

Моделирование схемных управляющих систем

Ю. С. Шуткин

Рассматривается общая проблема построения различных типов схемных управляющих систем и некоторые функционалы их сложности. Особое внимание уделяется сложности моделирования схем на компьютере. Предложен общий метод моделирования схем, реализующих булевы функции.

Ключевые слова: схемы из функциональных элементов, контактные схемы, сложность управляющих систем, моделирование.

1. Введение

Управляющие системы — понятие в кибернетике, используемое для обозначения класса объектов, описывающих способ вычисления некоторого заданного функционала с помощью определенного набора заданных базисных элементов. Как правило, имеющиеся в распоряжении базисные элементы представляют из себя некоторые максимально простые логические объекты, которые имеют определенную интерпретацию на практике, или же являются описанием некоторых реальных физических объектов.

Далее, имея в распоряжении набор базисных элементов и способ построения из них более сложных систем, мы получаем так называемую задачу синтеза управляющей системы, то есть получения требуемого функционала из имеющихся простейших функциональных частиц путем соединения их между собой. При этом мы хотим получить в некотором смысле оптимальную систему, то есть обладающую наилучшими свойствами из всех систем, построенных аналогичным путем и реализующих аналогичный функционал. Для этого вводится понятие сложности управляющей системы. Функционал, задающий

эту сложность, во-первых, должен позволять сравнить любые две системы между собой, таким образом, что мы всегда сможем сказать, что одна из них лучше другой в смысле этого функционала сложности, а во-вторых, он должен максимально точно описывать некоторое реальное свойство системы, которое нам интересно, и которое мы собираемся оптимизировать. В качестве такого функционала может выступать, например, размер системы.

Управляющие системы повсеместно используются в среде информационных технологий, как в области программного обеспечения, так и в hardware. Самыми общеизвестными примерами управляющих систем являются схемы из функциональных элементов (СФЭ, boolean circuits) — схемы, используемые для описания чипов, реализующих булевы функции; контактные схемы (contact scheme, contact network, contact circuit, switching circuit) — самый первый класс управляющих систем, придуманных Клодом Шенноном в 40-х годах XX века.

Условно можно поделить управляющие системы на два класса — системы с памятью (автоматы, программы, машины Тьюринга) и системы без памяти (схемы из функциональных элементов, контактные схемы, неветвящиеся программы). Второй тип систем называется также схемными управляющими системами. В рамках данной статьи мы более подробно будем рассматривать как раз этот тип управляющих систем.

Второй способ классификации управляющих систем состоит в определении функционала, который они реализуют. Это могут быть некоторые функции в определенном пространстве, алгоритмы вычисления, алгоритмы распознавания и т. д. Самым простым и базовым функционалом являются так называемые булевы функции — функции, принимающие на вход векторы из нулей и единиц и выдающие на выход также нули и единицы. Такой класс функций также хорош тем, что он достаточно хорошо подходит для описания многих физических процессов, таких, как передача сигнала (наличие или отсутствие сигнала), в теории информации нули и единицы кодируют биты информации.

2. Функционалы сложности управляющих систем

Изучение схемных управляющих систем начинается в еще в первой половине XX века, когда появляется необходимость математически описывать различные разработки в области электроники. Так, в 1938 году выходит работа Клода Шеннона о релейных схемах в электронике [15].

В России первые упоминания о схемных управляющих системах появляются несколько позже в работах О. Б. Лупанова [5] и С. В. Яблонского [13].

Основной целью этих работ является построение и анализ схем с наименьшим объемом, то есть схем, число элементов в которых минимально. Такой функционал схемной сложности называется объемной сложностью. Также в данных работах приводятся первые оценки объемной сложности схем.

Среди оценок сложности выделяют два типа оценок — верхняя оценка сложности задачи синтеза, которая характеризует максимальную сложность схемы, заведомо достаточную для реализации любой функции из определенного класса (как правило, класса всех булевых функций), а также нижняя оценка сложности, которая говорит о том, какое минимальное число элементов необходимо для того, чтобы реализовать самую сложную функцию из класса.

Для более точной формализации введенных выше понятий верхней и нижней оценки вводится так называемая функция Шеннона сложности.

Пусть $L(S)$ — сложность схемы S , которая реализует некоторую булеву функцию (б. ф.) f .

Сложностью б. ф. f (или сложностью реализации б. ф. f) назовем следующую величину.

$$L(f) = \min_{S \in H(f)} L(S),$$

где $H(f)$ — множество всех схем, реализующих функцию f .

Функцией Шеннона сложности реализации булевых функций схемами назовем величину

$$L(n) = \max_{f \in P_2^{(n)}} L(f),$$

где $P_2^{(n)}$ — класс всех булевых функций, зависящих не более, чем от n переменных.

Если мы хотим рассматривать задачу синтеза лишь для ограниченного класса A булевых функций, можем ввести функцию Шеннона для этого класса следующим образом.

$$L_A(n) = \max_{f \in A^{(n)}} L(f).$$

Таким образом, задача оценки сложности сводится к оценке функции Шеннона, соответствующей классу схем, выбранному функционалу сложности и классу булевых функций. Верхние оценки функции Шеннона как правило получаются за счет предъявления явного алгоритма построения схемы по заданной булевой функции, а нижние оценки получаются путем построения последовательности сложных функций с растущим числом аргументов, либо на основе мощностных соображений.

