Слайд 1. Название доклада.

Обобщенная модель системы автоматов.

Слайд 2. Напоминание.

Я напомню, что в первом докладе мы рассматривали составную систему, структура связей которой моделировалась ориентированным графом связей. Компоненты системы располагались в вершинах графа и моделировались конечными автоматами. Взаимодействие между автоматами понималось как обмен сообщениями по симплексным каналам связи, которым соответствовали дуги графа.

Дуга представляла собой очередь сообщений длины 1, то есть она либо была пуста, либо на ней находилось одно сообщение. Одна вершина графа была выделена и моделировала окружение системы. Дуга, ведущая из окружения в систему, называлась внешней входящей дугой; по ней окружение могло выдавать сообщения в систему. Дуга, ведущая из системы в окружение, называлась внешней выходящей дугой; с этих дуг окружение могло принимать сообщения от системы.

Наша задача сейчас – устранить недостатки модели системы, рассмотренной в 1-ом докладе.

1. Нужно так определить композицию, чтобы композиционный автомат можно было использовать как компонент более сложной системы.

2. Обобщить понятие дуги так, чтобы она реализовывала не только очередь длины 1, но также очередь любой конечной длины, очередь с приоритетами, стек и т.д. В общем, дуга – это будет некоторый автомат-посредник между автоматами, находящимися в вершинах.

3. Обобщить автомат в вершине так, чтобы он мог моделировать условный приём сообщений и условную выдачу сообщений. В частности, это позволит моделировать приём по приоритету и выдачу не всех, а части сообщений.

Слайд 3. Автомат дуги.

Дугу можно понимать как автомат с одним входом и одним выходом, который взаимодействует с двумя автоматами в вершине начала дуги и в вершине конца дуги. Это взаимодействие синхронно.

Вот здесь на слайде изображён граф переходов автомата дуги как очереди длины 1. У него есть состояние «пусто» и по одному состоянию на каждое сообщение, которое может находиться на дуге. В состоянии «пусто» дуга может принять сообщение m_i на свой вход и перейти в состояние m_i. В состоянии m_i дуга может выдать сообщение m_i по своему выходу и перейти в состояние «пусто».

В дальнейшем мы будем считать, что автомат дуги может быть и другим. Чтобы изобразить это, нарисуем на каждой дуге кружок, означающий автомат этой дуги.

Слайд 4. Автомат дуги.

У нас было окружение, 4 автомата в вершинах и 9 дуг. Теперь будет окружение 4+9=13 автоматов и 9*2=18 соединений между ними. Каждое такое соединение связывает выход одного автомата со входом другого автомата. Это либо соединение выхода автомата вершины со входом автомата выходящей дуги, либо соединение выхода автомата дуги со входом автомата вершины, для которой эта дуга входящая.

И вот теперь, до поры до времени, мы уже не будем различать автомат в вершине и автомат дуги. Более того, мы будем считать, что автомат дуги может иметь несколько входов и несколько выходов, как и автомат вершины.

После этого вершина графа связей соответствует автомату, а дуга графа связей соответствует соединению выхода одного автомата со входом другого автомата. Взаимодействие по соединению происходит синхронно.

Слайд 5. Обобщение автомата.

В конце первого доклада было упомянуто несколько примеров поведения автомата, которые не описываются в той модели автомата, которую мы использовали. Было предложено обобщить автомат, введя условный приём и условную выдачу сообщений.

Я напомню эти примеры.

Пример условного приёма сообщений – это приём сообщений по приоритету: принимается то сообщение, которое имеет наивысший приоритет среди всех тех сообщений, который автомат мог бы принять.

В общем виде условный приём сообщений означает, что автомат «приписывает» каждому своему входу некоторое сообщение или указывает, что сообщения нет. Если сообщения нет, то мы будем говорить, что есть пустое сообщение, которое будем обозначать буквой Λ. Такую «приписку» каждому входу своего сообщения будем называть *стимулом* и обозначать буквой x.

Отдельно автомат указывает те входы, с которых он принимает приписанные сообщения, эти сообщения должны быть непустыми. Множество этих входов будем называть *параметром приема* и обозначать буквой p.

Пример условной выдачи сообщений – это когда автомат хотел бы выдать несколько сообщений по разным своим выходам, но не все эти сообщения можно выдать. В той модели автомата, которая рассматривалась в первом докладе, в этой ситуации автомат не может сделать переход и стоит. Теперь же мы разрешим выдачу части сообщений.

