent with all existing rules in the table) such that the decision table is complete with respect to the conflict axiom.

Analysieren Sie die Anforderungen auf Vollständigkeit indem Sie die resultierende Entscheidungstabelle auf Vollständigkeit überprüfen.

Wenn die Entscheidungstabelle vollständig ist, geben Sie einen Beweis an; wenn nicht, schlagen Sie eine Umgebungsannahme in Form eines Konfliktaxioms vor, das mit allen in der Tabelle vorhandenen Regeln konsistent ist und so, daß die Entscheidungstabelle vollständig bezüglich des Konfliktaxioms ist.

(1)

Correction:

- 1.0 point for correct answer (including proof or conflict axiom)

4 Live Sequence Charts

Live Sequence Charts

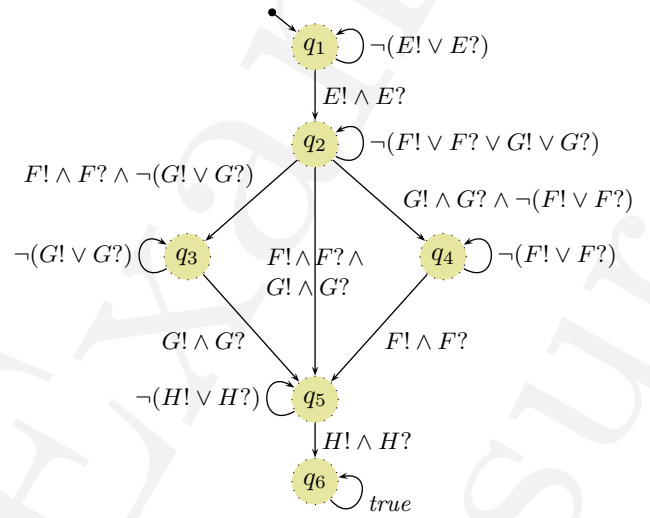
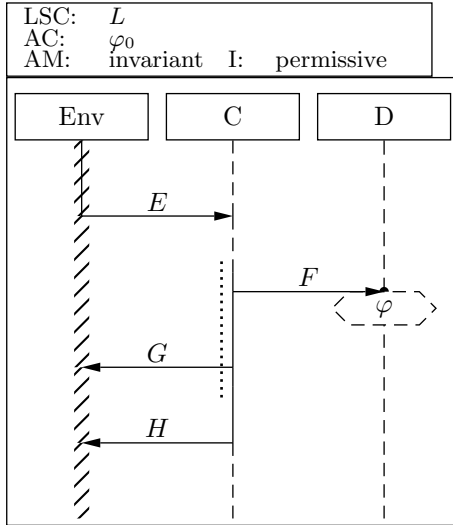


Figure 4: Live Sequence Chart and its TBA.
Live Sequence Chart und zugehöriger TBA.

Problem 4.1 (Locations and Cut Temperatures)

Aufgabe 4.1 (Locations und Cut Temperaturen)

- (i) What are the locations of the LSC L given in Figure 4 on page 6.
Clearly mark them in the LSC in Figure 4.

Geben Sie die Locations der in Abbildung 4 auf Seite 6 gegebenen LSC an.
Markieren Sie die Locations eindeutig in der LSC in Abbildung 4.

(3)

Correction:

- $\frac{3}{11}$ points per correct location

- (ii) According to the LSC semantics, which states of the TBA shown in Figure 4 need to be accepting and which not?
Mark the states in the TBA in Figure 4 with single or double outlines accordingly.

Welche Zustände des in Abbildung 4 gezeigten TBA sind entsprechend der LSC Semantik akzeptierend und welche nicht?

Markieren Sie die Zustände im TBA in Abbildung 4 entsprechend mit einfachem oder doppeltem Rand.

(2)

Correction:

- $\frac{1}{3}$ point for each correct mark

Problem 4.2 (Computation Paths)

Aufgabe 4.2 (Berechnungspfade)

- (i) Provide a computation path which trivially satisfies L .
- (ii) Provide a computation path which non-trivially satisfies L .
- (iii) Provide a computation path which violates L .

