

正誤情報

このたびは森北出版株式会社発行の書籍をお買い求めいただき、誠にありがとうございました。下記の書籍につきまして誤りのある箇所がございましたので、お詫びし訂正させていただきます。

2018年5月21日 森北出版株式会社 生産マネジメント部

タイトル

Coq/SSReflect/MathCompによる定理証明

正誤対象

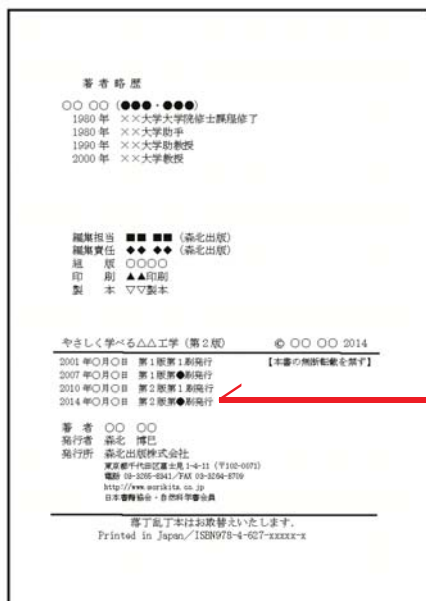
お手持ちの書籍の刷数をお調べのうえ、下の表をご覧ください。正誤表内の一番左に「対応刷数」という列がございます。該当する刷数の訂正情報をご参照下さい。

なお、刷数につきましては下記「刷数の調べ方」をご参照ください。

お持ちの本の刷数	
1	対応刷数 1 より 2 までをご参照ください
2	対応刷数 2 をご参照ください
それ以降	現在把握している訂正情報はございません

刷数の調べ方

本の一番後ろのページ(広告等除く)に下図のようなページがございます。ご参照いただき、お持ちの本の刷数をお調べください。



日付の最も新しい行に記載された数字がお持ちの本の刷数となります

対応 刷数	頁	行数, 図・ 表・式番号	誤	正
2	37	3 行目	…を「 <u>Z</u> 」に…	…を「 <u>X</u> 」に…
2	43	3~4 行目	…引数が同じであるときに True を返し, <u>そうでないときに False</u> を返します. …	…引数が <u>まったく同じであることを証明できれば</u> , <u>apply: eq_refl</u> で証明される関数です. …
2	44	下から 15 行目	… S (S (S <u>0</u>))…	…S (S (S <u>n</u>))…
2	50	下から 14 行目	2 * (<u>n.+ 1 + sum n</u>)=…	2 * n.+ 1 + <u>2 * sum n</u> =…
2	52	表 2.1 キャプション	表 2.1 論理記号の記法と関数名	表 2.1 論理記号の記法と関数名 <u>等</u>
2	52	表 2.1 列見出し	関数名	関数名 <u>等</u>
2	52	表 2.1	(同値-関数名) eq	iff
2	53	11 行目	…が <u>定める</u> ことを, …	…が <u>定まる</u> ことを, …
2	65	問 2.3	… mode … (2 箇所)	… move …
2	72	表 3.3	(apply. -代表的な使い方) …トップが(P1 -> P2 -> … -> <u>Pm</u>)であり, トップより右側が(P1 -> P2 -> … -> Pn)であるとき…	…トップが(P1 -> P2 -> … -> <u>Pn</u>)であり, トップより右側が(<u>Pm+1</u> -> <u>Pm+2</u> -> … -> Pn)であるとき…
2	74	図 3.9	(ゴールエリア (前)) 1 subgoal A, B, C : Prop AtoB : A -> B ----- (1/1) <u>((A -> B) -> C) -> C</u>	1 subgoal A, B, C : Prop AtoBtoC : <u>(A -> B) -> C</u> ----- (1/1) C
2	74	図 3.9	(タクティク) apply: AtoB=>a.	apply: AtoBtoC => a.
2	76	3.4.2 3 行目	…変数 (<u>もしくは関数</u>) です.	…変数です.

