List of Temporal Logic Laws

We give here a compact list of laws of the various temporal logics, particularly including those which are frequently used throughout the book. Furthermore, we note some of the corresponding formal systems.

Laws of Basic LTL

(T1) $\neg\bigcirc A \leftrightarrow \bigcirc\neg A$
(T2) $\neg\Box A \leftrightarrow \Diamond\neg A$
(T3) $\neg\Diamond A \leftrightarrow \Box\neg A$
(T4) $\Box A \rightarrow A$
(T5) $A \rightarrow \Diamond A$
(T6) $\Box A \rightarrow \bigcirc A$
(T7) $\bigcirc A \rightarrow \Diamond A$
(T8) $\Box A \rightarrow \Diamond A$
(T9) $\Diamond\Box A \rightarrow \Box\Diamond A$
(T10) $\Box\Box A \leftrightarrow \Box A$
(T11) $\Diamond\Diamond A \leftrightarrow \Diamond A$
(T12) $\Box\bigcirc A \leftrightarrow \bigcirc\Box A$
(T13) $\Diamond\bigcirc A \leftrightarrow \bigcirc\Diamond A$
(T14) $\bigcirc(A \rightarrow B) \leftrightarrow \bigcirc A \rightarrow \bigcirc B$
(T15) $\bigcirc(A \land B) \leftrightarrow \bigcirc A \land \bigcirc B$
(T16) $\bigcirc(A \lor B) \leftrightarrow \bigcirc A \lor \bigcirc B$
(T17) $\bigcirc(A \leftrightarrow B) \leftrightarrow (\bigcirc A \leftrightarrow \bigcirc B)$
(T18) $\Box(A \land B) \leftrightarrow \Box A \land \Box B$
(T19) $\Diamond(A \lor B) \leftrightarrow \Diamond A \lor \Diamond B$
(T20) $\Box\Diamond(A \lor B) \leftrightarrow \Box\Diamond A \lor \Box\Diamond B$
(T21) $\Diamond\Box(A \land B) \leftrightarrow \Diamond\Box A \land \Diamond\Box B$
(T22) $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
(T23) $\Box A \lor \Box B \rightarrow \Box(A \lor B)$
(T24) $(\Diamond A \rightarrow \Diamond B) \rightarrow \Diamond(A \rightarrow B)$

(T25) $\Diamond(A \land B) \rightarrow \Diamond A \land \Diamond B$
(T26) $\Box\Diamond(A \land B) \rightarrow \Box\Diamond A \land \Box\Diamond B$
(T27) $\Diamond\Box A \lor \Diamond\Box B \rightarrow \Diamond\Box(A \lor B)$
(T28) $\Box A \leftrightarrow A \land \bigcirc\Box A$
(T29) $\Diamond A \leftrightarrow A \lor \bigcirc\Diamond A$
(T30) $\Box(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B)$
(T31) $\Box(A \rightarrow B) \rightarrow (\Diamond A \rightarrow \Diamond B)$
(T32) $\Box A \rightarrow (\bigcirc B \rightarrow \bigcirc(A \land B))$
(T33) $\Box A \rightarrow (\Box B \rightarrow \Box(A \land B))$
(T34) $\Box A \rightarrow (\Diamond B \rightarrow \Diamond(A \land B))$
(T35) $\Box(\Box A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
(T36) $\Box(A \rightarrow \Diamond B) \rightarrow (\Diamond A \rightarrow \Diamond B)$
(T37) $\Diamond\Box\Diamond A \leftrightarrow \Box\Diamond A$
(T38) $\Box\Diamond\Box A \leftrightarrow \Diamond\Box A$

Laws for Binary Operators in LTL

(Tb1) $A\ \mathbf{until}\ B \leftrightarrow \bigcirc\Diamond B \land A\ \mathbf{unless}\ B$
(Tb2) $A\ \mathbf{unless}\ B \leftrightarrow \bigcirc(A\ \mathbf{unl}\ B)$
(Tb3) $A\ \mathbf{unl}\ B \leftrightarrow A\ \mathbf{unt}\ B \lor \Box A$
(Tb4) $A\ \mathbf{unt}\ B \leftrightarrow B \lor (A \land A\ \mathbf{until}\ B)$
(Tb5) $A\ \mathbf{unless}\ B \leftrightarrow B\ \mathbf{atnext}\ (A \rightarrow B)$
(Tb6) $A\ \mathbf{atnext}\ B \leftrightarrow B\ \mathbf{before}\ (\neg A \land B)$
(Tb7) $A\ \mathbf{before}\ B \leftrightarrow \neg(A \lor B)\ \mathbf{unless}\ (A \land \neg B)$
(Tb8) $\bigcirc A \leftrightarrow A\ \mathbf{atnext\ true}$
(Tb9) $\Box A \leftrightarrow A \land A\ \mathbf{unless\ false}$
(Tb10) $\Box A \leftrightarrow A\ \mathbf{unl\ false}$
(Tb11) $A\ \mathbf{until}\ B \leftrightarrow \bigcirc B \lor \bigcirc(A \land A\ \mathbf{until}\ B)$
(Tb12) $A\ \mathbf{unless}\ B \leftrightarrow \bigcirc B \lor \bigcirc(A \land A\ \mathbf{unless}\ B)$
(Tb13) $A\ \mathbf{unt}\ B \leftrightarrow B \lor (A \land \bigcirc(A\ \mathbf{unt}\ B))$
(Tb14) $A\ \mathbf{unl}\ B \leftrightarrow B \lor (A \land \bigcirc(A\ \mathbf{unl}\ B))$
(Tb15) $A\ \mathbf{atnext}\ B \leftrightarrow \bigcirc(B \rightarrow A) \land \bigcirc(\neg B \rightarrow A\ \mathbf{atnext}\ B)$
(Tb16) $A\ \mathbf{before}\ B \leftrightarrow \bigcirc\neg B \land \bigcirc(A \lor A\ \mathbf{before}\ B)$
(Tb17) $\neg(A\ \mathbf{unless}\ B) \leftrightarrow \bigcirc\neg B \land \bigcirc(\neg A \lor \neg(A\ \mathbf{unless}\ B))$
(Tb18) $\Box(\neg B \rightarrow A) \rightarrow A\ \mathbf{unl}\ B$
(Tb19) $\bigcirc(A\ \mathbf{unl}\ B) \leftrightarrow \bigcirc A\ \mathbf{unl}\ \bigcirc B$
(Tb20) $(A \land B)\ \mathbf{unl}\ C \leftrightarrow A\ \mathbf{unl}\ C \land B\ \mathbf{unl}\ C$
(Tb21) $A\ \mathbf{unl}\ (B \lor C) \leftrightarrow A\ \mathbf{unl}\ B \lor A\ \mathbf{unl}\ C$
(Tb22) $A\ \mathbf{unl}\ (B \land C) \rightarrow A\ \mathbf{unl}\ B \land A\ \mathbf{unl}\ C$
(Tb23) $A\ \mathbf{unl}\ (A\ \mathbf{unl}\ B) \leftrightarrow A\ \mathbf{unl}\ B$
(Tb24) $(A\ \mathbf{unl}\ B)\ \mathbf{unl}\ B \leftrightarrow A\ \mathbf{unl}\ B$
(Tb25) $\Box(B \rightarrow A) \rightarrow A\ \mathbf{atnext}\ B$
(Tb26) $\bigcirc(A\ \mathbf{atnext}\ B) \leftrightarrow \bigcirc A\ \mathbf{atnext}\ \bigcirc B$
(Tb27) $(A \land B)\ \mathbf{atnext}\ C \leftrightarrow A\ \mathbf{atnext}\ C \land B\ \mathbf{atnext}\ C$
(Tb28) $(A \lor B)\ \mathbf{atnext}\ C \leftrightarrow A\ \mathbf{atnext}\ C \lor B\ \mathbf{atnext}\ C$

