```java
//Robert Swartz
//Meta Theorem
//Copyright 2023
//Processor Design Capability


import java.awt.*;
import java.awt.datatransfer.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;


public class MetaTheorem
{
   public static void main(String[] args)
   {
      AppletImage metaApplet = new AppletImage();
      int w = 1065, h = 660, h2 = 75;
      metaApplet.pF.setLocation(200, 175);
      metaApplet.pF.setVisible(true);
      metaApplet.pF.setSize(w, h);
      metaApplet.pF.setResizable(false);
      metaApplet.mwa = new MyWindowAdapter(metaApplet);
      metaApplet.pF.addWindowListener(metaApplet.mwa);
      metaApplet.p0.setSize(w, h);
      metaApplet.pF.add(metaApplet.p0);
      metaApplet.p1.setSize(w, (h-h2)/2+h2);
      metaApplet.p1.setBackground(Color.lightGray);
      metaApplet.p2.setSize(w, h2);
      metaApplet.entry.setSize(w, (h-h2)/2);
      metaApplet.entry.setBackground(new Color(156, 248, 248));
      metaApplet.entry.setFont(metaApplet.myFont1);
      metaApplet.message.setEditable(false);
      metaApplet.message.setSize(w, (h-h2)/2);
      metaApplet.message.setBackground(new Color(224, 204, 176));
      metaApplet.message.setFont(metaApplet.myFont3);
      metaApplet.message.setText(metaApplet.helpMessage);
      metaApplet.syntaxMessage = new MyCanvas(metaApplet, 1);
      metaApplet.syntaxMessage.setSize(200, 26);
      metaApplet.syntaxMessage.setBackground(new Color(184, 112, 112));
      metaApplet.percentStatus = new MyCanvas(metaApplet, 2);
      metaApplet.percentStatus.setSize(200, 26);
      metaApplet.percentStatus.setBackground(new Color(64, 144, 172));
      metaApplet.mma0 = new MyMouseAdapter0(metaApplet);
      metaApplet.percentStatus.addMouseListener(metaApplet.mma0);
      metaApplet.mka1 = new MyKeyAdapter1(metaApplet);
      metaApplet.mma1 = new MyMouseAdapter1(metaApplet);
      metaApplet.start = new MyButton(metaApplet.spaces(18));
      metaApplet.start.myText = "Start";
      metaApplet.start.myLength = 10;
      metaApplet.start.setBackground(new Color(0, 208, 0));
      metaApplet.start.setFont(metaApplet.myFont2);
      metaApplet.start.addKeyListener(metaApplet.mka1);
      metaApplet.start.addMouseListener(metaApplet.mma1);
      metaApplet.p2.add(metaApplet.start);
      metaApplet.mka2 = new MyKeyAdapter2(metaApplet);
      metaApplet.mma2 = new MyMouseAdapter2(metaApplet);
      metaApplet.stop = new MyButton(metaApplet.spaces(17));
      metaApplet.stop.myText = "Stop";
```

```
        metaApplet.stop.myLength = 9;
        metaApplet.stop.setBackground(new Color(224, 64, 64));
        metaApplet.stop.setFont(metaApplet.myFont2);
        metaApplet.stop.addKeyListener(metaApplet.mka2);
        metaApplet.stop.addMouseListener(metaApplet.mma2);
        metaApplet.p2.add(metaApplet.stop);
        metaApplet.mka3 = new MyKeyAdapter3(metaApplet);
        metaApplet.mma3 = new MyMouseAdapter3(metaApplet);
        metaApplet.postfixB = new MyButton(metaApplet.spaces(23));
        metaApplet.postfixB.myText = "Postfix";
        metaApplet.postfixB.myLength = 12;
        metaApplet.postfixB.setBackground(new Color(125, 175, 175));
        metaApplet.postfixB.setFont(metaApplet.myFont2);
        metaApplet.postfixB.addKeyListener(metaApplet.mka3);
        metaApplet.postfixB.addMouseListener(metaApplet.mma3);
        metaApplet.p2.add(metaApplet.postfixB);
        metaApplet.mka4 = new MyKeyAdapter4(metaApplet);
        metaApplet.mma4 = new MyMouseAdapter4(metaApplet);
        metaApplet.modelB = new MyButton(metaApplet.spaces(25));
        metaApplet.modelB.myText = "Model";
        metaApplet.modelB.myLength = 12;
        metaApplet.modelB.setBackground(new Color(64, 156, 224));
        metaApplet.modelB.setFont(metaApplet.myFont2);
        metaApplet.modelB.addKeyListener(metaApplet.mka4);
        metaApplet.modelB.addMouseListener(metaApplet.mma4);
        metaApplet.p2.add(metaApplet.modelB);
        metaApplet.mka5 = new MyKeyAdapter5(metaApplet);
        metaApplet.mma5 = new MyMouseAdapter5(metaApplet);
        metaApplet.counterexampleB = new MyButton(metaApplet.spaces(37));
        metaApplet.counterexampleB.myText = "Counterexample";
        metaApplet.counterexampleB.myLength = 17;
        metaApplet.counterexampleB.setBackground(new Color(156, 112, 160));
        metaApplet.counterexampleB.setFont(metaApplet.myFont2);
        metaApplet.counterexampleB.addKeyListener(metaApplet.mka5);
        metaApplet.counterexampleB.addMouseListener(metaApplet.mma5);
        metaApplet.p2.add(metaApplet.counterexampleB);
