Лекция № 1

Под **системой** понимается некоторое сложное понятие, характеризующееся множеством различных описаний. Нас будут интересовать два понятия: **параметры q** системы и **процесс** z в системе.

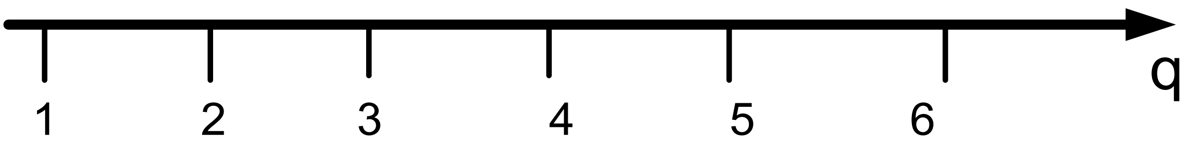
Под **параметром** q будем понимать двойку: **q = <a,b>**

где a - имя параметра, b – его значение.

Всю совокупность параметров системы, определяющих процесс функционирования или участвующих в нем, назовем *параметрическим множеством системы* $\boldsymbol{Q = \{q_i\}}$, где q_{i} – некоторый параметр.

Каждый параметр q_{i} принимает **множество значений**, обозначаемое в дальнейшем как $\boldsymbol{\sigma(q_{\mathrm{i}})}$.

Тогда, графически мы можем представить параметр, как некоторую ось



Определим ***пространство состояний*** *системы* как декартово произведение $\mathbf{S}=\prod_{\forall i} \boldsymbol{\sigma}(\mathbf{q}_{\mathrm{i}})$. В этом пространстве каждый параметр выступает в роли координаты, а размерность пространства равна мощности множества *Q. Многомерная точка пространства S есть возможное состояние системы*.

Например, выберем в качестве системы центральный процессор (ЦП) и жесткий диск (Д).

Для ЦП определим следующие параметры:

— количество процессоров (КП)

— производительность процессора (ПП)

— состояние процессора (СП)

— стоимость процессора (СтП).

Для Д определим следующие параметры:

— скорость записи и считывания (СЗС)

— состояние диска (СД).

Тогда множества значений параметров можно определить следующим образом:

Процессор:

σ(КП) = {1, 2};

σ(ПП) = {10, 50, 100, 500}, эти значения могут соответствовать какой-либо системе оценки производительности, например, обозначать среднее количество MFLOP операций за единицу времени;

σ(СП) = {“работает”, “простаивает”, “тестируется”};

σ(СтП) = {2 - 20}, эти значения стоимости могут быть выражены, например, в некоторых условных единицах;

Диск:

σ(СЗС) = {10 - 50}, эти показатели скорости операций на диске могут быть заданы аналогично параметру ПП;

σ(СД) = {“работает”, “простаивает”, “ремонтируется”}.

В рассмотренном выше примере размерность пространства состояний равна 6, а само пространство состояний имеет вид:

$$S=\sigma(\text{КП})\text{x}\sigma(\text{ПОП})\text{x}\sigma(\text{СП})\text{x}\sigma(\text{СтП})\text{x}\sigma(\text{СЗС})\text{x}\sigma(\text{СД})$$

Возможным состоянием системы $\mathbf{s}\in S$ может быть следующий кортеж:

$$\mathbf{s}=<2,\ 50,\ \text{“простаивает”},\ 15,\ 40,\ \text{“ремонтируется”}>$$

Понятие процесса

В дальнейшем нас будет интересовать изменения состояний системы во времени. Примем за основу определение процесса, предложенное в ряде работ

Процесс Z есть четверка:

$$Z=< S,\ T,\ F,\ \alpha > \qquad (1)$$

где:

S - пространство состояний системы, определенное ранее;

T - множество моментов времени *изменения* состояний системы;

F – график процесса, определяемый как отображение $T\rightarrow S$, причем это отображение должно быть функциональным (однозначным);

α- отношение линейного порядка на T.

Если множество T задано как упорядоченное, то в определении процесса α может быть не указано.

В общем случае множества T и S могут быть как дискретными, так и непрерывными.

Интервал времени $[t_{\text{Н}}, t_{\text{К}}]$, где $t_{\text{Н}}=\min\{\text{T}\}$, $t_{\text{К}}=\max\{\text{T}\}$, назовем *интервалом определения процесса*.

Поскольку пространство S координатного типа, то в случае необходимости подчеркнуть систему координат Q, на которой оно определено, будем обозначать его также S_Q.

В этих обозначениях, если множество Т задано, как упорядоченное, а пространство S определено на множестве параметров Q, определить процесс можно как: $Z=< S_Q, T, F >$.

Пример процесса решения задачи:

1) занять CPU
2) ждать окончания решения
3) занять диск
4) ждать окончания записи
5) занять принтер
6) ждать окончания печати
7) ждать освобождения CPU
8) занять CPU
9) ждать окончания решения

Введем понятие подпроцесса Z^i как плотное во времени подмножество процесса Z на интервале $[t_i; t_j]$ при условии, что $[t_i; t_j] \subseteq [t_{\text{Н}}, t_{\text{К}}]$. Плотность по времени означает, что на интервале $[t_i; t_j]$ нет ни одной точки, принадлежащей Т и не относящейся к подпроцессу Z^i. Этот интервал назовем *интервалом определения подпроцесса.* Понятие подпроцесса позволяет рассматривать процесс в виде последовательности подпроцессов и производить операции разделения и объединения фрагментов процесса.