(Tb29) A **atnext** $(B \lor C) \to A$ **atnext** $B \lor A$ **atnext** C

Laws for Fixpoint Operators in LTL

(Tμ1) $\Box A \ \leftrightarrow\ \nu u(A \land \bigcirc u)$
(Tμ2) $\Diamond A \ \leftrightarrow\ \mu u(A \lor \bigcirc u)$
(Tμ3) A **until** $B \ \leftrightarrow\ \mu u(\bigcirc B \lor \bigcirc(A \land u))$
(Tμ4) A **unless** $B \ \leftrightarrow\ \nu u(\bigcirc B \lor \bigcirc(A \land u))$
(Tμ5) A **unt** $B \ \leftrightarrow\ \mu u(B \lor (A \land \bigcirc u))$
(Tμ6) A **unl** $B \ \leftrightarrow\ \nu u(B \lor (A \land \bigcirc u))$
(Tμ7) A **atnext** $B \ \leftrightarrow\ \nu u(\bigcirc(B \to A) \land \bigcirc(\neg B \to u))$
(Tμ8) A **before** $B \ \leftrightarrow\ \nu u(\bigcirc\neg B \land \bigcirc(A \lor u))$

Laws for Propositional Quantification in LTL

(Tq1) $\forall u A \to A_u(B)$
(Tq2) $\forall u \bigcirc A \leftrightarrow \bigcirc \forall u A$
(Tq3) $\forall u \Box A \leftrightarrow \Box \forall u A$
(Tq4) $\exists u \Diamond A \leftrightarrow \Diamond \exists u A$
(Tq5) $\Box(A \lor B) \to \exists u \Box((A \land u) \lor (B \land \neg u))$

Laws for Past Operators in LTL

(Tp1) $\circledequal A \to \neg \ominus$ **false**
(Tp2) $\circledequal \neg A \to \neg \circledequal A$
(Tp3) $\neg \ominus A \leftrightarrow \circledequal \neg A$
(Tp4) $A \to \ominus \bigcirc A$
(Tp5) $A \to \bigcirc \circledequal A$
(Tp6) $\ominus(A \to B) \leftrightarrow \ominus A \to \ominus B$
(Tp7) $\ominus(A \land B) \leftrightarrow \ominus A \land \ominus B$
(Tp8) $\circledequal(A \land B) \leftrightarrow \circledequal A \land \circledequal B$

Laws of First-Order LTL

(T39) $\exists x \bigcirc A \leftrightarrow \bigcirc \exists x A$
(T40) $\forall x \bigcirc A \leftrightarrow \bigcirc \forall x A$
(T41) $\exists x \Diamond A \leftrightarrow \Diamond \exists x A$
(T42) $\forall x \Box A \leftrightarrow \Box \forall x A$
(Tb30) $\exists x(A$ **unl** $B) \leftrightarrow A$ **unl** $(\exists x B)$
if there is no free occurrence of x in A
(Tb31) $\forall x(A$ **unl** $B) \leftrightarrow (\forall x A)$ **unl** B
if there is no free occurrence of x in B
(Tb32) $\exists x(A$ **atnext** $B) \leftrightarrow (\exists x A)$ **atnext** B
if there is no free occurrence of x in B
(Tb33) $\forall x(A$ **atnext** $B) \leftrightarrow (\forall x A)$ **atnext** B
if there is no free occurrence of x in B

Derivation Rules of Linear Temporal Logic

(nex) $A \vdash \bigcirc A$

(alw) $A \vdash \Box A$

(ind) $A \rightarrow B, A \rightarrow \bigcirc A \vdash A \rightarrow \Box B$

(ind1) $A \rightarrow \bigcirc A \vdash A \rightarrow \Box A$

(ind2) $A \rightarrow B, B \rightarrow \bigcirc B \vdash A \rightarrow \Box B$

(som) $A \rightarrow \bigcirc B \vdash A \rightarrow \Diamond B$

(chain) $A \rightarrow \Diamond B, B \rightarrow \Diamond C \vdash A \rightarrow \Diamond C$

(indunless) $A \rightarrow \bigcirc C \vee \bigcirc(A \wedge B) \vdash A \rightarrow B$ **unless** C

(indunl) $A \rightarrow C \vee (B \wedge \bigcirc A) \vdash A \rightarrow B$ **unl** C

(indatnext) $A \rightarrow \bigcirc(C \rightarrow B) \wedge \bigcirc(\neg C \rightarrow A) \vdash A \rightarrow B$ **atnext** C

(indbefore) $A \rightarrow \bigcirc\neg C \wedge \bigcirc(A \vee B) \vdash A \rightarrow B$ **before** C

(μ-ind) $A_u(B) \rightarrow B \vdash \mu u A \rightarrow B$ if there is no free occurrence of u in B

(qltl-ind) $F \rightarrow \exists \mathbf{u}_2 \bigcirc((\mathbf{u}_2 \leftrightarrow \mathbf{u}_1) \wedge F_{\mathbf{u}_1}(\mathbf{u}_2))$
$\vdash F \rightarrow \exists \mathbf{u}_2((\mathbf{u}_2 \leftrightarrow \mathbf{u}_1) \wedge \Box F_{\mathbf{u}_1}(\mathbf{u}_2))$
if every occurrence of variables u_1^i in F is in the scope of at most one $\bigcirc$ operator and no other temporal operator

(indpast) $A \rightarrow B, A \rightarrow \ominus A \vdash A \rightarrow \boxminus B$

(indinit) **init** $\rightarrow A, A \rightarrow \bigcirc A \vdash A$

(wfr) $A \rightarrow \Diamond(B \vee \exists \bar{y}(\bar{y} \prec y \wedge A_y(\bar{y}))) \vdash \exists y A \rightarrow \Diamond B$
if B does not contain y,
for $y, \bar{y} \in \mathcal{X}_{WF}$

Laws of Generalized TLA

(GT1) $\Box\big[[A]_e \rightarrow A\big]_e$

(GT2) $\Box A \rightarrow \Box[\bigcirc A]_e$

(GT3) $\Box\big[[A]_e\big]_e \leftrightarrow \Box[A]_e$

(GT4) $\Box\big[\Box[A]_{e_1} \rightarrow [A]_{e_1}\big]_{e_2}$

(GT5) $\Box[A]_{e_1} \rightarrow \Box\big[[A]_{e_1}\big]_{e_2}$

(GT6) $\Box\big[[A]_{e_1}\big]_{e_2} \leftrightarrow \Box\big[[A]_{e_2}\big]_{e_1}$

