ЛИТЕРАТУРА

1. Новые методы экспериментального определения критериальных параметров динамических систем приводных механизмов: Монография / Я.О. Берестнев, Н.Н. Ишин. – Мн.: УП «Технопринт», 2004. – 117с.

2. Барков А.В., Баркова Н.А., Азовцев А.Ю. Мониторинг и диагностика роторных машин по вибрации. – СПб.: Изд. Центр СПбГМТУ, 2000. – 169 с.

3. Барков А.В., Баркова Н.А. Интеллектуальные системы мониторинга и диагностики машин по вибрации. - http://www.vibrotek.com/russian/bio/anb.htm

4. http://www.vibrotek.ru

5. Выявление взаимосвязей показателей качества зубчатых колес и возникающих при их работе явлений для установления критериев оценки технического состояния и технических требований к системам сбора и обработки информации при проведении контроля. Отчет о научно-исследовательской работе (Промежуточный) / БрГТУ / Драган А.В. / № ГБ 06/615 / № госрегистрации 20062631. – Брест, 2006. – 43 с.

6. Драган А. В. Диагностика технического состояния прямозубых зубчатых передач с использованием современных средств кинематического контроля: Дис....канд. техн. наук 05.02.02/ИНДМАШ НАН Беларуси. – Минск, 2000. – 145 с.

7. Драган А.В., Стецко И.П., Ромашко Д.А., Левкович Н.В. Новые аппаратно-программные средства для исследования и диагностики механических систем // Вестник Брестского государственного технического университета. – 2006. - №4. – С. 17-26.

8. Alexander V.Dragan, Andrew S. Scorohodov, Vera S. Alexandrova Kinematic Criteria of Smoothness of Work of Cylindrical Gear Wheels // Proceedings of the 11th World Congress in Mechanism and Machine Science, Tianjin university, 1–4 april 2004 / Edited by Tian Huang. – Tianjin (China), 2004. – Vol. 2. – P. 773-776.

УДК 519.714.7

Сковородкин С.В.

Научный руководитель: Тузик И.В.

ЗАДАЧА О НАХОЖДЕНИИ МИНИМАЛЬНОЙ БИНАРНОЙ ДИАГРАММЫ РЕШЕНИЙ

В программировании один из этапов разработки программы – это её тестирование. Для составления тестовых вариантов, позволяющих протестировать работу программы, используется потоковый граф. Потоковый граф (см.[2]) представляет собой структуру, аналогичную блок-схеме, однако принципиально отличается от блок-схем представлением условий.

Потоковый граф обладает рядом особенностей:

- С помощью потокового графа можно проверить все варианты работы программы.
- Потоковый граф состоит из операторных и предикатных узлов. Из предикатных узлов выходит две дуги, а из операторных одна дуга. Ряд подряд идущих линейных операторов заключается в один операторный узел, а условия заключаются в предикатные узлы, при этом каждый из таких узлов содержит только одно простое условие.
- Предикатные узлы соответствуют простым условиям в программе. Сложное условие необходимо разбить на ряд простых условий. В случае разбиения сложного условия на простые возникает структура, состоящая из ряда предикатных узлов. Эта структура представляется в виде бинарной диаграммы решений (БДР).

Такая структура будет являться правильной, но не оптимальной, так как в большинстве случаев её можно будет минимизировать.

Пример представления сложного условия в виде предикатных узлов: имеем сложное условие вида «А или В»:

```
If A or B then p1
Else p2;
```

Его представление с использованием только простых условий может иметь следующий вид:

```
If A then
  If B then p1
  Else p1
Else
  If B then p1
  Else p2;
```

Однако это представление можно максимально упростить (а соответствующую ему БДР - минимизировать, то есть сократить в ней количество вершин):

```
If A then p1
Else
  If B then p1
  Else p2;
```

Либо, как вариант, возможна такая запись:

```
If A then f:=True
Else f:=B;
  If f then p1
  Else p2;
```

БДР булевой функции N переменных $f(v_1,..., v_N)$ представляет собой дерево со следующими свойствами:

- Вершины соответствуют переменным, от которых зависит функция, и расположены по уровням. Каждому уровню соответствует одна переменная.
- Из каждой вершины выходит две дуги. Одна соответствует нулевому значению переменной, а вторая - единичному.
- БДР имеет 2^N листьев, каждый из которых соответствует одному из значений функции.