2	78	3.4.8 2行目	…move=> <u>Am</u> …	…move=> <u>Ai</u> …
1	79	3.5節 イラスト		p.82の3.6節のゾウのイラストと入れ替える
2	80	表 3.5	(<u>elim.</u> —代表的な使い方 の6行目) …定義される型 <u>XXX_ind</u> に…	…定義される <u>XXX_ind</u> の型に…
2	80	下から 7~6行目	…、型 <u>nat_ind</u> が	…、 <u>補題 nat_ind</u> の型が
1	82	3.6節 イラスト		p.79の3.5節のタコのイラストと入れ替える
2	89	3.7.4 4行目	… <u>apply BtoC.</u> と <u>apply AtoB.</u> を…	… <u>apply: BtoC.</u> と <u>apply: AtoB.</u> を…
2	89	図 3.34	(ゴールエリア (前) 4行目, ゴールエリア (後) 4行目) <u>BtoA : B -> C</u>	<u>BtoC : B -> C</u>
2	95	7行目	…注目します♦1.	…注目します♦1. <u>以下, ライブラリ <u>ssreflect, ssrbool, eqtype, ssrnat</u> をインポートした状況の解説です.</u>
2	95	8行目	Coq には型 <u>nat</u> の…	Coq には <u>一般の型上</u> の…
2	96	13行目	…Prop 型の = から bool 型の == に…	…Prop 型の = (<u>eq</u>) から bool 型の == (<u>eq_op</u>) に…
2	97	14~18行目	とくに, 「=」のかわりに <u>eqn</u> を使う利点も説明しました. MathComp では… … … …4.2節で行います.	…かわりに <u>eq_op</u> を使う利点も説明しました. (「MathComp では」以降「4.2節で行います。」まで4行削除)
2	107	2行目	…「 <u>Search _ (_ + _ = _).</u> 」…	…「 <u>Search (_ + _ = _).</u> 」…
2	118	2行目	… <u>機能的型</u> …	… <u>帰納的型</u> …
2	118	4行目	…ついた <u>型</u> のいくつか…	…ついた <u>補題 (もしくは関数)</u> のいくつか…
2	127	表 4.1	(==> - 意味) <u>b => c</u> …	<u>b ==> c</u> …
2	129	表 4.3 キャプション	ブール演算に関する補題例	ブール演算に関する補題例. <u>ただし a, b, c : bool</u>
2	130	表 4.4 キャプション	二項演算の命名規則. <u>ただし a, b, c : bool</u>	二項演算の命名規則