        metaApplet.mka6 = new MyKeyAdapter6(metaApplet);
        metaApplet.mma6 = new MyMouseAdapter6(metaApplet);
        metaApplet.reset = new MyButton(metaApplet.spaces(19));
        metaApplet.reset.myText = "Reset";
        metaApplet.reset.myLength = 10;
        metaApplet.reset.setBackground(new Color(240, 128, 176));
        metaApplet.reset.setFont(metaApplet.myFont2);
        metaApplet.reset.addKeyListener(metaApplet.mka6);
        metaApplet.reset.addMouseListener(metaApplet.mma6);
        metaApplet.p2.add(metaApplet.reset);
        metaApplet.mka7 = new MyKeyAdapter7(metaApplet);
        metaApplet.mma7 = new MyMouseAdapter7(metaApplet);
        metaApplet.paste = new MyButton(metaApplet.spaces(17));
        metaApplet.paste.myText = "Paste";
        metaApplet.paste.myLength = 9;
        metaApplet.paste.setBackground(new Color(96, 208, 208));
        metaApplet.paste.setFont(metaApplet.myFont2);
        metaApplet.paste.addKeyListener(metaApplet.mka7);
        metaApplet.paste.addMouseListener(metaApplet.mma7);
        metaApplet.p2.add(metaApplet.paste);
        metaApplet.mka8 = new MyKeyAdapter8(metaApplet);
        metaApplet.mma8 = new MyMouseAdapter8(metaApplet);
        metaApplet.clear = new MyButton(metaApplet.spaces(17));
        metaApplet.clear.myText = "Clear";
```

```
121         metaApplet.clear.myLength = 9;
122         metaApplet.clear.setBackground(new Color(160, 160, 224));
123         metaApplet.clear.setFont(metaApplet.myFont2);
124         metaApplet.clear.addKeyListener(metaApplet.mka8);
125         metaApplet.clear.addMouseListener(metaApplet.mma8);
126         metaApplet.p2.add(metaApplet.clear);
127         metaApplet.mka9 = new MyKeyAdapter9(metaApplet);
128         metaApplet.mma9 = new MyMouseAdapter9(metaApplet);
129         metaApplet.help = new MyButton(metaApplet.spaces(15));
130         metaApplet.help.myText = "Help";
131         metaApplet.help.myLength = 8;
132         metaApplet.help.setBackground(new Color(240, 128, 64));
133         metaApplet.help.setFont(metaApplet.myFont2);
134         metaApplet.help.addKeyListener(metaApplet.mka9);
135         metaApplet.help.addMouseListener(metaApplet.mma9);
136         metaApplet.p2.add(metaApplet.help);
137         GridBagLayout gbl = new GridBagLayout();
138         GridBagConstraints gbc = new GridBagConstraints();
139         metaApplet.p3 = new Panel(gbl);
140         metaApplet.p3.setSize(w, h2);
141         Canvas[] cs = new Canvas[5];
142         for (int i = 0; i < 5; i++)
143         {
144            cs[i] = new Canvas();
145            cs[i].setSize(100, 16);
146         }
147         cs[0].setSize(100, 8);
148         gbc.gridy = 0;
149         gbc.gridwidth = 1;
150         gbl.setConstraints(cs[0], gbc);
151         metaApplet.p3.add(cs[0]);
152         gbc.gridy = 1;
153         gbc.anchor = GridBagConstraints.WEST;
154         gbl.setConstraints(cs[1], gbc);
155         metaApplet.p3.add(cs[1]);
156         gbc.gridwidth = 4;
157         gbl.setConstraints(metaApplet.syntaxMessage, gbc);
158         metaApplet.p3.add(metaApplet.syntaxMessage);
159         gbc.gridwidth = 1;
160         gbc.anchor = GridBagConstraints.CENTER;
161         cs[2].setSize(175, 16);
162         gbl.setConstraints(cs[2], gbc);
163         metaApplet.p3.add(cs[2]);
164         gbc.gridwidth = 4;
165         gbc.anchor = GridBagConstraints.EAST;
166         gbl.setConstraints(metaApplet.percentStatus, gbc);
167         metaApplet.p3.add(metaApplet.percentStatus);
168         gbc.gridwidth = 1;
169         gbl.setConstraints(cs[3], gbc);
170         metaApplet.p3.add(cs[3]);
171         gbc.gridy = 2;
172         gbl.setConstraints(cs[4], gbc);
173         metaApplet.p3.add(cs[4]);
174         metaApplet.p1.add(metaApplet.entry, "North");
175         metaApplet.p1.add(metaApplet.p2, "Center");
176         metaApplet.p1.add(metaApplet.p3, "South");
177         metaApplet.p0.add(metaApplet.p1, "North");
178         metaApplet.p0.add(metaApplet.message, "Center");
179         metaApplet.pF.paintAll(metaApplet.pF.getGraphics());
180         metaApplet.entry.requestFocus();
```

```
      System.gc();
   }
}


class AppletImage
{
   public Theorem metaThread;
   public Timer timerThread;
   public Frame pF = new Frame("Meta Theorem");
   public Panel p0 = new Panel(new BorderLayout()), p1 = new Panel(new
      BorderLayout()), p2 = new Panel(new FlowLayout(FlowLayout.CENTER,
      16, 16)), p3;
   public TextArea entry = new TextArea("", 13, 13, TextArea.
      SCROLLBARS_VERTICAL_ONLY), message = new TextArea(" ", 13, 13, TextArea.
