

LOGIC

<p>TTable: truth table example</p> <p>Lex: examples</p>	<pre> 2: 1: A B C A 1 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 1 1 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 </pre> <p>TTable Lex LTable LsImp LxSF Lval</p>	<pre> 0: 1: A B C A 0 0 0 1 0 0 1 0 0 1 0 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 1 1 1 1 1 </pre> <p>choose logic example</p> <p>NOT NOT(A AND B) OR NOT C</p> <p>'A AND B AND NOT C OR C</p> <p>'NOR(A,B)</p> <p>'NAND(A,B)</p> <p>'EQUI(A,B)</p> <p>'IMPL(A,B)</p> <p>CANCEL OK</p>
<p>Ltable: logic table (2s)</p> <p>next example (4.8s)</p>	<pre> 0: 1: (NOT A OR B) AND (NOT B OR A) EQUI(A,B) A B A 0 0 1 0 1 0 1 0 0 1 1 1 </pre> <p>TTable Lex LTable LsImp LxSF Lval</p>	<pre> 0: 1: NOT NOT (A AND B) OR NOT C A B C A 0 0 0 1 0 0 1 0 0 1 0 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 1 1 1 1 1 </pre> <p>TTable Lex LTable LsImp LxSF Lval</p>
<p>Ltable: logic table (1.3s)</p> <p>LsImp: simplify expression (3s)</p>	<pre> 0: 1: A XOR B A B F 0 0 0 0 1 1 1 0 1 1 1 0 </pre> <p>Table EX1 EX2 LTable LsImp LxSF Lval</p>	<pre> 0: 1: A AND B OR A AND C A AND (B OR C) (NOT A OR B) AND (NOT B OR A) EQUI(A,B) NOT NOT (A AND B) OR NOT C IMPL(A,B) </pre> <p>Table EX1 EX2 LTable LsImp LxSF Lval</p>
<p>LxSF: logic to short form and back (0.5s, 1.5s)</p> <p>further examples (0.3s)</p>	<pre> 0: 1: NOT NOT (A AND B) OR NOT C ~(A+B)+~C NOT NOT (A AND B) OR NOT C A AND B AND NOT C OR A AND B A+B+C+A~B+C+~A+B+C+~A~B+C </pre> <p>TTable Lex LTable LsImp LxSF Lval</p>	<pre> 0: 1: NOR(A,B) ~(A+B) NOT (A OR B) EQUI(A,B) ~(A+B)+(~B+A) (NOT A OR B) AND (NOT B OR A) IMPL(A,B) ~A+B NOT A OR B </pre> <p>TTable Lex LTable LsImp LxSF Lval</p>
<p>Lval: value of expression (0.5s, 0.7s)</p> <p>Lexpor, Lexpor: expand logical expression (0.5s)</p>	<pre> 0: 1: EQUI(A,B) 0 1 3 0 NOT NOT (A AND B) OR NOT C 0 1 0 3 1 </pre> <p>Lexpor Lval ~Nor ~Nand ~Dnf ~Cnf</p>	<pre> 0: 1: A AND (B OR C) A AND B OR A AND C A OR B AND C (A OR B) AND (A OR C) </pre> <p>Lexpor Lval ~Nor ~Nand ~Dnf ~Cnf</p>
<p>→Nor: express by NOR (1s)</p> <p>→Nand: express by NAND (1s)</p>	<pre> 0: 1: A AND (B OR C) NOR(NOR(A,A),NOR(B,C)) A OR B NOR(NOR(A,B),NOR(A,B)) NOT A NOR(A,A) </pre> <p>Lexpor Lval ~Nor ~Nand ~Dnf ~Cnf</p>	<pre> 0: 1: NOT A NAND(A,A) A AND B NAND(NAND(A,B),NAND(A,B)) A OR B NAND(NAND(A,A),NAND(B,B)) </pre> <p>Lexpor Lval ~Nor ~Nand ~Dnf ~Cnf</p>
<p>→Dnf: table to disjunctive normal form (0.6s)</p> <p>→Cnf: table to conjunctive normal form (0.6s)</p>	<pre> 0: 1: NOT A AND NOT B OR A AND B </pre> <p>Lexpor Lval ~Nor ~Nand ~Dnf ~Cnf</p>	<pre> 0: 1: (NOT A OR NOT B OR NOT C) AND </pre> <p>Lexpor Lval ~Nor ~Nand ~Dnf ~Cnf</p>
<p>OR↔, AND↔: interchanges A and B (0.6s)</p> <p>NOR, NAND: evaluate NOR(A,B) with EVAL</p>	<pre> 0: 1: A OR B B OR A A AND B B AND A </pre> <p>Or↔ And↔ NOR NAND IMPL EQUI</p>	<pre> 0: 1: A B NOT (A OR B) NOR(A,B) NOT (A OR B) NAND(A,B) NOT (A AND B) </pre> <p>Or↔ And↔ NOR NAND IMPL EQUI</p>

IMPL: implication EQUI: equivalence evaluated with EVAL T->IV: term to independent variables (1s)	<pre> 9: 8: 7: 6: 5: 4: IMPL(A,B) 3: NOT A OR B 2: EQUI(A,B) 1: (NOT A OR B) AND (NOT B OR A) Or+ And+ NOR NAND IMPL EQUI LOGIC 'A'=PROPOSITION EXAMPLE 'B AND NOT C' Table [1] EX. TRUTH-TABLE EX1,2 EXAMPLE PROPOSITIONS Ltable 'A' + LOGICTABLE Lsimp 'A' + 'B' SIMPLIFY Lexpand 'A' + 'B' EXPAND: (A OR B) AND C, NOT(A AND/OR B) Lexpor 'A' + 'B' EXPAND: A OR (B AND C) Lval 'A' [1 0 ...] + 1,0 GRAPH </pre>	<pre> 9: EQUI(A,B) 8: (A B) 7: IMPL(A,B) 6: (A B) 5: NOR(A,A) 4: (A) 3: IMPL(A,A AND B) 2: 1: (A B) T->IV HelpL CST +nor 'A' + EXPRESS BY NOR +and 'A' + EXPRESS BY NAND +D(C)nf [1] + 'A' TRUTH-TABLE + DISJUNCTIVE NORMALFORM Or+ 'A OR B' + 'B OR A' And+ 'A AND B' + 'B AND A' NOR 'A' 'B' + 'NOR(A,B)' NAND 'A' 'B' + 'NAND(A,B)' IMPL 'A' 'B' + 'IMPL(A,B)' EQUI 'A' 'B' + 'EQUI(A,B)' T->IV 'A' + 1..3 TERM + INDEPENDENT VARS GRAPH </pre>
HelpLOGIC: help		