2	132	表 4.5	<code>eq_op</code> (全部で5箇所)	<code>eq_op</code>								
2	132	4.2.3 節 3行目	<code>eq_sym)</code> 」, 「…	<code>eq_sym)</code> 」, 「… (\の削除)								
2	136	表 4.8 eqnP 行	<table border="1"> <thead> <tr> <th>型</th> <th>補足</th> </tr> </thead> <tbody> <tr> <td><code>ssrbool.reflect (?x = ?y)(eqn ?x ?y)</code></td> <td></td> </tr> </tbody> </table>	型	補足	<code>ssrbool.reflect (?x = ?y)(eqn ?x ?y)</code>		<table border="1"> <thead> <tr> <th>型</th> <th>補足</th> </tr> </thead> <tbody> <tr> <td><code>ssrbool.reflect (x = y)(eqn x y)</code></td> <td><code>x, y : nat</code></td> </tr> </tbody> </table>	型	補足	<code>ssrbool.reflect (x = y)(eqn x y)</code>	<code>x, y : nat</code>
型	補足											
<code>ssrbool.reflect (?x = ?y)(eqn ?x ?y)</code>												
型	補足											
<code>ssrbool.reflect (x = y)(eqn x y)</code>	<code>x, y : nat</code>											
2	136	表 4.8 1eP 行 1tP 行	<table border="1"> <tbody> <tr> <td><code>ssrbool.reflect (?m <= ?n)%coq_nat (?m <= ?n)</code></td> <td></td> </tr> <tr> <td><code>ssrbool.reflect (?m < ?n)%coq_nat (?m < ?n)</code></td> <td></td> </tr> </tbody> </table>	<code>ssrbool.reflect (?m <= ?n)%coq_nat (?m <= ?n)</code>		<code>ssrbool.reflect (?m < ?n)%coq_nat (?m < ?n)</code>		<table border="1"> <tbody> <tr> <td><code>ssrbool.reflect (m <= n)%coq_nat (m <= n)</code></td> <td><u><code>m, n : nat</code></u></td> </tr> <tr> <td><code>ssrbool.reflect (m < n)%coq_nat (m < n)</code></td> <td><u><code>m, n : nat</code></u></td> </tr> </tbody> </table>	<code>ssrbool.reflect (m <= n)%coq_nat (m <= n)</code>	<u><code>m, n : nat</code></u>	<code>ssrbool.reflect (m < n)%coq_nat (m < n)</code>	<u><code>m, n : nat</code></u>
<code>ssrbool.reflect (?m <= ?n)%coq_nat (?m <= ?n)</code>												
<code>ssrbool.reflect (?m < ?n)%coq_nat (?m < ?n)</code>												
<code>ssrbool.reflect (m <= n)%coq_nat (m <= n)</code>	<u><code>m, n : nat</code></u>											
<code>ssrbool.reflect (m < n)%coq_nat (m < n)</code>	<u><code>m, n : nat</code></u>											
2	137	表 4.8 (続き) iterSr 行	<table border="1"> <thead> <tr> <th>型</th> </tr> </thead> <tbody> <tr> <td><code>forall m n : nat, maxn n (minn m n) = n</code></td> </tr> </tbody> </table>	型	<code>forall m n : nat, maxn n (minn m n) = n</code>	<table border="1"> <thead> <tr> <th>型</th> </tr> </thead> <tbody> <tr> <td><code>forall (T : Type) (n : nat) (f : T -> T) (x : T), iter n.+1 f x = iter n f (f x)</code></td> </tr> </tbody> </table>	型	<code>forall (T : Type) (n : nat) (f : T -> T) (x : T), iter n.+1 f x = iter n f (f x)</code>				
型												
<code>forall m n : nat, maxn n (minn m n) = n</code>												
型												
<code>forall (T : Type) (n : nat) (f : T -> T) (x : T), iter n.+1 f x = iter n f (f x)</code>												
2	138	表 4.9	(:: - 意味) … <code>x :: s</code> は <code>x cons s</code> を表す.	… <code>x :: s</code> は <code>cons x s</code> を表す.								
2	138	表 4.9	(nilp - 意味) …, <code>s=[::]</code> のとき <code>tt</code> を, そうでないとき <code>ff</code> を返す.	…, <code>s=[::]</code> のとき <code>true</code> を, そうでないとき <code>false</code> を返す.								
2	142	表 4.9 (続き)	(zip - 意味 1行目) … <code>[:: t1; t2; ...; xn]</code> は…	… <code>[:: t1; t2; ...; tn]</code> は…								
2	143	表 4.9 (続き)	(unzip2 - 意味) … <code>[:: t1; t2; ...; xn]</code> を返す.	<code>[:: t1; t2; ...; tn]</code> を返す.								
2	144	表 4.10	(nseqP - 型) … <code>(y = x / \ 1 > 0)</code> …	… <code>(y = x / \ n > 0)</code> …								
2	151	14行目	…, 計算結果は <code>idx op a_1 op a_2 op a_3</code> と…	…, 計算結果は <code>a_1 op (a_2 op (a_3 op idx))</code> と…								
2	159	1行目	…, 属していないことは <code>(A x)</code> として…	…, 属していないことは <code>~(A x)</code> として…								
2	159	5.2 4~5行目	<code>mySub M</code> (3箇所)	<code>mySet M</code>								
2	160	下から 2~1行目	…」により定義しましょう. 一方で, Coq の型としての等号が <code>mySet {M : Type}</code> に定められています. 「 <code>A = B</code> 」 …	…」と同値になるようにしていきます. <u>すでに Coq には関係 = (等号) が定義されています. そして「<code>A = B</code>」 …</u>								

2	161	14 行目	48 ... : A = A.	48 ... : eqmySet A A.
2	161	15 行目	49 ...by []....	49 ...by rewrite /eqmySet; apply: conj; apply: rfl_Sub.
2	161	17 行目	51 ... : A = B -> B = A.	51 ... : eqmySet A B -> eqmySet B A.
2	161	18 行目	52 move=> H. by rewrite H....	52 by rewrite /eqmySet; case.
2	162	下から 9 行目	右辺の <u>myEmptySet</u> 側	右辺の <u>myMotherSet</u> 側
2	183	7, 9, 11 行目	xRinvy (各行 1 箇所, 計 3 箇所)	xRy
2	185	表 6.1	(groupV - 言明) (x ¹ \in G)	(x ⁻¹ \in G)
2	187	下から 4 行目	xRinvy	xRy
2	188	5, 9 行目	xRinvy (各行 1 箇所, 計 2 箇所)	xRy
2	200	23 行目	...非負 <u>整数</u>非負 <u>実数</u> ...

最終更新 2018.5