В общем виде условная выдача сообщений означает, что автомат «приписывает» каждому своему выходу некоторое сообщение, быть может, пустое сообщение Λ. Такую «приписку» каждому выходу своего сообщения будем называть *реакцией* и обозначать буквой y.

Отдельно автомат указывает те выходы, по которым он выдаёт приписанные сообщения, эти сообщения должны быть непустыми. Множество этих выходов будем называть *параметром выдачи* и обозначать буквой q.

Формально автомат определяется на слайде. Автомат задаётся конечным множеством сообщений M, конечным множеством входов I, конечным множеством выходов J, конечным множеством состояний S, множеством переходов T и начальным состоянием s_0. Предполагается, что множества входов и выходов не пересекаются.

Слайд 6. Обобщение автомата.

Переход, ведущий из пресостояния s в постсостояние t, помечается стимулом x, параметром приема p, реакцией y и параметром выдачи q.

Если сам переход обозначается буквой a, то эту букву мы будем писать в нижнем индексе для s, x, p, y, q, t.

Стимул – это полностью определённое отображение множества входов на множество сообщений, к которому добавлено пустое сообщение лямбда.

Параметр приема p – это подмножество входов, с которых будут приниматься сообщения при выполнении перехода. Поэтому каждый из этих входов стимул должен отображать в непустое сообщение.

Реакция – это полностью определённое отображение множества выходов на множество сообщений, к которому добавлено пустое сообщение лямбда.

Параметр выдачи q – это подмножество выходов, по которым будут выдаваться сообщения при выполнении перехода. Поэтому каждый из этих выходов реакция должна отображать в непустое сообщение.

Когда выполняется переход, происходит следующее:
  принимаются сообщения, определяемые стимулом x,
    по входам, определяемым параметром приема p,
  выдаются сообщения, определяемые реакцией y,
    по выходам, определяемым параметром выдачи q,
  после чего автомат переходит в состояние t.

Если в текущем состоянии есть несколько переходов, то как выбирается один переход на выполнение?

Для этого нужно определить работу автомата в системе, то есть определить взаимодействие автоматов. А это мы сделаем с помощью формального определения композиции автоматов.

Слайд 7. Система автоматов.

Но сначала нужно дать формальное определение системы автоматов.

Пусть V – конечное множество автоматов в алфавите M.

Для удобства определения системы автоматов будем, без ограничения общности, считать, что входы, выходы и состояния автоматов разные.

Этого всегда можно добиться с помощью систематического переименования состояний, входов и выходов автоматов, в результате которого получаются автоматы изоморфные исходным.

Система автоматов – это набор из множества сообщений M, множества автоматов V и множества соединений E.

Соединение – это пара из выхода одного автомата и входа другого (или того же самого) автомата. При этом каждый выход каждого автомата соединён не более чем с одним входом, и с каждым входом каждого автомата соединён не более чем один выход. Множество соединений можно задать как биекцию из подмножества всех выходов всех автоматов на подмножество всех входов всех автоматов.

Если соединение это пара из выхода и входа одного и того же автомата, то такое соединение будем называть *петлёй*.

Вход автомата, с которым не соединён ни один выход, будем называть *внешним входом системы*, а выход, который не соединён ни с одним входом, будем называть *внешним выходом системы*.

Слайд 8. Классы связности.

Системе соответствует ориентированный граф с помеченными дугами, вершинами которого являются автоматы из множества V, а дуга ведёт из автомата A в автомат B и помечена соединением (j,i) тогда и только тогда, когда j – это выход автомата A, i – это вход автомата B, и выход j соединён со входом i. Будем называть это ориентированным графом связей.

Если снять ориентацию дуг, то получится неориентированный граф связей.

Будем говорить, что автоматы A и B *связаны* в системе, если в неориентированном графе связей существует путь из A в B (возможно, пустой, если $A=B$). В противном случае разные автоматы A и B *не связаны*.

Очевидно, что отношение связанности автоматов рефлексивно, симметрично и транзитивно, т.е. является отношением эквивалентности. Поэтому оно разбивает множество автоматов V на классы эквивалентности, которые будем называть классами связности.

Слайд 9. Состояние автомата – это множество.

Для удобства определения композиции будем считать, что каждое состояние каждого автомата в системе является множеством, причём состояния разных автоматов в системе являются непересекающимися множествами.

Если уже задана система, в которой это условие не выполнено, то можно преобразовать каждый автомат, заменив в нём каждое состояние s на соответствующий синглетон – множество, состоящее только из s.

Таким преобразованием будут получены автоматы изоморфные исходным.

