Теория вычислительных процессов и структур

Лекция №12. Последовательные взаимодействующие процессы

Процессы

Введение

Имя – название события.

Имя каждого события обозначает целый ***класс*** событий. Отдельные вхождения события внутри одного ***класса*** разделены во времени.

Множество имен событий, выбранных для конкретного описания объекта, называется его ***алфавитом***.

Словом ***процесс*** обозначают поведение объекта постольку, поскольку оно может быть описано в терминах ограниченного набора событий, выбранного в качестве его алфавита.

Простой торговый автомат (ПТА)

мон – опускание монеты
шок – появление шоколадки

Сложный торговый автомат (СТА)

м1 – опускание монеты номиналом в 1
м2 – опускание монеты номиналом в 2
мал – появление маленькой булочки
бол – появление большой булочки
сд1 – появление сдачи номиналом в 1

Основные обозначения

1) Имена событий слова из строчных букв: *мон*, *шок;* а так же буквами *a*, *b*, *c* – произвольные события

2) Имена конкретных процессов из прописных букв: *ПТА, СТА;* а так же буквами *P*, *Q*, *R* – произвольные процессы

3) Буквы *x*, *y*, *z* используются для переменных обозначающих события

4) Буквы *A*, *B*, *C* используются для обозначения множества событий.

5) Буквы *X*, *Y* используются для переменных, обозначающих процессы.

6) Алфавит процесса *P* обозначается α*P*, например:

$\alpha ПТА = \{мон, шок\}$

$\alpha СТА = \{м1, м2, мал, бол, сд1\}$

7) Процесс с алфавитом *A*, такой, что в нем не происходит ни одно событие из *A*, назовем $СТОП_A$.

Префиксы

Пусть x — событие, а P — процесс. Тогда

$(x \to P)$ /читается как "P за x"/

описывает объект, который вначале участвует в событии x, а затем ведет себя в точности как P.

Более формально:

$\alpha(x \to P) = \alpha P$, *если* $x \in \alpha P$

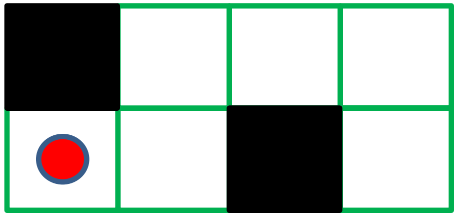
Примеры

1) Простой торговый автомат, который поглощает монету и ломается:

$(мон \to \text{СТОП}_{\alpha ПТА})$

2) Простой торговый автомат, который обслуживает двоих покупателей и ломается:

$(мон \to (шок \to (мон \to (шок \to \text{СТОП}_{\alpha ПТА}))))$



3) Фишка может двигаться по полю только вверх и вправо:

$\alpha ФИШ = \{вверх, вправо\}$

$ФИШ = (вправо \to вверх \to вправо \to вправо \to \text{СТОП}_{\alpha ФИШ})$

Рекурсия

Рассмотрим объект *ЧАСЫ*: α*ЧАСЫ* = {*тик*}; а теперь объект, который сначала издает один *тик,* а затем ведет себя так же как *ЧАСЫ*: (*тик* → *ЧАСЫ*). Объекты одинаковы, что позволяет сформулировать равенство *ЧАСЫ* = (*тик* → *ЧАСЫ*).

Описание процесса, начинающееся с префикса, называется ***предваренным***. Если $F(X)$ – предваренное выражение, содержащее имя процесса X, а A – алфавит X, то мы утверждаем, что уравнение $X = F(X)$ имеет единственное решение в алфавите A. Иногда удобнее обозначать это решение выражением $\mu X : A.F(X)$ Здесь X является локальным именем (связанной переменной) и может произвольно изменяться.