      SCROLLBARS_VERTICAL_ONLY);
   public MyButton start, stop, postfixB, modelB, counterexampleB, reset, paste,
      clear, help;
   public MyCanvas syntaxMessage, percentStatus;
   public Font myFont1 = new Font("Monospaced", Font.BOLD, 16), myFont2 = new
      Font("SansSerif", Font.BOLD, 16), myFont3 = new Font("SansSerif",
      Font.PLAIN, 17);
   public Color percentColor = new Color(64, 144, 172);
   public MyMouseAdapter0 mma0;
   public MyKeyAdapter1 mka1;
   public MyMouseAdapter1 mma1;
   public MyKeyAdapter2 mka2;
   public MyMouseAdapter2 mma2;
   public MyKeyAdapter3 mka3;
   public MyMouseAdapter3 mma3;
   public MyKeyAdapter4 mka4;
   public MyMouseAdapter4 mma4;
   public MyKeyAdapter5 mka5;
   public MyMouseAdapter5 mma5;
   public MyKeyAdapter6 mka6;
   public MyMouseAdapter6 mma6;
   public MyKeyAdapter7 mka7;
   public MyMouseAdapter7 mma7;
   public MyKeyAdapter8 mka8;
   public MyMouseAdapter8 mma8;
   public MyKeyAdapter9 mka9;
   public MyMouseAdapter9 mma9;
   public MyWindowAdapter mwa;
   public boolean complete = true, on_off = true, running, flag_e;
   public long k = Long.MIN_VALUE, lines, LEN2, modelCount, counterexampleCount;
   public boolean[] boolArray = new boolean[64];
   public String helpMessage =
      "Robert Swartz\nwww.mathapplets.net\nMeta Theorem\nCopyright 2023\n\n"+
      "This program tests whether a sentence of Boolean Logic is a theorem, a "+
      "contradiction, or a contingency.  A theorem is a sentence that is true "+
      "for all variable assignments, whereas a contradiction is a sentence that"
      +" is always false; a contingency is a sentence that can be true "+
      "sometimes or false.\n\nThere are 6 logical operations:  \'and\', \'or\',"
      +" \'xor\', \'implies\', \'iff\', \'not\'.  Enter these using the "+
      "following symbols:  &  |  *  >  =  !.  Logical variables are represented"
      +" by single letters, single digits, @, or #.  The logical constants "+
      "\'true\' and \'false\' are represented by + and -.  Therefore, this "+
      "program can handle sentences with up to 64 variables.  This program "+
      "evaluates all operations from left to right; use parentheses to change "
      +"the way a logical sentence is evaluated.  Blank spaces are parsed out."+
```

```
241       "\n\nFor example, the logical sentence \'(P implies Q) iff ((not P) or Q)"
242       +"\' would be entered as (P > Q) = (!P | Q).\n\nThe first thing that this"
243       +" program does is convert the sentence into postfix form.  Postfix form "
244       +"is more efficient especially if the sentence is to be evaluated many "+
245       "times.  For example, the sentence (P & (Q | R)) = ((P & Q) | (P & R)) "+
246       "would be converted to P Q R | & P Q & P R & | =.  During the conversion "
247       +"process, the syntax of the sentence is checked for errors.  Then, the "+
248       "program goes through the truth table of the sentence, starting from the"+
249       " all true case, and proceeding to all false.  It searches for 2 "+
250       "different variable assignments such that one makes the sentence true "+
251       "(model), and the other makes the sentence false (counterexample).  If "+
252       "after going through all 2^n lines of the truth table, the program "+
253       "doesn't find a model and a counterexample, then either the sentence is a"
254       +" theorem if only models were found, or it's a contradiction if only "
255       +"counterexamples were found.  Of course, if both a model and a "+
256       "counterexample were found, the sentence is a contingency.  Here, n is "+
257       "the number of variables in the sentence.\n\nWhile the applet is running,"
258       +" the percentage of progress through the truth table is displayed.  "
259       +"Scrolling through the truth table is possible with the Model and "+
260       "Counterexample buttons; the Reset button returns the pointer to the "+
261       "first line of the truth table.  It is advisable to reset the pointer "
262       +"when changing scroll modes.";
263
264    public String spaces(int size)
265    {
266       String text = "";
267       for (int i = 1; i <= size; i++)
268          text+=" ";
269       return text;
270    }
271 }
272
273
274 class Theorem extends Thread
275 {
276    public AppletImage metaApplet;
277    public int flag, tokens, len0, len, len2;
278    public String sentence, vars0 = "", postfix = "", N = "\n\n\n", T = "\t\t\t";
279    public StringTokenizer sentence2;
280    public StringBuffer sentence3, stackSentence0;
281    public char[] sentence4, stackSentence, vars;
282    public boolean[] stack;
283    public MyException myE = new MyException();
284
285    public Theorem(AppletImage a, int n)
286    {
287       metaApplet = a;
288       flag = n;
289    }
290
291    public void run()
292    {
293       long timeStamp = System.currentTimeMillis();
294       metaApplet.running = true;
295       metaApplet.flag_e = false;
296       metaApplet.lines = -1;
297       if (flag != 2)
298          metaApplet.timerThread = new Timer(metaApplet);
299       else
300          metaApplet.percentStatus.setText(" ");
```

```
301         metaApplet.message.setText(" ");
302         metaApplet.message.setFont(metaApplet.myFont1);
303         metaApplet.message.setText(N+"BUSY...");
304         boolean first = true, noTrue = true, noFalse = true;
305         String assignment = "", model = "", counterexample = "", X = "", time;
306         sentence = metaApplet.entry.getText();
307         len0 = sentence.length();
308         if (len0 == 0)
309            myStop1();
310         else
311         {
312            sentence2 = new StringTokenizer(sentence);
313            sentence = null;
314            tokens = sentence2.countTokens();
315            sentence3 = new StringBuffer(len0);
316            for (int i = 0; i < tokens; i++)
317               sentence3.append(sentence2.nextToken());
318            sentence2 = null;
319            if (sentence3.length() == 0)
320               myStop1();
321            else
322            {
323               System.gc();
324               sentence4 = new char[sentence3.length()];
325               sentence3.getChars(0, sentence3.length(), sentence4, 0);
326               sentence3 = null;
327               for (int i = 0; i < sentence4.length && vars0.length() < 64; i++)
328                  if (isVariable(sentence4[i]) && vars0.indexOf(sentence4[i]) < 0)
329                     vars0+=sentence4[i];
330               len2 = vars0.length();
331               metaApplet.LEN2 = len2;
332               vars = new char[len2];
333               vars0.getChars(0, len2, vars, 0);
334               if (len2 <= 62)
335               {
336                  metaApplet.lines = 1;
337                  for (int i = 0; i < len2; i++)
338                     metaApplet.lines*=2;
339                  metaApplet.lines+=Long.MAX_VALUE;
340               }
341               else if (len2 == 64)
342                  metaApplet.lines = Long.MAX_VALUE;
343               if (metaApplet.k == Long.MIN_VALUE)
344                  for (int i = 0; i < len2; i++)
345                     metaApplet.boolArray[i] = false;
346               stackSentence0 = new StringBuffer(sentence4.length);
347               System.gc();
348               try
349               {
350                  convert(0, sentence4.length-1);
351               }
352               catch (MyException e)
