Errata "Principles of Model Checking" (July 2013)

Thanks to the Model Checking Reading Club at the Radboud University of Nijmegen, The Netherlands (in particular David N. Jansen and Frits W. Vaandrager), Holger Hermanns and Moritz Hahn (Saarland University), Dave Parker (Oxford University, UK), Jeroen Ketema and Mark Timmer (University of Twente, NL), Stephen Siegel (University of Delaware, USA), René Thiemann (U. Innsbruck, AT), Martin Lange (University of Kassel and LMU, D), Ahmed Khademzadeh (Azad University of Mashhad, Iran), Moti Ben-Ari (Weizmann Institute, ISR), Verena Wolf (Saarland University), Prakash Panangaden (McGill, Canada), Andrzej Zbrzezny (Jan Dlugosz University), Tobias Schmit-Lechner (U. Freiburg), Stefan Leue (U. Konstanz), Johann Schuster (Uni Bundeswehr, Munich), Chris J. Myers (University of Utah, USA), Erika Abráhám, Christian Dehnert, Alexander Nyßen, Daniel Weber and Martin Zimmermann (RWTH Aachen University), Michael Ummels (ENS Cachan, France), Erik Wognsen (Aalborg University, Denmark) and the students at the RWTH Aachen University attending the "(Advanced) Model Checking" course.

Comments are provided as:

$\langle$ page number $\rangle$ $\langle$ line number $\rangle$ $\langle$ short quote of the wrong word(s) $\rangle$ $\triangleright$ $\langle$ correction $\rangle$

Line number $-n$ means line number n from the last line on that page.

Chapter 1: System Verification

pp. 1, l. -5, *Pentium II* $\triangleright$ Pentium

pp. 5, l. 9, *lines of code lines* $\triangleright$ lines of code

pp. 5, l. footnote, *much higher* $\triangleright$ as the number of lines of code in the "golden" version of Windows95 is about 15 million, the error rate is in fact lower than normal.

pp. 6, l. 4, *Pentium II* $\triangleright$ Pentium

Chapter 2: Modeling Concurrent Systems

pp. 20, l. 8, *arabic* $\triangleright$ Latin pp. 25, l. 11, *heading Example 2.8* $\triangleright$ Execution fragments of the Beverage Vending Machine

pp. 27, l. -15, *function* λ_y $\triangleright$ The function λ_y has no impact on the transitions (as suggested), but only affects the state labeling.

pp. 31, l. Fig. 2.3, *beer, soda* $\triangleright$ *bget* and *sget*, respectively

pp. 31, l. Fig. 2.3, *state with 1 beer, 2 soda* $\triangleright$ the grey circle should be a white circle.

pp. 34, l. 2, $\langle \ell, v \rangle$ $\triangleright$ $\langle \ell, \eta \rangle$

pp. 40, l. Def. 2.21, $\mathit{Effect}(\alpha, \eta) = \mathit{Effect}_i(\alpha, \eta)$ $\triangleright$ $\mathit{Effect}(\alpha, \eta)(v) = \begin{cases} \mathit{Effect}_i(\alpha, \eta|_{\mathit{Var}_i})(v) & \text{if } v \in \mathit{Var}_i \\ \eta(v) & \text{otherwise} \end{cases}$

pp. 42, l. -10, *interlock* $\triangleright$ interleave

pp. 46, l. Fig. 2.9, *locations in* PG_2 $\triangleright$ should be subscripted with 2 (rather than 1)

pp. 48, l. -1, $H = \mathit{Act}_1 \cap \mathit{Act}_2$ $\triangleright$ $H = (\mathit{Act}_1 \cap \mathit{Act}_2) \setminus \{ \tau \}$

pp. 51, l. Fig. 2.12, $T_1 \parallel T_2$ $\triangleright$ $TS_1 \parallel TS_2$ (this occurs twice)

pp. 51, l. Fig. 2.12, $\triangleright$ All downgoing transitions should be labeled with *request*, and all upgoing ones with *release*

pp. 51, l. -7, *all trains* $\triangleright$ the train

pp. 52, l. 3, *(above)* $\triangleright$ (page 54)

pp. 53, l. -1, *finite set of channels* $\triangleright$ set of channels

pp. 54, l. Fig. 2.16, *the transition labeled approach emanating from state* $\langle \mathit{far}, 3, \mathit{down} \rangle$ $\triangleright$ should be removed, and all the states that thus become unreachable

pp. 54, l. Fig. 2.16, *the transition labeled exit emanating from state* $\langle \mathit{in}, 1, \mathit{up} \rangle$ $\triangleright$ should be removed, and all the states that thus become unreachable

pp. 55, l. -10, $(\mathit{Cond}(\mathit{Var}) \times$ $\triangleright$ $\mathit{Cond}(\mathit{Var}) \times$

pp. 60, l. Definition 2.33, $\uplus$ *(occurs four times)* $\triangleright$ $\cup$

pp. 62, l. -3, *gen_msg(1)* $\triangleright$ *snd_msg(1)*

pp. 62, l. 4, *ack* $\triangleright$ message

pp. 65, l. Fig. 2.21, *second* **do** $\triangleright$ **od**

pp. 66, l. 8, *Staements build* $\triangleright$ Statements built

pp. 71, l. 15, *label in conclusion of inference rule* $c!e$ $\triangleright$ it is meant that the value of expression e is transferred; cf. Exercise 2.8, pp. 85

pp. 74, l. 1, $\xi[c := v_2 \ldots v_k]$ $\triangleright$ $\xi' = \xi[c := v_2 \ldots v_k]$

pp. 74, l. 1, $\xi[c := v_1 \ldots v_k v]$ $\triangleright$ $\xi' = \xi[c := v_1 \ldots v_k v]$

pp. 76, l. Figure 2.23 (top), $x \rhd\ x'$

pp. 78, l. 8, *800,000*$\cdot 2^{50} \rhd$ 8,000,000$\cdot 2^{50}$

pp. 79, l. -6,-8, $|dom(c)|^{cp(c)} \rhd\ |dom(c)|^{cap(c)}$

pp. 82, l. Exercise 2.2, line 2, $P_i is \rhd\ P_i$ is

Chapter 3: Linear-Time Properties

pp. 89, l. 9, *parallel systems* $\rhd$ reactive systems

pp. 90, l. 1, *Fault Designed Traffic Lights* $\rhd$ Faulty Traffic Lights

pp. 91, l. 7, *a deadlock occurs when all philosophers* $\rhd$ a deadlock may occur when all philosophers

pp. 92, l. Fig. 3.2, *request and release* $\rhd$ req and rel

pp. 92, l. 6, $request_4 \rhd\ req_{4,4}$; similar to the other request actions

pp. 93, l. -4,-5 and Fig. 3.3, Fig. 3.4, *state* $available_i \rhd\ available_{i,i}$