А поскольку в системе автоматов состояния разных автоматов разные, то будет выполнено условие попарного непересечения множеств состояний автоматов.

Слайд 10.Полезные обозначения

Прежде чем перейти к определению композиции введём ещё некоторые полезные обозначения.

Прежде всего определим операцию “прямой слэш”, операндами которой является произвольная функция f и произвольное множество N, а результатом – сужение функции на поддомен, равный разности её исходного домена и множества N.

Вот здесь на слайде дано формальное определение операции и указаны два достаточно очевидных её свойства.

Первое свойство для случая, когда множество N является объединением двух множеств.

А второе свойство для случая, когда сама функция является объединением двух функций с непересекающимися доменами, причём домен второй функции не пересекается с множеством N.

Будем считать, что операция “прямой слэш” имеет тот же приоритет, что разность множеств, которая обозначается как “обратный слэш”, и объединение множеств. Как следствие, в выражениях над множествами, где используются только операции “прямой слэш”, “обратный слэш” и объединение, мы будем использовать бесскобочную запись и предполагать, что операции выполняются слева направо.

Также на слайде приведено обозначение логической эквивалентности, которое мы будем использовать.

Слайд 11.Композиция переходов

Теперь мы можем перейти к определению композиции автоматов. Нам нужна такая композиция, которая учитывает наличие у автомата нескольких входов и выходов и моделирует взаимодействие автоматов через соединения, каждое из которых связывает выход одного автомата со входом другого автомата. Иными словами, это композиция автоматов по ориентированному графу связей системы автоматов.

Сначала мы определим композицию двух переходов по соединению.

Пусть в системе есть соединение (j,i) такое, что j – это выход автомата A, а i – это вход автомата B.

Определим композицию двух переходов: a маленькое и b маленькое. Функция f от a,j,i,b – задаёт условие существования композиционного перехода. Если она истинна, то определяется сам композиционный переход.

Композиция выполняется, если выполнены два условия.

Условие 1. Должны быть согласованы реакция y и стимул x для соединения (j,i). Сообщение (быть может, пустое), которое реакция y в переходе a маленькое приписывает выходу j, должно быть согласовано с сообщением, которое стимул x в переходе b маленькое приписывает входу i. А именно: эти сообщения должны совпадать.

Условие 2. Должны быть согласованы параметры выдачи q и приема p для соединения (j,i). А именно: автомат A выдает согласованное сообщение тогда и только тогда, когда автомат B его принимает. Это означает, что выход j принадлежит параметру выдачи q перехода a маленькое тогда и только тогда, когда вход i принадлежит параметру приема p перехода b маленькое.

Если условие существования композиционного перехода выполнено, то сам этот переход определяется так.

Пре- и постсостояния получаются объединением состояний операндов, понимаемых как множества. Здесь мы используем тот факт, что состояния разных автоматов не пересекаются, поэтому разные пары состояний операндов порождают разные состояния композиции. Объединение состояний удобнее декартового произведения, потому что объединение коммутативно и ассоциативно, а декартовое произведение нет.

Стимул композиции – это сообщения, приписанные входам обоих автоматов, кроме того входа i, которое входит в соединение (j,i), по которому происходит композиция.

Соответственно, принимаются сообщения со всех тех входов, с которых их принимают оба автомата, кроме входа i.

Аналогично реакция композиции – это сообщения, которые автоматы приписали своим выходам, кроме выхода j.

Соответственно, выдаются сообщения по всем тем выходам, по которым их выдают оба автомата, кроме выхода j.

Если соединение является петлёй, то есть автоматы A и B совпадают, то имеется дополнительное условие существования композиции: переходы a маленькое и b маленькое должны совпадать. Иными словами, при композиции по соединению-петле переход автомата компонуется сам с собой.

Определение самого перехода такое же.

Здесь мы опять пользуемся тем, что состояние автомата – это множество. Объединение множества с самим собой не меняет его. Поэтому пре и постсостояние перехода остаются такими же.

Вход i удаляется из домена стимула и из параметра приема p, а выход j – из домена реакции и параметра выдачи q.

Слайд 12. Композиция автоматов

Теперь определим композицию автоматов.

Но сначала заметим, что композиция переходов естественным образом распространяется на композицию множеств переходов. Такую композицию определим как множество попарных композиций переходов из множеств-операндов.

Мы определим композицию автоматов, в которой оставим только начальное состояние композиции и пре- и постсостояния переходов композиции. Это множество, очевидно, включает подмножество состояний, достижимых из начального состояния.

Для множества переходов T обозначим через $States$ функцию, дающую множество пре- и постсостояний этих переходов.