You may disregard the condition for this task.

- (i) Geben Sie einen Berechnungspfad an, der L trivial erfüllt.
- (ii) Geben Sie einen Berechnungspfad an, der L nicht-trivial erfüllt.
- (iii) Geben Sie einen Berechnungspfad an, der L verletzt.

Die Bedingung können Sie für diese Aufgabe ignorieren.

(3)

Correction:

- 1 point for each of (i), (ii), (iii)

Problem 4.3 (Cold Conditions)

Aufgabe 4.3 (Kalte Bedingungen)

- (i) The condition φ in LSC L has not been considered in the TBA shown in Figure 4. Complete the TBA such that the condition is properly represented.

Die Bedingung φ in LSC L wurde bei der Konstruktion des in Abbildung 4 gezeigten TBA nicht berücksichtigt. Vervollständigen Sie den TBA derart, daß die Bedingung korrekt repräsentiert wird.

(1)

Correction:

- $\frac{1}{7}$ point for each correct new transition and label (modification)

- (ii) Provide a computation path which satisfies L by taking a legal exit.

Geben Sie einen Berechnungspfad an, der L erfüllt, indem ein legaler Ausstieg genommen wird.

(1)

Correction:

- one point for correct path

5 Class Diagrams

Klassendiagramme



Figure 5: Class diagram.
Abbildung 5: Klassendiagramm.

Problem 5.1 (Signature)

Aufgabe 5.1 (Signatur)

Provide the (minimal) signature described by the class diagram in Figure 5.

Geben Sie die (kleinste) Signatur an, die vom Klassendiagramm in Abbildung 5 beschrieben wird.

(4)

Correction:

- 0.25 points for each correctly rewritten element
- max. -1 for notation errors

Problem 5.2 (Associations)

Aufgabe 5.2 (Assoziationen)

Assume we add $s : Segment_*$ to V and s to $atr(Train)$ to the signature of the class diagram shown in Figure 5. Extend the class diagram in Figure 5 such that this additional modelling element is properly *graphically* represented, i.e. do not just add s to the attributes compartment.

Nehmen Sie an, in der Signatur des in Abbildung 5 gezeigten Klassendiagramms wird $s : Segment_*$ zu V und s zu $atr(Train)$ hinzugefügt.

Erweitern Sie das Klassendiagramm in Abbildung 5 derart, daß dieses zusätzliche Modellelement korrekt *graphisch* repräsentiert wird, d.h. s soll nicht einfach dem Bereich für Attribute hinzugefügt werden.

(1)

Correction:

- 0.25 points each for: line between correct things, arrow at correct side, labels at correct sides

6 OCL

$$\sigma = \{u_1 \mapsto \{length \mapsto 27, n \mapsto u_2\}, u_2 \mapsto \{length \mapsto 13, n \mapsto \emptyset\}\}$$

Figure 6: System state.
Abbildung 6: Systemzustand.

Problem 6.1 (Objectdiagrams)

Aufgabe 6.1 (Objektdiagramme)

Provide an object diagram which completely represents the system state shown in Figure 6.

Geben Sie ein Objektdiagramm an, das den in Abbildung 6 gegebenen Systemzustand vollständig darstellt.

(3)

Correction:

- 0.5 points for each correctly represented piece of information
- max. -1 points for syntax mistakes

Problem 6.2 (Proto-OCL Proof)

Aufgabe 6.2 (Proto-OCL Beweis)

$$F := \forall self : Segment \bullet length(self) > length(n(self))$$

Figure 7: Proto-OCL formula.
Abbildung 7: Proto-OCL Formel.

It is claimed that the Proto-OCL formula F shown in Figure 7 evaluates to \perp in the system state from Figure 6, i.e. that $\mathcal{I}\llbracket F \rrbracket(\sigma, \emptyset) = \perp$.