Laws of Interval Temporal Logic

(IT1) **empty chop** $A \leftrightarrow A$

(IT2) $\odot A$ **chop** $B \leftrightarrow \odot(A$ **chop** $B)$

(IT3) $(A \vee B)$ **chop** $C \leftrightarrow A$ **chop** $C \vee B$ **chop** C

(IT4) A **chop** $(B \vee C) \leftrightarrow A$ **chop** $B \vee A$ **chop** C

(IT5) A **chop** $(B$ **chop** $C) \leftrightarrow (A$ **chop** $B)$ **chop** C

Laws of BTL and CTL

(BT1)	$\mathsf{E}\Box A \leftrightarrow A \land \mathsf{E}\bigcirc\mathsf{E}\Box A$
(BT2)	$\mathsf{E}\Diamond A \leftrightarrow A \lor \mathsf{E}\bigcirc\mathsf{E}\Diamond A$
(BT3)	$\mathsf{A}\Box A \leftrightarrow A \land \mathsf{A}\bigcirc\mathsf{A}\Box A$
(BT4)	$\mathsf{A}\Diamond A \leftrightarrow A \lor \mathsf{A}\bigcirc\mathsf{A}\Diamond A$
(BT5)	$\mathsf{A}\bigcirc A \rightarrow \mathsf{E}\bigcirc A$
(BT6)	$\mathsf{E}\Box A \rightarrow \mathsf{E}\bigcirc A$
(BT7)	$\mathsf{E}\Box\mathsf{E}\Box A \leftrightarrow \mathsf{E}\Box A$
(BT8)	$\mathsf{E}\bigcirc\mathsf{E}\Box A \rightarrow \mathsf{E}\Box\mathsf{E}\bigcirc A$
(BT9)	$\mathsf{E}\bigcirc(A \land B) \rightarrow \mathsf{E}\bigcirc A \land \mathsf{E}\bigcirc B$
(BT10)	$\mathsf{E}\bigcirc(A \rightarrow B) \leftrightarrow \mathsf{A}\bigcirc A \rightarrow \mathsf{E}\bigcirc B$
(BT11)	$\mathsf{E}\Diamond(A \lor B) \leftrightarrow \mathsf{E}\Diamond A \lor \mathsf{E}\Diamond B$
(BT12)	$\mathsf{E}\Box(A \land B) \rightarrow \mathsf{E}\Box A \land \mathsf{E}\Box B$
(CT1)	$A \ \mathbf{Eunt}\ B \leftrightarrow B \lor (A \land \mathsf{E}\bigcirc(A \ \mathbf{Eunt}\ B))$
(CT2)	$A \ \mathbf{Eunt}\ B \rightarrow \mathsf{E}\Diamond B$
(CT3)	$\mathsf{E}\bigcirc(A \ \mathbf{Eunt}\ B) \leftrightarrow \mathsf{E}\bigcirc A \ \mathbf{Eunt}\ \mathsf{E}\bigcirc B$
(CT4)	$A \ \mathbf{Eunt}\ C \lor B \ \mathbf{Eunt}\ C \rightarrow (A \lor B) \ \mathbf{Eunt}\ C$
(CT5)	$(A \land B) \ \mathbf{Eunt}\ C \rightarrow A \ \mathbf{Eunt}\ C \land B \ \mathbf{Eunt}\ C$
(CT6)	$A \ \mathbf{Eunt}\ (B \lor C) \leftrightarrow A \ \mathbf{Eunt}\ B \lor A \ \mathbf{Eunt}\ C$
(CT7)	$A \ \mathbf{Eunt}\ (B \land C) \rightarrow A \ \mathbf{Eunt}\ B \land A \ \mathbf{Eunt}\ C$

Derivation Rules of Branching Time Temporal Logic

(nexb)	$A \rightarrow B \vdash \mathsf{E}\bigcirc A \rightarrow \mathsf{E}\bigcirc B$
(indb1)	$A \rightarrow B, A \rightarrow \mathsf{E}\bigcirc A \vdash A \rightarrow \mathsf{E}\Box B$
(indb2)	$A \rightarrow \neg B, A \rightarrow \mathsf{A}\bigcirc(A \lor \neg\mathsf{E}\Diamond B) \vdash A \rightarrow \neg\mathsf{E}\Diamond B$
(indc)	$A \rightarrow \neg C, A \rightarrow \mathsf{A}\bigcirc(A \lor \neg(B \ \mathbf{Eunt}\ C)) \vdash A \rightarrow \neg(B \ \mathbf{Eunt}\ C)$

The Formal System Σ_{LTL}

(taut)	All tautologically valid formulas
(ltl1)	$\neg\bigcirc A \leftrightarrow \bigcirc\neg A$
(ltl2)	$\bigcirc(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B)$
(ltl3)	$\Box A \rightarrow A \land \bigcirc\Box A$
(mp)	$A, A \rightarrow B \vdash B$
(nex)	$A \vdash \bigcirc A$
(ind)	$A \rightarrow B, A \rightarrow \bigcirc A \vdash A \rightarrow \Box B$

Additional Axioms and Rules for Extensions of LTL

(until1)	$A \ \mathbf{until}\ B \leftrightarrow \bigcirc B \lor \bigcirc(A \land A \ \mathbf{until}\ B)$
(until2)	$A \ \mathbf{until}\ B \rightarrow \bigcirc\Diamond B$
(unless1)	$A \ \mathbf{unless}\ B \leftrightarrow \bigcirc B \lor \bigcirc(A \land A \ \mathbf{unless}\ B)$

(unless2) $\bigcirc\Box A \to A$ **unless** B

(atnext1) A **atnext** $B \leftrightarrow \bigcirc(B \to A) \land \bigcirc(\neg B \to A$ **atnext** $B)$
(atnext2) $\bigcirc\Box\neg B \to A$ **atnext** B

(before1) A **before** $B \leftrightarrow \bigcirc\neg B \land \bigcirc(A \lor A$ **before** $B)$
(before2) $\bigcirc\Box\neg B \to A$ **before** B

(μ-rec) $A_u(\mu u A) \to \mu u A$
(μ-ind) $A_u(B) \to B \vdash \mu u A \to B$ if there is no free occurrence of u in B

(qltl1) $A_u(B) \to \exists u A$
(qltl2) $\exists u \bigcirc A \leftrightarrow \bigcirc \exists u A$
(qltl3) $\exists u(u \land \bigcirc\Box\neg u)$
(qltl-part) $A \to B \vdash \exists u A \to B$ if there is no free occurrence of u in B
(qltl-ind) $F \to \exists \mathbf{u}_2 \bigcirc((\mathbf{u}_2 \leftrightarrow \mathbf{u}_1) \land F_{\mathbf{u}_1}(\mathbf{u}_2))$
$\vdash F \to \exists \mathbf{u}_2((\mathbf{u}_2 \leftrightarrow \mathbf{u}_1) \land \Box F_{\mathbf{u}_1}(\mathbf{u}_2))$
if every occurrence of variables u_1^i in F is in the scope of at most one $\bigcirc$ operator and no other temporal operator

