

Б.П. Кузнецов об автоматном программировании

Автоматное программирование ничуть не лучше любого другого ни по числу допускаемых ошибок, ни по срокам трудоемкости создания и отладки программ. В этом я убедился и в период активного использования как табличного, так и спискового (Switch, Любченко, Зюбин) задания автоматов в программах (к месту и не к месту) и в последующий период «безавтоматного программирования» с 2003 г. по нынешний день (да и до этого, когда «вспомнил» об автоматах). Более того, автоматное программирование доступно отнюдь не большинству, менее понятно и более трудоемко.

Перечислю типичные ошибки, сопровождающие автоматное программирование, как из собственного, так и заимствованного опыта, только лишь на примере составления диаграммы состояний конечного автомата:

1. Не учтенные состояния автомата, вызванные незнанием предметной области;
2. Дублирование (избыточность) состояний, приводящее к непредсказуемому поведению программы;
3. Не учтенные переходы;
4. Лишние переходы;
5. Неверно ориентированные переходы;
6. Неортогональность входного алфавита;
7. Неверное назначение приоритетов переходов при неортогональном алфавите;
8. Не учтенные входные воздействия и неполнота входного алфавита;
9. Не полный учет букв входного алфавита;
10. Путаница, связанная с неверным отождествлением входных воздействий и буквами входного алфавита (наиболее распространенная ошибка);
11. Неверные булевы формулы, отождествляемые с буквами входного алфавита и помечающие переходы (см. п.6);
12. Не учтенные выходные воздействия, в особенности как реакция на ошибочное поведение управляемого объекта и самого управляющего автомата и связи управляющего и операционного автоматов с объектом управления;
13. Неверная пометка состояний и переходов буквами выходного алфавита;
14. Неверное отождествление букв выходного алфавита с выходными воздействиями;
15. Забытое обнуление или продление выходного воздействия;
16. Путаница в отметке переходов и состояний буквами выходного алфавита при использовании совмещенной модели (Мили и Мура) автоматов;
17. Не прослеживаются полные пути в диаграмме состояний;
18. Всевозможные ошибки в полных путях при использовании обратных связей в диаграмме состояний
19. Другие ошибки.

Только одного этого перечня вполне достаточно, чтобы скомпрометировать «непорочность» автоматного программирования. И его мнимые достоинства связаны с тем, что «каждый кулик хвалит свое болото», чем и я в свое время безапелляционно сообщал научному и инженерному сообществам в своих публикациях (см., например, мою статью «Психология автоматного программирования»).

PS. Мне кажется, что приведенный длинный перечень возможных ошибок только подтверждает такое достоинство автоматного программирования как формализация задания логики программы. Все изложенное можно проверять (автоматически и вручную), повышая качество программы. Интересно, как бы выглядел этот перечень для программ, которые пишутся традиционно? Мне кажется, что была бы одна строчка – в логике программы могут быть ошибки... Что с этим делать? А то, что автоматные программы, в отличие написанных иначе, удобно верифицировать методом *Model Checking* – это разве не достоинство. Графы переходов можно обсуждать с Заказчиками, а программы нельзя. И т. д., и т. п. А.А. Шалыто