Примеры

1) Вечные часы:

   ЧАСЫ = μX : {*тик*}.(*тик* → X)

2) Вечный простой торговый автомат:

   ПТА = (*мон* → (*шок* → *ПТА*))

   ПТА = μX : {*мон*, *шок*}. (*мон* → (*шок* → X))

3) Два автомата по размену монеты номиналом в 5 с одинаковым алфавитом:

   α*РАЗМ5* = {*м5, сд2, сд1*}

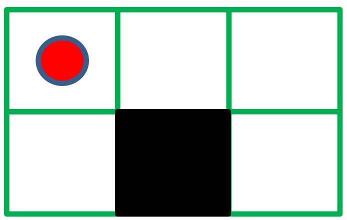
   РАЗМ5А = (*м5* → *сд1* → *сд2* → *сд2* → *РАЗМ5А*)

   РАЗМ5В = (*м5* → *сд1* → *сд1* → *сд1* → *сд2* → *РАЗМ5В*)

Выбор

Если x и y – различные события, то $(x \rightarrow P \mid y \rightarrow Q)$ описывает объект, который сначала участвует в одном из событий x, y. Последующее же поведение объекта описывается процессом P, если первым произошло событие x, или Q, если первым произошло событие y. Вертикальную черту | следует читать как "или" : "P за x или Q за y".

Примеры



1) Возможные перемещения фишки по полю описываются процессом:

$$(\textit{вниз} \rightarrow \textit{СТОП} \mid \textit{вправо} \rightarrow \textit{вправо} \rightarrow \textit{вниз} \rightarrow \textit{СТОП})$$

2) Автомат по размену монеты номиналом в 5 с двумя вариантами размена:

$$\textit{РАЗМ5С} = \textit{м5} \rightarrow (\textit{сд1} \rightarrow \textit{сд2} \rightarrow \textit{сд2} \rightarrow \textit{РАЗМ5С} \mid$$
$$\textit{сд1} \rightarrow \textit{сд1} \rightarrow \textit{сд1} \rightarrow \textit{сд2} \rightarrow \textit{РАЗМ5С})$$

3) Автомат предлагающий на выбор шоколадку или ириску:

$$\textit{ТАШИ} = \mu X : \textit{мон} \rightarrow (\textit{шок} \rightarrow X \mid \textit{ирис} \rightarrow X)$$

Определение выбора легко обобщить на случай более чем двух альтернатив, например:

$$(x \to P \mid y \to Q \mid \ldots \mid z \to R)$$

В общем случае если B – некоторое множество событий, а $P(x)$ – выражение, определяющее процесс для всех различных x из B, то запись

$$(x : B \to P(x))$$

определяет процесс, который сначала предлагает на выбор любое событие y из B, а затем ведет себя как $P(y)$**.** Запись следует читать как «P от x за x из P». Множество B определяет начальное ***меню*** процесса, потому что в нем предлагается набор действий, из которого осуществляется первоначальный выбор.

Процесс, который в каждый момент времени может участвовать в любом событии из своего алфавита A:

$$\alpha \text{ИСП}_{\text{A}} = A$$

$$\text{ИСП}_{\text{A}} = (x : A \to \text{ИСП}_{\text{A}})$$

В особом случае, когда в начальном меню содержится лишь одно событие

$$(x : \{e\} \to P(x)) = (e \to P(e))$$

потому что e является единственным возможным начальным событием. В еще более специальном случае, когда начальное меню пусто, вообще ничего не происходит, и поэтому

$$(x : \{\ \} \to P(x)) = \mathit{СТОП}$$

Двуместный оператор выбора | можно определить и в более общих обозначениях:

$$(a \to P \mid b \to Q) = (x : B \to R(x)), \text{ где } B = \{a, b\}, \text{ а } R(x) = \mathbf{if}\ x = a\ \mathbf{then}\ P\ \mathbf{else}\ Q.$$

Взаимная рекурсия

Рекурсия позволяет определить единственный процесс как решение некоторого единственного уравнения. Эта техника легко обобщается на случай решения систем уравнений с более чем одним неизвестным. Для достижения желаемого результата необходимо, чтобы правые части всех уравнений были предваренными, а каждый неизвестный процесс входил ровно один раз в правую часть одного из уравнений.

Примеры

1) Автомат с газированной водой имеющий два режима – лимон и апельсин.