2. Операции над процессами

<u>Операция свертки процесса</u>

Пусть задан процесс $Z=<S, T, F, \alpha>$

Процесс $Z_1=<S_1, T_1, F_1, \alpha_1>$ является **сверткой** процесса Z, если он получен в результате следующих преобразований:

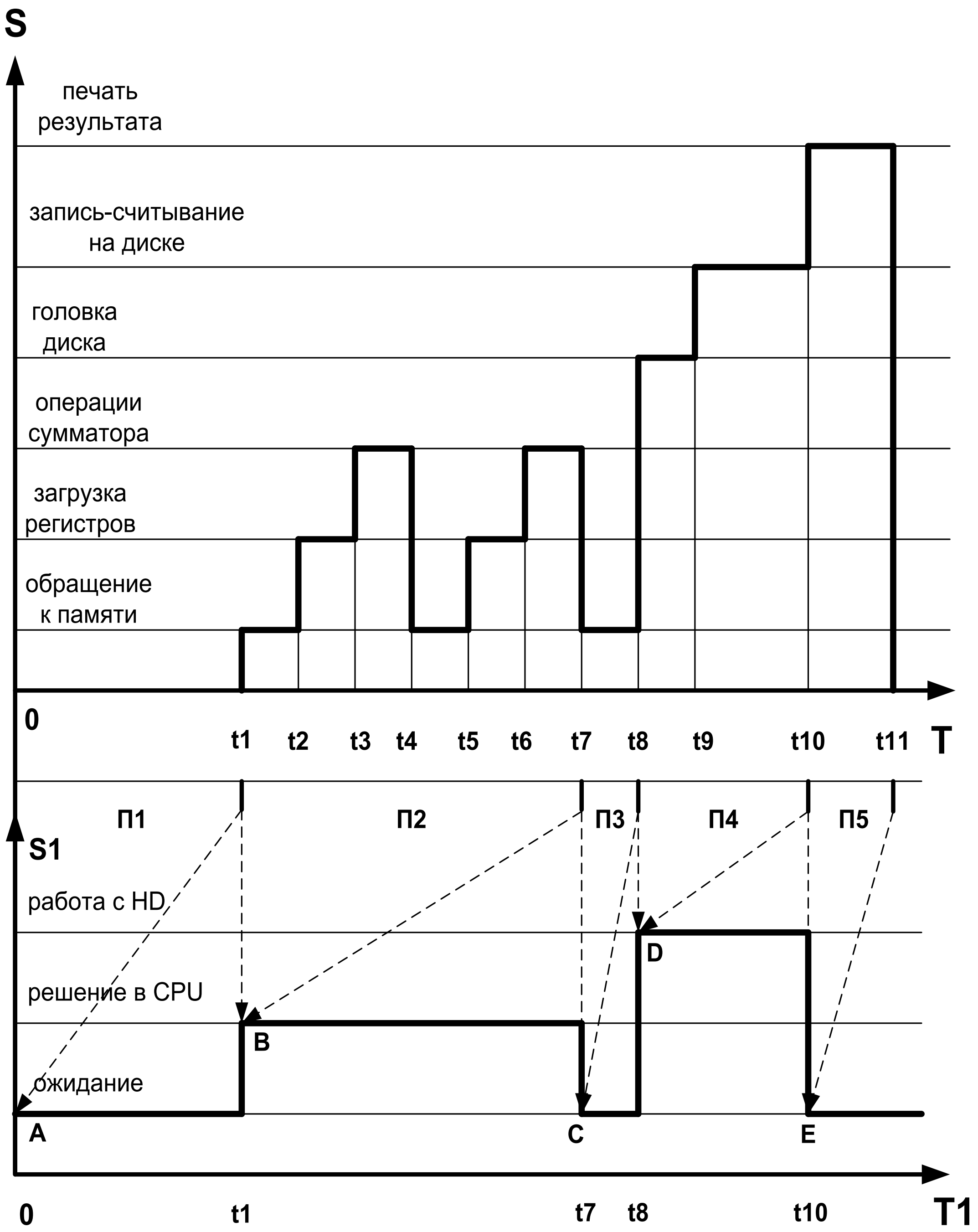
а) произведено полное разбиение интервала определения процесса Z на n непересекающихся подинтервалов $[\tau_j, \tau_{j+1}]$, где j=1..n, причем $\tau_1=t_H$, $\tau_{n+1}=t_K$. В результате получим разбиение процесса Z на n подпроцессов Z^j (j=1..n);

б) поставим в соответствие каждому подпроцессу Z^j одно значение состояния s_1^j из множества S_1 и одно значение времени β^j из интервала $[\tau_j, \tau_{j+1}]$. В результате получим дискретное множество $T_1 = \{\beta^j\}_{j=1}^n$, график $F_1 = \{<\beta^j, s_1^j>\}_{j=1}^n$, отношение $\alpha_1 \subset \alpha$.

Таким образом, получим новый процесс Z_1, который и называется сверткой процесса Z. Очевидно, процесс Z_1 дискретен во времени. Никаких ограничений на характер пространства состояний S_1 не накладывается. Однако на практике при проведении операции свертки пространство S_1, как правило, оказывается значительно меньшей мощности, чем исходное пространство S.

Рассмотрим для примера процесс решения задачи с использованием CPU и жесткого диска HD (рис. 1).

Пусть исходный процесс Z описан в пространстве S, имеющем следующие состояния: {обращение к памяти, загрузка регистров, операции сумматора, головка диска, запись-считывание на диске, печать результата}. Множество моментов времени изменения состояния T={0, t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11}.