Пусть опять в системе есть соединение (j,i) такое, что j – это выход автомата A, а i – это вход автомата B.

Композиционный автомат – это автомат в том же алфавите сообщений M. Я напомню, что все автоматы в системе – это автоматы в одном алфавите.

Входы композиционного автомата – это все входы автоматов A и B, кроме входа i, а выходы – это все выходы автоматов A и B, кроме выхода j.

Множество переходов – это композиция множеств переходов автоматов A и B.

Начальное состояние – это объединение начальных состояний автоматов A и B.

А множество состояний определяется как начальное состояние плюс множество пре- и постсостояний переходов композиции.

Композиция автоматов удовлетворяет всем условиям, которые налагаются на автомат в его определении.

Кроме того, выполняется введённое нами дополнительное требование, чтобы состояние автомата было множеством.

Слайд 13. Композиция системы автоматов

Когда два автомата компонуются по связывающему их соединению, соответствующим образом меняется система автоматов: в ней вместо этих двух автоматов остаётся один композиционный автомат.

Алфавит сообщений не меняется.

А из множества соединений удаляется соединение, по которому происходит композиция.

Композиция системы удовлетворяет всем требованиям, предъявляемым к системе автоматов.

Кроме того, состояния разных автоматов в системе, по-прежнему, являются непересекающимися множествами.

Слайд 14. Ассоциативность композиции

Определенная нами композиция обладает важным свойством ассоциативности. Ассоциативность композиции важна для того, чтобы работа системы зависела только от множества её автоматов и соединений и не зависела от какого-либо упорядочивания этих множеств.

Пусть в системе есть два разных соединения (j,i) и (l,k), и четыре автомата A, B, C, D такие, что j – выход автомата A, i – вход автомата B, l – выход автомата C и k – вход автомата D. Эти автоматы не обязательно разные.

Если в этих четырёх автоматах выбрать по одному переходу, то результат композиции этих переходов по двум соединениям, а также условия существования композиционного перехода не зависят от того, в каком порядке производится композиция: сначала по одному соединению, а потом по другому, или наоборот. Это ассоциативность композиции переходов.

Автоматы, которые получаются из автоматов A, B, C, D при композиции по двум соединениям, тоже не зависят от того, в каком порядке производится композиция.

И, наконец, композиционная система не зависит от порядка соединений, по которым делается композиция.

Слайд 15. Композиция без соединений

Из последнего утверждения следует, что если мы скомпонуем систему по последовательности соединений, в которую каждое соединение входит ровно один раз, то получится система без соединений, причём эта система не зависит от порядка композиции.

В такой системе все входы и выходы являются внешними.

Однако если в исходной системе были несвязанные автоматы, то композиция будет состоять из нескольких автоматов. А именно: по одному автомату на каждый класс связности.

Поэтому такую систему нельзя использовать как компонент более сложной системы, поскольку по нашему определению системы её компонент – это один автомат.

Для того чтобы обойти это неудобство достаточно определить композицию несвязанных автоматов.

Семантика такой композиции достаточно прозрачна: если два автомата не связаны, то работа системы из этих автоматов эквивалентна работе одного автомата, в котором объединяются входы и выходы обоих автоматов, а любые два переходы из разных автоматов объединяются в один переход.

На слайде приведены формальные определения композиции переходов без соединения, композиции несвязанных автоматов и композиции системы, в которой соединений нет.

Такая композиция тоже ассоциативна для переходов, автоматов и системы.

Каждая композиция уменьшает на 1 число автоматов в системе. Поэтому через конечное число композиций система превратится в систему без соединений, состоящую из одного автомата. Этот автомат представляет систему в целом и может использоваться как компонент более сложной системы.

Слайд 16. Тупики в классе связности

В композиции, которую мы определили, нет «асинхронных» переходов, все переходы «синхронные»: переход в каждом автомате выполняется одновременно с переходами в автоматах, связанных с ним по соединениям.

Это означает, что если какой-то автомат «стоит», т.е. в текущем своем состоянии не может выполнить никакого перехода, то «стоят» все автоматы из того же класса связности, и это «стояние» продолжается бесконечно до рестарта системы. Это ситуация тупика (deadlock).

Тупик возникает из-за несогласованности требований автоматов, связанных соединением. Эти требования определяются условиями композиции переходов, которые приведены на слайде. При композиции по петле оба перехода берутся из одного автомата и они должны совпадать. Рассмотрим остальные два условия с точки зрения возможности тупика.