pp. 93, l. -4,-5 and Fig. 3.3, Fig. 3.4, *state* $available_{i+1} \rhd\ available_{i,i+1}$

pp. 93, l. 10, *The corresponding is* $\rhd$ The corresponding condition is

pp. 94, l. Fig. 3.4, *falls* $x_i \rhd\ x_i$

pp. 96, l. 3, *finite paths* $\rhd$ finite path fragments

pp. 96, l. 4, *infinite path* $\rhd$ infinite path fragment

pp. 98, l. 1, *Trace and Trace Fragment* $\rhd$ Trace

pp. 100, l. 9, *(over AP)* $\rhd$ (over 2^{AP})

pp. 101, l. -3, $red_1\ green_2 \rhd\ red_1,\ green_2$

pp. 103, l. 11, $lwait_i \rhd\ wait_i$

pp. 103, l. 11, $\exists k \geqslant j.\ wait_i \in A_k \rhd\ \exists k > j.\ crit_i \in A_k$

pp. 105, l. Example 3.16, line 4, $\langle \text{wait}_1, \text{wait}_2, y = 1\rangle \to \langle \text{wait}_1, \text{crit}_2, y = 0\rangle \rhd \langle w_1, w_2, y = 1\rangle \to \langle w_1, c_2, y = 0\rangle$

pp. 111, l. Theorem 3.21, $M = \sum_{s \in S} |Post(s)| \rhd\ M = \sum_{s \in Reach(TS)} |Post(s)|$

pp. 111, l. 22, *The time needed to check* $s \models \Phi$ *is linear in the length of* $\Phi \rhd$ Add: This implicitly assumes that $a \in L(s)$ can be checked in $\mathcal{O}(1)$ time.

pp. 112, l. -2, $\rhd$ A minimal bad prefix is one such that the first occurrence of Φ is the last symbol in the word.

pp. 113, l. Figure 3.9, $s_0 \xrightarrow{yellow} s_1$ ▷ $s_0 \xrightarrow{yellow \wedge \neg red} s_1$

pp. 115, l. Lemma 3.27, *Proof* ▷ add the following sentence to the beginning of the proof: First note that for $P = (2^{AP})^\omega$ the claim trivially holds, since $closure(P) = P$ and the fact that P is a safety property since $\overline{P}$ is empty. In the remainder of the proof we consider $P \neq (2^{AP})^\omega$.

pp. 117, l. -3, *for any transition system TS.* ▷ for any transition system TS without terminal states.

pp. 118, l. 10,11, $\pi^{m_0}\pi^{m_1}\pi^{m_2}\ldots$ *of* $\pi^0\pi^1\pi^2\ldots$ *such that* ▷ $\pi^{m_0}, \pi^{m_1}, \pi^{m_2}, \ldots$ of $\pi^0, \pi^1, \pi^2, \ldots$ such that

pp. 124, l. -3, *By definition* ▷ By Lemma 3.27

pp. 130, l. 3, *without being taken beyond* ▷ without being taken infinitely often beyond

pp. 131, l. 17, *assignment* $x = -1$ ▷ assignment $x := -1$

pp. 132, l. 2, *an execution fragment ... but not strongly A-fair.* ▷ an execution fragment that visits infinitely many states in which no A-action is enabled is weakly A-fair (as the premise of weak A-fairness does not hold) but may not be strongly A-fair.

pp. 134, l. 10, *any finite trace is fair by default* ▷ any finite trace is strongly or weakly fair by default

pp. 136, l. -5, *strong fairness property* ▷ fairness property

pp. 138, l. 4, *It forces synchronization actions to happen infinitely often.* ▷ It forces synchronization actions to happen infinitely often provided they are enabled infinitely often.

pp. 138, l. 9, *(3.1) does not permit this.* ▷ (3.1) does not permit this (except if $Syn_{i,j}$ is a singleton set).

pp. 138, l. -14, *This requires that ... is enabled.* ▷ This requires that infinitely often a synchronization takes place when such synchronization is infinitely often enabled.

pp. 138, l. -2, *Weak fairness is appropriate for the internal actions* $\alpha \in Act_i \setminus Syn_i$*, as the ability to perform an internal action is preserved until it will be executed.* ▷ should be deleted

pp. 141, l. 5, *the set of properties that has* ▷ the property that has

pp. 145, l. Exercise 3.5(g), *between zero and two* ▷ between zero and non-zero

Chapter 4: Regular Properties

pp. 152, l. 2, *an alphabet* ▷ a finite alphabet pp. 157, l. -11, $w = A_1 \ldots A_n \in \Sigma \triangleright$ $w = A_1 \ldots A_n \in \Sigma^*$

pp. 157, l. -10, *starts in* Q_0 $\triangleright$ starts in state Q_0

pp. 157, l. -4, $Q_0 \triangleright$ $\{ Q_0 \}$

pp. 158, l. -14, *NFAs can be much more efficient.* $\triangleright$ NFAs can be much smaller.

pp. 161, l. -9, *(2) ... for all* $1 \leqslant i < n$ $\triangleright$ ... for all $0 \leqslant i < n$. (Note: the invariant false has minimal bad prefix ε.)

pp. 161, l. -8, $1 \leqslant i < n \triangleright$ $0 \leqslant i < n$

pp. 163, l. Example 4.15, *Minimal bad prefixes for this safety property constitute the language* $\{ pay^n\, drink^{n+1} \mid n \geqslant 0 \}$ $\triangleright$ Bad prefixes for this safety property constitute the language $\{ \sigma \in \left(2^{\{pay,drink\}}\right)^* \mid w(\sigma, drink) > w(\sigma, pay) \}$ where $w(\sigma, a)$ denotes the number of occurrences of a in σ.

pp. 164, l. 5,6, *two NFAs intersect.* $\triangleright$ the languages of two NFAs intersect.

pp. 164, l. -8, *path fragment* π $\triangleright$ initial path fragment π

pp. 164, l. -6, $TS \otimes \mathcal{A}$ *which has an initial state* $\triangleright$ $TS \otimes \mathcal{A}$ such that there exists an initial state

pp. 167, l. 7, 11, -4, $P_{inv(A)} \triangleright$ $P_{inv(\mathcal{A})}$

pp. 167, l. -2, $q_1, \ldots, q_n \notin F \triangleright$ Note: this condition is not necessary.