Пример операции свертки процесса

Допустим, что нам необходимо построить процесс Z1, отражающий лишь длительность занятия задачей CPU и жесткого диска HD. Для этого зададим новое пространство S1, имеющее следующие состояния: {ожидание, решение в CPU, работа с HD}. Разобьем интервал определения процесса Z на подинтервалы: П1, П2, П3, П4, П5. Это разбиение определяет также соответствующие подпроцессы процесса Z. Выполним отображение подпроцессов процесса Z на фазовую плоскость процесса Z1: П1 отображается в точку A, П2 – в точку B, П3 – в точку C, П4 – в точку D, П5 – в точку E. В результате получим процесс Z1 с пространством состояний S1, множеством моментов времени изменения состояния T1={0, t1, t7, t8, t10}. График этого процесса приведен на этом же рисунке.

Как видно из примера, операция свертки порождает новый процесс, дискретный во времени, поскольку подпроцессы процесса Z, имеющие конечную длительность, отображаются лишь на одну точку фазовой плоскости нового процесса Z1.

Операция свертки относится к классу операций анализа.

Операция развертки

Операция развертки обратна по отношению к операции свертки: процесс Z является разверткой процесса Z_1. При выполнении этой операции необходимо каждую точку $\langle \beta^j, s_1^j \rangle$ процесса Z_1 развернуть в подпроцесс Z^j.

Поставим каждому β^j в соответствие интервал $[\tau_j, \tau_{j+1}]$, при условии, что:

$$\tau_j \le \beta^j \le \tau_{j+1} \quad \text{и} \quad \bigcap_{\forall j} [\tau_j, \tau_{j+1}] = \varnothing. \qquad (2)$$

Зададим отображение $B_j: [\tau_j, \tau_{j+1}] \to S$. Отображение B_j позволяет получить фазовую траекторию подпроцесса Z^j. Для построения процесса Z в

целом необходимо задать все B_j $(j=1,..,n)$. Операция развертки позволяет восстановить исходный процесс на основе некоторых представлений о свернутых процессах.

Операция развертки относится к классу операций синтеза.

Операция проецирования

Процесс Z_1 является *проекцией* процесса Z на координатное пространство S_{Q_1} (обозначение $Z_1 = Пр_{S_{Q_1}} Z$), если $Q_1 \subseteq Q$ и процесс построен по следующей процедуре:

1) каждую точку графика F проецируем на пространство S_{Q_1}. В результате получаем множество $\overline{F}$. Мощность множества $\overline{F}$ равна мощности множества F
2) упорядочиваем множество $\overline{F}$ в соответствие с α. Результат действий 1) и 2) будем называть *отображением* процесса Z на пространство S_{Q_1};

3) вводим отношение эквивалентности на множестве $\overline{F}$ такое, что **r** *подряд расположенных* точек f_{i+1} ,… f_{i+r} множества $\overline{F}$ ($f_{i+1}= <t^{i+1}, s_1^{i+1}>$,… $f_{i+r}=<t^{i+r}, s_1^{i+r}>$) считаются эквивалентными, если $s_1^{i+1}= s_1^{i+2} = \ldots=s_1^{i+r}$

r может принимать любые положительные целые значения, начиная от 1. Таким образом формируются классы эквивалентных значений $K_Э$. При **r**=1 класс содержит одну единственную точку.

4) каждому классу эквивалентности $K_Э$ на $\overline{F}$ ставим в соответствие одну точку $f_{экв}= <t_{\min}, s_{экв}>$ где $t_{min} = \min\limits_{\forall t_i \in K_Э}\{t\}$, $s_{экв}=s_1^{i}=\ldots$ для всего класса $K_Э$

5) формируем множество F_1 из элементов $f_{экв}$ по всем классам эквивалентности на $\overline{F}$, мощность F_1 равна количеству классов эквивалентности на $\overline{F}$.

6) проецируя F_1 на T, получим множество T_1. Очевидно, что $T_1 \subseteq T$, сужение отношения α на T_1 обозначим α_1.

В результате выполнения вышеуказанных операций получим процесс $Z_1 = Пр_{S_{Q_1}} Z$:

$$Z_1=< S_{Q_1} , T_1, F_1, \alpha_1>.$$

Пользуясь операцией проецирования, можно переопределить понятие подпроцесса:

$$Z_{[t_1,t_2]} = Пр_{[t_1,t_2]} Z$$

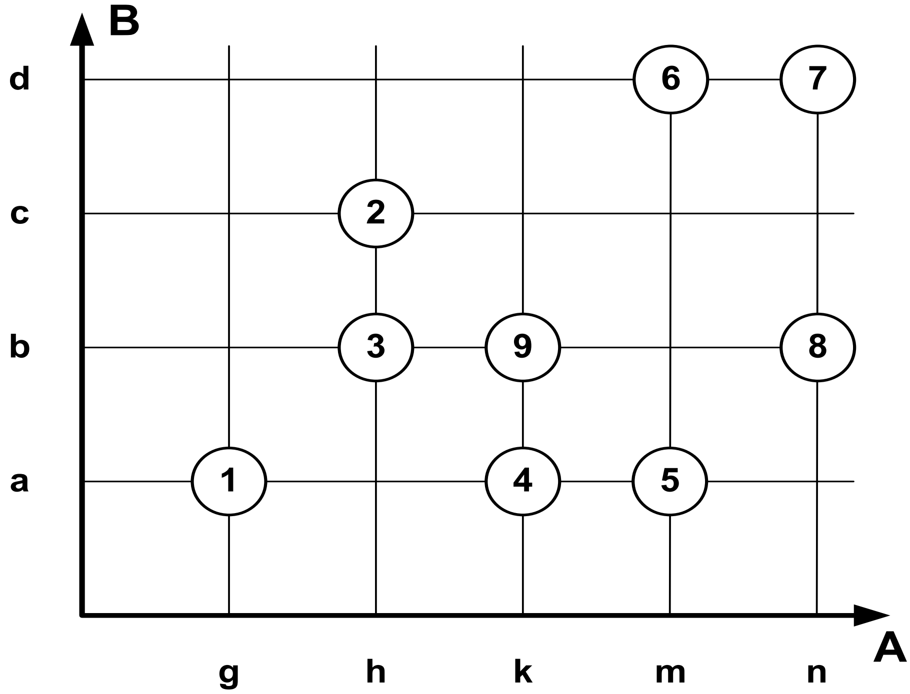


Рис. 2

Процесс Z в двухпараметрическом пространстве

Пример операции проецирования приведен на рис. 2. На нем показан исходный процесс Z, заданный в двухпараметрическом пространстве S=σ(A)xσ(B), где σ(A)={g, h, k, m, n}, σ(B)={a, b, c, d}. Ось времени не показана, однако значения моментов времени изменения состояний указаны в кружочках, обозначающих соответствующее состояние. Таким образом, как видно из рисунка, множество T={1, 2, 3, 4, 5, 6, 7, 8, 9}.

График процесса F={<1,<g,a>, <2,<h,c,>, <3,<h,b>, <4,<k,a>, <5,<m,a>, <6,<m,d>, <7,<n,d>, <8,<n,b>, <9,<k,b>}

Построим процесс $Z_A = Пр_{S_A} Z$, являющийся проекцией процесса Z на пространство σ(A), в соответствие с алгоритмом выполнения операции проецирования.

1) Строим график $\overline{F}_A$={<1,g>, <2,h>, <3,h>, <4,k>, <5,m>, <6,m>, <7,n>, <8,n>, <9,k>}

2) Множество $\overline{F}_A$ уже упорядочено по времени. Получаем отображение процесса Z на пространство σ(A), показанное на рис. 3.

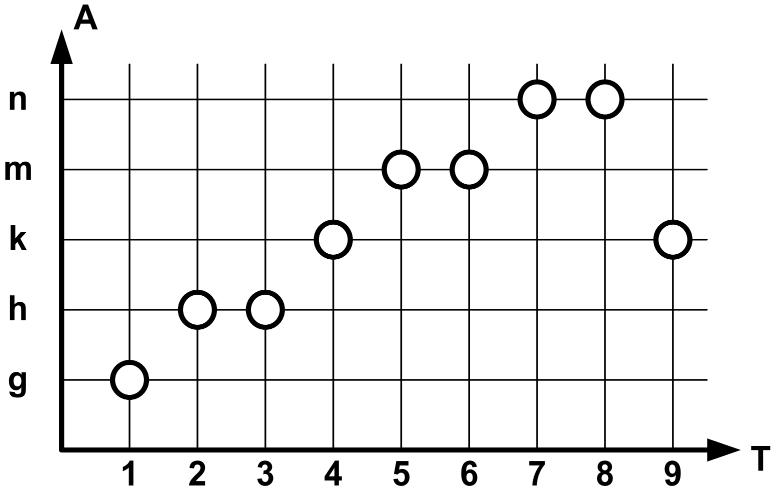


Рис. 3

Отображение процесса Z на параметр A

3) На множестве $\overline{F}_A$ определяем классы эквивалентности: $К_1$ – точка 1, $К_2$ – точки 2,3, $К_3$ – точка 4, $К_4$ – точки 5,6, $К_5$ – точки 7,8, $К_6$ – точка 9.

4) Ставим в соответствие классу $К_2$ точку <2,h>, классу $К_4$ точку <5,m>, классу $К_5$ точку <7,n>.

5) Формируем график F_A={<1,g>, <2,h>, <4,k>, <5,m>, <7,n>, <9,k>}

6) Формируем множество времен изменения состояний T_A=<1,2,4,5,7,9>

Полученный в результате проведенных операций процесс $Z_A = Пр_{S_A} Z$ показан на рис. 4.

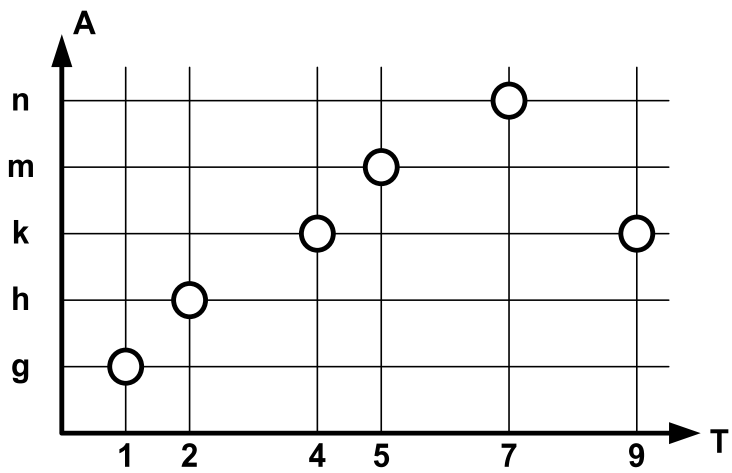


Рис. 4.

Процесс Z_A – проекция процесса Z на параметр A

Аналогично строится и проекция процесса Z на параметр B (пространство, состоящее из единственного параметра). Процесс Z_B показан на рис. 5.