$\alpha АГАЗ = \alpha G = \alpha W = \{устлимон, устапельсин, мон, лимон, апельсин\}$

$АГАЗ = (устлимон \to G \mid устапельсин \to W)$

$G = (мон \to лимон \to G \mid устапельсин \to W)$

$W = (мон \to апельсин \to W \mid устлимон \to G)$

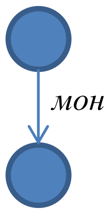
2) Объект двигающийся с земли вертикально вверх или вниз, а находясь на земле - вокруг. Пусть n – некоторое число из натурального ряда {0, 1, 2, ...}. Для каждого такого n введем пронумерованное имя CTn, с помощью которого будем описывать поведение объекта, находящегося в n шагах от земли. Начальное поведение определяется уравнением

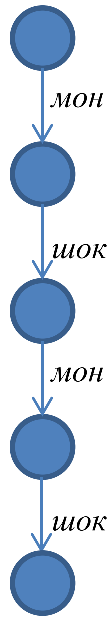
$CT_0 = (вверх \to CT_1 \mid вокруг \to CT_0)$

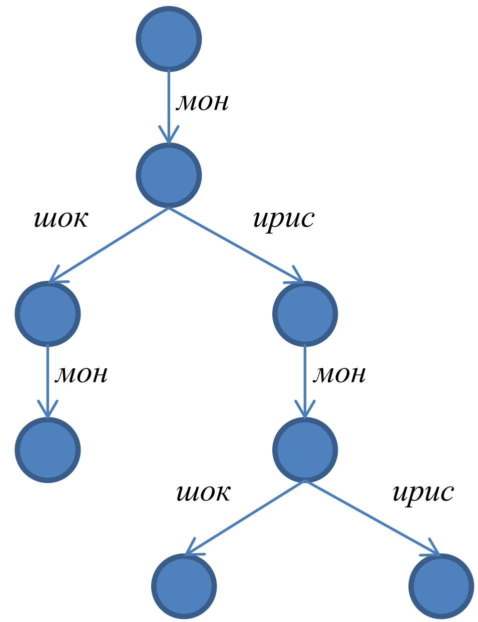
а остальные уравнения образуют бесконечную систему

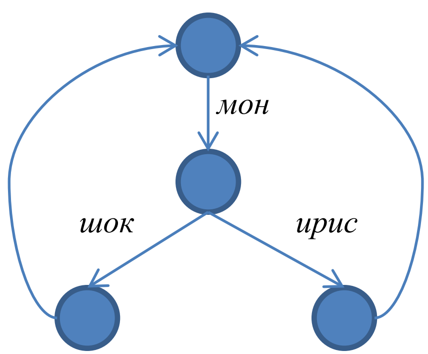
$CT_{n+1} = (вверх \to CT_{n+2} \mid вниз \to CT_n)$, где n пробегает натуральный ряд 0,1,2, …

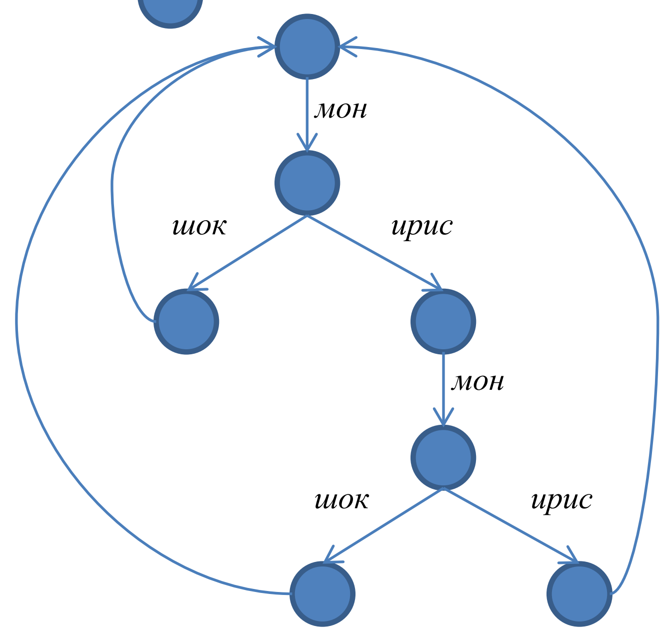
Рисунки


мон


мон
шок
мон
шок


мон
шок
ирис
мон
мон
шок
ирис


мон
шок
ирис


мон
шок
ирис
мон
шок
ирис

Законы

Два процесса, определенные с помощью оператора выбора, одинаковы, если на первом шаге они предлагают одинаковые альтернативы и после одинакового первого шага ведут себя одинаково.