353               {
354                  metaApplet.flag_e = true;
355               }
356               if (metaApplet.running)
357               {
358                  sentence4 = null;
359                  len = stackSentence0.length();
360                  stackSentence = new char[len];
```

```java
                stackSentence0.getChars(0, len, stackSentence, 0);
                stackSentence0 = null;
                System.gc();
                stack = new boolean[len];
                metaApplet.syntaxMessage.setText("Valid Syntax");
                if (flag == 1)
                {
                   metaApplet.timerThread.start();
                   if (metaApplet.complete)
                   {
                      metaApplet.complete = false;
                      metaApplet.k = Long.MIN_VALUE;
                      for (int i = 0; i < len2; i++)
                         metaApplet.boolArray[i] = false;
                   }
                   do
                   {
                      update();
                      crunch();
                      if (metaApplet.running)
                      {
                         if (metaApplet.k == Long.MIN_VALUE)
                            first = stack[0];
                         if (stack[0])
                            noTrue = false;
                         else
                            noFalse = false;
                      }
                      metaApplet.k++;
                      if (!metaApplet.running)
                      {
                         if (metaApplet.k > Long.MIN_VALUE)
                         {
                            reverseUpdate();
                            metaApplet.k--;
                         }
                         break;
                      }
                      if ((len2 < 64 && metaApplet.k > metaApplet.lines) ||
                          (len2 == 64 && metaApplet.k == metaApplet.lines))
                         break;
                   }
                   while (metaApplet.running && (noFalse || noTrue));
                   if (len2 == 64 && metaApplet.k == metaApplet.lines)
                   {
                      update();
                      crunch();
                      if (metaApplet.running)
                      {
                         if (stack[0])
                            noTrue = false;
                         else
                            noFalse = false;
                      }
                   }
                   if (metaApplet.running)
                   {
                      metaApplet.complete = true;
                      if (!(noFalse || noTrue))
                      {
```

```
421                     for (int j = 0; j < len2/4; j++)
422                     {
423                        for (int i = 0; i < 4; i++)
424                           assignment+=vars[4*j+i]+" = "
425                              +metaApplet.boolArray[4*j+i]+T;
426                        assignment+="\n";
427                     }
428                     for (int i = 0; i < 3; i++)
429                        if (len2%4 > i)
430                           assignment+=vars[len2-(len2%4-i)]+" = "
431                              +metaApplet.boolArray[len2-(len2%4-i)]+T;
432                     assignment = assignment.trim();
433                     model = assignment;
434                     counterexample = "All variables are true.";
435                     if (first)
436                     {
437                        model = "All variables are true.";
438                        counterexample = assignment;
439                     }
440                     metaApplet.message.setText("\n\nThis is a contingency."
441                        +"\n\nModel:\n"+model+"\n\nCounterexample:\n"
442                        +counterexample);
443                  }
444                  else if (!noTrue)
445                     metaApplet.message.setText(N+"This is a theorem.");
446                  else
447                     metaApplet.message.setText(N+"This is a contradiction.")
448                        ;
449                  time = ""+(System.currentTimeMillis()-timeStamp)/10/100.0;
450                  if (time.length()-time.indexOf('.') == 2)
451                     time+=0;
452                  metaApplet.message.append("\n\n"+time+" seconds");
453               }
454            }
455            else if (flag == 2)
456            {
457               metaApplet.flag_e = true;
458               for (int i = 0; i < len; i++)
459                  postfix+=stackSentence[i]+" ";
460               metaApplet.message.setText(N+postfix);
461            }
462            else if (flag == 3 && len2 > 0)
463            {
464               metaApplet.timerThread.start();
465               if ((len2 < 64 && metaApplet.k <= metaApplet.lines) ||
466                  (len2 == 64 && metaApplet.k <  metaApplet.lines))
467               {
468                  do
469                  {
470                     update();
471                     crunch();
472                     metaApplet.k++;
473                     if (metaApplet.running && stack[0])
474                     {
475                        noTrue = false;
476                        break;
477                     }
478                     if (!metaApplet.running)
479                     {
480                        if (metaApplet.k > Long.MIN_VALUE)
```

```
                        {
                            reverseUpdate();
                            metaApplet.k--;
                        }
                        break;
                    }
                }
                while (metaApplet.running && ((len2 < 64 && metaApplet.k <=
                    metaApplet.lines) || (len2 == 64 && metaApplet.k <
                    metaApplet.lines)));
            }
            if (len2 == 64 && metaApplet.k == metaApplet.lines)
            {
                update();
                crunch();
                if (metaApplet.running && stack[0])
                    noTrue = false;
            }
            if (metaApplet.running)
            {
                if (!noTrue)
                {
                    metaApplet.modelCount++;
                    assignment = "\nModel #"+metaApplet.modelCount+":\n\n";
                    for (int j = 0; j < len2/4; j++)
                    {
                        for (int i = 0; i < 4; i++)
                            assignment+=vars[4*j+i]+" = "
                                +metaApplet.boolArray[4*j+i]+T;
                        assignment+="\n";
                    }
                    for (int i = 0; i < 3; i++)
                        if (len2%4 > i)
                            assignment+=vars[len2-(len2%4-i)]+" = "
                                +metaApplet.boolArray[len2-(len2%4-i)]+T;
                    metaApplet.message.setText("\n\n"+assignment.trim());
                }
                else if (metaApplet.k >= metaApplet.lines)
                {
                    if (metaApplet.modelCount == 0)
                        metaApplet.message.setText(N+"No models.");
                    else
                    {
                        X = N+"No more models.\n\n"+metaApplet.modelCount;
                        if (metaApplet.modelCount == 1)
                            metaApplet.message.setText(X+" model was found.");
                        else
                            metaApplet.message.setText(X+" models were found."