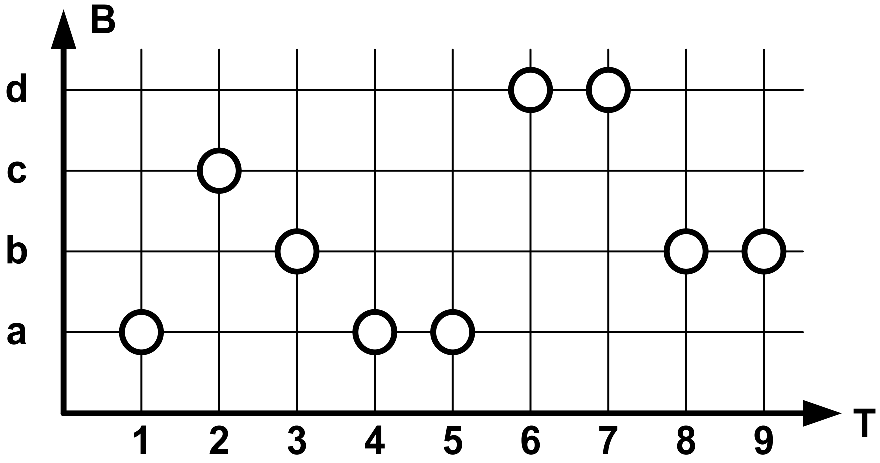


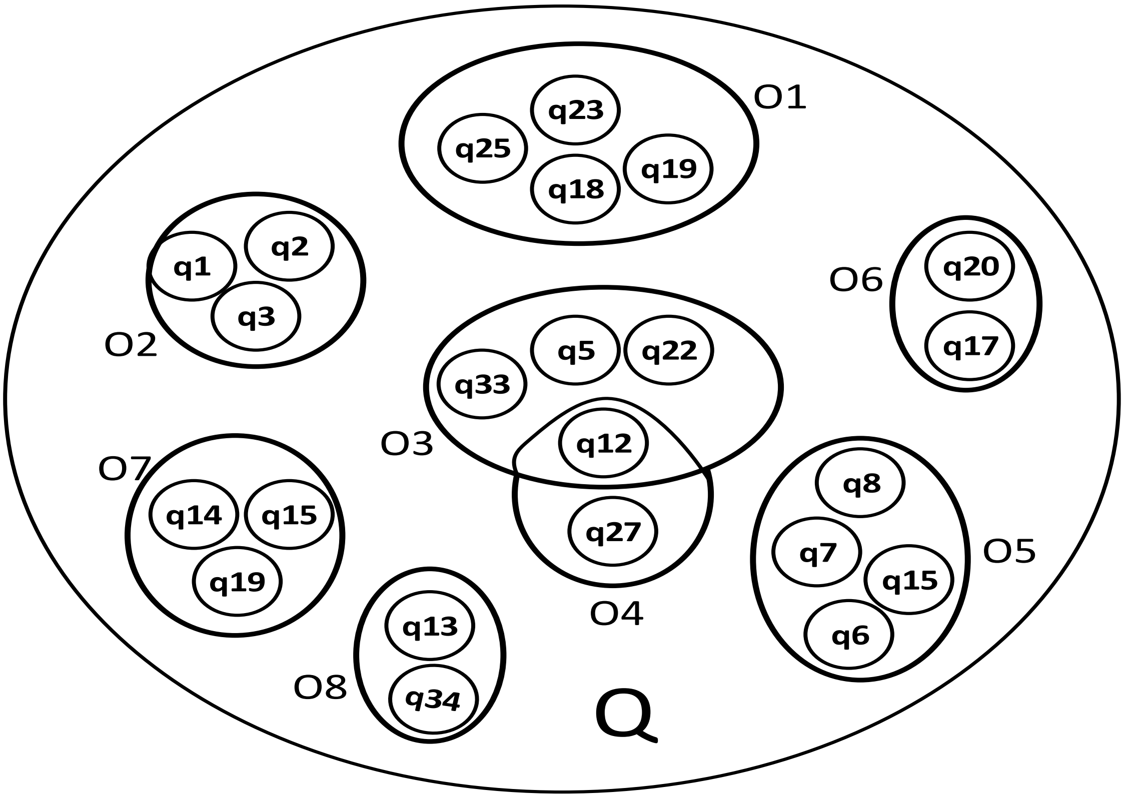
Рис. 5

Отображение процесса Z на параметр B

Лекция № 2

Объекты

Будем рассматривать объект O, как часть системы, как подмножество системы, у которого п параметрическое множество есть подмножество параметрического множества системы: $O \subset Q$.



На рисунке показано параметрическое множество системы **Q** в виде некоторого множества параметров qi, и разбиение этого множества на подмножества, каждое из которых характеризует множество параметров некоторого объекта **O**j . Разбиение на объекты может быть полным или частичным. Если система имеет полное разбиение на объекты, то $\bigcup\limits_{\forall l} O_l = Q$. На рисунке представлен пример полного разбиения системы на объекты. Параметрические множества объектов могут иметь общие параметры между собой. Например, на рисунке объекты O3 и O4 имеют общий параметр q12. Такие объекты называются *пересекающимися*. На практике большинство объектов являются пересекающимися.

Поскольку объект есть полное подмножество системы, то все наши определения по системе точно так же применимы и к объекту.

Так, процесс в объекте O определяется аналогично процессу в системе, как:

$$\boldsymbol{Z_O = \langle S_O, T_O, F_O, \alpha \rangle} \quad \text{где:} \quad S_o = \mathop{\Pi}\limits_{\forall q \in O} \sigma(q) \;\; ; \;\; \boldsymbol{F_O : T_O \rightarrow S_O}$$

Тогда можно дать определение процесса в объекте, как проекцию процесса в системе на пространство объекта:

$$\boldsymbol{Z_O} = \mathop{\Pi p}\limits_{O} Z$$

Операция объединения

Эта операция позволяет объединять процессы в нескольких объектах в один процесс.

Пространство S_Q называется *склейкой* пространств S_{Q_1} и S_{Q_2}, если $Q=Q_1 \cup Q_2$. Процессы Z_1 и Z_2, допускающие операцию объединения, называются *согласованными*.

Не будем давать строгие математические доказательства свойств склейки, а сформулируем несколько достаточно понятных утверждений.