pp. 168, l. 1, $0 \leqslant i \leqslant n \triangleright$ $0 < i \leqslant n$

pp. 169, l. Theorem 4.22, $|TS|\cdot|\mathcal{A}| \triangleright$ $|TS \otimes \mathcal{A}|$

pp. 171, l. 8, *single word* $\triangleright$ a set contaning a single word

pp. 171, l. 11, *in* Σ $\triangleright$ in $\mathcal{I}$

pp. 177, l. -7, *Example 4.13 on page 161* $\triangleright$ Example 4.14 on page 162

pp. 183, l. -3, -1, $\mathcal{L}_{q_1q_3} = \ldots \triangleright$ $\mathcal{L}_{q_1q_3} = C^*AB(B + BC^*AB)^*$

pp. 188, l. -2, *containing a b* $\triangleright$ containing only a b

pp. 191, l. -5, *no accepting run that starts in* q_2 $\triangleright$ no accepting run that reaches q_2

pp. 196, l. Example 4.57, *page 193* $\triangleright$ page 194

pp. 200, l. -7, $\bigwedge_{q\in Q} \triangleright$ $\bigwedge_{q\in F}$

pp. 202, l. Fig. 4.22, $\triangleright$ The two states should be labeled s_0 and s_1, respectively

pp. 203, l. 4, $\overline{P}$ = *"eventually forever* $\neg$ *green* $\triangleright$ P = infinitely often *green*

pp. 206, l. Proof:, $TS = (S, Act, \rightarrow, I, AP) \triangleright$ $TS = (S, Act, \rightarrow, I, AP, L)$

pp. 207, l. -4, *We now DFS-based cycle checks ... checking* $\triangleright$ We now present a DFS-based algorithm for persistence checking that searches backwards edges to check for cycles.

pp. 212, l. 6, *ignores all states in* T $\triangleright$ does not revisit the states in T

pp. 215, l. Figure, $t \in R_{in}$ $\triangleright$ $t \in T$

pp. 218, l. 10, *Regula r* $\triangleright$ Regular

Chapter 5: Linear Temporal Logic

pp. 230, l. 5, *eventually in the future* $\triangleright$ now or eventually in the future

pp. 236, l. Figure 5.2, $\triangleright$ It is assumed that $\sigma = A_0 A_1 A_2 \ldots$

pp. 240, l. -10, $\delta_{r_2} = \neg r_1$ $\triangleright$ $\delta_{r_2} = \neg r_2$

pp. 241, l. Fig. 5.6, $\triangleright$ Note that the inputs of the r registers are on the right, and their outputs on the left.

pp. 243, l. -1, $\Box\Big(\bigvee_{1\leqslant i\leqslant N} \mathit{leader}_i \wedge \bigwedge_{\substack{1\leqslant j\leqslant N\\ j\neq i}} \neg\, \mathit{leader}_j\Big)$ $\triangleright$ $\Box \bigvee_{1\leqslant i\leqslant N}\Big(\mathit{leader}_i \wedge \bigwedge_{\substack{1\leqslant j\leqslant N\\ j\neq i}} \neg\, \mathit{leader}_j\Big)$

pp. 244, l. 7, $\Box\Diamond\Big(\bigvee_{1\leqslant i\leqslant N} \mathit{leader}_i \wedge \bigwedge_{\substack{1\leqslant j\leqslant N\\ j\neq i}} \neg\, \mathit{leader}_j\Big)$ $\triangleright$ $\Box\Diamond \bigvee_{1\leqslant i\leqslant N}\Big(\mathit{leader}_i \wedge \bigwedge_{\substack{1\leqslant j\leqslant N\\ j\neq i}} \neg\, \mathit{leader}_j\Big)$

pp. 256, l. -3, $(\sigma[i..] \models \varphi) \ \wedge \ \forall k \leqslant i.\sigma[k..] \models \psi$ $\triangleright$ $(\sigma[i..] \models \varphi \ \wedge \ \forall k \leqslant i.\sigma[k..] \models \psi$

pp. 263, l. 20, $\mathit{Act}' = \mathit{Act} \uplus \{\, \mathit{begin} \,\}$ $\triangleright$ $\mathit{Act}' = \mathit{Act}$ with $\mathit{begin} \notin \mathit{Act}$

pp. 266, l. -8, *interlocked* $\triangleright$ interleaved

pp. 267, l. 7, *as soon as* $\triangleright$ before

pp. 270, l. Fig. 5.15, $\triangleright$ The bottom cell should be white and not gray.

pp. 276, l. -11, $\psi \in B$*if and only if* ... $\triangleright$ $\psi \in B$ if and only if ...

pp. 278, l. Proof of Theorem 5.37, $\triangleright$ It is assumed that $\sigma = A_0 A_1 A_2 \ldots$ is such that $A_i \subseteq \mathit{closure}(\varphi)$, i.e., $A_i = B_i \cap AP$ means $A_i \cap \mathit{closure}(\varphi) = B_i \cap AP$

pp. 281, l. 1-5, *For* $B_0 B_1 B_2 \ldots$ *a sequence ... we have for all* $\psi \in \mathit{closure}(\varphi)$*:* $\psi \in B_0 \Leftrightarrow A_0 A_1 A_2 \ldots \models \psi$ $\triangleright$ For all $\psi \in \mathit{closure}(\varphi)$ and $B_0 B_1 B_2 \ldots$ a sequence ... we have: $\psi \in B_0 \Leftrightarrow A_0 A_1 A_2 \ldots \models \psi$

pp. 283, l. 10, $\neq \bigcirc \psi \in B$ *if and* ... $\triangleright$ $\neg \bigcirc \psi \in B$ if and ...

pp. 283, l. 17, *and* $\varphi = \bigcirc a \in B_1, B_2$ $\triangleright$ and $\varphi = a \in B_1, B_2$

pp. 284, l. -14, $B_3\, B_3\, B_1 B_4^\omega$ $\triangleright$ $B_3\, B_3\, B_1 B_5^\omega$

pp. 287, l. -5, $|\neg(\mathit{fair} \rightarrow \varphi)| = |\mathit{fair}| + |\varphi|$ $\triangleright$ $|\neg(\mathit{fair} \rightarrow \varphi)| = |\neg(\neg \mathit{fair} \vee \varphi)| = |\mathit{fair}| + |\varphi| + 3$

pp. 289, l. 11, *a new vertex b to G* $\triangleright$ a new vertex b to TS

pp. 292, l. Figure 5.23, $\triangleright$ the self-loop at state $P(n)$ should be omitted pp. 292, l. -1, $\bigcirc^{2i-1}(q,A,i) \rightarrow \rhd$ *begin* $\wedge$ $\bigcirc^{2i-1}(q,A,i) \rightarrow$

pp. 294, l. -6, $\mathcal{G}_varphi \rhd \mathcal{G}_\varphi$

pp. 297, l. 7, *Membership to* $\rhd$ Membership in

pp. 303, l. Exercise 5.7(b), $\mathsf{W} \rhd \mathsf{Y}$ (to avoid confusion with unless)

pp. 303, l. Exercise 5.8(a), $\varphi_1 \wedge \varphi_2 \rhd \varphi_1 \,\mathsf{R}\, \varphi_2$

