МОДЕЛЬ И ПОДХОД К СИНТЕЗУ УПРАВЛЕНИЯ ДИСКРЕТНОЙ СИСТЕМОЙ

Садовская О. И., Герман О. В.
Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: ign_olga@mail.ru, ovgerman@tut.by

Рассматривается задача синтеза управления в дискретной системе многофазного многопроцессного типа, которая обобщает задачу Джонсона об $n > 1$ станках в нескольких аспектах. Делается попытка свести задачу к случаю $n = 2$ станкам. Излагается общий принцип решения и его реализация.

Постановка задачи

Пусть дано множество процессов $P = \{p_1, p_2, \ldots, p_z\}$ и технологическая линия, представленная упорядоченным множеством ресурсов $R = \{r_1, r_2, \ldots, r_k\}$. Каждый процесс p_i имеет в общем случае собственную программу прохождения ресурсов Sh_i. Программа Sh_i задается последовательностью номеров ресурсов с указанием времени, в течение которого данный процесс удерживается на этом ресурсе. Ресурсы могут повторяться (и не один раз) в технологических программах ресурсов. В системе может одновременно находиться несколько процессов, но в каждый момент времени на каждом ресурсе может находиться не более одного процесса. Элементы программ Sh_i будем называть фазами. Таким образом, правомочно говорить «на фазе k процесс p_i использует ресурс r_m», что означает следующее: k -м элементом Sh_i является r_m. Состоянием (маркировкой) $S(t)$ системы в момент t назовем распределение процессов по ресурсам системы с указанием времен, определяющих сколько им осталось на занимаемых ресурсах находиться. Задача состоит в том, чтобы определить последовательность номеров ресурсов и моменты их запуска в систему, которая доставляет минимальное время выполнения всех процессов из P. Эта задача обобщает известную задачу Джонсона об $n > 1$ станках [1] в нескольких аспектах. Во-первых, процессы могут реализовывать несовпадающие технологические программы Sh_i. Во-вторых, могут возникать состояния тупиков – блокировок, при которых невозможно дальнейшее перемещение процессов по технологической линии. В-третьих, времена использования даже однотипных ресурсов у разных процессов могут быть разными. Таким образом, имеем в некотором смысле обобщенную задачу Джонсона.

Имеются эвристические алгоритмы для решения задачи Джонсона. В частности, делается попытка свести задачу к случаю $n = 2$ станкам. Однако, ввиду указанных выше специфических особенностей, этот подход не работает в нашем случае. Далее излагается общий принцип решения и его реализация. В этом изложении мы упрощаем содержание задачи допущением, что время захвата ресурса любым процессом одно и то же и равно единице. Акцент в данной работе делается на предупреждение тупиков и поиск расписания, при котором обеспечивается наивысшая производительность системы.

I. Описание подхода к решению

Прежде всего, имеем в любом состоянии несколько возможных продолжений, что можно описать системой логических формул (продукций) следующего вида: $q_j(t,k)\&Sh_i(t+1,m)\&fr(t,m) \rightarrow notfr(t+1,m)\&fr(t+1,k)\&not(q_j(t+1,k))\&q_j(t+1,m)$. Эта формула читается следующим образом: если на такте t процесс i удерживает ресурс k $(q_j(t,k))$ и на следующем такте ему требуется ресурс $m(Sh_i(t+1,m))$, причем этот ресурс свободен $fr(t,m)$, то процесс переходит на следующую фазу, освобождая текущий ресурс и захватывая следующий: $notfr(t+1,m)\&fr(t+1,k)$ соответственно изменяя свое состояние в системе: $not(q_j(t+1,k))\&q_j(t+1,m)$. Обратим внимание, что для всех процессов будем иметь семейство формул данного типа, причем t играет роль временного параметра. Второй аргумент формулы, если он имеется, выполняет роль константы. Нам остается передать факт, что на каждом ресурсе может быть не более одного процесса (за исключением конечного накопительного ресурса F) и что один и тот же процесс не может быть одновременно на нескольких ресурсах. Первое условие передается таким образом: для любых несовпадающих i, j $q_i(t,k)\&q_j(t,k) \rightarrow false$. Второе условие передается таким образом: для любых несовпадающих k, l $q_i(t,k)\&q_i(t,l) \rightarrow false$. Наконец, последняя группа условий означает, что как минимум один процесс должен перейти на следующий шаг: $fr(t,m)\&not - fr(t+1,m) \vee not - fr(t,m)\&fr(t+1,m)$. Полученная таким образом логическая спецификация «схватывает» функционирование системы во времени. Исходное состояние системы можно передать как $S_0 = q_1(0,0)\&q_2(0,0)\&\ldots\&q_z(0,0)\&fr(0,1)\&fr(0,2)$ $\&\ldots\&fr(0,k)$. Конечное состояние можно передать формулой $q_1(n,F)\&q_2(n,F)\&\ldots\&qz(n,F)$ $\&n > 0$.