                                );
                    }
                }
            }
        }
        else if (flag == 4 && len2 > 0)
        {
            metaApplet.timerThread.start();
            if ((len2 < 64 && metaApplet.k <= metaApplet.lines) ||
                (len2 == 64 && metaApplet.k <  metaApplet.lines))
            {
                do
```

```
                    {
                        update();
                        crunch();
                        metaApplet.k++;
                        if (metaApplet.running && !stack[0])
                        {
                            noFalse = false;
                            break;
                        }
                        if (!metaApplet.running)
                        {
                            if (metaApplet.k > Long.MIN_VALUE)
                            {
                                reverseUpdate();
                                metaApplet.k--;
                            }
                            break;
                        }
                    }
                    while (metaApplet.running && ((len2 < 64 && metaApplet.k <=
                        metaApplet.lines) || (len2 == 64 && metaApplet.k <
                        metaApplet.lines)));
                }
                if (len2 == 64 && metaApplet.k == metaApplet.lines)
                {
                    update();
                    crunch();
                    if (metaApplet.running && !stack[0])
                        noFalse = false;
                }
                if (metaApplet.running)
                {
                    if (!noFalse)
                    {
                        metaApplet.counterexampleCount++;
                        assignment = "\nCounterexample #"
                            +metaApplet.counterexampleCount+":\n\n";
                        for (int j = 0; j < len2/4; j++)
                        {
                            for (int i = 0; i < 4; i++)
                                assignment+=vars[4*j+i]+" = "
                                    +metaApplet.boolArray[4*j+i]+T;
                            assignment+="\n";
                        }
                        for (int i = 0; i < 3; i++)
                            if (len2%4 > i)
                                assignment+=vars[len2-(len2%4-i)]+" = "
                                    +metaApplet.boolArray[len2-(len2%4-i)]+T;
                        metaApplet.message.setText("\n\n"+assignment.trim());
                    }
                    else if (metaApplet.k >= metaApplet.lines)
                    {
                        if (metaApplet.counterexampleCount == 0)
                            metaApplet.message.setText(N+"No counterexamples.");
                        else
                        {
                            X = N+"No more counterexamples.\n\n"+
                                metaApplet.counterexampleCount;
                            if (metaApplet.counterexampleCount == 1)
                                metaApplet.message.setText(X+
```

```
                                    " counterexample was found.");
                            else
                                metaApplet.message.setText(X+
                                    " counterexamples were found.");
                        }
                    }
                }
                else if (flag == 3 && len2 == 0)
                    metaApplet.message.setText(N+"Models don't apply!");
                else
                    metaApplet.message.setText(N+"Counterexamples don't apply!");
            }
            else
                metaApplet.percentStatus.setText(" ");
        }
    }
    metaApplet.running = false;
}

public void convert(int pos, int end)throws MyException
{
    int i = 0, parens = 0, nots = 0;
    boolean skip = false, error = false;
    String oper = "";
    while (pos <= end)
    {
        if (!metaApplet.running)
            throw myE;
        skip = false;
        error = true;
        nots = 0;
        parens = 0;
        while (pos <= end && sentence4[pos] == '!')
        {
            nots++;
            pos++;
        }
        if (pos > end)
            myStop2();
        if (pos != 0 && sentence4[pos-1] == '!' && !(sentence4[pos] == '(' ||
            isVariable(sentence4[pos]) || sentence4[pos] == '+' ||
            sentence4[pos] == '-'))
            myStop2();
        if (nots > 0)
            error = false;
        if (sentence4[pos] == '(')
        {
            i = pos;
            parens++;
            pos++;
            error = false;
            while (parens > 0 && pos <= end)
            {
                if (sentence4[pos] == '(')
                    parens++;
                if (sentence4[pos] == ')')
                    parens--;
                pos++;
            }
```

```
661             if (parens != 0 || i == pos-2)
662                myStop2();
663             convert(i+1, pos-2);
664             if (nots%2 == 1)
665                stackSentence0.append('!');
666             stackSentence0.append(oper);
667             skip = true;
668          }
669          if (pos <= end && (isVariable(sentence4[pos]) || sentence4[pos] == '+'
670             || sentence4[pos] == '-'))
671          {
672             if (skip)
673                myStop2();
674             stackSentence0.append(sentence4[pos]);
675             if (nots%2 == 1)
676                stackSentence0.append('!');
677             stackSentence0.append(oper);
678             error = false;
679             pos++;
680          }
681          if (error)
682             myStop2();
683          if (pos <= end)
684          {
685             if (!isOperator(sentence4[pos]) || pos == end)
686                myStop2();
687             oper = ""+sentence4[pos];
688          }
689          pos++;
690       }
691    }
692
693    public void crunch()
694    {
695       int pointer = -1;
696       for (int n = 0; n < len && metaApplet.running; n++)
697       {
698          if (isVariable(stackSentence[n]))
699             stack[++pointer] = metaApplet.boolArray[myIndex(stackSentence[n])];
700          else
701             switch (stackSentence[n])
702             {
703                case '&' ->
704                   stack[--pointer] &= stack[pointer+1];
705                case '|' ->
706                   stack[--pointer] |= stack[pointer+1];
707                case '>' ->
708                   stack[--pointer] = !stack[pointer] || stack[pointer+1];
709                case '=' ->