Prove this claim using the semantics of Proto-OCL.

Es wird behauptet, daß die in Abbildung 7 gegebene Proto-OCL Formel F in dem durch das Objektdiagramm in Abbildung 6 beschriebenen System State zu \perp ausgewertet wird, d.h. daß $\mathcal{I}\llbracket F \rrbracket(\sigma, \emptyset) = \perp$.

Beweisen Sie diese Behauptung unter Verwendung der Proto-OCL Semantik.

(5)

- $\frac{1}{6}$ for each correct step using the Proto-OCL semantics

Problem 6.3 (Proto-OCL Examples)

Aufgabe 6.3 (Proto-OCL Beispiele)

Provide two system states σ_1 and σ_2 such that $\mathcal{I}[F](\sigma_1, \emptyset) = \text{true}$ and $\mathcal{I}[F](\sigma_2, \emptyset) = \text{false}$.

Geben Sie zwei System States σ_1 und σ_2 an so, daß $\mathcal{I}[F](\sigma_1, \emptyset) = \text{true}$ und $\mathcal{I}[F](\sigma_2, \emptyset) = \text{false}$ gilt.

(4)

Correction:

- σ_1 : (1 point, if correct and notation right)
- σ_2 : (3 points, if correct and notation right)
- can be given as object diagram (say: complete, otherwise possibly fewer points), or in system state notation

Problem 6.4 (Proto-OCL Formalisation)

Aufgabe 6.4 (Proto-OCL Formalisierung)

Formalise the following requirements using Proto-OCL formulae:

- The value of the basic type attribute of each object of class *Segment* should be strictly bigger than 10.
- The value of the *length* attribute of two different *Segment* instances to which a *Train* instance refers by an *s*-link differs by at most 1.

Formalisieren Sie die folgenden Anforderungen mit Hilfe von Proto-OCL Formeln:

- Der Wert des Basistyp-Attributs jedes Objektes der Klasse *Segment* soll streng größer als 10 sein.
- Die Werte des *length* Attributs zweier verschiedener *Segment*-Instanzen, auf die eine *Train*-Instanz mit einem *s*-Link verweist, unterscheiden sich um höchstens 1.

(3)

Correction:

- 0.5 points for each element marked in the reference solution
- 0.125 points for each element marked in the reference solution

7 Communicating Finite Automata

Communicating Finite Automata

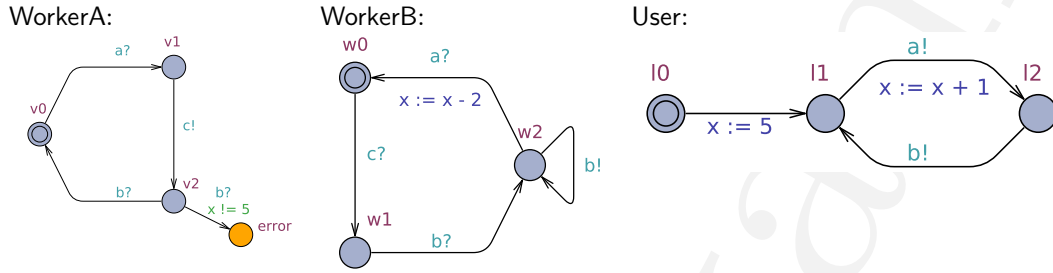


Figure 8: Network of Communicating Finite Automata (x is a global variable).
Abbildung 8: Netzwerk Kommunizierender Endlicher Automaten (x ist eine globale Variable).

Problem 7.1 (Computation Paths)

Aufgabe 7.1 (Berechnungspfade)

Consider the network \mathcal{N} of communicating finite automata given in Figure 8.

- (i) It is claimed that $\mathcal{N} \models E \langle \rangle \text{WorkerA.error}$ holds.

Prove this claim by providing an appropriate computation path.

Es wird behauptet, daß $\mathcal{N} \models E \langle \rangle \text{WorkerA.error}$ gilt.