Для того чтобы тупик был невозможен из-за нарушения условия 1, достаточно потребовать, чтобы

либо A) в каждом автомате в каждом состоянии был определен переход по каждому стимулу x,

либо B) в каждом автомате в каждом состоянии был определен переход по каждой реакции y.

Для того чтобы тупик был невозможен из-за нарушения условия 2, достаточно потребовать, чтобы

либо A) в каждом автомате вместе с каждым переходом имеются переходы с теми же пресостоянием, стимулом, параметром приёма и реакцией y и со всеми возможными значениями параметра выдачи, определяющими по какому выходу j сообщение $y(j)$ передаётся, а по какому нет,

либо B) в каждом автомате вместе с каждым переходом имеются переходы с теми же пресостоянием и стимулом x и всеми возможными параметрами приема, определяющими по какому входу i сообщение $x(i)$ принимается, а по какому нет.

Мы выбираем варианты A для обоих условий. Это формализовано в определении вполне определенного автомата.

Слайд 17. Вполне определённые автоматы

Автомат будем называть *определённым по всем стимулам*, если в каждом состоянии есть переход по каждому стимулу.

Автомат будем называть *определенным по всем параметрам выдачи*, если вместе с каждым переходом имеются переходы с теми же пресостоянием, стимулом, параметром приема и реакцией и со всеми возможными значениями параметра выдачи.

Автомат будем называть *вполне определенным*, если он определён по всем стимулам и по всем параметрам выдачи.

Слайд 18. Правила умолчания

Вместо условий вполне определённости можно использовать *правила умолчания*.

Для этого при задании автомата указываются не все, а некоторое подмножество T^* явных переходов.

Это подмножество дополняется до множества T всех переходов согласно правилам умолчания. Эти правила такие.

1) Если нет явного перехода в некотором состоянии по некоторому стимулу, то это означает, что такой переход определён неявно без приёма и выдачи сообщений и без изменения состояния.

   В некотором смысле такой неявный переход моделирует «стояние» автомата, поскольку состояние автомата не меняется, сообщения не принимаются и не передаются.

2) Если есть явный переход, но для некоторого возможного значения параметра выдачи нет явного перехода с теми же пресостоянием, стимулом, параметром приема и реакцией, то такой переход определён неявно без изменения состояния.

На слайде написано формальное определение множества всех переходов T через подмножество явных переходов T^* по этим правилам умолчания.

Здесь важно отметить, что при композиции автоматов нужно выполнять композицию всех переходов из множества T, а не только явных переходов из множества T^*.

Слайд 19.Детерминизм автоматов

Для целей тестирования мы ограничимся исследованием детерминированных автоматов.

Детерминированность будет означать, что каждый автомат в системе в своём текущем состоянии может выполнить не более одного перехода.

Если автомат вполне определён, то будет ровно один переход.

Формально, в детерминированном автомате должны быть выполнены два условия:

1. Пресостояние и стимул однозначно определяют параметр приема и реакцию:
2. Одинаково помеченные переходы, ведущие из одного пресостояния, совпадают, т.е. ведут в одно постсостояние:

Заметим, что если выполнено условие 1, условие 2 можно переписать короче.

Слайд 20. Возможный недетерминизм системы

В то же время, даже если все автоматы системы детерминированные, их композиция может оказаться недетерминированной.

Легче всего это показать на примере композиции автомата с самим собой по соединению-петле *(j,i)*.

Пусть в автомате есть два перехода *a* и *a*\` из одного пресостояния *s* по стимулам *x* и *x*\`. Пусть эти стимулы разные, но отличаются только на входе *i*. Постсостояния *t* и *t*\` имеют право быть тоже разными.

Поскольку здесь только один автомат, то каждый переход компонуется сам с собой. Пусть каждый из переходов *a* и *a*\` компонуется сам с собой.

Сравним результаты композиции. У них одинаковые пресостояния *s*, одинаковые стимулы – *x* на домене без входа *i*, но разные постсостояния *t* и *t*\`.

Если в композиционных переходах получились разные параметры приема или разные реакции, то нарушается 1-ое правило детерминизма: пресостояние и стимул неоднозначно определяют параметр приема или реакцию.

Если же эти параметры композиционных переходов равны, а, кроме того, равны параметры выдачи, то есть *q* минус *j* равно *q*\` минус *j*, то нарушается 2-ое правило детерминизма: одинаково помеченные переходы из одного пресостояния ведут в разные постсостояния *t* и *t*\`.