Утверждение 1.

Два процесса: Z_1 с пространством состояний S_{Q_1} и Z_2 с пространством состояний S_{Q_2} согласованы, если $Q_1 \cap Q_2 = \varnothing$.

Утверждение 2.

Если $Z_1 = Пр_{S_{Q_1}} Z$ и $Z_2 = Пр_{S_{Q_2}} Z$, то процессы Z_1 и Z_2 согласованы.

Утверждение 3.

Пусть заданы процесс Z_1, определенный на интервале $\left[t_H^1, t_K^1\right]$, и Z_2, определенный на интервале $\left[t_H^2, t_K^2\right]$.

Если $\left[t_H^1, t_K^1\right] \cap \left[t_H^2, t_K^2\right] = \varnothing$, то процессы Z_1 и Z_2 согласованы.

Сформулированные выше утверждения позволяют выработать практически важные рекомендации по управлению процессами с тем, чтобы осуществить их объединение.

Лекция № 3

Оператор общего вида

Как строить отображение F?

Пусть имеем систему с параметрами Q. Определим оператор генерации процесса Z в системе, как   s = H (A, ω, t)   где:

H - оператор ;

S - пространство состояний системы, определенное ранее;

T - множество моментов времени *изменения* состояний системы s – элемент множества S $s \in S$ $t \in T$

А - множество аргументов оператора H ; **важно** , что $A \subseteq Q$;

ω - случайное число

Интерес представляет описание процесса в объектах.

Допустим, имеется объект О в системе Q. Тогда генерация процесса Z_O в объекте О может быть выполнена путем задания оператора H^O :

$$s = H^O (A^O, \omega ,t)$$

где:

s - элемент множества S^O ($s \in S^O$)

$t \in T^O$

A^O - множество аргументов оператора: H^O .

при этом необходимо отметить, что $A^O \subseteq \mathbf{Q}$;

ω - случайное число.

Важно обратить внимание на то, что, если пространство состояния объекта определяется на параметрах $O \subset Q$, то множество аргументов является самостоятельным подмножеством параметров всей системы Q. При этом необходимо учитывать, что состав элементов множества аргументов *в общем случае зависит от времени.* (см. рисунок А)

Отношение сцепленности и зависимости объектов.

Рассмотрим два объекта O_l и O_m в системе Q. Пусть $O_l \cap O_m = \varnothing$, а процессы в них заданы следующими операторами:

$$s_{t_i}^{O_l} = H^{O_l}\left(A_{t_i}^{O_l}, t_i, \omega\right); \qquad s_{t_i}^{O_m} = H^{O_m}\left(A_{t_i}^{O_m}, t_i, \omega\right).$$

Если $O_m \cap A_{t_i}^{O_l} = \varnothing$ и $O_l \cap A_{t_i}^{O_m} = \varnothing$, то такие процессы и объекты называются *не сцепленными* в момент времени t_i.

Если $O_l \cap A_{t_i}^{O_m} \neq \varnothing$, то объект O_m *сцеплен* с объектом O_l в момент времени t_i. То же относится и к их процессам. Это означает, что для определения состояния объекта O_m в момент времени t_i, необходимо знание состояния объекта O_l в это же время.

Обозначим отношение сцепления как $O_l \to O_m$. Если $O_m \cap A_{t_i}^{O_l} \neq \varnothing$, то объект O_l *сцеплен* с объектом O_m в момент времени t_i: $O_m \to O_l$.

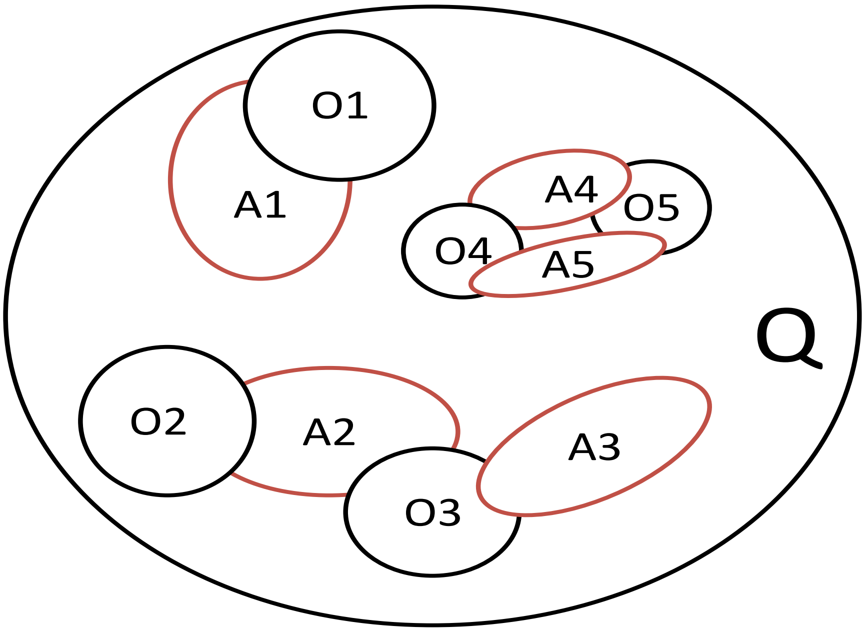


Рисунок А.

Если одновременно $O_m \cap A_{t_i}^{O_l} \neq \varnothing$ и $O_l \cap A_{t_i}^{O_m} \neq \varnothing$, то объекты O_m и O_l *взаимно-сцеплены* в момент времени t_i: $O_m \leftrightarrow O_l$. При операторном способе описания процессов всегда нежелательна модель, приводящая к появлению взаимно - сцепленных объектов, поскольку возникающую неопределенность приходится раскрывать путем решения в общем случае систем нелинейных уравнений, что может привести к непреодолимым трудностям. В дальнейшем будем стремиться создавать модели, не приводящие к взаимному сцеплению объектов.