Beweisen Sie diese Behauptung, indem Sie einen entsprechenden Berechnungspfad angeben.

(9)

Correction:

- 1 point for each correct configuration and transition (in shortest path)

- (ii) It is claimed that $\mathcal{N} \models A[] \text{WorkerA.error} \vee \text{not deadlock}$ does not hold.

Prove this claim by providing an appropriate computation path.

Es wird behauptet, daß $\mathcal{N} \models A[] \text{WorkerA.error} \vee \text{not deadlock}$ nicht gilt.

Beweisen Sie diese Behauptung, indem Sie einen entsprechenden Berechnungspfad angeben.

(3)

Correction:

- $\frac{3}{13}$ points for each correct configuration and transition (in shortest path)

- (iii) It is claimed that $\mathcal{N} \models E[] \text{not WorkerA.error}$ holds for an infinite computation path.

Prove this claim by providing an appropriate computation path.

Es wird behauptet, daß $\mathcal{N} \models E[] \text{not WorkerA.error}$ für einen unendlichen Berechnungspfad gilt.

Beweisen Sie diese Behauptung, indem Sie einen entsprechenden Berechnungspfad angeben.

(3)

Correction:

- $\frac{2}{13}$ points for each correct configuration and transition (in shortest path)
- 1 point for indicating the loop

8 Formal Verification

Formale Verifikation

Problem 8.1 (Check PD Calculus Proof)

Aufgabe 8.1 (PD Kalkül-Beweis prüfen)

Consider the proof sketch for function **SOMESUM** in Figure 10 on page 13.

Betrachten Sie die Beweisskizze für die Funktion **SOMESUM** in Abbildung 10 auf Seite 13.

- (i) Complete the proof sketch by filling in the missing three formulae and label the marked steps by the axioms and rules uses for the step.
(The first step is already marked with axiom (A2) as an example.)

Vervollständigen Sie die Beweisskizze indem Sie die fehlenden drei Formeln einfügen und beschriften Sie die markierten Schritte mit dem Axiom bzw. der Formel, die für diesen Schritt verwendet wird.

(Der erste markierte Schritt ist als Beispiel bereits mit Axiom (A2) beschriftet.)

(6)

- (ii) Prove the step marked with (*).

You may use the propositions (1) and (2) given below.

Beweisen Sie den mit (*) markierten Schritt.

Sie können hierzu die unten angegebenen Beziehungen (1) und (2) verwenden.

(1)

$$(1) \sum_{k=1}^n k = \frac{1}{2}(n \cdot (n + 1))$$

$$(2) \sum_{k=1}^n k^2 = \frac{1}{6}n \cdot (n + 1) \cdot (2 \cdot n + 1)$$

Correction:

- $\frac{3}{4}$ point for each correct answer; maybe 0.5 for the missing formulae if they are half-right
- correct proof of (*): 1 point

Problem 8.2 (Prove with PD)

Aufgabe 8.2 (Beweisen mit PD)

$\{true\} \ y := x; y = ((x - 1) \cdot x + y) + z; \text{while } z \text{ do skip } \{y = x^2\}$

Figure 9: A simple program.

Abbildung 9: Ein einfaches Programm.

Prove that the Hoare triple given in Figure 9 holds for partial correctness using the PD calculus.

Beweisen Sie mit Hilfe des PD-Kalküls, daß das in Abbildung 9 angegebene Hoare-Triple im Sinne von partieller Korrektheit gilt.