Для решения задачи можно использовать модифицированный метод подстановок [3]. Суть этого метода заключается в том, что отыскиваются подстановки для переменных, определенных для момента времени $t + 1$ относительно значений в момент t. С помощью этих подстановок можно последовательно строить системы формул (дизъюнктов), определяющих решения в моменты $t = 1$, $t = 2$, $t = 3$ и т.д. Относительно этих решений устанавливаем, выполнима ли в них формула для конечного состояния, т.е. $q_1(n, F)\&q_2(n, F)\&\ldots\&q_z(n, F)\&n > 0$. Нетрудно заметить, что процесс порождения новых систем гарантированно сходится за конечное число шагов к результату при условии неповторения ранее пройденных состояний, так как число различных состояний дискретной системы фиксированно. Описанный подход имеет ощутимое преимущество в сравнении с известным методом на основе определения булевого дифференциала [2], поскольку последний отыскивает все возможные допустимые состояния системы в явном виде, чтобы построить граф переходов на множестве состояний системы. Число же состояний в реальных задачах может быть огромным. Таким образом, реализованный подход может быть использован для задачи синтеза управления в дискретной системе многофазного многопроцессного типа.

II. Программная модель

Программная модель реализована на базе C#. В модели порождаются транзакты-состояния в соответствии с продукциями. Каждый транзакт (объект) несет значение некоторой маркировки системы. Ядро модели составляет цикл, в котором последовательно просматривается коллекция транзактов, из которых выбирается «наиболее перспективный». Для этого транзакта порождаются новые транзакты-состояния согласно логике системы при условии, что новый транзакт свободен от блокировок, не повторяет ранее пройденных состояний, и функция-оценка достигаемой в этом состоянии производительности системы не ниже минимальной. Начиная с момента насыщения модели процессами, транзакты с минимальной достигнутой производительностью удаляются из коллекции. Порожденные транзакты добавляются в коллекцию транзактов и процесс ведется до тех пор, пока коллекция не станет пустой либо не будет достигнуто конечное состояние.

III. Список литературы

1. Михайлова, Л. В. Формирование и оперативное управление производственными системами на базе поточно-группового производства в автоматизированном режиме / Л. В. Михайлова, Ф. И. Парамонов, А. В. Чудин –Москва: ИТЦ МАТИ, 2002. – 60 с.
2. Бохманн, Д. Двоичные динамические системы / Д. Бохманн, Х. Постхофф. Москва: Энергоатомиздат, 1986. – 400 с.
3. Герман, О. В. Метод решения задачи динамической булевой оптимизации / О. В. Герман, О. И. Садовская // Информационные технологии и системы 2012 (ИТС 2012): материалы междунар. науч. конф., Минск, 24 октября 2012г. / БГУИР; редкол.: Л. Ю. Шилин [и др.]. –Минск, 2012. – С. 38-40.