710                   stack[--pointer] = !(stack[pointer] ^ stack[pointer+1]);
711                case '!' ->
712                   stack[pointer] = !stack[pointer];
713                case '+' ->
714                   stack[++pointer] = true;
715                case '-' ->
716                   stack[++pointer] = false;
717                default ->
718                   stack[--pointer] ^= stack[pointer+1];
719             }
720       }
```

```
721    }
722
723    public void update()
724    {
725       int i = 0;
726       try
727       {
728          do
729          {
730             metaApplet.boolArray[i] = !metaApplet.boolArray[i];
731             i++;
732          }
733          while (metaApplet.boolArray[i-1]);
734       }
735       catch (ArrayIndexOutOfBoundsException e)
736       {
737          for (i = 0; i < len2; i++)
738             metaApplet.boolArray[i] = true;
739       }
740    }
741
742    public void reverseUpdate()
743    {
744       int i = 0;
745       try
746       {
747          do
748          {
749             metaApplet.boolArray[i] = !metaApplet.boolArray[i];
750             i++;
751          }
752          while (!metaApplet.boolArray[i-1]);
753       }
754       catch (ArrayIndexOutOfBoundsException e)
755       {
756          for (i = 0; i < len2; i++)
757             metaApplet.boolArray[i] = false;
758       }
759    }
760
761    public int myIndex(char x)
762    {
763       for (int index = 0; ; index++)
764          if (x == vars[index])
765             return index;
766    }
767
768    public boolean isVariable(char x)
769    {
770       return (x >= '@' && x <= 'z') || (x >= '0' && x <= '9') || x == '#';
771    }
772
773    public boolean isOperator(char x)
774    {
775       return x == '&' || x == '|' || x == '>' || x == '=' || x == '*';
776    }
777
778    public void myStop1()
779    {
780       metaApplet.message.setText(" ");
```

```
781         metaApplet.syntaxMessage.setText("Invalid Syntax!!!");
782         metaApplet.flag_e = true;
783         metaApplet.running = false;
784      }
785
786      public void myStop2()throws MyException
787      {
788         myStop1();
789         throw myE;
790      }
791   }
792
793
794   class MyException extends Exception{}
795
796
797   class Timer extends Thread
798   {
799      public AppletImage metaApplet;
800
801      public Timer(AppletImage a)
802      {
803         metaApplet = a;
804      }
805
806      public void run()
807      {
808         double X;
809         String s;
810         while (metaApplet.running && !metaApplet.flag_e)
811         {
812            X = ((double)(metaApplet.k+Long.MAX_VALUE+1))/Math.pow(2.0, (double)
813               metaApplet.LEN2);
814            if (X < 0.000001 || (X >= 1.0 && metaApplet.LEN2 == 0))
815               X = 0.0;
816            else
817               X = Math.round(1000000.0*X)/10000.0;
818            if (metaApplet.k <= metaApplet.lines)
819            {
820               s = ""+X;
821               if (X >= 0.0001 && X < 0.001)
822                  s = "0.000"+s.substring(0, 1);
823               else
824               {
825                  for (int i = 1; i <= 3; i++)
826                     if ((s.substring(s.indexOf('.')+1, s.length())).length() < 4)
827                        s += '0';
828               }
829               metaApplet.percentStatus.setText(s+"%");
830            }
831            else
832               metaApplet.percentStatus.setText("100%");
833            metaApplet.percentStatus.paint(metaApplet.percentStatus.getGraphics());
834            try
835            {
836               sleep(333);
837            }
838            catch (InterruptedException e){}
839         }
840         if (!metaApplet.flag_e)
```

```
841         {
842            X = ((double)(metaApplet.k+Long.MAX_VALUE+1))/Math.pow(2.0, (double)
843               metaApplet.LEN2);
844            if (X < 0.000001 || (X >= 1.0 && metaApplet.LEN2 == 0))
845               X = 0.0;
846            else
847               X = Math.round(1000000.0*X)/10000.0;
848            if (metaApplet.k <= metaApplet.lines)
849            {
850               s = ""+X;
851               if (X >= 0.0001 && X < 0.001)
852                  s = "0.000"+s.substring(0, 1);
853               else
854               {
855                  for (int i = 1; i <= 3; i++)
856                     if ((s.substring(s.indexOf('.')+1, s.length())).length() < 4)
857                        s+='0';
858               }
859               metaApplet.percentStatus.setText(s+"%");
860            }
861            else
862               metaApplet.percentStatus.setText("100%");
863            metaApplet.percentStatus.clearBlock();
864            metaApplet.percentStatus.paint(metaApplet.percentStatus.getGraphics());
865         }
866         else
867            metaApplet.percentStatus.setText(" ");
868      }
869   }
870
871
872   class MyCanvas extends Canvas
873   {
874      public AppletImage metaApplet;
875      public int flag;
876      public String textNew = " ", textOld = " ";
877
878      public MyCanvas(AppletImage a, int f)
879      {
880         metaApplet = a;
881         flag = f;
882      }
883
884      public void setText(String s)
885      {
886         Graphics g = getGraphics();
887         textNew = s;
888         if (flag == 1)
889         {
890            g.setFont(new Font("Tahoma", Font.BOLD, 16));
891            g.setColor(new Color(184, 112, 112));
892            g.drawString(textOld, getSize().width/2-(int)(textOld.length()*4),
893               getSize().height-8);
894            g.setColor(Color.black);
895            g.drawString(textNew, getSize().width/2-(int)(textNew.length()*4),
896               getSize().height-8);
897         }
898         else if (flag == 2 && metaApplet.on_off)
899         {
900            g.setFont(new Font("Verdana", Font.BOLD, 16));
```

```
901            g.setColor(metaApplet.percentColor);
902            g.drawString(textOld, getSize().width/2-(int)(textOld.length()*5.5),
903               getSize().height-8);
904            g.setColor(Color.black);
905            g.drawString(textNew, getSize().width/2-(int)(textNew.length()*5.5),
906               getSize().height-8);
907         }
908         textOld = textNew;
909      }
910
911      public void paint(Graphics g)
912      {
913         if (g != null)
914         {
915            if (flag == 1)
916            {
917               g.setFont(new Font("Tahoma", Font.BOLD, 16));
918               g.drawString(textNew, getSize().width/2-(int)(textNew.length()*4)