Chapter 6: Computation Tree Logic

pp. 318, l. -10, $\wedge, \rightarrow \rhd \vee, \rightarrow$

pp. 320, l. -4, *state formula* $\rhd$ State formula

pp. 327, l. -12, *since* $\exists(\varphi \,\mathsf{U}\, \psi \vee \Box\varphi) \rhd$ since $\forall(\varphi \,\mathsf{U}\, \psi \vee \Box\varphi)$

pp. 333, l. 10, $\neg\exists\Diamond\neg\Phi = \neg\exists(\mathit{true} \,\mathsf{U}\, \Phi) \rhd \neg\exists\Diamond\neg\Phi \equiv \neg\exists(\mathrm{true} \,\mathsf{U}\, \neg\Phi)$

pp. 336, l. 11, *CTL formulae* $\exists\Diamond(a \wedge \forall\bigcirc a)$ *and* $\Diamond(a \wedge \bigcirc a) \rhd$ CTL formula $\exists\Diamond(a \wedge \forall\bigcirc a)$ and LTL formula $\Diamond(a \wedge \bigcirc a)$

pp. 338, l. 5, $TS_n = (S'_n, \ldots \rhd TS'_n = (S'_n, \ldots$

pp. 338, l. -5 and -6, $\rhd$ transitions to s'_{n-1} are non-existing for n=0

pp. 340, l. -10, *and* $\varphi = \forall\Diamond\exists\Diamond a \rhd$ and $\varphi = \forall\Box\exists\Diamond a$

pp. 342, l. Algorithm 13, and -8 and -4, *maximal genuine* $\rhd$ maximal proper

pp. 343, l. 4, *subformula of* $\Psi \rhd$ subformula of Ψ'

pp. 345, l. -2, $Sat(\exists(\Phi \,\mathsf{U}\, \Psi) \rhd Sat(\exists(\Phi \,\mathsf{U}\, \Psi))$

pp. 345, l. proof of (g)(ii), *Let* $\pi = s_0 s_1 s_2 \ldots$ *be a path starting in* s=s_0. *(As TS has no terminal states, such a path exists.)* $\rhd$ Delete.

pp. 349, l. -9, -7, $(a = c) \wedge (a \neq b) \rhd (a \leftrightarrow c) \wedge (a \not\leftrightarrow b)$

pp. 349, l. -8, *Algorithm 14 (see page 348)* $\rhd$ Algorithm 15

pp. 351, l. Algorithm 15, $\rhd$ comments in the first two lines of algorithm need to be swapped while replacing E by T and T by E

pp. 354, l. Example 6.28, *see the gray states in Figure 6.13(a).* $\rhd$ cf. Figure 6.13(b).

pp. 354, l. Example 6.28, *Figure 6.13(b), Figure 6.13(c)* $\rhd$ Figure 6.13(c), Figure 6.13(d)

pp. 358, l. 11, $\rhd$ Note that the length of $\Phi_n \in \mathcal{O}(n!)$.

pp. 361, l. Example 6.35, $\Rightarrow$ *in the formulas* $\rhd \rightarrow$

pp. 371, l. -6, *ifstatement* $\rhd$ if statement pp. 372, l. Algorithm 19, line 4, $C \cap Sat(b_j) \neq \varnothing$ ▷ $C \cap Sat(b_i) \neq \varnothing$

pp. 374, l. 1, *path formula of the form* $\exists\varphi$ ▷ state formula of the form $\exists\varphi$

pp. 374, l. 6, *counterxamples* ▷ counterexamples

pp. 378, l. -6, *Eaxmple* ▷ Example

pp. 380, l. 12, $(a \wedge a') \,\mathsf{U}\, (\neg a \wedge \neg a' \wedge a_{fair})$ ▷ $(a \wedge \neg a') \,\mathsf{U}\, (\neg a \wedge \neg a' \wedge a_{fair})$

pp. 381, l. 9, $\Box \Diamond (q \wedge r) \rightarrow \Box \Diamond \neg(q \vee r)$ ▷ $\Box \Diamond (a \wedge b) \rightarrow \Box \Diamond \neg(a \vee b)$

pp. 381, l. 9 and 12, $b = c$ ▷ $b \leftrightarrow c$

pp. 383, l. 12, $0 \leqslant n \leqslant m \leqslant k$ ▷ $1 \leqslant n \leqslant m \leqslant k$

pp. 383, l. 9 and 10, $\ldots z_m$ ▷ $\ldots, z_m$

pp. 386, l. 13, 15 (twice) and 19, $s\{\overline{y} \leftarrow \overline{z}\}$ ▷ $s\{\overline{z} \leftarrow \overline{y}\}$

pp. 387, l. 18, $t\{\bar{x}/\bar{x}'\}$ ▷ $t\{\bar{x}' \leftarrow \bar{x}\}$

pp. 388, l. 7, x' ▷ x'_1

pp. 388, l. 7, $\bigwedge_{j<i\leqslant n}(x_j \leftrightarrow x'_j)$ ▷ $\bigwedge_{i+1<j\leqslant n}(x_j \leftrightarrow x'_j)$

pp. 388, l. 7-8, ▷ add conjunct $\wedge \left(\neg x_1 \rightarrow x'_1 \wedge \bigwedge_{1<j\leqslant n}(x_j \leftrightarrow x'_j)\right)$

pp. 388, l. 9, $\chi_B(\overline{x}) = x_1$ ▷ $\chi_B(\overline{x}) = \neg x_1$

pp. 388, l. 14–17, ▷ x and x' should be swapped

pp. 388, l. Example 6.58 (four times), $\{x \leftarrow x'\}$ ▷ $\{x' \leftarrow x\}$

pp. 389, l. 9, $\bigwedge_{1\leqslant i\leqslant n} \Delta_i(\overline{x}_i, \overline{x}'_i)$ ▷ $\bigwedge_{1\leqslant i\leqslant m} \Delta_i(\overline{x}_i, \overline{x}'_i)$

pp. 389, l. 16, $\bigvee_{1\leqslant i\leqslant m}$ ▷ $\bigvee_{1\leqslant i\leqslant m}$

pp. 390, l. 8, $\exists s' \in S s.t. s' \in Post(s)$ ▷ $\exists s' \in S.\, s' \in Post(s)$

pp. 390, l. Algorithm 20, line 4, $f_{j+1}(\bar{x}) := f_{j+1}(\bar{x}) \vee \ldots$ ▷ $f_{j+1}(\bar{x}) := f_j(\bar{x}) \vee \ldots$