Не следует смешивать отношение сцепления и зависимости. Так, если $O_m \to O_l$ и $O_l \to O_k$, то вовсе не обязательно, чтобы $O_m \to O_k$. Таким образом, отношение сцепления не является транзитивным.

Если к отношению сцепления добавить полное транзитивное замыкание, то получим *отношение зависимости*. Если O_l зависит от O_m, а O_k зависит от O_l, то O_k зависит и от O_m. Таким образом, *отношение сцепления можно определить как отношение непосредственной зависимости.*

Алгоритмическая модель процесса (АМП)

Общая идея

Поскольку, в общем случае, задание процесса в виде единого оператора (1) либо затруднительно, либо невозможно, то предлагается задавать оператор H в виде некоторой алгоритмической структуры.

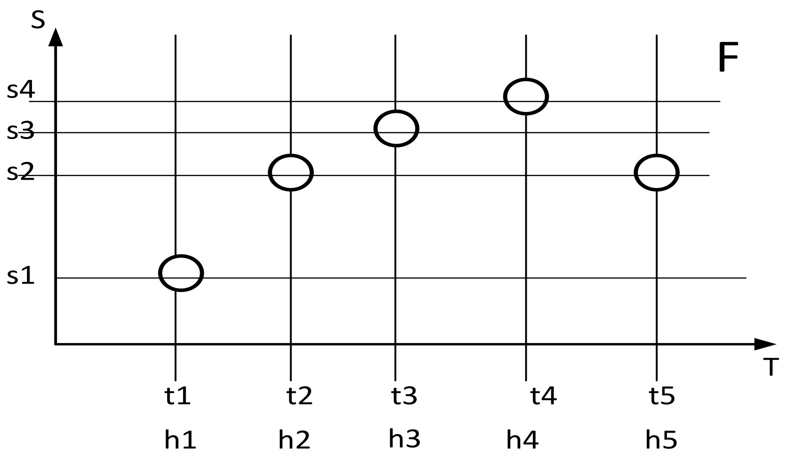
Рассмотрим дискретный процесс Z. Поставим в соответствие каждой i-ой точке процесса (момент времени изменения состояния t_i) некоторый оператор h_i^c. Оператор h_i^c вычисляет значение состояния $s_i \in S$ в момент времени t_i:

$$s_i = h_i^c\left(A_i, t_i, \omega\right)$$

Оператор h_i^c описывает вычисление только одной i-й точки процесса Z. В силу этого условия будем в дальнейшем называть этот оператор *элементарным*.

Таким образом, если график процесса содержит n точек, то мы должны задать линейную последовательность элементарных операторов:

$$h_1^c, h_2^c \ldots, h_i^c, \ldots h_n^c.$$



Инициатор

Введем новый элемент модели - *инициатор*. Будем полагать, что инициатор - это объект, обладающий следующими фундаментальными свойствами:

а) независимостью: может существовать самостоятельно без операторов;

б) динамичностью: инициатор имеет возможность перемещаться от оператора к оператору; будем называть попадание инициатора на оператор *сцеплением инициатора с элементарным оператором*;

в) инициативностью: в момент сцепления инициатора с оператором происходит выполнение (инициирование) элементарного оператора, что соответствует вычислению нового состояния процесса. Будем в дальнейшем полагать, что выполнение элементарного оператора происходит мгновенно.

Будем в дальнейшем полагать, что выполнение элементарного оператора происходит *мгновенно*. Таким образом, описание процесса может быть выполнено путем задания линейной последовательности операторов $\left\langle h_i^c \right\rangle_{i=1}^n$ и перемещения по этой последовательности инициатора ***I***, сцепляющегося с элементарными операторами h_i^c в заданные моменты времени t_i , изменения состояния процесса.

<u>Задание условий движения инициатора</u>

Предлагаемая модель описания процесса предполагает, что моменты сцепления инициатора с элементарными операторами определяют сами элементарные операторы. С этой целью введем в состав элементарного оператора h_i^c оператор h_i^y , который определяет условие, при выполнении которого инициатор покидает оператор h_i^c и сцепляется со следующим оператором h_{i+1}^c. Возможны следующие варианты задания такого условия:

а) указание момента времени сцепления инициатора с оператором h_{i+1}^c;

б) определение логического условия, при выполнении которого инициатор сцепляется с оператором h_{i+1}^c;

в) комбинированная форма, включающая варианты а) и б).

Таким образом:

$$h_i^y \in \left\{ h_i^t, h_i^л, h_i^{t,л} \right\},\ \text{где}$$

h^y_i - оператор условия продвижения инициатора;

h^t_i - оператор временного условия (соответствует варианту а);

$h^л_i$ - оператор логического условия (соответствует варианту б);

$h^{t,л}_i$ - оператор комбинированного условия (соответствует варианту в).

В наших описаниях на языке описания процессов будем использовать оператор **ждать** .

Расширим понятие элементарного оператора, добавив к нему помимо оператора h^c_i оператор h^y_i. Таким образом, определим элементарный оператор h_i, как двойку:

$$h_i = \left\langle h_i^c, h_i^y \right\rangle.$$

Теперь можно определить понятие *алгоритмической модели процесса* (в дальнейшем АМП), в виде тройки:

$$\text{АМП} = \left\langle \{h_i\}_{i=1}^{n}, \beta, I \right\rangle,$$

где: $\{h_i\}_{i=1}^{n}$ - множество элементарных операторов;

β- линейный порядок на $\{h_i\}_{i=1}^{n}$;

I- инициатор.