919                  ,getSize().height-8);
920            }
921            else if (flag == 2 && metaApplet.on_off)
922            {
923               g.setFont(new Font("Verdana", Font.BOLD, 16));
924               g.drawString(textNew, getSize().width/2-(int)(textNew.length()*5.5)
925                  ,getSize().height-8);
926            }
927         }
928      }
929
930      public void clearBlock()
931      {
932         getGraphics().clearRect(0, 0, getSize().width, getSize().height);
933      }
934   }
935
936
937   class MyMouseAdapter0 extends MouseAdapter
938   {
939      public AppletImage metaApplet;
940
941      public MyMouseAdapter0(AppletImage a)
942      {
943         metaApplet = a;
944      }
945
946      public void mousePressed(MouseEvent e)
947      {
948         if (e.getButton() == MouseEvent.BUTTON1)
949         {
950            metaApplet.on_off = !metaApplet.on_off;
951            if (metaApplet.on_off)
952               metaApplet.percentStatus.setText(metaApplet.percentStatus.textNew);
953            else
954               metaApplet.percentStatus.clearBlock();
955         }
956      }
957
958      public void mouseEntered(MouseEvent e)
959      {
960         metaApplet.percentStatus.setBackground(new Color(64, 172, 144));
```

```
961         metaApplet.percentColor = new Color(64, 172, 144);
962      }
963
964      public void mouseExited(MouseEvent e)
965      {
966         metaApplet.percentStatus.setBackground(new Color(64, 144, 172));
967         metaApplet.percentColor = new Color(64, 144, 172);
968      }
969   }
970
971
972   class MyButton extends Button
973   {
974      public String myText;
975      public int myLength;
976
977      public MyButton(String mySpaces)
978      {
979         super(mySpaces);
980      }
981
982      public void paint(Graphics g)
983      {
984         g.setFont(new Font("Tahoma", Font.BOLD, 16));
985         g.drawString(myText, (myLength-myText.length())*5, 20);
986      }
987   }
988
989
990   class MyKeyAdapter1 extends KeyAdapter
991   {
992      public AppletImage metaApplet;
993
994      public MyKeyAdapter1(AppletImage a)
995      {
996         metaApplet = a;
997      }
998
999      public void keyReleased(KeyEvent e)
1000     {
1001        if ((e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
1002           KeyEvent.VK_SPACE) && !metaApplet.running)
1003        {
1004           metaApplet.metaThread = new Theorem(metaApplet, 1);
1005           System.gc();
1006           metaApplet.metaThread.start();
1007        }
1008     }
1009  }
1010
1011
1012  class MyMouseAdapter1 extends MouseAdapter
1013  {
1014     public AppletImage metaApplet;
1015
1016     public MyMouseAdapter1(AppletImage a)
1017     {
1018        metaApplet = a;
1019     }
1020
```

```
1021    public void mousePressed(MouseEvent e)
1022    {
1023       if (e.getButton() == MouseEvent.BUTTON1 && !metaApplet.running)
1024       {
1025          metaApplet.metaThread = new Theorem(metaApplet, 1);
1026          System.gc();
1027          metaApplet.metaThread.start();
1028       }
1029    }
1030 }
1031
1032
1033 class MyKeyAdapter2 extends KeyAdapter
1034 {
1035    public AppletImage metaApplet;
1036
1037    public MyKeyAdapter2(AppletImage a)
1038    {
1039       metaApplet = a;
1040    }
1041
1042    public void keyPressed(KeyEvent e)
1043    {
1044       if (e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
1045          KeyEvent.VK_SPACE)
1046       {
1047          metaApplet.message.setText("");
1048          if (!metaApplet.running)
1049             metaApplet.syntaxMessage.setText("");
1050          metaApplet.running = false;
1051       }
1052    }
1053 }
1054
1055
1056 class MyMouseAdapter2 extends MouseAdapter
1057 {
1058    public AppletImage metaApplet;
1059
1060    public MyMouseAdapter2(AppletImage a)
1061    {
1062       metaApplet = a;
1063    }
1064
1065    public void mousePressed(MouseEvent e)
1066    {
1067       if (e.getButton() == MouseEvent.BUTTON1)
1068       {
1069          metaApplet.message.setText("");
1070          if (!metaApplet.running)
1071             metaApplet.syntaxMessage.setText("");
1072          metaApplet.running = false;
1073       }
1074    }
1075 }
1076
1077
1078 class MyKeyAdapter3 extends KeyAdapter
1079 {
1080    public AppletImage metaApplet;
```

```
1081
1082    public MyKeyAdapter3(AppletImage a)
1083    {
1084       metaApplet = a;
1085    }
1086
1087    public void keyReleased(KeyEvent e)
1088    {
1089       if ((e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
1090          KeyEvent.VK_SPACE) && !metaApplet.running)
1091       {
1092          metaApplet.metaThread = new Theorem(metaApplet, 2);
1093          System.gc();
1094          metaApplet.metaThread.start();
1095       }
1096    }
1097 }
1098
1099
1100 class MyMouseAdapter3 extends MouseAdapter
1101 {
1102    public AppletImage metaApplet;
1103
1104    public MyMouseAdapter3(AppletImage a)
1105    {
1106       metaApplet = a;
1107    }
1108
1109    public void mousePressed(MouseEvent e)
1110    {
1111       if (e.getButton() == MouseEvent.BUTTON1 && !metaApplet.running)
1112       {
1113          metaApplet.metaThread = new Theorem(metaApplet, 2);
1114          System.gc();
1115          metaApplet.metaThread.start();
1116       }
1117    }
1118 }
1119
1120
1121 class MyKeyAdapter4 extends KeyAdapter
1122 {
1123    public AppletImage metaApplet;
1124
1125    public MyKeyAdapter4(AppletImage a)
1126    {
1127       metaApplet = a;
1128    }
1129
1130    public void keyPressed(KeyEvent e)
1131    {
1132       if ((e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
1133          KeyEvent.VK_SPACE) && !metaApplet.running)
1134       {
1135          metaApplet.metaThread = new Theorem(metaApplet, 3);
1136          System.gc();
1137          metaApplet.metaThread.start();
1138       }
1139    }
1140 }
```

```
1141
1142
1143 class MyMouseAdapter4 extends MouseAdapter
1144 {
1145    public AppletImage metaApplet;
1146
1147    public MyMouseAdapter4(AppletImage a)
1148    {
1149       metaApplet = a;
1150    }
1151
1152    public void mousePressed(MouseEvent e)
1153    {
1154       if (e.getButton() == MouseEvent.BUTTON1 && !metaApplet.running)
1155       {
1156          metaApplet.metaThread = new Theorem(metaApplet, 3);
1157          System.gc();
1158          metaApplet.metaThread.start();
1159       }
1160    }
1161 }
1162
1163
1164 class MyKeyAdapter5 extends KeyAdapter
1165 {
1166    public AppletImage metaApplet;
1167
1168    public MyKeyAdapter5(AppletImage a)