pp. 391, l. Algorithm 21, line 4, $f_{j+1}(\bar{x}) := f_{j+1}(\bar{x}) \wedge \ldots$ ▷ $f_{j+1}(\bar{x}) := f_j(\bar{x}) \wedge \ldots$

pp. 391, l. Algorithm 21, line 4, *return* ▷ **return**

pp. 391, l. 19, *can be ruled as* ▷ can be ruled out as

pp. 393, l. Figure 6.21 (right), *solid line between* z_3 *and 0* ▷ dashed line between z_3 and 0

pp. 395, l. 5, *or* $i = j$ ▷ or $z_i = z_j$

pp. 396, l. -15, *The semantics* ▷ The semantics of

pp. 396, l. -1, $f_{succ_b(v)}|_{z=c}$ ▷ $f_{succ_b(v)}|_{z=b}$

pp. 396, l. Def. 6.65, *for node* ▷ for node v

pp. 398, l. 9, *left subtree* ▷ right subtree pp. 400, l. -9, $\langle var(v), succ_1(v), succ_0(v)\rangle = \langle var(w), succ_1(w), succ_0(w)\rangle$ ▷ $var(v) = var(w)$ and $f_{succ_0(v)} = f_{succ_0(w)}$ and $f_{succ_1(v)} = f_{succ_1(w)}$

pp. 402, l. 6, $f_v = (\neg z \wedge f_{succ_0(v)}) \wedge (z \wedge f_{succ_1(v)}) = (\neg z \wedge f_w) \wedge (z \wedge f_w) = f_w$ ▷ $f_v = (\neg z \wedge f_{succ_0(v)}) \vee (z \wedge f_{succ_1(v)}) = (\neg z \wedge f_w) \vee (z \wedge f_w) = f_w$

pp. 402, l. 8, $f_v = (\neg z \wedge f_{succ_0(v)}) \wedge (z \wedge f_{succ_1(v)}) = (\neg z \wedge f_{succ_0(w)}) \wedge (z \wedge f_{succ_1(w)}) = f_w.$ ▷ $f_v = (\neg z \wedge f_{succ_0(v)}) \vee (z \wedge f_{succ_1(v)}) = (\neg z \wedge f_{succ_0(w)}) \vee (z \wedge f_{succ_1(w)}) = f_w.$

pp. 403, l. 10, *ismorphism* ▷ isomorphism

pp. 405, l. 2, $z_m = a_m, z_m = b_m, \ldots, z_i = a_i, z_i = b_i$ ▷ $z_m = a_m, y_m = b_m, \ldots, z_i = a_i, y_i = b_i$

pp. 405, l. 3, $z_m = a_m, z_m = b_m, \ldots, z_{i+1} = a_{i+1}, z_{i+1} = b_{i+1}, z_i = a_i$ ▷ $z_m = a_m, y_m = b_m, \ldots, z_{i+1} = a_{i+1}, y_{i+1} = b_{i+1}, z_i = a_i$

pp. 405, l. -5, $I_{\overline{b}} = \{i \in \{1, \ldots, m\} \mid b_i = 1$ ▷ $I_{\overline{b}} = \{i \in \{1 \ldots, m\} \mid b_i = 1\}$

pp. 405, l. -4, *As f* $\overline{b}$, $\overline{\mathfrak{c}} \in \{0,1\}^m$ ▷ As $\overline{b}$, $\overline{\mathfrak{c}} \in \{0,1\}^m$

pp. 409, l. -12, $info(v) = \langle var(v), succ_0(v), succ_0(v)\rangle$ ▷ $info(v) = \langle var(v), succ_1(v), succ_0(v)\rangle$

pp. 412, l. 7, u ▷ v

pp. 412, l. Algorithm 22, line -4, *rule (?)* ▷ rule

pp. 413, l. 13, $f_2 z_1 = b_1, \ldots, z_i = b_i$ ▷ $f_2|_{z_1=b_1,\ldots,z_i=b_i}$

pp. 414, l. Algorithm 23, line -2, **return** *node w* ▷ should be just before final **fi**

pp. 417, l. heading Algorithm 24, $(v, \overline{x} \leftarrow \overline{x}')$ ▷ $(v, \overline{x}' \leftarrow \overline{x})$

pp. 417, l. Algorithm 24, ▷ swap $\overline{x}$ and $\overline{x}'$

pp. 417, l. Algorithm 24, line 4, *ist* ▷ is a

pp. 417, l. Algorithm 24, ▷ replace z by x and u by v

pp. 418, l. -9, $f\{x \leftarrow x'\}$ ▷ $f\{x' \leftarrow x\}$

pp. 418, l. -6, $f|_{x=\overline{b}}$ ▷ $f|_{x=b}$

pp. 420, l. Algorithm 26, line 10 and 11, $\bar{x}, \bar{x}'$); ▷);

pp. 420, l. Algorithm 26, line 6,7,9,10, $v|_{x_i=0}$ *and* $v|_{x_i=1}$, *respectively* ▷ $v|_{x_i'=0}$ and $v|_{x_i'=1}$, respectively

pp. 420, l. Algorithm 26, line -5, $w_0 := RelProd(u|_{x_i'=0}, v); w_1 := RelProd(u|_{x_i'=1}, v);$ ▷ $w_0 := RelProd(u|_{x_i'=0}, v|_{x_i'=0}); w_1 := RelProd(u|_{x_i'=1}, v|_{x_i'=1});$

pp. 426, l. -1, $\exists\Diamond(a \wedge \exists\Diamond b) \wedge \exists\Diamond(b \wedge \exists\Diamond a)$ ▷ $\exists\Diamond(a \wedge \exists\Diamond b) \vee \exists\Diamond(b \wedge \exists\Diamond a)$

Chapter 7: Equivalences and Abstraction

pp. 454, l. 3, *Sssume* $\triangleright$ Assume

pp. 459, l. 7, $[s]_\sim \xrightarrow{\tau} [s']_\sim$ $\triangleright$ $[s]_\sim \xrightarrow{\tau}{}' [s']_\sim$

pp. 464, l. Figure 7.9, *arrows* $n_1\,c_2$ *to* $w_1 w_2$ *and* $c_1 n_2$ *to* $w_1 w_2$ $\triangleright$ should be omitted

pp. 466, l. 8, $H = Act_1 \cap Act_2$ $\triangleright$ $H = (Act_1 \cap Act_2) \setminus \{ \tau \}$

pp. 467, l. 8, $Act = 2^{AP} \cup \{ \tau \}$ $\triangleright$ $Act = 2^{AP}$

pp. 467, l. 10, $s \xrightarrow{\tau}_{act} t$ $\triangleright$ $s \xrightarrow{L(s)}_{act} t$