(3)

Correction:

- $\frac{1}{3}$ points for each rule application and each proof necessary for (R6)

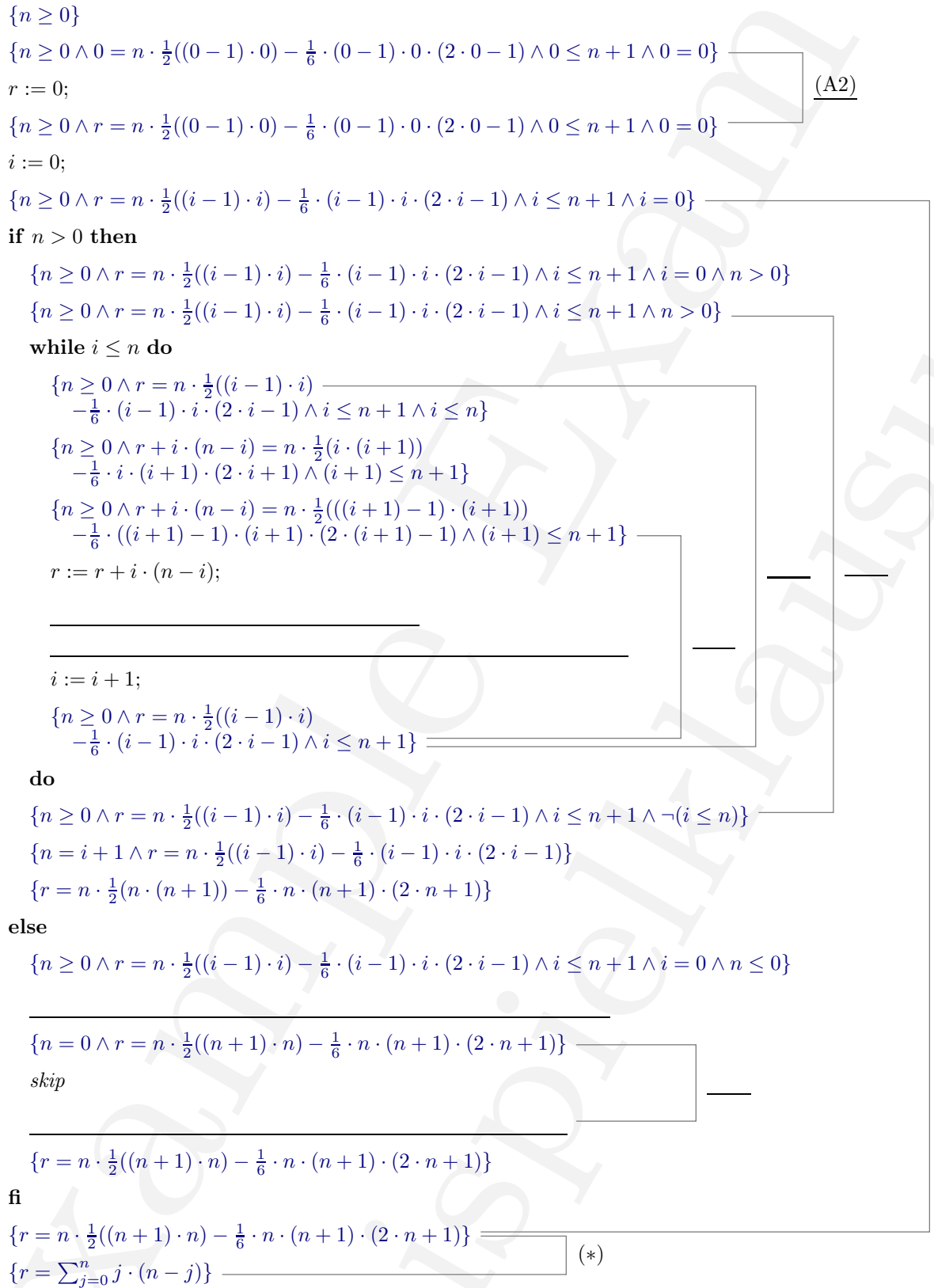


Figure 10: Incomplete proof sketch for function SOMESUM.
 Abbildung 10: Unvollständige Beweisskizze für Funktion SOMESUM.

9 Software Testing

Software Tests

Problem 9.1 (Statement Coverage)

Aufgabe 9.1 (Anweisungsüberdeckung)

Provide a test suite which achieves 100 % statement coverage for function `maxminavg` shown in Figure 11 on page 16. (The requirements on `maxminavg` are given in the caption of Figure 11.) For each test case, point out which statement it covers.