Для объемной сложности наиболее сильные оценки сложности были получены О. Б. Лупановым [6]. Более подробно о работах Лупанова в целом и об оценках объемной сложности можно посмотреть в [8].

В данной работе мы попытаемся сделать обзор всех остальных функционалов сложности, отличных от объемной сложности, а также обзор результатов, полученных для этих функционалов. Особенно нас будет интересовать так называемая «временная» сложность, которая характеризует скорость моделирования схемы на компьютере.

Итак, кроме объемной сложности достаточно интенсивно изучались следующие функционалы.

В качестве обобщения объемной сложности для схем из функциональных элементов рассматривался функционал взвешенной объемной сложности, когда каждый базовый элемент из набора имеет свой собственный вес [6]. Такое обобщение является вполне естественным, так как в реальных схемах элементы действительно могут иметь различную площадь.

Для схем из функциональных элементов рассматривалось такое понятие, как глубина схемы [7, 2] и ее обобщение — задержка схемы [7, 9]. Эти две величины отвечают за отклик схемы на входные данные, то есть время, через которое на выходе схемы появится вы-

ходное значение, если считать, что на все входы схемы значения поступают одновременно. Легко заметить, что такой функционал также очень важен при моделировании на компьютере, особенно если вычисление базисных элементов осуществляется параллельно.

Еще одним функционалом сложности схем из функциональных элементов является мощность схемы. Основные результаты для данного функционала были получены М. Н. Вайнцвайгом [1] и О. М. Касим-Заде [4]. О. М. Касим-Заде также рассматривал вопрос об одновременной оптимизации объемной и мощностной сложности, то есть вопрос построения схем, одновременно оптимальных по этим двум функционалам [3]. Заметим, что получение таких оценок является более сложным по сравнению с оценками для одного функционала, а иногда и вообще невозможным.

Если пытаться понять, какой функционал сложности будет описывать скорость моделирования схем из функциональных элементов, то окажется, что ничего лучше объемной сложности на эту роль не подходит, так как для вычисления значения функции мы должны вычислить значения всех входов последнего функционального элемента, для него мы должны знать выходы всех предпоследних элементов и так далее, до самих входов схемы, то есть мы должны вычислить абсолютно все элементы схемы.

Получается, что моделирование схем из функциональных элементов в том виде, в котором они определяются классически, является экспоненциально сложной задачей от числа переменных булевой функции (так как объем схемы растет экспоненциально от числа переменных). В связи с этим возникает необходимость модифицировать определение схемы из функциональных элементов таким образом, чтобы оно допускало оптимизацию при моделировании.

Одной из попыток такой модификации было рассмотрение неветвящихся программ с условной остановкой А. В. Чашкиным [10]. В данной работе вводится дополнительный оператор условной остановки, который позволяет выдавать значение схемы, даже если еще не все элементы схемы вычислены, однако выход схемы уже определен однозначно. Однако А. В. Чашкин показал, что для почти всех булевых функций данного оператора недостаточно, чтобы снизить сложность схем до неэкспоненциальной, порядок модифицированной сложности схем по-прежнему растет как экспонента.

Данный эффект удалось преодолеть для другого класса схем — для контактных схем.

Контактные схемы отличаются от схем из функциональных элементов тем, что для них нет необходимости вычислять всю схему целиком, чтобы получить значение схемы на входном наборе, так как необходимо вычислить только те части схемы, до которых доходит сигнал. Оказывается, что доля контактов, до которых сигнал действительно доходит на одном конкретном входном наборе может быть гораздо меньше объема всей схемы, что позволяет осуществить моделирование схемы за гораздо меньшее число вычислений, чем ее объем [12].

Более того, автором показано, что существует метод синтеза контактных схем, одновременно асимптотически оптимальный как в смысле объемной сложности, так и в смысле скорости моделирования, то есть временной сложности [11].

Кроме того, функционал сложности, рассматриваемый в работе [12], также можно интерпретировать, как мощность контактной схемы, то есть среднее число возбужденных контактов в схеме, что будет аналогом мощности схем из функциональных элементов, рассмотренной в [4]. Как и в работе О. М. Касим-Заде, оценка мощностной сложности получается линейной от числа переменных функции.

Основной идеей метода синтеза, дающего линейную оценку временной сложности [12], является направление сигнала в определенное очень узкое русло внутри контактной схемы путем присоединения к каждой вершине схемы противоположных контактов, лишь один из которых может пропускать входной сигнал.

Такой метод соответствует разложению булевой функции по ее переменным и, вообще говоря, не зависит от определенной схемной реализации функции, то есть может быть применен также и для любого другого класса схем, допускающего такое разделение. Основной чертой данного метода является то, что каждый элемент входного вектора участвует в вычислении не более ограниченного числа раз, то есть каждая переменная проверяется конечное число раз. Это и гарантирует линейную зависимость сложности моделирования от числа переменных.