(pltl1) $\circleddash\!\!\!\!=\,\neg A \to \neg \circleddash\!\!\!\!=\, A$
(pltl2) $\ominus(A \to B) \to (\ominus A \to \ominus B)$
(pltl3) $\boxminus A \to A \land \ominus\boxminus A$
(pltl4) $\diamondminus\ominus$ **false**
(pltl5) $A \to \ominus\bigcirc A$
(pltl6) $A \to \bigcirc\circleddash\!\!\!\!=\, A$
(prev) $A \vdash \ominus A$
(indpast) $A \to B, A \to \ominus A \vdash A \to \boxminus B$

(iltl) $\bigcirc\neg$**init**
(init) **init** $\to \Box A \vdash A$

(since) A **since** $B \leftrightarrow \circleddash\!\!\!\!=\, B \lor \circleddash\!\!\!\!=\,(A \land A$ **since** $B)$

(backto) A **backto** $B \leftrightarrow \ominus B \lor \ominus(A \land A$ **backto** $B)$

(atlast) A **atlast** $B \leftrightarrow \ominus(B \to A) \land \ominus(\neg B \to A$ **atlast** $B)$

(after) A **after** $B \leftrightarrow \ominus\neg B \land \ominus(A \lor A$ **after** $B)$

The Formal System $\Sigma_{\mathbf{FOLTL}}$

(taut) All tautologically valid formulas
(ltl1) $\neg\bigcirc A \leftrightarrow \bigcirc\neg A$
(ltl2) $\bigcirc(A \to B) \to (\bigcirc A \to \bigcirc B)$
(ltl3) $\Box A \to A \land \bigcirc\Box A$
(ltl4) $A_x(t) \to \exists x A$ if t is substitutable for x in A
(ltl5) $\bigcirc\exists x A \to \exists x \bigcirc A$
(ltl6) $A \to \bigcirc A$ if A is rigid
(eq1) $x = x$
(eq2) $x = y \to (A \to A_x(y))$ if A is non-temporal
(mp) $A, A \to B \vdash B$

(nex) $A \vdash \bigcirc A$
(ind) $A \rightarrow B, A \rightarrow \bigcirc A \vdash A \rightarrow \Box B$
(par) $A \rightarrow B \vdash \exists x A \rightarrow B$ if there is no free occurrence of x in B

The Formal System $\Sigma_{\mathbf{pGTLA}}$

(taut) All tautologically valid formulas
(taut$_{pf}$) $\Box[A]_e$ if A is a tautologically valid pre-formula
(gtla1) $\Box A \rightarrow A$
(gtla2) $\Box A \rightarrow \Box[A]_e$
(gtla3) $\Box A \rightarrow \Box[\bigcirc\Box A]_e$
(gtla4) $\Box[A \rightarrow B]_e \rightarrow (\Box[A]_e \rightarrow \Box[B]_e)$
(gtla5) $\Box[e' \neq e]_e$
(gtla6) $\Box[\neg\bigcirc A \leftrightarrow \bigcirc\neg A]_e$
(gtla7) $\Box[\bigcirc(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B)]_e$
(gtla8) $\Box\big[\Box[A]_{e_1} \rightarrow [A]_{e_1}\big]_{e_2}$
(gtla9) $\Box[A]_{e_1} \rightarrow \Box\big[\bigcirc\Box[A]_{e_1}\big]_{e_2}$
(gtla10) $\Box\big[[A]_{e_1} \wedge \bigcirc\Box[A]_{e_1} \rightarrow \Box[A]_{e_1}\big]_{e_2}$
(gtla11) $\Box\big[\bigcirc\Box A \rightarrow \Box[\bigcirc A]_{e_1}\big]_{e_2}$
(mp) $A, A \rightarrow B \vdash B$
(alw) $A \vdash \Box A$
(ind$_{pf}$) $A \rightarrow B, \Box[A \rightarrow \bigcirc A]_{\mathsf{U}(A)} \vdash A \rightarrow \Box B$

The Formal System $\Sigma_{\mathbf{BTL}}$

(taut) All tautologically valid formulas
(btl1) $\mathsf{E}\bigcirc\mathbf{true}$
(btl2) $\mathsf{E}\bigcirc(A \vee B) \leftrightarrow \mathsf{E}\bigcirc A \vee \mathsf{E}\bigcirc B$
(btl3) $\mathsf{E}\Box A \leftrightarrow A \wedge \mathsf{E}\bigcirc\mathsf{E}\Box A$
(btl4) $\mathsf{E}\Diamond A \leftrightarrow A \vee \mathsf{E}\bigcirc\mathsf{E}\Diamond A$
(mp) $A, A \rightarrow B \vdash B$
(nexb) $A \rightarrow B \vdash \mathsf{E}\bigcirc A \rightarrow \mathsf{E}\bigcirc B$
(indb1) $A \rightarrow B, A \rightarrow \mathsf{E}\bigcirc A \vdash A \rightarrow \mathsf{E}\Box B$
(indb2) $A \rightarrow \neg B, A \rightarrow \mathsf{A}\bigcirc(A \vee \neg\mathsf{E}\Diamond B) \vdash A \rightarrow \neg\mathsf{E}\Diamond B$

The Formal System $\Sigma_{\mathbf{CTL}}$

(taut) All tautologically valid formulas
(btl1) $\mathsf{E}\bigcirc\mathbf{true}$
(btl2) $\mathsf{E}\bigcirc(A \vee B) \leftrightarrow \mathsf{E}\bigcirc A \vee \mathsf{E}\bigcirc B$
(btl3) $\mathsf{E}\Box A \leftrightarrow A \wedge \mathsf{E}\bigcirc\mathsf{E}\Box A$
(ctl) $A\ \mathbf{Eunt}\ B \leftrightarrow B \vee (A \wedge \mathsf{E}\bigcirc(A\ \mathbf{Eunt}\ B))$
(mp) $A, A \rightarrow B \vdash B$

(nexb) $A \to B \vdash \mathsf{E}\bigcirc A \to \mathsf{E}\bigcirc B$
(indb1) $A \to B, A \to \mathsf{E}\bigcirc A \vdash A \to \mathsf{E}\Box B$
(indc) $A \to \neg C, A \to \mathsf{A}\bigcirc(A \lor \neg(B \text{ } \mathbf{\mathsf{E}unt} \text{ } C)) \vdash A \to \neg(B \text{ } \mathbf{\mathsf{E}unt} \text{ } C)$