pp. 469, l. Remark 7.19, line 10, $s_2 \models \varphi$*, but* $s_1 \not\models \varphi$ $\triangleright$ $s_2 \not\models \neg\varphi$, but $s_1 \models \neg\varphi$

pp. 475, l. Corollary 7.27 (c), $\equiv_{CTL}$ $\triangleright$ $\equiv_{CTL^*}$

pp. 478, l. 11, *fo* $\triangleright$ of

pp. 489, l. Algorithm 32, line 6+7, $\triangleright$ these lines need to be swapped

pp. 489, l. Algorithm 32, $\Pi_{old} := \Pi$ $\triangleright$ $\Pi_{old} := \Pi_{old} \setminus \{ C' \} \cup \{ C, C' \setminus C \}$

pp. 512, l. Definition 7.65, 2nd clause, $s \xrightarrow{\alpha} s'$ *and* $s \xrightarrow{\alpha} s''$ $\triangleright$ $s \longrightarrow s'$ and $s \longrightarrow s''$

pp. 513, l. 9, $\{ a \}\varnothing \notin Traces(TS_1)$ $\triangleright$ $\{ a \}\varnothing \notin Traces(TS_2)$

pp. 518, l. 8, $\forall\Phi \in \forall CTL^*$ $\triangleright$ $\forall\Phi \in \forall$CTL

pp. 519, l. -10, *fragment of* CTL^* $\triangleright$ fragment of CTL

pp. 522, l. -4, $|Post(s_2)|$ $\triangleright$ $|Post(s_2) \cap Sim(s_1')$

pp. 528, l. -9, $s_1 \in Pre(s_2')$ $\triangleright$ $s_1 \in Pre(s_1')$

pp. 532, l. -1, $(TS)2$ $\triangleright$ $TS_2)$

pp. 537, l. -5, $\langle c_2, n_1 \rangle$ $\triangleright$ $\langle n_1, c_2 \rangle$

pp. 539, l. 2, $\mathcal{R}$ *on* $(S_1 \times S_2) \cup (S_1 \times S_2)$ $\triangleright$ $\mathcal{R}$ on $TS_1 \oplus TS_2$

pp. 541, l. Figure 7.36, *transition from* ℓ_3 *to* ℓ_4 $\triangleright$ should be deleted

pp. 541, l. -2, $\{ (s, s') \mid s' \in [s]_\mathcal{R}), s \in S \}$ $\triangleright$ $\{ (s, [s]_\mathcal{R}) \mid s \in S \}$

pp. 542, l. 5, $\langle c_2, n_1 \rangle$ $\triangleright$ $\langle n_1, c_2 \rangle$

pp. 544, l. 11, *finer that* $\triangleright$ finer than

pp. 546, l. 13, s_2 *is* $\approx^{div}_{TS}$*-divergent whereas* s_0 *and* s_1 *are not.* $\triangleright$ s_2 is not $\approx^{div}_{TS}$-divergent whereas s_0 and s_1 are.

pp. 546, l. after Example 7.110, *where the state labelling is indicated by the grey scale* $\triangleright$

pp. 549, l. Definition 7.116, clause 2, $\widehat{\pi}_1$ $\triangleright$ $\widehat{\pi}_i$

pp. 554, l. 8, *amounts* $\triangleright$ amounts to pp. 556, l. Figure 7.45, v_1 *and* v_2 ▷ t_1 and t_2

pp. 556, l. Figure 7.45 (rechts), s_1 ▷ s_2

pp. 557, l. -8, *since* s_2 *and* u_2 *are* $\mathcal{R}$*-equivalent* ▷ since s_1 and u_2 are $\mathcal{R}$-equivalent

pp. 562, l. 1, *and* $s_1 \exists\varphi$ ▷ and $s_1 \models \exists\varphi$

pp. 563, l. 4, $\Phi_B \cup \Phi_C$ *is a* $CTL_{\setminus\bigcirc}$ *formula* ▷ $\exists(\Phi_B \cup \Phi_C)$ is a $\text{CTL}_{\setminus\bigcirc}$ state-formula

pp. 564, l. Figure 7.46, *the states* $(0,1)$ *and* $(1,0)$ *with self-loop* ▷ should be included in the "chain", and not be seperate deadlock states

pp. 566, l. 16, $\ell_2 : \langle \textbf{if } (\mathit{free} > 0) \textbf{ then } i := 0; \mathit{free}-- \textbf{ fi} \rangle$ ▷ $\ell_2 : \langle \textbf{if } (\mathit{free} > 0) \textbf{ then } i := 0; \mathit{free}-- \textbf{ fi} \rangle \,; \textbf{goto } \ell_0$

pp. 566, l. -3, $\langle \ell_0, \ell_2', 2, 0, 0\rangle \rightarrow \langle \ell_0, \ell_0', 2, 0, 0\rangle$ ▷ $\langle \ell_1, \ell_2', 2, 0, 0\rangle \rightarrow \langle \ell_1, \ell_0', 2, 0, 0\rangle$

pp. 568, l. Definition 7.134, condition 1., $B \cap Pre(C) \neq \varnothing$ ▷ $B \cap Pre^*_\Pi(C) \neq \varnothing$

pp. 569, l. Lemma 7.135 (ii), $B \cap Pre(C) \neq \varnothing$ ▷ $B \cap Pre^*_\Pi(C) \neq \varnothing$

pp. 569, l. 7, *there are some states in* B *that cannot reach* C *by only visiting states in* B*. For such states, the only possibility is to reach* C *via some other block* $D \neq B, C$*.* ▷ C can only be reached via paths that entirely go through B.

pp. 572, l. 11, $t \in Exit(B)$ ▷ $t \in Bottom(B)$

pp. 577, l. -2, *quotient space* $S/\cong$ ▷ quotient space $S/\cong^{div}$

pp. 578, l. 4, $E = \{\, (s,t) \in S \times S \mid L(s) = L(t) \,\}$ ▷ $E = \{\, (s,t) \in S \times S \mid L(s) = L(t) \wedge s \xrightarrow{\alpha} t \text{ for some } \alpha \in Act \,\}$

pp. 578, l. item 3., *self-loops* $[s]_{div} \rightarrow [s]_{div}$ ▷ self-loops $[s] \rightarrow [s]$

pp. 592, l. Exercise 7.29, item (B)(2), $s_1, s_1') \notin \mathcal{R}$ ▷ $s_1', s_2) \notin \mathcal{R}$

pp. 592, l. Exercise 7.29, item (c), $TS_1 \neg \trianglelefteq TS_2$ ▷ $TS_1 \not\trianglelefteq TS_2$