L1) $(x : A \rightarrow P(x)) = (y : B \rightarrow Q(y)) \equiv (A = B \ \& \ \forall x \in A . P(x) = Q(x))$

L1A) $СТОП \neq (d \rightarrow P)$

L1B) $(c \rightarrow P) \neq (d \rightarrow Q)$ если $c \neq d$

L1C) $(c \rightarrow P \mid d \rightarrow Q) = (d \rightarrow Q \mid c \rightarrow P)$

L1D) $(c \rightarrow P) = (c \rightarrow Q) \equiv P = Q$

Примеры

1) $(мон \rightarrow шок \rightarrow мон \rightarrow шок \rightarrow СТОП) \neq (мон \rightarrow СТОП)$

Доказательство следует из L1D и L1A.

2) $\mu X.(мон \rightarrow (шок \rightarrow X \mid ирис \rightarrow X)) = \mu X.(мон \rightarrow (ирис \rightarrow X \mid шок \rightarrow X))$

Доказательство следует из L1C.

Всякое должным образом предваренное рекурсивное уравнение имеет единственное решение.

L2) Если $F(X)$ – предваренное выражение, то $(Y = F(Y)) \equiv (Y = \mu X.F(X))$

L2A) $\mu X.F(X) = F(\mu X.F(X))$

Пример

1) Пусть *TA1* = (*мон* → *TA2*), а *TA2* = (*шок* → *TA1*). Требуется доказать, что *TA1* = *ПТА* [*ПТА*= (*мон* → (*шок* → *ПТА*))]

TA1 = (*мон* → *TA2*)

TA1 = (*мон* → (*шок* → *TA1*))

Соответственно *TA1* и *ПТА* являются решением одного и того же рекурсивного уравнения, а так как оно предваренное, оно имеет единственное решение, что, в свою очередь, означает, что *TA1* и *ПТА* – разные имена одного и того же процесса.

Закон L2 можно распространить на случай взаимной рекурсии. Систему рекурсивных уравнений можно записать в общем виде, используя индексы

$X_i = F(i, X)$ для всех i из S.

где S – множество индексов, взаимно однозначно соответствующих множеству уравнений, X – ***массив*** процессов с индексами из S, а $F(i, X)$ – предваренное выражение

L3) При соблюдении вышеописанных условий если $(\forall i : S.(X_i = F(i, X)\ \&\ Y_i = F(i, Y)))$, то $X = Y$

Протоколы

Протоколом поведения процесса называется конечная последовательность символов, фиксирующая события, в которых процесс участвовал до некоторого момента времени.

Мы будем обозначать протокол последовательностью символов, разделенной запятыми и заключенной в угловые скобки:

<*x*, *y*> состоит из двух событий — *x* и следующего за ним *y*.

<x> состоит из единственного события x.

<> пустая последовательность, не содержащая событий.

Примеры

1) ПТА в момент завершения обслуживания первых двух покупателей:

<*мон, шок, мон, шок*>

2) СТА имеет семь различных протоколов длинны два и менее:

<>

<*м1*> <*м2*>

<*м2, бол*> <*м2, мал*> <*м1, м1*> <*м1, мал*>

Из четырех протоколов длины два для данного устройства фактически может произойти только один. Его выбор по своему желанию осуществляет первый покупатель.

Операции над протоколами

Введем следующие соглашения:

s, t, u обозначают протоколы,

S, T, U обозначают множества протоколов,

f, g, h обозначают функции.

Конкатенация

Конкатенация – построение нового протокола из пары операндов $s\hat{}t$. Например:

<*мон, шок*>^<*мон, ирис*> = <*мон, шок, мон, ирис*>

Самые важные свойства конкатенации — это ее ассоциативность и то, что пустой протокол $\langle\rangle$ служит для нее единицей.