References

1. Abadi, M. The power of temporal proofs. *Theoretical Computer Science 65*, 1 (June 1989), pp. 35–84. See Corrigendum in TCS 70 (1990), p. 275.
2. Abadi, M. and Manna, Z. Temporal logic programming. In *Symp. Logic Programming* (San Francisco, California, 1987), IEEE Computer Society, pp. 4–16.
3. Abrial, J.-R. *The B-Book: Assigning Programs to Meanings*. Cambridge University Press, Cambridge, UK, 1996.
4. Alpern, B. and Schneider, F. B. Defining liveness. *Information Processing Letters 21*, 4 (1985), pp. 181–185.
5. Alpern, B. and Schneider, F. B. Recognizing safety and liveness. *Distributed Computing 2* (1987), pp. 117–126.
6. Alur, R. and Henzinger, T. A. A really temporal logic. *Journal of the ACM 41* (1994), pp. 181–204.
7. Andrews, G. R. *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, 2000.
8. Apt, K. R. and Olderog, E.-R. *Verification of sequential and concurrent programs*. Springer, New York, 1991.
9. Back, R. and von Wright, J. *Refinement calculus – A systematic introduction*. Springer, New York, 1998.
10. Ball, T. and Rajamani, S. K. The SLAM project: Debugging system software via static analysis. In *29th Ann. Symp. Principles of Programming Languages* (Portland, Oregon, 2002), pp. 1–3.
11. Banieqbal, B. and Barringer, H. Temporal logic with fixpoints. In *Temporal Logic in Specification* (Altrincham, UK, 1987), B. Banieqbal, H. Barringer, and A. Pnueli, Eds., vol. 398 of *Lecture Notes in Computer Science*, Springer, pp. 62–74.
12. Barringer, H., Fisher, M., Gabbay, D. M., Gough, G., and Owens, R. METATEM: A framework for programming in temporal logic. In *Stepwise Refinement of Distributed Systems* (Mook, The Netherlands, 1989), J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, Eds., vol. 430 of *Lecture Notes in Computer Science*, Springer, pp. 94–129.
13. Bartlett, K. A., Scantlebury, R. A., and Wilkinson, P. T. A note on reliable full-duplex transmission over half-duplex links. *Communications of the ACM 12* (1969), pp. 260–261.
14. Ben-Ari, M. *Principles of Concurrent and Distributed Programming*, 2nd ed. Addison-Wesley, Harlow, UK, 2006.

15. Ben-Ari, M., Pnueli, A., and Manna, Z. The temporal logic of branching time. *Acta Informatica 20* (1983), pp. 207–226.
16. Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., and Schnoebelen, P. *Systems and Software Verification. Model-Checking Techniques and Tools*. Springer, 2001.
17. Bhat, G. and Peled, D. Adding partial orders to linear temporal logic. *Fundamenta Informaticae 36*, 1 (1998), pp. 1–21.
18. Biere, A., Cimatti, A., Clarke, E., Strichman, O., and Zhu, Y. Bounded model checking. In *Highly Dependable Software*, vol. 58 of *Advances in Computers*. Academic Press, 2003.
19. Bjørner, D. and Jones, C. B. *Formal Specification and Software Development*. Prentice Hall, 1982.
20. Bjørner, N., Browne, A., Colon, M., Finkbeiner, B., Manna, Z., Sipma, H., and Uribe, T. Verifying temporal properties of reactive systems: A STeP tutorial. *Formal Methods in System Design 16* (2000), pp. 227–270.
21. Blackburn, P., de Rijke, M., and Venema, Y. *Modal Logic*, vol. 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 2001.
22. Bowman, H. and Thompson, S. A decision procedure and complete axiomatization of finite interval temporal logic with projection. *Journal of Logic and Computation 13* (2003), pp. 195–239.
23. Bryant, R. E. Symbolic boolean manipulations with ordered binary decision diagrams. *ACM Computing Surveys 24*, 3 (1992), pp. 293–317.
24. Büchi, J. R. On a decision method in restricted second-order arithmetics. In *Intl. Cong. Logic, Method and Philosophy of Science* (1962), Stanford University Press, pp. 1–12.
25. Burstall, M. Program proving as hand simulation with a little induction. In *IFIP Congress 1974* (Stockholm, Sweden, 1974), North-Holland, pp. 308–312.
26. Caires, L. and Cardelli, L. A spatial logic for concurrency (part I). *Information and Computation 186*, 2 (2003), pp. 194–235.
27. Cansell, D., Méry, D., and Merz, S. Diagram refinements for the design of reactive systems. *Journal of Universal Computer Science 7*, 2 (2001), pp. 159–174.
28. Chandy, K. M. and Misra, J. *Parallel Program Design: A Foundation*. Addison-Wesley, 1988.
29. Chomicki, J. and Toman, D. Temporal logic in information systems. BRICS Lecture Series LS-97-1, Department of Computer Science, University of Aarhus, 1997.
30. Clarke, E. M. and Emerson, E. A. Synthesis of synchronization skeletons for branching time temporal logic. In *Workshop Logic of Programs* (Yorktown Heights, N.Y., 1981), D. Kozen, Ed., vol. 131 of *Lecture Notes in Computer Science*, Springer, pp. 52–71.
31. Clarke, E. M., Grumberg, O., Jha, S., Lu, Y., and Veith, H. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM 50*, 5 (2003), pp. 752–794.
32. Clarke, E. M., Grumberg, O., and Long, D. E. Model checking and abstraction. *ACM Trans. Program. Lang. Syst. 16*, 5 (1994), pp. 1512–1542.
33. Clarke, E. M., Grumberg, O., and Peled, D. *Model Checking*. MIT Press, Cambridge, Mass., 1999.
34. Clarke, E. M., Jha, S., Enders, R., and Filkorn, T. Exploiting symmetry in temporal logic model checking. *Formal Methods in System Design 9*, 1-2 (1996), pp. 77–104.