Chapter 8: Partial-Order Reduction

pp. 596, l. 19, *consists* ▷ consists of

pp. 597, l. 11, *of state space* ▷ of the state space

pp. 601, l. -11, *TS be action-deterministic* ▷ *TS* be an action-deterministic

pp. 602, l. 5, *independent on* ▷ independent of

pp. 605, l. $s_i = \alpha(t_i)$, $t_i = \alpha(s_i)$ ▷ pp.
, l. [, *1* ▷ e

x] 609(A2)If α depends on $ample(s)$If $\alpha \notin \alpha(s)$ depends on $\alpha(s)$ pp. 610, l. 3, *all ample actions* $\triangleright$ all actions

pp. 610, l. 4, *Note that for n=0, condition (A2) is false, as the existential quantification (over i) ranges over an empty domain.* $\triangleright$ Note that condition (A2) is false if there is an execution fragment $s \xrightarrow{\alpha} t$ such that $\alpha \notin ample(s)$ and α depends on $ample(s)$. After all, in that case $n = 0$ and the existential quantification (over i) ranges over an empty domain. This observation will be formalized in Lemma 8.14.

pp. 610, l. 6, *any finite execution in TS* $\triangleright$ any finite execution in TS ending after the first ample action

pp. 610, l. 14, $s_1 \xrightarrow{\beta_1} s_2 \xrightarrow{\beta_2} \ldots$ *with* β_i *independent of* $ample(s)$ *for* $0 < i \leqslant n$ $\triangleright$ $s \xrightarrow{\beta_1} s_1 \xrightarrow{\beta_2} \ldots$ with β_i independent of $ample(s)$ for $0 < i$

pp. 611, l. (A2), *If* α *depends on* $ample(s)$ $\triangleright$ If $\alpha \notin \alpha(s)$ depends on $\alpha(s)$

pp. 611, l. 6, *cycle* $s_0 s_2 s_2$ $\triangleright$ cycle $s_2 s_2$

pp. 612, l. -8 and -10, $Reach(TS)$ $\triangleright$ $Reach(\hat{TS})$

pp. 613, l. Lemma 8.15, *then for all actions* $\triangleright$ then all actions

pp. 613, l. Lemma 8.15, *adddition* $\triangleright$ addition

pp. 613, l. 8, *constraints (A1) and (A2)* $\triangleright$ constraint (A2)

pp. 613, l. 10, $Act(\beta(s_0)) = Act(s_1)$ $\triangleright$ $Act(\beta_1(s_0)) = Act(s_1)$

pp. 613, l. below Notation 8.16, *necessary* $\triangleright$ almost sufficient

pp. 616, l. -9, $ample(\langle s_0, t_i \rangle) = \{ \alpha_{i+1} \}$, *for* i=1, 2, 3 $\triangleright$ $ample(\langle s_0, t_i \rangle) = \{ \alpha_{i+1} \}$, for i=0, 1, 2

pp. 623, l. -10 and -4, *Section 5.2* $\triangleright$ Section 4.4.2

pp. 624, l. 5, *Section 5.2* $\triangleright$ Section 4.4.2

pp. 625, l. Algorithm 39, line 3, $TS \models \Box \Phi$ $\triangleright$ $TS \models \Diamond \Box \Phi$

pp. 626, l. Algorithm 40, line 14, $push(t, V)$ $\triangleright$ delete this line

pp. 629, l. -5, $\varrho = s_0 \to \ldots \to t \xrightarrow{\alpha} trap$ $\triangleright$ $\varrho = s_0 \to' \ldots \to' t \xrightarrow{\alpha}{}' trap$

pp. 634, l. Algorithm 41, line 8, $(\exists j \neq i.\, Act_i(s) \times Act_j(s) \cap D = \varnothing)$ $\triangleright$ $(\forall j \neq i.\, Act_i(s) \times Act_j) \cap D = \varnothing)$

pp. 645, l. -4 and pp. 646, line 2, $n_1 \xleftrightarrow{x<N:x:=x+1} \ell_1 \xleftrightarrow{b:=\neg b} n_1$ $\triangleright$ $n_1 \xleftrightarrow{x<N:x:=x+1} \ell_1 \xleftrightarrow{b:=\neg b} n_1$

pp. 646, l. Fig. 8.16 (right), *edge label* β_2 *from* ℓ_2 *to* n_2 $\triangleright$ edge label γ_2 from ℓ_2 to n_2

pp. 647, l. Fig. 8.18 (right), *edge label* β_2 *from* ℓ_2 *to* n_2 $\triangleright$ edge label γ_2 from ℓ_2 to n_2

pp. 648, l. Algorithm 42, line 8, $\bigcup_{j<k\leqslant n} Act_k$ ▷ $\bigcup_{i<k\leqslant n} Act_k$

pp. 650, l. -4, *Figure 8.19 (left)* ▷ Figure 8.19 (top)

pp. 650, l. -1, *Figure 8.19 (right)* ▷ Figure 8.19 (bottom)

pp. 651, l. 5, *a-state in TS* ▷ a-state in $\hat{TS}$

pp. 651, l. 12, *Since* $\hat{TS}$ *does not contain* s ▷ Since $\hat{TS}$ does not contain s_0

pp. 652, l. 10, *Figure 8.19 (left)* ▷ Figure 8.19

pp. 653, l. (A2), *If* α *depends on* $ample(s)$ ▷ If $\alpha \notin \alpha(s)$ depends on $\alpha(s)$

pp. 654, l. -4, *Natural* $\nu_2(\ldots)$ ▷ Natural number $\nu_2(\ldots)$

pp. 666, l. Exercise 8.6, $ample(s_9) = \{\, \alpha, \beta, \delta \,\}$ ▷ $ample(s_9) = \{\, \eta, \beta, \delta \,\}$

Chapter 9: Timed Automata

pp. 674, l. -12, *is more an intuitive than* ▷ is more intuitive than

pp. 678, l. Definition 9.3, 5th bullet, *is a transition relation* ▷ is a finite transition relation

pp. 680, l. -1, *transition label true* $: x \geqslant 2, \{\, x \,\}$ ▷ transition label $x \geqslant 2 : \tau, \{\, x \,\}$

pp. 681, l. Figure 9.6, *coming down and going up* ▷ *comingdown* and *goingup*

pp. 683, l. -9, $\ldots || TA_n$ ▷ $\ldots ||_H TA_n$

pp. 685, l. Figure 9.9, $\langle far, 0, up)\rangle \rightarrow \langle near, 1, up\rangle$, $reset(x, y)$ ▷ $reset(z, y)$

pp. 687, l. Definition 9.10, *clause for* $\neg g$ ▷ should be omitted

pp. 696, l. 2, $\eta \not\models g_j$ *or* $Inv(\ell_j)$ ▷ $\eta \not\models g_j$ or $\eta \not\models Inv(\ell_j)$

pp. 696, l. 12, η_{i-1} ▷ η_{j-1} (this occurs twice!)