Суть задачи о минимизации БДР состоит в том, чтобы минимизировать число вершин в БДР. При этом порядок расположения вершин по уровням может быть произвольный.

Существует ряд алгоритмов нахождения минимальной БДР. Один из них, описанный в [1], состоит в следующем. Для фиксированного порядка расположения переменных по уровням:

1. Строим полную БДР.
2. Двигаясь по уровням сверху вниз, для каждого уровня выполняем 2 действия:
   а) находим вершины, от прохождения которых не зависит значение функции, и удаляем их;
   б) находим одинаковые поддеревья и из нескольких одинаковых оставляем одно.
3. Приводим БДР к конечному виду (для заданного порядка переменных она будет содержать минимальное число вершин).
4. Меняем порядок переменных и выполняем шаги 1-3 для нового порядка расположения переменных по уровням. Эти действия выполняем для всех возможных перестановок переменных, от которых зависит функция.
5. Выбираем тот порядок расположения переменных по уровням, который будет оптимальным (т.е. при котором БДР функции будет содержать минимальное количество вершин), если же таких вариантов несколько, то выбираем любой из них.

В данной работе автором предлагается модификация этого алгоритма, основанная на минимизации количества дуг в БДР функции. Ниже приводится описание работы этого модифицированного алгоритма, с учетом его программной реализации.

1. Проводим проверку функции на фиктивность переменных.
2. Если все переменные фиктивны, то решением будет являться любой из наборов переменных по уровням, иначе избавляемся от фиктивных переменных (если таковые

имеются) и переходим к следующему шагу.

3. Переменной N присваиваем количество переменных, от которых зависит функция.

4. Переменной Min, которая будет содержать в себе число ветвей минимальной БДР, присваиваем количество ветвей полной БДР. Для функции N переменных эта величина рассчитывается по формуле:

$$\sum_{i=1}^{N} 2^i = 2^{N+1} - 2, \tag{1}$$

следовательно, Min= 2^{N+1}-2;

5. Находим количество ветвей для минимизированной БДР (при текущем порядке расположения переменных по уровням) по следующему алгоритму:

Переменной Count, которая будет содержать число ветвей БДР, присваиваем то же значение, что и переменной Min в пункте 4.

Строковой переменной S присваиваем значения заданной функции N переменных: S[1]=f(0,...,0), S[2]=f(0,...,0,1),...,S[2^N]=f(1,...,1). Например, для функции, представленной на рисунке 1, переменная S = '10100110'.

Для дальнейшего описания алгоритма введем два определения.

Определение 1. Две идентичные по своей структуре поддиаграммы, корни которых являются концами нулевой и единичной дуг одной и той же вершины любого уровня, будем называть «соседями».

Определение 2. Две идентичные по своей структуре поддиаграммы, корни которых являются концами дуг двух различных вершин одного и того же уровня, будем называть «не соседями».

На рисунках нулевые дуги подписаны словом «ложь», а единичные – словом «истина».

Двигаясь по уровням сверху вниз, начиная со 2-ого уровня (т.к. на первом уровне листья «соседей» не могут быть равны, потому что в пп.1-2 была проверка на фиктивность переменных), выполняем следующие действия:

а) Обрабатываем на i-м уровне одинаковые наборы, образованные значениями листьев «соседей», записав их в строковые переменные А и В.