35. Clarke, E. M. and Schlingloff, H. Model checking. In *Handbook of Automated Deduction*, A. Robinson and A. Voronkov, Eds., vol. II. Elsevier Science, 2000, pp. 1635–1790.
36. Clifford, J. Tense logic and the logic of change. *Logique et Analyse 34* (1966), pp. 219–230.
37. Dams, D., Grumberg, O., and Gerth, R. Abstract interpretation of reactive systems: Abstractions preserving $\forall$CTL*, $\exists$CTL* and CTL*. In *IFIP Work. Conf. Programming Concepts, Methods, and Calculi* (Amsterdam, The Netherlands, 1994), E.-R. Olderog, Ed., Elsevier Science, pp. 561–581.
38. de Alfaro, L., Manna, Z., Sipma, H. B., and Uribe, T. Visual verification of reactive systems. In *Third Intl. Workshop Tools and Algorithms for the Construction and Analysis of Systems* (Enschede, The Netherlands, 1997), E. Brinksma, Ed., vol. 1217 of *Lecture Notes in Computer Science*, Springer, pp. 334–350.
39. de Roever, W.-P., de Boer, F., Hannemann, U., Hooman, J., Lakhnech, Y., Poel, M., and Zwiers, J. *Concurrency Verification: Introduction to Compositional and Noncompositional Methods*. Cambridge University Press, Cambridge, UK, 2001.
40. de Roever, W.-P., Langmaack, H., and Pnueli, A., Eds. *Compositionality: The Significant Difference* (1998), vol. 1536 of *Lecture Notes in Computer Science*, Springer.
41. Dijkstra, E. W. Self-stabilizing systems in spite of distributed control. *Communications of the ACM 17*, 11 (1974), pp. 643–644.
42. Dijkstra, E. W. *A Discipline of Programming*. Prentice Hall, 1976.
43. Ebbinghaus, H., Flum, J., and Thomas, W. *Einführung in die Mathematische Logik*. Wissenschaftliche Buchgesellschaft, Darmstadt, Germany, 1978.
44. Emerson, E. A. Alternative semantics for temporal logics. *Theoretical Computer Science 26* (1983), pp. 121–130.
45. Emerson, E. A. Temporal and modal logic. In *Handbook of theoretical computer science*, J. van Leeuwen, Ed., vol. B: Formal Models and Semantics. Elsevier, 1990, pp. 997–1071.
46. Emerson, E. A. and Sistla, A. P. Symmetry and model checking. *Formal Methods in System Design 9*, 1-2 (1996), pp. 105–131.
47. Esparza, J. Model checking using net unfoldings. *Science of Computer Programming 23* (1994), pp. 151–195.
48. Floyd, R. Assigning meaning to programs. In *Symposium on Applied Mathematics 19, Mathematical Aspects of Computer Science* (New York, 1967), J. T. Schwartz, Ed., American Mathematical Society, pp. 19–32.
49. Francez, N. *Fairness*. Springer, New York, 1986.
50. Frege, G. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Louis Nebert, Halle, Germany, 1879.
51. French, T. and Reynolds, M. A sound and complete proof system for QPTL. *Advances in Modal Logic 4* (2002), pp. 1–20.
52. Gabbay, D. M., Hodkinson, I., and Reynolds, M. *Temporal Logic: Mathematical Foundations and Computational Aspects*, vol. 1. Clarendon Press, Oxford, UK, 1994.
53. Gabbay, D. M., Pnueli, A., Shelah, S., and Stavi, J. On the temporal analysis of fairness. In *7th Ann. Symp. Principles of Programming Languages* (Las Vegas, Nevada, 1980), pp. 163–173.
54. Gastin, P. and Oddoux, D. Fast LTL to Büchi automata translation. In *13th Intl. Conf. Computer Aided Verification* (Paris, France, 2001), G. Berry, H. Comon, and A. Finkel, Eds., vol. 2102 of *Lecture Notes in Computer Science*, Springer, pp. 53–65.

55. GERTH, R., PELED, D., VARDI, M. Y., AND WOLPER, P. Simple on-the-fly automatic verification of linear temporal logic. In *Protocol Specification, Testing, and Verification* (Warsaw, Poland, 1995), Chapman & Hall, pp. 3–18.
56. GODEFROID, P. AND WOLPER, P. A partial approach to model checking. *Information and Computation 110*, 2 (1994), pp. 305–326.
57. GÖDEL, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme. *Monatshefte für Mathematik und Physik 38* (1931), pp. 173–198.
58. GOLDBLATT, R. *Logics of Time and Computation*, vol. 7 of *CSLI Lecture Notes*. CSLI, Stanford, California, 1987.
59. GRAF, S. AND SAÏDI, H. Construction of abstract state graphs with PVS. In *9th Intl. Conf. Computer Aided Verification* (Haifa, Israel, 1997), O. Grumberg, Ed., vol. 1254 of *Lecture Notes in Computer Science*, Springer, pp. 72–83.
60. HAMILTON, A. G. *Logic for Mathematicians*, revised ed. Cambridge University Press, Cambridge, UK, 1988.
61. HENZINGER, T. A., JHALA, R., MAJUMDAR, R., AND MCMILLAN, K. L. Abstractions from proofs. In *31st Symp. Principles of Programming Languages* (Venice, Italy, 2004), ACM Press, pp. 232–244.
62. HOARE, C. A. R. An axiomatic basis for computer programming. *Communications of the ACM 12* (1969), pp. 576–580.
63. HODKINSON, I., WOLTER, F., AND ZAKHARYASCHEV, M. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic 106* (2000), pp. 85–134.
64. HOLZMANN, G. AND PELED, D. An improvement in formal verification. In *IFIP Conf. Formal Description Techniques* (Bern, Switzerland, 1994), Chapman & Hall, pp. 197–214.
65. HOLZMANN, G. J. *The SPIN Model Checker*. Addison-Wesley, 2003.
66. HUGHES, G. E. AND CRESSWELL, M. J. *An Introduction to Modal Logic*. Methuen, London, UK, 1968.
67. HUTH, M. AND RYAN, M. D. *Logic in Computer Science: Modelling and Reasoning about Systems*, 2nd ed. Cambridge University Press, Cambridge, UK, 2004.
68. IP, C. N. AND DILL, D. L. Better verification through symmetry. *Formal Methods in System Design 9*, 1-2 (1996), pp. 41–75.
69. KAMINSKI, M. Invariance under stuttering in a temporal logic of actions. *Theoretical Computer Science 368*, 1-2 (2006), pp. 50–63.
70. KAMP, H. W. *Tense logic and the theory of linear order*. PhD thesis, UCLA, Los Angeles, California, 1968.
71. KELLER, R. M. Formal verification of parallel programs. *Communications of the ACM 19* (1976), pp. 371–384.
72. KESTEN, Y. AND PNUELI, A. Verification by augmented finitary abstraction. *Information and Computation 163*, 1 (2000), pp. 203–243.
73. KESTEN, Y. AND PNUELI, A. Complete proof system for QPTL. *Journal of Logic and Computation 12*, 5 (2002), pp. 701–745.
74. KNAPP, A., MERZ, S., WIRSING, M., AND ZAPPE, J. Specification and refinement of mobile systems in MTLA and Mobile UML. *Theoretical Computer Science 351*, 2 (2006), pp. 184–202.
75. KOZEN, D. Results on the propositional mu-calculus. *Theoretical Computer Science 27* (1983), pp. 333–354.
76. KRIPKE, S. A. Semantical analysis of modal logic I. *Z. Math. Logik Grundlagen Math. 9* (1963), pp. 67–96.
77. KRÖGER, F. Logical rules of natural reasoning about programs. In *Intl. Coll. Automata, Logic and Programming* (Edinburgh, UK, 1976), Edinburgh University Press, pp. 87–98.