L1) $s\hat{}\langle\rangle = \langle\rangle\hat{}s = s$

L2) $s\hat{}(t\hat{}u) = (s\hat{}t)\hat{}u$

L3) $s\hat{}t = s\hat{}u \equiv t = u$

L4) $s\hat{}t = u\hat{}t \equiv s = u$

L5) $s\hat{}t = \langle\rangle \equiv s = \langle\rangle \ \&\ t = \langle\rangle$

Пусть f — функция, отображающая протоколы в протоколы. Мы говорим, что функция ***строгая***, если она отображает пустой протокол в пустой протокол:

$f(\langle\rangle) = \langle\rangle$

Будем говорить, что функция ***дистрибутивна,*** если

$f(s\hat{}t) = f(s)\hat{}f(t)$

Если n — натуральное число, то t^n будет обозначать конкатенацию n копий протокола t. Ее легко определить индукцией по n:

L6) $t^0 = \langle\rangle$

L7) $t^{n+1} = t\,^\wedge t^n$

L8) $t^{n+1} = t^n\,^\wedge t$

L9) $(s\,^\wedge t)^{n+1} = s\,^\wedge(t\,^\wedge s)^n\,^\wedge t$

Сужение

Выражение ($t \upharpoonright A$) обозначает протокол t, суженный на множество символов A; он строится из t, отбрасыванием всех символов не принадлежащих A. Например:

$\langle$*вокруг, вверх, вниз, прямо, вниз*$\rangle \upharpoonright \langle$*вверх, вниз*$\rangle = \langle$*вверх, вниз, вниз*$\rangle$

L1) $\langle\rangle \upharpoonright A = \langle\rangle$

L2) $(s\,^\wedge t) \upharpoonright A = (s \upharpoonright A)\,^\wedge\,(t \upharpoonright A)$

L3) $\langle x\rangle \upharpoonright A = \langle x\rangle$ если $x \in A$

L4) $\langle y\rangle \upharpoonright A = \langle\rangle$ если $y\ !\!\in A$

L5) $s \upharpoonright \{\} = \langle\rangle$

L6) $(s \upharpoonright A) \upharpoonright B = s \upharpoonright (A \cap B)$

Голова и хвост

Если s — непустая последовательность, обозначим ее первый элемент s_0, а результат, полученный после его удаления – s' Например:

$\langle x,y,z\rangle_0 = x$; $\langle x,y,z\rangle' = \langle y,z\rangle$

Звездочка

Множество A^* —это набор всех конечных протоколов (включая $\langle\rangle$), составленных из элементов множества A. После сужения на A такие протоколы остаются неизменными. Отсюда следует простое определение:

$$A^* = \{s \mid s \upharpoonright A = s\}$$

L1) $\langle\rangle \in A^*$

L2) $\langle x\rangle \in A^* \equiv x \in A$

L3) $(s\frown t) \in A^* \equiv s \in A^* \;\&\; t \in A^*$

L4) $A^* = \{t \mid t = \langle\rangle \vee (t_0 \in A \;\&\; t' \in A^*)\}$

Порядок

Если s – копия некоторого начального отрезка t то можно найти такое продолжение u последовательности s, что $s\frown u = t$. Поэтому мы определим отношение порядка

$$s \le t = (\exists\, u.s\frown u = t)$$

и будем говорить, что s является префиксом t.

L1) $\langle\rangle \le s$

L2) $s \le s$

L3) $s \le t \;\&\; t \le s \Rightarrow t = s$

L4) $s \le t \;\&\; t \le u \Rightarrow s \le u$

L5) $(\langle x\rangle\frown s) \le t \equiv t \ne \langle\rangle \;\&\; x = t_0 \;\&\; s \le t'$

L6) $s \le u$ & $t \le u \Rightarrow s \le t$ & $t \le s$

Если s является подпоследовательностью t (не обязательно начальной), мы говорим, что s находится в t это можно определить так:

s в $t = (\exists\, u, v. t = u\hat{\ }s\hat{\ }v)$

L7) $(\langle x\rangle\hat{\ }s)$ в $t \equiv t \neq \langle\rangle$ & $((x = t_0$ & $s \le t') \vee (\langle x\rangle\hat{\ }s)$ в $t')$

Будем говорить, что функция f из множества протоколов во множество протоколов ***монотонна,*** если она сохраняет отношение порядка $\le$ т. е. $f(s) \le f(t)$ всегда, когда $s \le t$.