Например для бинарных диаграмм решений (ordered binary decision diagrams) [14] такой подход изначально заложен в строение схе-

мы, что гарантирует вычисление ответа за линейное время от числа переменных.

Для схем из функциональных элементов также возможна модификация, когда каждая переменная участвует в вычислении всего один раз. Если предположить, что мы реализуем метод разложения функции по ее переменным, то есть представляем функцию в виде

$$f(x_1, x_2, \dots, x_n) = x_1 f(1, x_2, \dots, x_n) \vee \bar{x}_1 f(0, x_2, \dots, x_n),$$

и потом реализуем функции $f(1, x_2, \dots, x_n)$ и $f(0, x_2, \dots, x_n)$ отдельно, то получается, что нам необходимо уметь пропускать вычисления одной из них в зависимости от значения переменной x_1 на входном наборе, то есть если значение переменной равно 1, то вычислять требуется только значение функции $f(1, x_2, \dots, x_n)$, а если оно равно 0, то только значение функции $f(0, x_2, \dots, x_n)$. Таким образом, для получения линейной сложности нам необходимо лишь уметь «пропускать» вычисления ненужных нам функций.

Если вернемся к представлению схемы из функциональных элементов в виде неветвящейся программы, то есть последовательности операторов, входы которых являются выходами предыдущих операторов или входами схемы (а такое представление, очевидно, возможно), то добавление оператора *goto* в набор возможных операторов позволит составлять неветвящиеся программы таким образом, что мы сможем пропускать ненужные нам части с помощью этого оператора. Легко видеть, что это даст нам линейную сложность моделирования схемы в силу ранее озвученных утверждений.

Однако стоит заметить, что данный метод синтеза неветвящихся программ, примененный в лоб, скорее всего даст объемную сложность, отличную от оптимальной, и вопрос одновременной оптимизации объемной и временной сложности по-прежнему остается открытым.

3. Заключение

Замечено, что для достижения линейной от числа переменных скорости моделирования булевых функций достаточно применять при синтезе схем метод разделения по переменным реализуемой

функции. Это гарантирует направление потока вычислений лишь в ограниченной русло, сильно отличное от объема всей схемы.

Данный метод напрямую применяется для бинарных диаграмм решений, для контактных схем, а также может быть применен для любых других классов схем с помощью несущественной модификации этого класса. В качестве примера в работе рассмотрено применение метода для схем из функциональных элементов.

Заметим, что построение схем, допускающих быстрое моделирование, является очень эффективным при сложной разработке, так как не требует последующей верификации модели, как в случае, когда схема строится отдельно, а «быстрая» модель уже существует до или разрабатывается после этого.

Список литературы

- [1] Вайнцвайг М. Н. О мощности схем из функциональных элементов // ДАН СССР. — 1961. Т. 139, № 2. — С. 320–323.
- [2] Гашков С. Б. О глубине булевых функций // Проблемы кибернетики. — 1978. Т. 34. — С. 265–268.
- [3] Касим-Заде О. М. Об одновременной минимизации сложности и мощности схем из функциональных элементов // Проблемы кибернетики. — 1978. Т. 33. — С. 215–220.
- [4] Касим-Заде О. М. Об одной мере сложности схем из функциональных элементов // Проблемы кибернетики. — 1981. Т. 38. — С. 117–179.
- [5] Лупанов О. Б. О вентильных и контактно-вентильных схемах // ДАН СССР. — 1956. Т. 111, № 6. — С. 1171–1174.
- [6] Лупанов О. Б. О синтезе некоторых классов управляющих систем // Проблемы кибернетики. — 1963. Т. 10.
- [7] Лупанов О. Б. О схемах из функциональных элементов с задержками // Проблемы кибернетики. — 1970. Т. 23. — С. 73–97.
- [8] Памяти Олега Борисовича Лупанова (1932–2006) // Дискретн. анализ и исслед. опер. Сер. 1. — 2007. Т. 14, № 1. — С. 3–20.
- [9] Храпченко В. М. Различие и сходство между задержкой и глубиной // Проблемы кибернетики. — 1979. Т. 35. — С. 141–168.

- [10] Чашкин А. В. О среднем времени вычисления значений булевых функций // Дискретн. анализ и исслед. опер. — 1997. Т. 4, № 1. — С. 60–78.
- [11] Шуткин Ю. С. Об одновременной минимизации объемной и временной сложности контактных и вентильных схем // Интеллектуальные системы. — 2010. Т. 14, вып. 1–4. — С. 595–615.
- [12] Шуткин Ю. С. Асимптотически оптимальная реализация булевых функций информационными графами // Дискретная математика. — 2011. Т. 23, № 4. — С. 80–102.
- [13] Яблонский С. В. Основные понятия кибернетики // Проблемы кибернетики. — 1959. Т. 2. — С. 7–38.
- [14] Lee C. Y. Representation of Switching Circuits by Binary-Decision Programs // Bell Systems Technical Journal. — 1959. Vol. 38. — P. 985–999.
- [15] Shannon C. E. A Symbolic Analysis of Relay and Switching Circuits // Trans. AIEE. — 1938. Vol. 57, no. 12. — P. 713–723.