В этой части алгоритма используем символ '-' (или любой другой символ, отличный от '0' и '1'), чтобы заменять им каждый символ в А или каждый символ в В, в случае, когда А=В. Присутствие в А или В символа '-' означает, что поддиаграммы, соответствующие позициям символа '-' в строке S, выбывают из рассмотрения как избыточные.

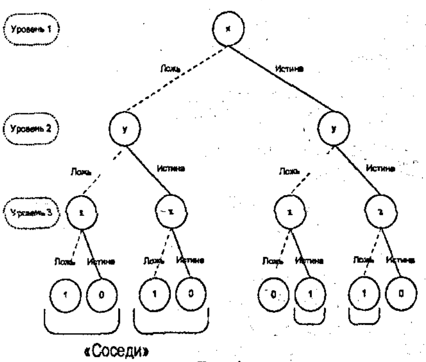

Рис.1

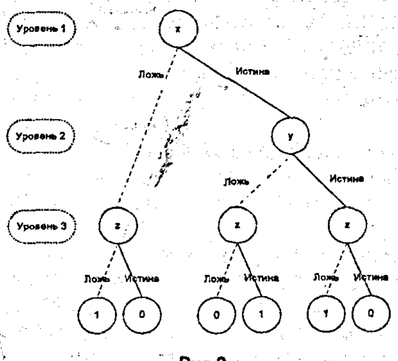

Рис.2

Если значения листьев «соседей» одинаковы (т.е. А=В) и в А и В нет символа '-', то «выбрасываем» избыточные поддиаграммы (записывая в соответствующие позиции строки символ '-'). Тогда количество ветвей в БДР становится равным

$$Count = Count - 2^{N+1-I}, \qquad (2)$$

где i – уровень, на котором ведется сравнение, а в строке S часть значений, соответствующая либо переменной A, либо переменной B, оказывается замененной символами '-'.

Например, на рисунке 1 видно, что на втором уровне есть «соседи» с идентичными значениями функции, следовательно, после удаления избыточных узлов, строка S примет вид '10--0110', а БДР будет выглядеть так, как представлено на рис. 2.

Используя такой подход, перебираем все «соседние» значения функции на i-ом уровне и в случае необходимости используем формулу (2).

б) Обрабатываем на i-м уровне одинаковые наборы, образованные значениями листьев «не соседей».

В случае, если текущей уровень не равен N (т.к. на уровне N, шаг 5б) уже не нужен) и такие наборы найдены, и если ни один из найденных наборов не состоит полностью из единиц или нулей (потому что такие наборы упрощаются на шаге 5а) на уровнях, ниже текущего, делаем замену соответствующих поддиаграмм одной эквивалентной. Это означает, что число ветвей становится равным

$$Count = Count - (2^{N+1-I} - 2), \qquad (3)$$

где i – уровень, на котором сравниваем значения листьев «не соседей», а в строке S все символы одного из сравниваемых наборов оказываются замененными символом '-'.

Строка S в нашем примере примет вид S='10--01--', а соответствующая БДР показана на рис. 3.

Таким образом, двигаясь по уровням сверху вниз и используя формулы (2) и (3), мы получаем число ветвей, которое будет соответствовать минимизированной БДР для данного порядка переменных.

6. Сравниваем полученное число Count с Min, и если Count≤Min, тогда значению Min присваиваем значение Count: Min=Count

7. Если ещё есть порядки расположения переменных по уровням, для которых мы ещё не находили число ветвей минимизированной БДР, то текущему порядку расположения переменных по уровням присваиваем следующий порядок расположения переменных по уровням и пересчитываем для него значения функции.

Иначе ответом будет являться тот порядок расположения переменных по уровням, которому соответствует минимальное количество ветвей (Min).