Geben Sie eine Menge von Testfällen an, die für die Funktion `maxminavg` aus Abbildung 11 auf Seite 16 eine Anweisungsüberdeckung von 100 % erreicht. (Die Anforderungen an `maxminavg` sind in der Bildunterschrift von Abbildung 11 gegeben.) Geben Sie für jeden Testfall an, welche Anweisungen dieser überdeckt.

(4)

Correction:

- $\frac{2}{3}$ points for covering each basic block (4.0)
- one point less if no soll-values!

Problem 9.2 (Branch Coverage)

Aufgabe 9.2 (Verzweigungsüberdeckung)

Provide a test suite which achieves the maximal possible branch coverage for function `maxminavg` shown in Figure 11. For each test case, point out which branches it covers.

What is the maximum value for branch coverage?

Geben Sie eine Menge von Testfällen an, die für die Funktion `maxminavg` aus Abbildung 11 die maximale Verzweigungsüberdeckung erreicht. Geben Sie für jeden Testfall an, welche Verzweigungen dieser überdeckt.

Wie groß ist die maximale Verzweigungsüberdeckung?

(4)

Correction:

- 0.75 points for covering each pair of positive/negative branch
- one point for correct answer

Problem 9.3 (Term Coverage)

Aufgabe 9.3 (Termüberdeckung)

Provide a test suite which achieves the maximal possible term coverage for the condition of the if-statement in line 24 of Figure 11. For each test case, point out how it contributes to term coverage.

What is the maximum value for term coverage here? Why?

Geben Sie eine Menge von Testfällen an, die für die Bedingung der if-Anweisung in Zeile 24 von Abbildung 11 die maximale Termüberdeckung erreicht. Geben Sie für jeden Testfall an, wie dieser zur Termüberdeckung beiträgt.

Wie groß ist die maximale Termüberdeckung hier? Warum?

(4)

Correction:

- 1 point for each covered case
- 2 point for correct explanation (0.5 points for each individual reason)

Problem 9.4 (Undetected Errors)

Aufgabe 9.4 (Unerkannte Fehler)

Provide

- a test suite which achieves 100% statement coverage for function `maxminavg` shown in Figure 11 on page 16, and
- a modification of `maxminavg`

such that the modification does not satisfy the requirements on `maxminavg` and all test cases are executed unsuccessfully.

Explain your solution!

Geben Sie

- eine Menge von Testfällen, die für die Funktion `maxminavg` aus Abbildung 11 auf Seite 16 eine Anweisungsüberdeckung von 100% erreicht und
- eine Modifikation von `maxminavg` an

so, daß die Modifikation die Anforderungen an `maxminavg` nicht erfüllt und alle Testfälle erfolglos ausgeführt werden.

Erklären Sie Ihre Lösung!

(2)

Correction:

- one point for correct modification
- one point for explanation (why are requirements no longer met, why not detected by test suite?)

Problem 9.5 (Path Coverage)

Aufgabe 9.5 (Pfadüberdeckung)

There is a notion of *path coverage* where 100% is achieved if each path from an entry to an exit point is covered, including skipping loops and executing the loop body at least once.

Do you see a relation between this notion of path coverage and the McCabe complexity metrics? Explain.

Es gibt einen Begriff von *Pfadüberdeckung* bei dem für 100% Überdeckung jeder Pfad von einem Eingangs- zu einem Ausgangspunkt überdeckt wird. Insbesondere Schleifenkörper sollen dabei einmal übersprungen und mindestens ein- oder mehrmals durchlaufen werden.

Sehen Sie einen Zusammenhang zwischen diesem Begriff von Pfadüberdeckung und der Komplexitätsmetrik von McCabe? Erklären Sie.