Следует обратить внимание на то, что АМП содержит *один и только один* инициатор, т.е. каждому процессу соответствует один инициатор. В этом смысле инициатор является представителем процесса, при его потери либо отсутствии развитие процесса прекращается.

Линейную последовательность элементарных операторов назовем треком TR:

$$TR = \left\langle \{h_i\}_{i=1}^{n}, \beta \right\rangle$$

Тогда можно АМП определить также как двойку:

АМП = <TR,I>

Структура трека

Пусть задан некоторый трек TR. В реальных приложениях трек содержит достаточно много элементарных операторов, выполняющих одни и те же операции над аргументами. Операторы эквивалентны, если при одних и тех же значениях аргументов они вычисляют одинаковые результаты. Это свойство трека позволяет задать отношение эквивалентности на множестве $\{h_i\}_{i=1}^{n}$ элементарных операторов трека TR.

Назовем *структурой* свертку трека TR по отношению эквивалентности элементарных операторов.

Пример. Пусть задан некоторый трек TR (рисунок 1).

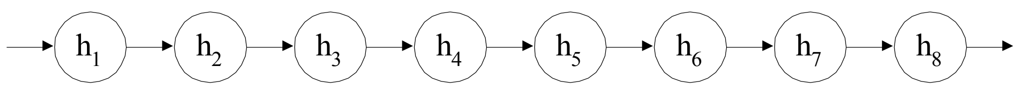


Рисунок 1. Пример трека

Пусть отношение эквивалентности элементарных операторов имеет вид:

$$\{(h_1, h_3), (h_2, h_5, h_6, h_8), (h_4, h_7)\}$$

Тогда структура имеет вид графа (рисунок 2).

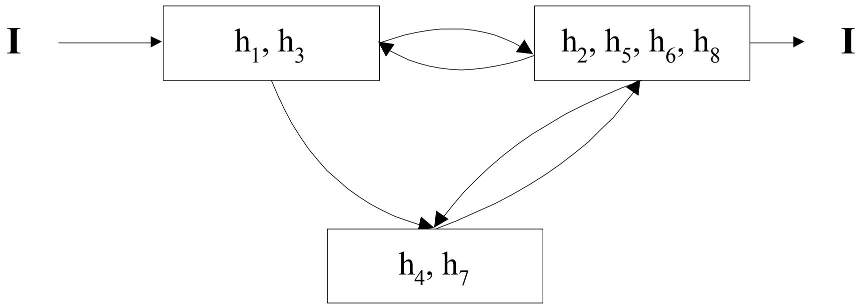


Рисунок 2. Свертка трека в структуру

Очевидно, если заданы трек и отношение эквивалентности операторов, то всегда возможно построение структуры. Однако обратное восстановление трека по структуре является неоднозначной операцией. Эта операция относится к классу операций развертки. С тем, чтобы операцию построения трека из структуры сделать однозначной, введем еще один тип элементарного оператора - *навигационный* элементарный оператор. Навигационный оператор определяется так же, как и элементарный оператор, однако в результате его выполнения определяется тот элементарный оператор в структуре, который должен выполняться следующим. Выполнение навигационного оператора инициируется инициатором. Поскольку время на выполнение навигационного оператора, как и всех элементарных операторов, равно нулю, то использование его не сказывается на времени реализации процесса. В общем случае навигационный оператор должен следовать за каждым элементарным оператором в структуре.

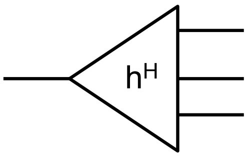


Рисунок 3. Обозначение навигационного оператора

Если навигационный оператор обозначить, как показано на рисунке 3, то структура из вышеописанного примера будет иметь вид (рисунок 4):

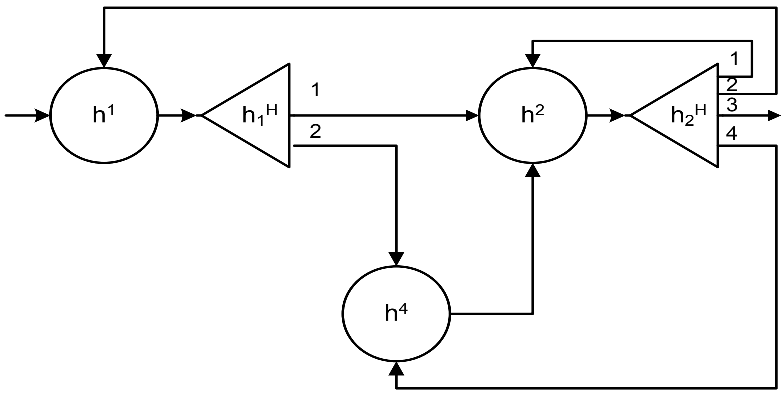


Рисунок 4. Вид структуры с навигационными операторами

Здесь операторы h_1, h_2, h_4 являются представителями своего класса эквивалентности. Как видно, после операторов h_1 и h_2 стоят навигационные операторы $h^{н}{}_1$ и $h^{н}{}_2$, в то время как после оператора h_4 нет необходимости в использовании навигационного оператора. Навигационный оператор используется также и для организации циклов (оператор $h^{н}{}_2$, выход 1).

Использование структуры по сравнению с треком позволяет значительно снизить размерность описания процесса. Однако необходимо иметь в виду, что процесс определен только в случае задания трека, а поэтому структура есть лишь способ более компактного описания трека, генерация самого трека остается необходимой операцией. На практике задание структуры с навигационными операторами для последующей генерации трека используется часто и повсеместно, где необходима генерация процесса.

Определение элементарного оператора в составе структуры можно представить в виде: $h = \langle h^{c}, h^{у}, h^{н} \rangle$.