78. KRÖGER, F. LAR: A logic of algorithmic reasoning. *Acta Informatica 8* (1977), pp. 243–266.
79. KRÖGER, F. A uniform logical basis for the description, specification and verification of programs. In *IFIP Work. Conf. Formal Description of Programming Concepts* (St. Andrews, Canada, 1978), North-Holland, pp. 441–457.
80. KRÖGER, F. On temporal program verification rules. *RAIRO Informatique Théorique et Applications 19* (1985), pp. 261–280.
81. KRÖGER, F. On the interpretability of arithmetic in temporal logic. *Theoretical Computer Science 73* (1990), pp. 47–60.
82. KRÖGER, F. A generalized nexttime operator in temporal logic. *Journal of Computer and Systems Sciences 29* (1984), pp. 80–98.
83. KRÖGER, F. *Temporal Logic of Programs*. Springer, Berlin-Heidelberg, 1987.
84. KUPFERMAN, O. AND VARDI, M. Y. Weak alternating automata are not so weak. In *5th Israeli Symp. Theory of Computing and Systems* (1997), IEEE Computer Society, pp. 147–158.
85. KUPFERMAN, O. AND VARDI, M. Y. Complementation constructions for nondeterministic automata on infinite words. In *11th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems* (Edinburgh, UK, 2005), N. Halbwachs and L. Zuck, Eds., vol. 3440 of *Lecture Notes in Computer Science*, Springer, pp. 206–221.
86. LAMPORT, L. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering SE-3(2)* (1977), pp. 125–143.
87. LAMPORT, L. 'Sometime' is sometimes 'not never'. In *7th Ann. Symp. Principles of Programming Languages* (Las Vegas, Nevada, January 1980), ACM Press, pp. 174–185.
88. LAMPORT, L. The Temporal Logic of Actions. *ACM Trans. Program. Lang. Syst. 16*, 3 (1994), pp. 872–923.
89. LANGE, M. *Temporal Logics Beyond Regularity*. Habilitationsschrift, Ludwig-Maximilians-Universität München, Munich, Germany, 2007.
90. LEMMON, E. J. AND SCOTT, D. *An Introduction to Modal Logic*, vol. 11 of *American Philosophical Quarterly Monograph Series*. Basil Blackwell, Oxford, UK, 1977. edited by K. Segerberg.
91. LESSKE, F. Constructive specifications of abstract data types using temporal logic. In *2nd Intl. Symp. Logical Foundations of Computer Science* (Tver, Russia, 1992), A. Nerode and M. A. Taitslin, Eds., vol. 620 of *Lecture Notes in Computer Science*, Springer, pp. 269–280.
92. LICHTENSTEIN, O., PNUELI, A., AND ZUCK, L. The glory of the past. In *Logics of Programs* (Brooklyn College, New York, 1985), R. Parikh, Ed., vol. 193 of *Lecture Notes in Computer Science*, Springer, pp. 196–218.
93. LIPECK, U. W. AND SAAKE, G. Monitoring dynamic integrity constraints based on temporal logic. *Information Systems 12* (1987), pp. 255–269.
94. LOISEAUX, C., GRAF, S., SIFAKIS, J., BOUAJJANI, A., AND BENSALEM, S. Property preserving abstractions for the verification of concurrent systems. *Formal Methods in System Design 6* (1995), pp. 11–44.
95. MANNA, Z. AND PNUELI, A. Verification of concurrent programs: Temporal proof principles. In *Workshop Logic of Programs* (Yorktown Heights, New York, 1981), D. Kozen, Ed., vol. 131 of *Lecture Notes in Computer Science*, Springer, pp. 200–252.
96. MANNA, Z. AND PNUELI, A. Verification of concurrent programs: The temporal framework. In *The correctness problem in computer science*, R. S. Boyer and J. S. Moore, Eds. Academic Press, 1982, pp. 215–273.

97. MANNA, Z. AND PNUELI, A. How to cook a temporal proof system for your pet language. In *10th Ann. Symp. Principles of Programming Languages* (Austin, Texas, 1983), pp. 141–154.
98. MANNA, Z. AND PNUELI, A. Proving precedence properties: The temporal way. In *10th Intl. Coll. Automata, Languages and Programming* (Barcelona, Spain, 1983), J. Diaz, Ed., vol. 154 of *Lecture Notes in Computer Science*, Springer, pp. 491–512.
99. MANNA, Z. AND PNUELI, A. Verification of concurrent programs: A temporal proof system. In *Foundations of computer science IV*, vol. 159 of *Mathematical Centre Tracts*. CWI, Amsterdam, 1983, pp. 163–255.
100. MANNA, Z. AND PNUELI, A. The anchored version of the temporal framework. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, Eds., vol. 354 of *Lecture Notes in Computer Science*. Springer, 1989, pp. 201–284.
101. MANNA, Z. AND PNUELI, A. A hierarchy of temporal properties. In *9th Symp. Principles of Distributed Programming* (Vancouver, Canada, 1990), pp. 377–408.
102. MANNA, Z. AND PNUELI, A. *The Temporal Logic of Reactive and Concurrent Systems – Specification*. Springer, New York, 1992.
103. MANNA, Z. AND PNUELI, A. Temporal verification diagrams. In *Intl. Conf. Theoretical Aspects of Computer Software* (Sendai, Japan, 1994), M. Hagiya and J. C. Mitchell, Eds., vol. 789 of *Lecture Notes in Computer Science*, Springer, pp. 726–765.
104. MANNA, Z. AND PNUELI, A. *The Temporal Logic of Reactive and Concurrent Systems – Safety*. Springer, New York, 1995.
105. MANZANO, M. *Extensions of First Order Logic*, vol. 19 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1996.
106. MCMILLAN, K. L. *Symbolic Model Checking*. Kluwer Academic Publishers, 1993.
107. MERZ, S. Decidability and incompleteness results for first-order temporal logics of linear time. *Journal of Applied Non-Classical Logic 2*, 2 (1992).
108. MERZ, S. Efficiently executable temporal logic programs. In *Executable Modal and Temporal Logics* (Chambéry, France, 1995), M. Fisher and R. Owens, Eds., vol. 897 of *Lecture Notes in Computer Science*, Springer, pp. 69–85.
109. MERZ, S. A more complete TLA. In *World Cong. Formal Methods* (Toulouse, France, 1999), J. M. Wing, J. Woodcock, and J. Davies, Eds., vol. 1709 of *Lecture Notes in Computer Science*, Springer, pp. 1226–1244.
110. MOSZKOWSKI, B. C. *Executing Temporal Logic*. Cambridge University Press, Cambridge, UK, 1986.
111. MOSZKOWSKI, B. C. A complete axiomatization of interval temporal logic with infinite time. In *15th Ann. Symp. Logics in Computer Science* (Santa Barbara, California, 2000), IEEE Computer Society, pp. 241–252.
112. MULLER, D. E., SAOUDI, A., AND SCHUPP, P. E. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *3rd IEEE Symp. on Logic in Computer Science* (Edinburgh, UK, 1988), IEEE Press, pp. 422–427.
113. NIEBERT, P. A ν-calculus with local views for systems of sequential agents. In *20th Intl. Symp. Mathematical Foundations of Computer Science* (Prague, Czech Republic, 1995), J. Wiedermann and P. Hájek, Eds., vol. 969 of *Lecture Notes in Computer Science*, Springer, pp. 563–573.
114. ORGUN, M. A. AND MA, W. An overview of temporal and modal logic programming. In *First Intl. Conf. Temporal Logic* (Bonn, Germany, 1994), D. M. Gabbay and H. J. Ohlbach, Eds., vol. 827 of *Lecture Notes in Computer Science*, Springer, pp. 445–479.