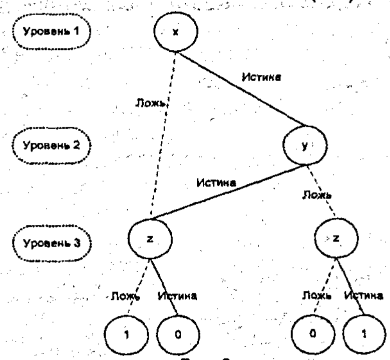

Рис. 3

Новому порядку расположения переменных по уровням будет соответствовать таблица значений R, а исходному порядку $f(v_1,...,v_N)$ будет соответствовать таблица A. Если для нового порядка расположения переменных по уровням переменной первого уровня будет соответствовать переменная v_i, $i=\overline{1,N}$, то первому столбцу таблицы R присвоим значения i-ого столбца таблицы А. Аналогичные действия проведём для оставшихся N-1 столбцов таблицы R. Затем, когда всем столбцам будут присвоены значения, вычислим значения функции для нового порядка расположения переменных: для i-ой строки, $i=\overline{0,N-1}$, запишем слева направо значения её переменных, образуя, таким образом, число в двоичной форме записи. Затем переводим это число в десятичную форму записи и получаем номер строки j, $j=\overline{0,N-1}$, которая будет содержать значение функции для строки i, таблицы R, в столбце значений таблицы А.

Предложенный алгоритм позволяет найти порядок расположения переменных по уровням, при котором БДР будет минимальной. Алгоритм имеет экспоненциальную сложность, как и другие существующие в данное время алгоритмы решения этой задачи (см. [1], [3]). Данный алгоритм не является универсальным средством решения задач данного типа, т.к. он ограничен вычислительной мощностью компьютера, а также имеет ограничения, связанные со средой разработки программ, в которой реализуется алгоритм.

Алгоритм может быть применён при автоматизированной оптимизации структуры программ.

ЛИТЕРАТУРА

1. Карпов, Ю.Г. Теория автоматов/ Ю.Г. Карпов – СПб.: Питер, 2002.
2. Орлов, С.А. Технологии разработки программного обеспечения. Учебное пособие. / С.А. Орлов. – СПб.: Питер, 2003.
3. Prasad, P.W.C. Binary Decision Diagrams: An Improved Variable Ordering using Graph Representation of Boolean Functions/ P.W.C. Prasad, A.Assi, A. Harb, V.C. Prasad – International Journal of Computer Science Volume 1 Number 1.

УДК 621.9.06

Рудиюк А.Н.

Научный руководитель: доцент Горбунов В.П.

ВЛИЯНИЕ КОНСТРУКЦИИ ПЕРЕДАЧИ ВИНТ-ГАЙКА КАЧЕНИЯ НА ТОЧНОСТЬ КООРДИНАТНЫХ ПЕРЕМЕЩЕНИЙ РАБОЧИХ ОРГАНОВ В СТАНКАХ С ЧПУ

В настоящее время одной из основных целей в технологии механической обработки является достижение требуемой точности изготовления деталей: точность её размеров, формы, взаимного расположения и шероховатость поверхности. К тому же ужесточились и сами параметры точности. Данные параметры точности обработки деталей формируются точностью самого станка, а именно – геометрической точностью, кинематической точностью, жёсткостью и точностью позиционирования.

В автоматизированном производстве наибольшее значение имеет использование станков с ЧПУ, где точность обработки должна обеспечиваться автоматически за счёт: точного перемещения рабочих органов станка, использования систем автоматического управления и других факторов [3].

Точность координатных перемещений на станках с ЧПУ характеризуются точностью позиционирования $\Delta_{поз}$, под которой понимается отклонение действительного положения рабочего органа станка X_i от запрограммированного $X_{прог}$ при его многократном двухстороннем позиционировании в различных точках по пути его перемещения по одной из координатных осей [2]. Точность позиционирования формируется всем комплексом станка с ЧПУ (его механической частью и системой управления) и зависит от многих факторов: погрешности блоков и элементов устройства ЧПУ, погрешности привода подач, геометрических погрешностей станка, погрешностей измерительных преобразований и др.