Заметим, что если автомат вполне определён, то при равенстве реакций *y* и *y*\` на всех выходах, кроме, быть может, выхода *j*, всегда найдутся такие переходы, что в композиционных переходах параметры выдачи равны, и 2-ое правило детерминизма будет нарушено.

Слайд 21. Возможный недетерминизм системы

Когда в цикле несколько автоматов и соединений, то аналогичное нарушение детерминизма возникает, если в автоматах этого цикла есть две «параллельные», но разные, цепочки компонуемых переходов, ведущих из одинаковых состояний и отличающихся в стимулах только по входам соединений цикла.

Слайд 22. Возможный недетерминизм системы

Когда мы компонуем по соединениям такого цикла, то рано или поздно остаётся только одно соединение, то есть цикл соединений превращается в петлю. Только здесь уже пары переходов a и $a`$ – это композиционные переходы автомата, получившегося после композиции по всем соединениям цикла, кроме одного последнего.

В то же время можно заметить, что если хотя бы в одном автомате цикла, в переходах из двух «параллельных» цепочек в реакции указано одно и то же сообщение для выхода, участвующего в соединении цикла, то недетерминизм не возникает.

Слайд 23. Возможный недетерминизм системы

Посмотрим, почему так происходит для случая петли (j,i).

Из условия композиции переходов следует, что если $y(j)=y`(j)$, то должно быть $x(i)=x`(i)$, а тогда будет $x=x`$.

А тогда, по 1-ому правилу детерминизма для автомата A будет $p=p`$ и $y=y`$. Поэтому правило 1 не нарушается для композиционного автомата.

Если $p=p`$, то вход $i$ принадлежит $p$ тогда и только тогда, когда он принадлежит $p`$.

А по условию композиции, вход i принадлежит p тогда и только тогда, когда выход j принадлежит q, и вход i принадлежит $p`$ тогда и только тогда, когда выход $j$ принадлежит $q`$.

Тем самым, выход j принадлежит q тогда и только тогда, когда выход j принадлежит $q`$.

И если в остальном q и $q`$ тоже совпадают, то просто $q=q`$.

Но тогда по 2-ому правилу детерминизма для автомата A должно быть $t=t`$. Поэтому 2-ое правило тоже не нарушается для композиционного автомата.

Слайд 24.Композиция разных автоматов

Теперь я приведу без доказательства основные утверждения о композиции вполне определённых и детерминированных автоматов.

Сначала три леммы о композиции разных автоматов.

<u>Лемма 1a</u>: композиция разных детерминированных автоматов тоже детерминирована.

<u>Лемма 1b</u>: композиция разных вполне определённых автоматов тоже вполне определена.

<u>Лемма 1c</u>: композиция разных автоматов, определённых по всем параметрам выдачи, тоже определена по всем параметрам выдачи.

Заметим, что композиция разных автоматов, определенных только по всем стимулам, может не быть определенной по всем стимулам.

Это объясняется тем, что для какой-то пары стимулов этих автоматов может быть нарушено 2-ое условие композиции о согласованности параметров приема и выдачи, то есть возникнет deadlock. Это означает, что в композиционном автомате нет перехода по композиционному стимулу, получающемуся из этих двух стимулов.

Слайд 25. Композиция по петле со свободным контактом

Мы уже знаем, что композиция детерминированного автомата по петле *(j,i)* может оказаться недетерминированной. Но если потребовать, чтобы реакция приписывала выходу *j* одно и то же сообщение, то будет детерминизм.

Если *k* вход, а *l* выход автомата *A*, то пару *(k,l)* будем называть *контактом* в автомате *A*.

Контакт *(k,l)* в автомате *A* будем называть *свободным*, если сообщение на выходе *l* не зависит от сообщения на входе *k*.

Пусть *(i,j)* свободный контакт в автомате *A*, и в системе автоматов имеется соединение-петля *(j,i)*. Тогда имеют место следующие три утверждения.

<u>Лемма 2a</u>: Если автомат *A* детерминированный, то композиция тоже детерминирована.

<u>Лемма 2b</u>: Если автомат *A* вполне определённый, то композиция тоже вполне определена.

<u>Лемма 2c</u>: Если автомат *A* определён по всем параметрам выдачи, то композиция тоже определена по всем параметрам выдачи.

Заметим, что если автомат *A* определен по всем стимулам, то композиция *A[j,i]A* может не быть определенной по всем стимулам.

Слайд 26.Свободные циклы

Кроме петель, в системе могут быть и другие циклы соединений. А как мы уже видели, в процессе композиции такой цикл становится петлёй, когда выполняется композиция по всем соединениям цикла, кроме одного. Поэтому важным является сохранение свободности контакта при композиции.