pp. 696, l. proof of Lemma 9.24, ▷ The variables i, j and x depend on the cycle in π. For the sake of simplicity, this dependency is not treated here.

pp. 696, l. -5, *when going from location off to on* ▷ when going from location *on* to *off*

pp. 697, l. Notation 9.26, line 1, *infinitely many actions are* ▷ infinitely many actions from Act are

pp. 699, l. -4, *more than 2 minutes* ▷ at least two minutes

pp. 699, l. -3, $\forall\Diamond^{>2}\neg on$ ▷ $\forall\Diamond^{\leqslant 2}\neg on$

pp. 702, l. Figure 9.16, $x > 3 : reset(x)$ ▷ $x \geqslant 3 : reset(x)$

pp. 702, l. -5, *TCTLsemantics* ▷ TCTL semantics pp. 705, l. -12, *Remark 9.35* $\triangleright$ Lemma 9.35

pp. 707, l. -1, $\pi \in s_0 \xrightarrow{d_0} s_1 \xrightarrow{d_1} s_2 \xrightarrow{d_2}$ $\triangleright$ $\pi \in s_0 \xrightarrow{d_0} s_1 \xrightarrow{d_1} s_2 \xrightarrow{d_2} \ldots$

pp. 709, l. -10, *of the form* $x \leqslant c$ *or* $x < c$ $\triangleright$ of the form $x \leqslant c$, $x < c$, $x \geqslant c$ or $x > c$

pp. 710, l. -12, *Figure 9.18)* $\triangleright$ Figure 9.18

pp. 711, l. -2, $\eta_1 \cong_2 \eta_2$ *iff* $\eta_1 \cong_1 \eta_2$ $\triangleright$ $\eta \cong_2 \eta'$ iff $\eta \cong_1 \eta'$

pp. 713, l. Definition 9.42, line 3, *if and only if either* $\triangleright$ if and only if either for all $x \in C$ (in the two bullets the universal quantification over x needs to be deleted)

pp. 716, l. -3, *consraint (C)* $\triangleright$ constraint (C)

pp. 717, l. Proof of Theorem 9.46, *open intervals like* $]0, 1[$ $\triangleright$ $(0, 1)$

pp. 725, l. 4, *delay transitions* $\triangleright$ action transitions

pp. 726, l. -3, $\triangleright$ add: where I is the set of initial states of $TS(TA)$

pp. 730, l. 4, $\forall\Diamond\, a$ $\triangleright$ $a \,\mathsf{U}\, b$

pp. 730, l. 19, $\Diamond\, a$ $\triangleright$ $a \,\mathsf{U}\, b$

pp. 730, l. 21, *time-convergent* $\triangleright$ time-divergent

pp. 730, l. -4, *for* $i \leqslant j$ $\triangleright$ for all $i \leqslant j$

pp. 730, l. -4, $\pi \models_{TCTL} \Diamond\, a$ $\triangleright$ $\pi \models_{\mathrm{TCTL}} a \,\mathsf{U}\, b$

pp. 731, l. Example 9.63, *with* $\eta(x) > 1$ $\triangleright$ with $\eta(x) = 2$

pp. 732, l. 6, *several state regions* $\triangleright$ several states

pp. 740, l. Exercise 9.1, edge label at location *on*, $x \geqslant 2$: *sw_on, reset*(x) $\triangleright$ $x \geqslant 2$: *switch_on*, *reset*(x)

Chapter 10: Probabilistic Systems

pp. 749, l. Example 10.2, *senf off* $\triangleright$ sent off

pp. 753, l. Notation 10.6, l. 1, $Post^*(s)$ $\triangleright$ $Post(s)$

pp. 774, l. 3, *any successor of* t $\triangleright$ any state reachable from t

pp. 776, l. -3, *absorbing states* $\triangleright$ states pp. 778, l. 4, $\mathbf{P}'(s,t) = \ldots \rhd$

$$\mathbf{P}'(s,t) = \begin{cases} 1 & \text{if } s = t \text{ and } s \in B \cup S \setminus (C \cup B) \\ 0 & \text{if } s \neq t \text{ and } s \in B \cup S \setminus (C \cup B) \\ \mathbf{P}(s,t) & \text{otherwise.} \end{cases}$$

pp. 778, l. -9, $\{ 4, 5, 6, won \} \rhd \{ won \}$

pp. 780, l. 12, *it can be shown that* $Pr(n \models \Diamond 0) < 1 \rhd$ it can be shown that $Pr(n \models \Diamond 0) > 0$

pp. 782, l. -3, *expresses in addition that almost surely the player will always win* $\rhd$ expresses that within five steps, the player reaches a state from which he will win almost surely

pp. 821, l. 13, *time complexity of the size* $\rhd$ time complexity in the size

pp. 851, l. Theorem 10.100, $\rhd$ Add the following condition: $\sum_{s \in S} x_s$ is minimal.

pp. 857, l. 2, $\sum_{s \in S_? \setminus \{s\}} \mathbf{P}(s, \alpha, t) \cdot x_t \rhd - \sum_{s \in S_? \setminus \{s\}} \mathbf{P}(s, \alpha, t) \cdot x_t$

pp. 863, l. Algorithm 46, **return** $T \rhd$ **return** $S \setminus T$

pp. 865, l. Lemma 10.113 + succeeding paragraph, $\rhd$ should be after Theorem 10.109

pp. 870, l. Lemma 10.119, *any* $s \in S \rhd$ any $s \in T$

pp. 876, l. 11, $U_{\Box\Diamond P} \rhd U_{\Box\Diamond B}$

pp. 883, l. Theorem 10.129 and just before, *is in 2EXPTIME* $\rhd$ is 2EXPTIME-complete (twice)

pp. 903, l. Exercise 10.14, $\varphi = \Box \Diamond a \rhd \varphi = \Diamond \Box a$

pp. 903/904, l. Exercise 10.17, *Markov chain* $\mathcal{M} \rhd$ Markov chain $\mathcal{M}$ where all states are equally labeled

pp. 905, l. Exercise 10.22, $\rhd$ Compute also the values $y_s = Pr^{\max}(s \models C \cup B)$ with $C = S \setminus \{ s_3 \}$ and $B = \{ s_6 \}$

pp. 905, l. Exercise 10.23, *(a), 1. and (b)* $\rhd$ (a), (b), (c)

Appendix

pp. 912, l. footnote, $\sigma = A_1 A_2 A_3 \ldots \rhd \sigma = A_0 A_1 A_2 \ldots$

pp. 918, l. 8, *not to 1* $\rhd$ not to n

pp. 925, l. 1, *they are composed of simple paths* ▷ they are composed of paths, at least one of which is simple.