Errata

Rigorous Software Development: an introduction to program verification

September, 2011

Chapter 3

pag. 53 – Proposition 3.24 (5).

Reads: 5. $\Gamma \models A \vee B$ iff $\Gamma \models A$ or $\Gamma \models B$

Should read: 5. $\Gamma \models A \vee B$ if $\Gamma \models A$ or $\Gamma \models B$

pag. 53 – Proposition 3.24: Additional clause in proposition.

Should read: 5'. $\Gamma \models C$ if $\Gamma \models A \vee B$ and $\Gamma, A \models C$ and $\Gamma, B \models C$

pag. 59 – proof of Proposition 3.24: extraneous prime in conclusion of rule (W) (displayed equation).

Reads: $\Gamma', B \vdash C$

Should read: $\Gamma, B \vdash C$

pag. 62 – proof of Theorem 3.30: case of "rule ($E_\vee$)".

Reads: From induction hypothesis $\Gamma \models B \vee C$, by Proposition 3.24 (5) we have either $\Gamma \models B$ or $\Gamma \models C$. Reasoning by cases with Proposition 3.17 (3) and the remaining hypotheses $\Gamma, B \models A$ and $\Gamma, C \models A$, we obtain $\Gamma \models A$.

Should read: From induction hypothesis $\Gamma \models B \vee C$, $\Gamma, B \models A$ and $\Gamma, C \models A$. By Proposition 3.24 (5'), we obtain $\Gamma \models A$.

pag. 63 – proof of Theorem 3.31:

Reads: If $\Gamma \models A$ then, by Proposition 3.18 (2), $\Gamma, \neg A \models \bot$.

Should read: If $\Gamma \models A$ then, by Proposition 3.18 (1), $\Gamma, \neg A \models \bot$.

pag. 69 – beginning of the last paragraph of Section 3.5.2: missing letter "A".

Reads: s an example

Should read: As an example

pag. 76 – Exercise 3.8:

Reads: $A \to B \wedge B \to A$

Should read: $(A \to B) \wedge (B \to A)$

pag. 77 – Exercise 3.15 (e):

Reads: (e) $\Gamma \models A \vee B$ iff $\Gamma \models A$ or $\Gamma \models B$

Should read: (e) $\Gamma \models A \vee B$ if $\Gamma \models A$ or $\Gamma \models B$

pag. 77 – Exercise 3.15: additional clause in exercise:

Should read: (f) $\Gamma \models C$ if $\Gamma \models A \vee B$ and $\Gamma, A \models C$ and $\Gamma, B \models C$

pag. 79 – titles of references [4] and [5] from bibliography of chap. 3: word "sat" should be typeset as "SAT"

Chapter 4

pag. 90 – Definition 4.20 (4): word "refutable" should be typeset in italic.

pag. 90 – Definition 4.20 (5): words "satisfy Γ with α" should be typeset in italic.

pag. 95 – Proposition 4.30 (4):

Reads: 4. $\Gamma \models \phi \vee \psi$ iff $\Gamma \models \phi$ or $\Gamma \models \psi$

Should read: 4. $\Gamma \models \phi \vee \psi$ if $\Gamma \models \phi$ or $\Gamma \models \psi$

pag. 95 – Proposition 4.30: Additional clause in proposition.

Should read: 4'. $\Gamma \models \theta$ if $\Gamma \models \phi \vee \psi$ and $\Gamma, \phi \models \theta$ and $\Gamma, \psi \models \theta$

pag. 104 – beginning of section 4.5.2: addition of remark restricting validity/satisfiability to sentences.

Reads: Let us now focus on what is perhaps the most distinctive feature of first-order logic: quantifiers. We have already seen that quantifiers in first-order formulas can always be moved to the outermost position in prenex normal forms (Proposition 4.48). The following constructions allow us to further restrict the roles of these quantifiers in first-order formulas.

Should read: Let us now focus on what is perhaps the most distinctive feature of first-order logic: quantifiers. In the remainder of this section we will restrict ourselves to the problem of verifying validity or satisfiability of sentences (i.e. formulas without free variables). Notice that, from a semantic point of view, free variables in formulas behave as if they were universally quantified. Concretely, a formula ϕ is valid (reps. satisfiable) if and only if its universal closure is valid (reps. satisfiable) – see Exercise 4.6. Hence, to check validity/satisfiability of a formula with free variables, we will consider instead the corresponding problem in its universal closure (which is a sentence). Working with sentences has the additional advantage that, as in propositional logic, validity and satisfiability are interchangeable by negation: a sentence ϕ is valid iff $\neg\phi$ is not satisfiable (we leave as an exercise to the reader to check that, in the presence of free variables, this is not necessarily the case).

We have already seen that quantifiers in first-order formulas can always be moved to the outermost position in prenex normal forms (Proposition 4.48). The following constructions allow us to further restrict the roles of these quantifiers in first-order formulas.

pag. 110 – penultimate line of Section 4.6.1: extraneous word "write" in sentence.

Reads: the formula write $\exists x_1, x_2.\neg(x_1 = x_2)$

Should read: the formula $\exists x_1, x_2.\neg(x_1 = x_2)$

pag. 117 – footnote 3:

Reads: Note that in fact reflexivity is redundant as it follows from symmetry and transitivity

Should Read: Actually this set of axioms is not minimal, since symmetry and transitivity follows from reflexivity and congruence axioms

pag. 126 – Exercise 6: last sentence of (b)

Reads: the set $\Psi(\rho)$ is satisfiable

Should read: the set $\Phi(\rho)$ is satisfiable

pag. 126 – Exercise 4.12: second line of (c)

Reads: $\Gamma \models \phi \vee \psi$ iff $\Gamma \models \phi$ or $\Gamma \models \psi$

Should read: $\Gamma \models \phi \vee \psi$ if $\Gamma \models \phi$ or $\Gamma \models \psi$

pag. 126 – Proposition 4.12 (c): Additional clause in exercise.

Should read: $\Gamma \models \theta$ if $\Gamma \models \phi \vee \psi$ and $\Gamma, \phi \models \theta$ and $\Gamma, \psi \models \theta$

pag. 127 – Exercise 4.15: formula appearing in (f)

Reads: (f) $\vdash \neg((\exists x\,.\,P(x)) \to \forall x\,.\,P(x))$

Should read: (f) $\vdash \neg(\exists x\,.\,\neg P(x)) \to \forall x\,.\,P(x)$

pag. 127 – title of reference 7 from bibliography of chap. 4: word "smt" should be typeset as "SMT"

Chapter 5

pag. 133 – Figure 5.2 (7):

Reads: $(\textbf{while}\, b\, \textbf{do}\, C, s) \rightsquigarrow_{\mathcal{M}} s$

Should read: $(\textbf{while}\ b\ \textbf{do}\, \{\theta\}\, C, s) \rightsquigarrow_{\mathcal{M}} s.$

pag. 151 – in the conclusion of while rule for total correctness.

Reads: $[\theta]\, \textbf{while}\ V\ \textbf{do}\, \{\theta, V\}\, b\, [\theta \wedge \neg b]$

Should read: $[\theta]\, \textbf{while}\ b\ \textbf{do}\, \{\theta\}\, C\, [\theta \wedge \neg b]$

Chapter 10

pag. 249 – third line: symbol "@" is missing from the beginning of line.

pag. 255 – title of reference 3 from bibliography of chap. 10: words "coq" and "why" should be typeset as"Coq" and "Why" respectively.