Формально последовательность соединений образует цикл, если вход одного соединения и выход следующего соединения принадлежат одному автомату, а также вход последнего соединения и выход первого соединения принадлежат одному автомату.

Цикл будем называть свободным, если в нём есть свободный контакт как пара из входа одного соединения и выхода следующего соединения или как пара из входа последнего соединения и выхода первого соединения.

В этих терминах нам нужно, чтобы в системе все циклы были свободные и оставались свободными при композиции.

Слайд 27.Сохранение свободных циклов

Лемма 3 утверждает, что если в системе все циклы свободные, а все автоматы детерминированы и определены по всем стимулам, то в композиции тоже все циклы свободные.

Это графически изображено на слайде справа вверху.

Также графически изображены утверждения предыдущих лемм.

Слайд 28.Сохранение свободных циклов

Из этого непосредственно следует вот такая теорема: если в системе все циклы свободные, а все автоматы детерминированные и вполне определенные, то в композиции тоже все циклы свободные и все автоматы тоже детерминированные и вполне определенные.

Слайд 29.Основное утверждение

Как следствие из этой теоремы получаем основное утверждение: если в системе все циклы свободные, а все автоматы детерминированы и вполне определены, то композиция по всем соединениям состоит из автоматов, которые не связаны друг с другом, детерминированы и вполне определены.

Выполняя далее композицию без соединений, получаем один детерминированный и вполне определенный автомат.

Здесь можно отметить, что для целей тестирования композицию без соединений делать не нужно, даже вредно. Имея набор несвязанных автоматов, мы можем тестировать каждый из этих автоматов независимо от другого.

Композицию без соединений нужно делать, если мы хотим получить один автомат как компонент более сложной системы.

Слайд 30.Генерация тестов

Для обобщенной модели тесты генерируются аналогично тому, как рассказывалось в первом докладе.

Генерируются тесты для композиционного автомата всей системы (или для композиционного автомата класса связности) и фильтруются:

Тест сохраняется, если он покрывает хотя бы один переход хотя бы одного исходного автомата в системе, который не покрывается уже сохранёнными тестами.

Иначе тест отбрасывается.

После окончания генерации и фильтрации переходы автоматов системы, которые не покрываются сохранёнными тестами, - это недостижимые переходы.

Слайд 31. Автоматы вершин и дуг (из первого доклада)

Теперь посмотрим как связана рассмотренная нами обобщенная модель системы с моделью системы из первого доклада, я буду называть её первой моделью.

Напомню, что в первой модели у нас был граф связей с вершинами и дугами. Автомат находился в вершине, а дуга была очередью сообщений длины 1.

Дуга как очередь длины 1 в обобщенной модели описывается вполне определенным детерминированным автоматом, который изображен на слайде.

Как работал автомат вершины?

Переход в автомате помечался только стимулом x и реакцией y.

Стимул x был частичным отображением: он выделял часть входов и указывал непустые сообщения, которые должны приниматься с этих входов при выполнении перехода. Состояния остальных входов были несущественны и сообщения с них не принимались.

Реакция y была частичным отображением: она выделяла часть выходов и указывала непустые сообщения, которые должны выдаваться на эти выходы при выполнении перехода. На остальные выходы сообщения не передавались, то есть передавались пустые сообщения.

Определим преобразование автомата, которое будет интерпретировать автомат первой модели как обобщённый автомат.

Переход будет соответствует множеству кратных переходов. В каждом таком переходе параметр приема p – это домен исходного стимула x, а параметр выдачи q – это домен исходной реакции y.

Стимул x` – это любой стимул, совпадающий с исходным стимулом $x$ на его домене, а на остальных входах указываются какие-то сообщения, в том числе и пустые. Реакция $y$` совпадает с исходной реакцией y на ее домене, а на остальных выходах указывается пустое сообщение.

Кроме того, если в состоянии нет перехода по некоторому стимулу, то добавим переход, моделирующий «стояние» автомата: параметр приема p пустой, на все выходы выдаются пустые сообщения, а состояние не меняется.

Такой автомат вершины будет определен по всем стимулам. Если исходный автомат был детерминирован в смысле первой модели, то после интерпретирующего преобразования получится автомат детерминированный в смысле второй модели.

При такой интерпретации у нас возникают две проблемы.