Длина

Длину протокола t будем обозначать $\#t$ Например,

$\#\langle x,y,z\rangle = 3$

Следующие законы определяют операцию # :

L1) $\#\langle\rangle = 0$

L2) $\#\langle x\rangle = 1$

L3) $\#(s\hat{\ }t) = (\#s) + (\#t)$

Число вхождений в t символов из A вычисляется выражением $\#(t \upharpoonright A)$.

L4) $\#(t \upharpoonright (A \cup B)) = \#(t \upharpoonright A) + \#(t \upharpoonright B) - \#(A \cap B))$

L5) $s \le t \Rightarrow \#s \le \#t$

L6) $\#(t^n) = n \times (\#t)$

Протоколы процессов

Мы ввели понятие протокола как последовательной записи поведения процесса вплоть до некоторого момента времени. До начала процесса неизвестно, какой именно из возможных протоколов будет записан: его выбор зависит от внешних по отношению к процессу факторов. Однако полный набор всех возможных протоколов процесса P может быть известен заранее, и мы введем функцию ***протоколы* (*P*)** для обозначения этого множества.

Примеры

1) Единственным возможным протоколом процесса *СТОП* является $\langle\rangle$.

   $\textit{протоколы (СТОП)} = \{\langle\rangle\}$

2) Автомат, поглощающий монету, прежде чем сломаться имеет два протокола:

   $\textit{протоколы} (\textit{мон} \to \textit{СТОП}) = \{\langle\rangle, \langle \textit{мон} \rangle\}$

3) Тикающие часы:

   $\textit{протоколы} (\mu X.\textit{тик} \to X) = \{\langle\rangle, \langle \textit{тик} \rangle, \langle \textit{тик}, \textit{тик} \rangle, \ldots\} = \{\textit{тик}^*\}$

4) ПТА:

   $\textit{протоколы} (\mu X.\textit{мон} \to \textit{шок} \to X) = \{s \mid \exists\, n.s \leq \langle \textit{мон}, \textit{шок} \rangle^n\}$

Законы

Законы помогают вычислить множество протоколов любого процесса, определенного с помощью уже введенных обозначений. Протокол процесса *СТОП:*

L1) *протоколы* (*СТОП*) $= \{t \mid t = \langle\rangle\} = \{\langle\rangle\}$

Протокол процессы $(c \rightarrow P)$:

L2) *протоколы* $(c \rightarrow P) = \{t \mid t = \langle\rangle \vee (t_0 = c \;\&\; t' \in$ *протоколы*$(P))\}$

$= \{\langle\rangle \cup \{\langle c\rangle \wedge t \mid t \in$ *протоколы*$(P)\}\}$

Протокол процесса выбора:

L3) *протоколы* $(c \rightarrow P \mid d \rightarrow Q) = \{t \mid t = \langle\rangle \vee (t_0 = c \;\&\; t' \in$ *протоколы*$(P))$

$\vee (t_0 = d \;\&\; t' \in$ *протоколы*$(\mathrm{Q}))$

Конструкция выбора в общем случае:

L4) *протоколы* $(x : B \rightarrow P(x)) = \{t \mid t = \langle\rangle \vee (t_0 = B \;\&\; t' \in$ *протоколы*$(P(t_0)))\}$

Протокол рекурсивно определенного процесса:

L5) *протоколы* $(\mu X : A.F(X)) = \bigcup_{n \geq 0}$ протоколы$(F^n(\text{СТОП}_\text{A}))$

Протокол – это последовательность символов, фиксирующих события, в которых процесс *P* участвовал до некоторого момента времени. Отсюда следует, что $\langle\rangle$ является протоколом любого процесса до момента наступления его первого события. Кроме того, если (*s*^*t*) протокол процесса до некоторого момента, то *s* должен быть протоколом того же процесса до некоторого более раннего момента времени. Наконец, каждое происходящее событие должно содержаться в алфавите процесса.