(1)

Correction:

- one point for pointing out the existing relation

```

1 public static void maxminavg( int[] values ) {
2
3     max = Double.MIN_VALUE;
4     min = Double.MAX_VALUE;
5
6     int    count = 0;
7     double total = 0;
8
9     for (int i = 0; i < values.length; ++i) {
10
11         double v = values[i];
12
13         if (v > max) {
14             max = v;
15         }
16         if (v < min) {
17             min = v;
18         }
19
20         count++;
21         total += v;
22     }
23
24     if (count == 0 || count < 0)
25         avg = -27;
26     else
27         avg = total / count;
28 }

```

Figure 11: Function `maxminavg`. For input $a[0], \dots, a[n]$, $n \geq 0$, with $\sum_{i=0}^n a[i] \leq \text{Double.MAX_VALUE}$, `maxminavg` should store the minimum value in the global variable `min`, the maximum number in `max`, and the average (up to the precision of Java's `double`) in `avg`. In case the input array is empty, `max` and `min` should get the values `Double.MIN_VALUE` and `Double.MAX_VALUE`, respectively, and `avg` the value `-27`.

Abbildung 11: Funktion `maxminavg`. Für Eingabe $a[0], \dots, a[n]$, $n \geq 0$, mit $\sum_{i=0}^n a[i] \leq \text{Double.MAX_VALUE}$, soll `maxminavg` das Minimum der Werte in der globalen Variablen `min`, das Maximum in `max`, und das arithmetische Mittel (in der Präzision von Java's `double`) in `avg` speichern. Wenn die Eingabe das leere Feld ist, sollen `max` und `min` die Werte `Double.MIN_VALUE` bzw. `Double.MAX_VALUE` erhalten, und in `avg` die `-27` gespeichert werden.

Softwaretechnik / Software Engineering

<http://swt.informatik.uni-freiburg.de/teaching/SS2015/swtv1>

Example Exam

Beispielklausur

Name:

Matriculation number:

Immatrikulationsnummer:

- **Date:** next greek calend

Datum: St. Nimmerlein

- **Time:** 11:55

Zeit: 11:55

- **Room:** 0-00-000

Raum: 0-00-000

- **Duration:** 90 minutes

Dauer: 90 Minuten

- **Language:** English and German

Sprache: Deutsch und Englisch

- Use an indelible ink pen
(like ball pen, not: pencil).

Verwenden Sie einen dokumentenechten Stift
(z.B. Kugelschreiber, nicht: Bleistift).

- Please fill out **name and matriculation number**.

Bitte ausfüllen: **Name und Immatrikulationsnummer**.

- You may want to make yourself familiar with the **appendix**.

Problem Aufgabe	Max. Points Punkte	Achieved Points erreichte Punkte
1.1	3	
1.2	2	
2.1	5	
3.1	5	
3.2	5	
4.1	5	
4.2	3	
4.3	2	
5.1	4	
5.2	1	
6.1	3	
6.2	5	
6.3	4	
6.4	3	
7.1	15	
8.1	7	
8.2	3	
9.1	4	
9.2	4	
9.3	4	
9.4	2	
9.5	1	
Σ	90	

Grade:

Note:

Date of the review of the exam:

Datum der Korrektur der Klausur:

Appendix

A Proto-OCL Semantics

Proto-OCL Semantik

- $\mathcal{I}[c](\sigma, \beta) = \beta(c)$,
- $\mathcal{I}[v(F)](\sigma, \beta) = \sigma(\mathcal{I}[F](\sigma, \beta)) (v)$ if $\mathcal{I}[F](\sigma, \beta) \neq \perp$, and \perp otherwise,
 $\mathcal{I}[v(F)](\sigma, \beta) = \sigma(\mathcal{I}[F](\sigma, \beta)) (v)$ falls $\mathcal{I}[F](\sigma, \beta) \neq \perp$, und \perp sonst,
- $\mathcal{I}[f(F_1, \dots, F_n)](\sigma, \beta) = f_{\mathcal{I}}(\mathcal{I}[F_1](\sigma, \beta), \dots, \mathcal{I}[F_n](\sigma, \beta))$,
- $\mathcal{I}[\forall c : C \bullet F](\sigma, \beta) = \begin{cases} true & , \text{ if } \mathcal{I}[F](\sigma, \beta[c := u]) = true \text{ for all } u \in \text{dom}(\sigma) \cap \mathcal{D}(C) \\ false & , \text{ if } \mathcal{I}[F](\sigma, \beta[c := u]) = false \text{ for some } u \in \text{dom}(\sigma) \cap \mathcal{D}(C) \\ \perp & , \text{ otherwise} \end{cases}$
- $\mathcal{I}[\forall c : C \bullet F](\sigma, \beta) = \begin{cases} true & , \text{ falls } \mathcal{I}[F](\sigma, \beta[c := u]) = true \text{ für alle } u \in \text{dom}(\sigma) \cap \mathcal{D}(C) \\ false & , \text{ falls } \mathcal{I}[F](\sigma, \beta[c := u]) = false \text{ für ein } u \in \text{dom}(\sigma) \cap \mathcal{D}(C) \\ \perp & , \text{ sonst} \end{cases}$
- $\sigma_{\mathcal{I}}(u)(v) = \begin{cases} \perp & , \text{ if } u \notin \text{dom}(\sigma) \text{ or } v \notin \text{dom}(\sigma(u)) \\ u' & , \text{ if } \sigma(u)(v) = \{u'\} \text{ and } v : C_{0,1} \text{ for some } C \\ \perp & , \text{ if } \sigma(u)(v) = \emptyset \text{ and } v : C_{0,1} \text{ for some } C \\ \sigma(u)(v) & , \text{ otherwise} \end{cases}$
- $\sigma_{\mathcal{I}}(u)(v) = \begin{cases} \perp & , \text{ falls } u \notin \text{dom}(\sigma) \text{ oder } v \notin \text{dom}(\sigma(u)) \\ u' & , \text{ falls } \sigma(u)(v) = \{u'\} \text{ und } v : C_{0,1} \text{ für ein } C \\ \perp & , \text{ falls } \sigma(u)(v) = \emptyset \text{ und } v : C_{0,1} \text{ für ein } C \\ \sigma(u)(v) & , \text{ sonst} \end{cases}$
- Example for semantics of a function symbol (here: \in):
 Beispiel für die Semantik von Funktionssymbolen (hier: \in):
 $\in_{\mathcal{I}}(\cdot, \cdot) : \mathcal{D}(\mathcal{C}) \dot{\cup} \{\perp\} \times 2^{\mathcal{D}(\mathcal{C})} \dot{\cup} \{\perp\} \rightarrow \mathbb{B} \dot{\cup} \{\perp\}$

$$\in_{\mathcal{I}}(u, U) = \begin{cases} u \in U & , \text{ if } u \neq \perp \text{ and } U \neq \perp \quad (\text{falls } u \neq \perp \text{ und } U \neq \perp) \\ \perp & , \text{ otherwise} \quad (\text{sonst}) \end{cases}$$

B Calculus PD

Kalül PD

$$\{p\} \text{ skip } \{p\} \tag{A1}$$

$$\{p[u := t]\} u := t \{p\} \tag{A2}$$

$$\frac{\{p\} S_1 \{r\}, \{r\} S_2 \{q\}}{\{p\} S_1; S_2 \{q\}} \tag{R3}$$

$$\frac{\{p \wedge B\} S_1 \{q\}, \{p \wedge \neg B\} S_2 \{q\}}{\{p\} \text{ if } B \text{ then } S_1 \text{ else } S_2 \text{ fi } \{q\}} \tag{R4}$$

$$\frac{\{p \wedge B\} S \{p\}}{\{p\} \text{ while } B \text{ do } S \text{ do } \{p \wedge \neg B\}} \tag{R5}$$

$$\frac{p \rightarrow p_1, \{p_1\} S \{q_1\}, q_1 \rightarrow q}{\{p\} S \{q\}} \tag{R6}$$