Слайд 32.Первая проблема: автомат вершины не определен по всем параметрам выдачи

Первая проблема заключается в том, что автомат вершины не определен по всем параметрам выдачи. Это вопрос принципиальный, потому что семантика первой модели предполагает, что должны быть выданы все сообщения, описываемые реакцией. Эта семантика работает по принципу «всё или ничего». Условная выдача части сообщений не разрешается. Если не удаётся выдать все сообщения, автомат «стоит».

Слайд 33. Вторая проблема: асинхронные (альтернативные) переходы в автомате дуги

Вторая проблема заключается в том, что когда автомат вершины «стоит», автоматы входящих и выходящих дуг, тем не менее, могут работать. Входящая дуга, если она пустая, может принять сообщение от автомата в её начальной вершине. Выходящая дуга, если она занята каким-то сообщением, может выдать это сообщение автомату в её конечной вершине.

В противоположность этому семантика обобщённой модели предполагает, что на каждом такте либо каждый автомат из класса связности выполняет ровно один переход, либо все они «стоят».

Иными словами, обобщённая модель допускает только синхронные переходы, тогда как в первой модели возможны несимметричные асинхронные переходы. Несимметричность в том, что «стоять» может только автомат вершины, а асинхронный переход выполняется в автоматах связанных с ним дуг. Более точно здесь следовало бы говорить не об асинхронных, а об альтернативных переходах. Если автомат вершины не «стоит», автомат дуги выполняет один переход, а если «стоит», то, вообще говоря, другой переход. Но этот другой асинхронный переход может быть выполнен только в том случае, когда автомат вершины «стоит».

Эти две проблемы тесно связаны между собой. Почему? Потому что для того, чтобы определить, может ли автомат вершины выполнить переход или он будет «стоять», нужно «знать» состояния всех выходящих дуг: пусты они или заняты. Если не всех, то, по крайней мере, тех дуг, на которые выдаются непустые сообщения.

Слайд 34. Специальная асинхронная композиция автомата вершины с автоматами всех выходящих дуг

А это означает, что нельзя выполнить композицию автомата вершины с автоматом только одной выходящей дуги, если сохранять семантику первой модели после интерпретирующего преобразования. По сути, нужно было бы определить специальную асинхронную композицию автомата вершины с автоматами всех выходящих дуг. При выполнении такой композиции мы знаем, будет автомат вершины выполнять переход или будет «стоять». И это уже позволяет определить выполняемые переходы в автоматах выходящих дуг.

Результатом такой асинхронной композиции автомата вершины со всеми автоматами выходящих дуг является уже вполне определенный автомат. И он детерминирован, если был детерминирован автомат вершины. После того, как мы скомпонуем каждый автомат вершины с автоматами всех выходящих дуг, у нас получится система, состоящая только из вполне определенных детерминированных автоматов.

Условие свободности циклов тоже будет выполнено. Почему? Потому что в автомате дуги реакция зависит только от её состояния, то есть не зависит от стимула. Тем самым и в композиции автомата вершины с автоматами всех выходящих дуг реакция тоже зависит только от состояния композиции, потому что это совокупная реакция автоматов всех выходящих дуг. А из этого следует, что все контакты свободные.

Этот прием со специальной асинхронной композицией можно распространить и на более общий случай, когда дуга не обязательно является очередью сообщений длины 1. Достаточно, чтобы автомат дуги имел один вход (выходов может быть уже несколько) и был вполне определенным детерминированным автоматом, в котором реакция зависит только от состояния.

Слайд 35. Специальная асинхронная композиция двудольного графа автоматов вершин и автоматов дуг, замкнутого по выходам автоматов вершин

Если же автомат дуги имеет несколько входов, то с таким автоматом дуги придётся компоновать автоматы сразу всех вершин, соединённых со входами этого автомата дуги. В общем получится, что в специальной асинхронной композиции каждый автомат вершины участвует вместе с автоматами всех выходящих дуг, а каждый автомат дуги – вместе с автоматами всех вершин, соединённых со входами этого автомата дуги. Получится двудольный подграф системы автоматов, замкнутый по выходам автоматов вершин.

В связи с этим можно отметить, что не случайно композиция для первой модели в первом докладе определялась сразу для всей системы.

Какой вывод можно сделать из этого анализа?

К сожалению, получилось, что обобщённая модель, с одной стороны, позволяет описывать существенно более широкий класс поведений автоматов, чем первая модель, но, с другой стороны, она кое-что «потеряла» по сравнению с первой моделью.

Было бы неплохо попробовать разработать новую модель взаимодействия автоматов, которая совмещала бы в себе достоинства семантики первой модели с асинхронными, точнее, альтернативными переходами семантики обобщённой модели.