Три этих факта находят свое формальное выражение в законах:

L6) $<> \in$ *протоколы*(P)

L7) $s \hat{} t \in$ *протоколы*(P) => $s \in$ *протоколы*(P)

L8) *протоколы*$(P) \in (\alpha P)^*$

После

Если $s \in$ *протоколы{P),* то P/s (Р после s) – это процесс ведущий себя так, как ведет себя P с момента завершения всех действий, записанных в протоколе s. Если s не является протоколом P, то (P/s) не определено.

Следующие законы раскрывают значение операции / . Ничего не сделав, процесс остается неизменным:

L1) $P/<> = P$

Поведение процесса P после завершения событий из $s \hat{} t$ совпадает с поведением (P/s) после завершения событий из t:

L2) $P/(s \hat{} t) = (P/s)/t$

Если процесс участвовал в событии c, его дальнейшее поведение определяется именно этим начальным выбором:

L3) $(x : B \rightarrow \mathrm{P}(x))/<c> = P(c)$ если $c \in B$

L4) *протоколы* $(P/s) = \{t \mid s \hat{} t \in$ *протоколы*$(P) \}$ если $s \in$ *протоколы*(P)

По определению процесс P — ***циклический***, если при любых обстоятельствах он может вернуться в начальное состояние, т. е.

$\forall s :$ *протоколы*(P). $\exists t.(P/(s \hat{} t) = P)$

Дальнейшие операции над протоколами

Замена символа

Пусть *f* – функция, отображающая символы из множества *A* в множество *B*. С помощью *f* можно построить новую функцию *f**, отображающую последовательность символов из *A** в последовательность символов из *B**, путем применения *f* к каждому элементу последовательности. Например, если *удвоить* – функция, удваивающая свой целый аргумент, то *удвоить**(<1,2,4,9>) = <2,4,8,18>

Конкатенация

Пусть *s* – последовательность, каждый элемент которой в свою очередь является последовательностью. Тогда ^/*s* получается из *s* конкатенацией всех ее элементов в их исходном порядке, например: ^/<<1,3>, <>, <2,7>> = <1,3>^<>^<2,7> = <1,3,2,7>

Чередование

Последовательность *s* является чередованием последовательностей *t* и *u*, если ее можно разбить на серию подпоследовательностей, которые, чередуясь, представляют собой подпоследовательности из *t* и *u*. Например: *s* = <1,2,4,7,8,3,9> является чередованием *t* = <1,2,7,9> и *u* = <4,8,3>

Индекс

Если $0 \leq i \leq \#s$, то мы используем привычную запись $s[i]$ для обозначения i-го элемента последовательности s, как это описано:
$s[0] = s_0$ & $s[i+1] = s'[i]$ при условии, что $s \neq \langle\rangle$

Обратный порядок

Если s – последовательность, то $\overline{s}$ получается из s взятием ее элементов в обратном порядке. Например: $\overline{\langle 3,6,99 \rangle} = \langle 99,6,3 \rangle$

Выборка

Если s – последовательность пар, определим $s \downarrow x$ как результат выборки из s всех пар, первый элемент которых есть x**,** и замены каждой такой пары на ее второй элемент. Первый и второй элементы пары будем разделять точкой. Так, если
$s = \langle a.9, b.7, a.5, c.8 \rangle$, то $s \downarrow a = \langle 9, 5 \rangle$, а $s \downarrow d = \langle\rangle$

Композиция

Пусть символ √ означает успешное завершение процесса. Значит, этот символ может появиться только в конце протокола. Пусть *t* – протокол, отражающий последовательность событий с того момента, как *s* успешно завершился. Композицию *s* и *t* обозначим (*s; i*)**.** Если в *s* отсутствует символ √ , то *t* не может начаться. Если же символ √ присутствует в конце *s*, то он удаляется а *t* присоединяется к результату. Символ √ можно рассматривать как своего рода "клей", склеивающий *s* и *t*; при отсутствии клея *t* не может прилипнуть. Если символ √ встречается (что ошибочно) в середине протокола, то из соображений общности условимся, что все символы после его первого вхождения следует считать ошибочными и опустить.