115. PELED, D. Combining partial order reductions with on-the-fly model-checking. *Formal Methods in System Design 8*, 1 (1996), pp. 39–64.
116. PELED, D. AND WILKE, T. Stutter-invariant temporal properties are expressible without the next-time operator. *Information Processing Letters 63*, 5 (1997), pp. 243–246.
117. PENCZEK, W. Branching time and partial order in temporal logics. In *Time and Logic – A Computational Approach*, L. Bolc and A. Szalas, Eds. UCL Press, London, 1994, pp. 179–228.
118. PETERSON, G. L. Myths about the mutual exclusion problem. *Information Processing Letters 12* (1981), pp. 115–116.
119. PETRI, C. A. *Kommunikation mit Automaten*. Schriften des Institutes für Instrumentelle Mathematik, Bonn, Germany, 1962.
120. PNUELI, A. The temporal logic of programs. In *18th Ann. Symp. Foundations of Computer Science* (Providence, Rhode Island, 1977), IEEE, pp. 46–57.
121. PNUELI, A. The temporal semantics of concurrent programs. *Theoretical Computer Science 13* (1981), pp. 45–60.
122. PNUELI, A. In transition from global to modular temporal reasoning about programs. In *Logics and Models of Concurrent Systems*, K. R. Apt, Ed., vol. 193 of *Lecture Notes in Computer Science*. Springer, 1985, pp. 123–144.
123. PNUELI, A. System specification and refinement in temporal logic. In *Foundations of Software Technology and Theoretical Computer Science* (New Delhi, India, 1992), R. K. Shyamasundar, Ed., vol. 652 of *Lecture Notes in Computer Science*, Springer, pp. 1–38.
124. PRATT, V. R. A decidable μ-calculus: Preliminary report. In *22nd Ann. Symp. Foundations of Computer Science* (Nashville, Tennessee, 1981), IEEE Computer Society, pp. 421–427.
125. PRIOR, A. N. *Time and modality*. Oxford University Press, Oxford, UK, 1957.
126. PRIOR, A. N. *Past, Present and Future*. Oxford University Press, Oxford, UK, 1967.
127. QUEILLE, J. P. AND SIFAKIS, J. Specification and verification of concurrent systems in Cesar. In *5th Intl. Symp. Programming* (Torino, Italy, 1981), vol. 137 of *Lecture Notes in Computer Science*, Springer, pp. 337–351.
128. QUEILLE, J. P. AND SIFAKIS, J. Fairness and related properties in transition systems – a temporal logic to deal with fairness. *Acta Informatica 19* (1983), pp. 195–220.
129. REISIG, W. *Petri Nets: An Introduction*. Springer, Berlin-Heidelberg, 1985.
130. RESCHER, N. AND URQUHART, A. *Temporal Logic*. Springer, New York, 1971.
131. SAFRA, S. On the complexity of ω-automata. In *29th IEEE Symp. Foundations of Computer Science* (White Plains, New York, 1988), IEEE Computer Society, pp. 319–327.
132. SCHLINGLOFF, H. *Beweistheoretische Untersuchungen zur temporalen Logik*. Diplomarbeit, Technische Universität München, Institut für Informatik, Munich, Germany, 1983.
133. SCHNEIDER, F. B. *On Concurrent Programming*. Springer, New York, 1997.
134. SCHNEIDER, K. *Verification of Reactive Systems*. Springer, New York, 2004.
135. SCHOBBENS, P.-Y., RASKIN, J.-F., AND HENZINGER, T. A. Axioms for real-time logics. *Theoretical Computer Science 274*, 1-2 (2002), pp. 151–182.
136. SEGERBERG, C. On the logic of tomorrow. *Theoria 33* (1967), pp. 45–52.
137. SHOENFIELD, J. R. *Mathematical Logic*. Addison-Wesley, Reading, Mass., 1967.
138. SISTLA, A. P. *Theoretical issues in the design of distributed and concurrent systems*. PhD thesis, Harvard Univ., Cambridge, MA, 1983.
139. SISTLA, A. P., VARDI, M. Y., AND WOLPER, P. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science 49* (1987), pp. 217–237.

140. STIRLING, C. *Modal and Temporal Properties of Processes*. Springer, New York, 2001.
141. SZALAS, A. Concerning the semantic consequence relation in first-order temporal logic. *Theoretical Computer Science 47* (1986), pp. 329–334.
142. SZALAS, A. Arithmetical axiomatization of first-order temporal logic. *Information Processing Letters 26* (1987), pp. 111–116.
143. SZALAS, A. A complete axiomatic characterization of first-order temporal logic of linear time. *Theoretical Computer Science* (1987), pp. 199–214.
144. SZALAS, A. Towards the temporal approach to abstract data types. *Fundamenta Informaticae 11* (1988), pp. 49–64.
145. SZALAS, A. Temporal logic of programs: a standard approach. In *Time and Logic – A Computational Approach*, L. Bolc and A. Szalas, Eds. UCL Press, London, UK, 1994, pp. 1–50.
146. SZALAS, A. AND HOLENDERSKI, L. Incompleteness of first-order temporal logic with until. *Theoretical Computer Science 57* (1988), pp. 317–325.
147. THOMAS, W. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, J. van Leeuwen, Ed., vol. B: Formal Models and Semantics. Elsevier Science, Amsterdam, 1990, pp. 133–194.
148. THOMAS, W. Languages, automata, and logic. In *Handbook of Formal Language Theory*, G. Rozenberg and A. Salomaa, Eds., vol. III. Springer, New York, 1997, pp. 389–455.
149. THOMAS, W. Complementation of Büchi automata revisited. In *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, J. Karhumäki, Ed. Springer, 2000, pp. 109–122.
150. VALMARI, A. The state explosion problem. In *Lectures on Petri Nets I: Basic Models*, vol. 1491 of *Lecture Notes in Computer Science*. Springer, 1998, pp. 429–528.
151. VAN BENTHEM, J. *The Logic of Time*, vol. 156 of *Synthese Library*. Reidel, Dordrecht, The Netherlands, 1983. Revised and expanded edition, 1991.
152. VARDI, M. Y. Verification of concurrent programs: The automata-theoretic framework. In *Second Symp. Logic in Computer Science* (Ithaca, New York, 1987), IEEE, pp. 167–176.
153. VARDI, M. Y. Alternating automata and program verification. In *Computer Science Today*, J. van Leeuwen, Ed., vol. 1000 of *Lecture Notes in Computer Science*. Springer, 1995, pp. 471–485.
154. VARDI, M. Y. Branching vs. linear time – final showdown. In *Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems* (Genova, Italy, 2001), T. Margaria and W. Yi, Eds., vol. 2031 of *Lecture Notes in Computer Science*, Springer, pp. 1–22.
155. VARDI, M. Y. AND WOLPER, P. Reasoning about infinite computations. *Information and Computation 115*, 1 (1994), pp. 1–37.
156. VON WRIGHT, G. H. Always. *Theoria 34* (1968), pp. 208–221.
157. WALUKIEWICZ, I. Completeness of Kozen's axiomatisation of the propositional μ-calculus. *Information and Computation 157*, 1-2 (2000), pp. 142–182.
158. WHITEHEAD, A. N. AND RUSSELL, B. *Principia Mathematica (3 vols)*. Cambridge University Press, Cambridge, UK, 1910–13. 2nd edition 1925–27.
159. WOLPER, P. Temporal logic can be more expressive. *Information and Control 56* (1983), pp. 72–93.
160. WOLPER, P. The tableau method for temporal logic: an overview. *Logique et Analyse 28* (1985), pp. 119–136.
161. ZAPPE, J. *Towards a Mobile Temporal Logic of Actions*. PhD thesis, Ludwig-Maximilians-Universität München, Munich, Germany, 2005.

Index