

ECE 474A/574A Exam3 Cheat Sheet

<pre> UNATE_COMPLEMENT(F) begin M ← M(F) V ← MONOTONE(F) M' ← PERS_UNATE_COMPLEMENT(M) R ← TRANSLATE(M', V) end </pre>	<pre> PERS_UNATE_COMPLEMENT(M) begin M' ← Φ (M', T) ← SPECIAL_CASES(M) if (T=1) return (M') j ← UCOMP_SELECT(M) (M¹, M⁰) ← PERS_COFACTORS(M, j) M¹' ← PERS_UNATE_COMPLEMENT(M¹) M⁰' ← PERS_UNATE_COMPLEMENT(M⁰) Return(M' ← MERGE(M¹', M⁰') end </pre>	<pre> COMPLEMENT(F, D) begin R ← Φ for (i=1, ..., m) begin (F_i, D_i) ← EXTRACT(F, D, i) R_i ← COMP1(F_i u D_i) R ← R, R_i end Return (R) end </pre>
--	---	--

<pre> COMP1(F) begin if (row of all 2's) Return (R ← Φ) if (F unate) Return (R ← UNATE_COMPLEMENT(F)) c ← F¹ for(j=1, ..., n) begin for(i=2, ..., F) if(c_i ≠ F_j) then c_j ← 2 end R ← UNATE_COMPLEMENT((c)) F ← F_c j ← BINATE_SELECT(F) R ← R, MERGE_WITH_CONTAINMENT(COMP1(F_{x_j}), COMP1(F_{x_j'})) Return end </pre>	<pre> EXPAND(F, R) begin F ← DECREASING_ORDER(F) for (j=1, ..., F) begin (W, F_j) ← EXPAND1(F_j, R, F) F ← (F u {F_j}) - W end Return(F) end </pre>	<pre> EXPAND1(c, F, R) begin (B, C) ← MATRICIES(c, F, R) L ← Φ RA ← Φ N ← NCOLS(B) while (L + RA < N and B≠Φ and C≠Φ) begin X_E ← ESSENTIAL(B) L ← L u X_E (B, C) ← ELIM1(B, C, X_E) Jⁱ ← MFC(B, C) if(Jⁱ = 0) Jⁱ ← EG(C) X_I ← INESSENTIAL(B) RA ← RA u Jⁱ u X_I (B,C) ← ELIM2(B, C, Jⁱ u X_I) end if (B≠Φ) then L ← MINLOW(B) W ← {Fⁱ ∈ F C_{ij} = 0 for all j ∈ L} return (W, c+(L,c)) end </pre>
--	---	---

<p>Two argument operators expressed in terms of ITE</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Table</th> <th>Name</th> <th>Expression</th> <th>Equivalent Form</th> </tr> </thead> <tbody> <tr><td>0000</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0001</td><td>AND(F, G)</td><td>FG</td><td>ITE(F, G, 0)</td></tr> <tr><td>0010</td><td>F > G</td><td>FG'</td><td>ITE(F, G', 0)</td></tr> <tr><td>0011</td><td>F</td><td>F</td><td>F</td></tr> <tr><td>0100</td><td>F < G</td><td>F'G</td><td>ITE(F, 0, G)</td></tr> <tr><td>0101</td><td>G</td><td>G</td><td>G</td></tr> <tr><td>0110</td><td>XOR(F, G)</td><td>F ⊕ G</td><td>ITE(F, G', G)</td></tr> <tr><td>0111</td><td>OR(F, G)</td><td>F + G</td><td>ITE(F, 1, G)</td></tr> <tr><td>1000</td><td>NOR(F, G)</td><td>(F + G)'</td><td>ITE(F, 0, G')</td></tr> <tr><td>1001</td><td>XNOR(F, G)</td><td>(F ⊕ G)'</td><td>ITE(F, G, G')</td></tr> <tr><td>1010</td><td>NOT(G)</td><td>G'</td><td>ITE(G, 0, 1)</td></tr> <tr><td>1011</td><td>F ≥ G</td><td>F + G'</td><td>ITE(F, 1, G')</td></tr> <tr><td>1100</td><td>NOT(F)</td><td>F'</td><td>ITE(F, 0, 1)</td></tr> <tr><td>1101</td><td>F ≤ G</td><td>F' + G</td><td>ITE(F, G, 1)</td></tr> <tr><td>1110</td><td>NAND(F, G)</td><td>(FG)'</td><td>ITE(F, G', 1)</td></tr> <tr><td>1111</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	Table	Name	Expression	Equivalent Form	0000	0	0	0	0001	AND(F, G)	FG	ITE(F, G, 0)	0010	F > G	FG'	ITE(F, G', 0)	0011	F	F	F	0100	F < G	F'G	ITE(F, 0, G)	0101	G	G	G	0110	XOR(F, G)	F ⊕ G	ITE(F, G', G)	0111	OR(F, G)	F + G	ITE(F, 1, G)	1000	NOR(F, G)	(F + G)'	ITE(F, 0, G')	1001	XNOR(F, G)	(F ⊕ G)'	ITE(F, G, G')	1010	NOT(G)	G'	ITE(G, 0, 1)	1011	F ≥ G	F + G'	ITE(F, 1, G')	1100	NOT(F)	F'	ITE(F, 0, 1)	1101	F ≤ G	F' + G	ITE(F, G, 1)	1110	NAND(F, G)	(FG)'	ITE(F, G', 1)	1111	1	1	1	<pre> ITE(F, G, H){ (result, terminal_case) = TERMINAL_CASE(F, G, H) if (terminal_case) return (result) (result, in_computed_table) = COMPUTED_TABLE_HAS_ENTRY(F, G, H) if (in_computed_table) return (result) v = TOP_VARIABLE(F, G, H) T = ITE(F_v, G_v, H_v) E = ITE(F_v', G_v', H_v') R = FIND_OR_ADD_UNIQUE_TABLE(v, T, E) INSERT_COMPUTED_TABLE((F, G, H), R) return (R) } </pre>
Table	Name	Expression	Equivalent Form																																																																		
0000	0	0	0																																																																		
0001	AND(F, G)	FG	ITE(F, G, 0)																																																																		
0010	F > G	FG'	ITE(F, G', 0)																																																																		
0011	F	F	F																																																																		
0100	F < G	F'G	ITE(F, 0, G)																																																																		
0101	G	G	G																																																																		
0110	XOR(F, G)	F ⊕ G	ITE(F, G', G)																																																																		
0111	OR(F, G)	F + G	ITE(F, 1, G)																																																																		
1000	NOR(F, G)	(F + G)'	ITE(F, 0, G')																																																																		
1001	XNOR(F, G)	(F ⊕ G)'	ITE(F, G, G')																																																																		
1010	NOT(G)	G'	ITE(G, 0, 1)																																																																		
1011	F ≥ G	F + G'	ITE(F, 1, G')																																																																		
1100	NOT(F)	F'	ITE(F, 0, 1)																																																																		
1101	F ≤ G	F' + G	ITE(F, G, 1)																																																																		
1110	NAND(F, G)	(FG)'	ITE(F, G', 1)																																																																		
1111	1	1	1																																																																		