1169    {
1170       metaApplet = a;
1171    }
1172
1173    public void keyPressed(KeyEvent e)
1174    {
1175       if ((e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
1176          KeyEvent.VK_SPACE) && !metaApplet.running)
1177       {
1178          metaApplet.metaThread = new Theorem(metaApplet, 4);
1179          System.gc();
1180          metaApplet.metaThread.start();
1181       }
1182    }
1183 }
1184
1185
1186 class MyMouseAdapter5 extends MouseAdapter
1187 {
1188    public AppletImage metaApplet;
1189
1190    public MyMouseAdapter5(AppletImage a)
1191    {
1192       metaApplet = a;
1193    }
1194
1195    public void mousePressed(MouseEvent e)
1196    {
1197       if (e.getButton() == MouseEvent.BUTTON1 && !metaApplet.running)
1198       {
1199          metaApplet.metaThread = new Theorem(metaApplet, 4);
1200          System.gc();
```

```
1201            metaApplet.metaThread.start();
1202         }
1203      }
1204   }
1205
1206
1207   class MyKeyAdapter6 extends KeyAdapter
1208   {
1209      public AppletImage metaApplet;
1210
1211      public MyKeyAdapter6(AppletImage a)
1212      {
1213         metaApplet = a;
1214      }
1215
1216      public void keyPressed(KeyEvent e)
1217      {
1218         if ((e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
1219            KeyEvent.VK_SPACE) && !metaApplet.running)
1220         {
1221            metaApplet.percentStatus.setText(" ");
1222            metaApplet.complete = true;
1223            metaApplet.k = Long.MIN_VALUE;
1224            metaApplet.modelCount = 0;
1225            metaApplet.counterexampleCount = 0;
1226         }
1227      }
1228   }
1229
1230
1231   class MyMouseAdapter6 extends MouseAdapter
1232   {
1233      public AppletImage metaApplet;
1234
1235      public MyMouseAdapter6(AppletImage a)
1236      {
1237         metaApplet = a;
1238      }
1239
1240      public void mousePressed(MouseEvent e)
1241      {
1242         if (e.getButton() == MouseEvent.BUTTON1 && !metaApplet.running)
1243         {
1244            metaApplet.percentStatus.setText(" ");
1245            metaApplet.complete = true;
1246            metaApplet.k = Long.MIN_VALUE;
1247            metaApplet.modelCount = 0;
1248            metaApplet.counterexampleCount = 0;
1249         }
1250      }
1251   }
1252
1253
1254   class MyKeyAdapter7 extends KeyAdapter
1255   {
1256      public AppletImage metaApplet;
1257
1258      public MyKeyAdapter7(AppletImage a)
1259      {
1260         metaApplet = a;
```

```
1261    }
1262
1263    public void keyPressed(KeyEvent e)
1264    {
1265       String clip = " &\n(";
1266       if (e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
1267          KeyEvent.VK_SPACE)
1268       {
1269          try
1270          {
1271             if (metaApplet.entry.getText().equals(""))
1272                clip = "(";
1273             metaApplet.entry.append(clip+Toolkit.getDefaultToolkit()
1274                .getSystemClipboard().getContents(" ").getTransferData(new
1275                DataFlavor(String.class, "class=java.lang.String")).toString()+
1276                ")");
1277          }
1278          catch (IOException e2){}
1279          catch (HeadlessException e3){}
1280          catch (UnsupportedFlavorException e4){}
1281       }
1282    }
1283 }
1284
1285
1286 class MyMouseAdapter7 extends MouseAdapter
1287 {
1288    public AppletImage metaApplet;
1289
1290    public MyMouseAdapter7(AppletImage a)
1291    {
1292       metaApplet = a;
1293    }
1294
1295    public void mousePressed(MouseEvent e)
1296    {
1297       String clip = " &\n(";
1298       if (e.getButton() == MouseEvent.BUTTON1)
1299       {
1300          try
1301          {
1302             if (metaApplet.entry.getText().equals(""))
1303                clip = "(";
1304             metaApplet.entry.append(clip+Toolkit.getDefaultToolkit()
1305                .getSystemClipboard().getContents(" ").getTransferData(new
1306                DataFlavor(String.class, "class=java.lang.String")).toString()+
1307                ")");
1308          }
1309          catch (IOException e2){}
1310          catch (HeadlessException e3){}
1311          catch (UnsupportedFlavorException e4){}
1312       }
1313    }
1314 }
1315
1316
1317 class MyKeyAdapter8 extends KeyAdapter
1318 {
1319    public AppletImage metaApplet;
1320
```

```
1321    public MyKeyAdapter8(AppletImage a)
1322    {
1323       metaApplet = a;
1324    }
1325
1326    public void keyPressed(KeyEvent e)
1327    {
1328       if (e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
1329          KeyEvent.VK_SPACE)
1330       {
1331          metaApplet.entry.setText(" ");
1332          metaApplet.entry.setText("");
1333       }
1334    }
1335 }
1336
1337
1338 class MyMouseAdapter8 extends MouseAdapter
1339 {
1340    public AppletImage metaApplet;
1341
1342    public MyMouseAdapter8(AppletImage a)
1343    {
1344       metaApplet = a;
1345    }
1346
1347    public void mousePressed(MouseEvent e)
1348    {
1349       if (e.getButton() == MouseEvent.BUTTON1)
1350       {
1351          metaApplet.entry.setText(" ");
1352          metaApplet.entry.setText("");
1353       }
1354    }
1355 }
1356
1357
1358 class MyKeyAdapter9 extends KeyAdapter
1359 {
1360    public AppletImage metaApplet;
1361
1362    public MyKeyAdapter9(AppletImage a)
1363    {
1364       metaApplet = a;
1365    }
1366
1367    public void keyReleased(KeyEvent e)
1368    {
1369       if ((e.getKeyCode() == KeyEvent.VK_ENTER || e.getKeyCode() ==
1370          KeyEvent.VK_SPACE) && !metaApplet.running)
1371       {
1372          metaApplet.message.setText(" ");
1373          metaApplet.message.setFont(metaApplet.myFont3);
1374          metaApplet.message.setText(metaApplet.helpMessage);
1375       }
1376    }
1377 }
1378
1379
1380 class MyMouseAdapter9 extends MouseAdapter
```

```
{
   public AppletImage metaApplet;

   public MyMouseAdapter9(AppletImage a)
   {
      metaApplet = a;
   }

   public void mousePressed(MouseEvent e)
   {
      if (e.getButton() == MouseEvent.BUTTON1 && !metaApplet.running)
      {
         metaApplet.message.setText(" ");
         metaApplet.message.setFont(metaApplet.myFont3);
         metaApplet.message.setText(metaApplet.helpMessage);
      }
   }
}


class MyWindowAdapter extends WindowAdapter
{
   public AppletImage metaApplet;

   public MyWindowAdapter(AppletImage a)
   {
      metaApplet = a;
   }

   public void windowClosing(WindowEvent e)
   {
      metaApplet.running = false;
      metaApplet.pF.dispose();
   }
}
```