УДК 004.415.52

Правила доказательства корректности предикатных программ

В.И. Шелехов

Институт Систем Информатики СО РАН, Новосибирский государственный университет

Описывается система правил доказательства корректности предикатной программы. Данное руководство ориентировано на применение для дедуктивной верификации реальных производственных программ.

***Ключевые слова**: тотальная корректность программы, дедуктивная верификация, формальная операционная семантика*

Введение

Определяемая в данной работе система правил доказательства корректности собрана из различных работ по предикатному программированию [8-18].

1. Язык предикатного программирования

Полная предикатная программа состоит из набора рекурсивных *предикатных программ* на языке P [3] следующего вида:

<имя программы>(<описания аргументов>**:** <описания результатов>)
pre <предусловие>
post <постусловие>
measure <выражение>
{ <оператор> }

Предусловие и постусловие являются формулами на языке исчисления предикатов. Они обязательны при дедуктивной верификации [8, 9]. Мера задается только для рекурсивных программ.

Ниже представлены основные конструкции языка P: оператор присваивания, блок (оператор суперпозиции), параллельный оператор, условный оператор, вызов программы и описание переменных, используемое для аргументов, результатов и локальных переменных.

<переменная> = <выражение>
{<оператор1>; <оператор2>}
<оператор1> || <оператор2>
if (<логическое выражение>) <оператор1> **else** <оператор2>
<имя программы>(<список аргументов>**:** <список результатов>)
<тип> <пробел> <список имен переменных>

Всякая переменная характеризуется *типом* – множеством допустимых значений. *Описание типа* **type** T(p) = D с возможными параметрами p связывает имя типа T с его изображением D. Типы **bool**, **int**, **real** и **char** являются *примитивными*. Значением типа

array(T_e, T_i) является *массив* с *элементами массива* типа T_e и *индексами* конечного типа T_i. Тип массива является предикатным типом, его значения (массивы) являются тотальными и однозначными предикатами.

Пусть E(x) – логическое выражение. Тип **subtype**(T x**:** E(x)) определяет *подтип* типа T при истинном предикате E(x), т.е. множество {x∈T | E(x)}. Определенный в языке P тип целых чисел **nat** представляется описанием:

type nat = **subtype**(**int** x: x ≥ 0) .

Допускаются подтипы, параметризуемые переменными. Примером является тип *диапазона* целых чисел:

type Diap(**nat** n) = **subtype**(**int** x: x ≥ 1 & x ≤ n) .

В языке P для изображения типа диапазона используется конструкция 1..n.

Описания типов переменных являются частью спецификации программы. Описание переменной T x есть утверждение x∈T, которое становится частью предусловия, если x – аргумент предикатной программы, или частью постусловия, если x – результат программы. При этом утверждение x∈T обычно не пишется в составе предусловия или постусловия, хотя предполагается.

Гиперфункция – программа с несколькими *ветвями* результатов. Гиперфункция B(x**:** y**:** z) имеет две ветви результатов y и z. Исполнение гиперфункции завершается одной из ветвей с вычислением результатов по этой ветви; результаты других ветвей не вычисляются.

Вызов гиперфункции записывается в виде B(x**:** y #M1**:** z #M2). Здесь M1 и M2 – метки программы, содержащей вызов. Операторы перехода #M1 и #M2 встроены в ветви вызова. Исполнение вызова либо завершается первой ветвью с вычислением y и переходом на метку M1, либо второй ветвью с вычислением z и переходом на метку M2.

Вызов гиперфункции может комбинироваться с операторами обработки ветвей:

B(x**:** y #M1**:** z #M2) **case** M1: C(y**:** u) **case** M2: D(z**:** u) .

Вызов вида B(x**:** y #M1**:** z #M2); M1**: ...** может быть представлен оператором B(x**:** y**:** z #M2).

Формально гиперфункция определяется через предикатную программу следующего вида:

```
B(x: y, z, e)
pre P(x) post e = E(x) & (E(x) ⇒ S(x, y)) & (¬E(x) ⇒ R(x, z))
{ ... };
```

Здесь x, y и z – непересекающиеся возможно пустые наборы переменных; P(x), E(x), S(x, y) и R(x, z) – логические утверждения. Предположим, что все присваивания вида e = **true** и e = **false** – последние исполняемые операторы в теле программы B. Программа B может быть заменена следующей программой в виде *гиперфункции*:

```
B(x: y #1: z #2)
pre P(x)   pre 1: E(x) post 1: S(x, y) post 2: R(x, z)
{ ... };
```

В теле гиперфункции каждое присваивание e = **true** заменено оператором перехода #1, а e = **false** – на #2. *Метки* 1 и 2 – дополнительные параметры, определяющие два различных *выхода* гиперфункции.

Спецификация гиперфункции состоит из двух частей. Утверждение после "**pre** 1" есть предусловие первой ветви; предусловие второй ветви – отрицание предусловия первой

ветви. Утверждения после “**post** 1” и “**post** 2” есть постусловия для первой и второй ветвей, соответственно.

Аппарат *гиперфункций* является более общим и гибким по сравнению с известным механизмом обработки исключений, например, в таких языках, как Java и C++. Традиционные подходы в реализации обработки аварийных ситуаций предполагают заведение дополнительных структур, усложняющих программу. Этого удается избежать при использовании гиперфункций. Использование гиперфункций делает программу короче, быстрее и проще для понимания [10, 11].

2. Формула тотальной корректности

Предикатная программа H(x: y) с аргументами x и результатами y есть предикат в форме вычислимого оператора.

Для языка предикатного программирования **P** [3] построена формальная операционная семантика [4], определяющая сильнейший предикат $\mathcal{R}(H)$, истинный после завершения исполнения программы H(x: y). Точнее, формальная семантика определена предикатом:

$\mathcal{R}(H)(x, y) \cong$ для значения набора x исполнение программы H всегда завершается и существует исполнение программы, при котором результатом вычисления является значение набора y.

Здесь учитывается специфика неоднозначных программ. Для таких программ будут нужны дополнительные условия корректности. Отметим, что предикатная программа всегда однозначна. Неоднозначность возникнет при введении дополнительной неоднозначной операции, например такой, как датчик случайных чисел.

Спецификацией программы H(x: y) являются два предиката: *предусловие* P(x) и *постусловие* Q(x, y). Спецификация записывается в виде: [P(x), Q(x, y)]. *Тотальная корректность* программы H(x: y) относительно спецификации обозначается как

H(x: y) **corr** [P(x), Q(x, y)]

и определяется формулой:

$$\forall x.\ P(x) \Rightarrow [\forall y.\ \mathcal{R}(H)(x, y) \Rightarrow Q(x, y)\ ]\ \&\ \exists y.\ \mathcal{R}(H)(x, y) \qquad (1)$$

Данная формула получена конкатенацией формулы *частичной корректности* $P(x)\ \&\ \mathcal{R}(H)(x, y) \Rightarrow Q(x, y)$ и *условия завершения* $P(x) \Rightarrow \exists y.\ \mathcal{R}(H)(x, y)$.

Для предикатных программ определяется тотальная корректность, а не частичная, поскольку предикатные программы обязаны всегда завершаться. Предикатные программы принадлежат классу программ-функций [6], не взаимодействующих с внешним окружением программы. Такие программы должны всегда завершатся, поскольку бесконечно работающие и невзаимодействующие программы бесполезны.

Доказано тождество $\mathcal{R}(H) = H$ [4]. Учитывая это, далее в правилах будем опускать $\mathcal{R}$, подразумевая, что оператор понимается в смысле соответствующего предиката формальной семантики.

Обозначение H(x: y) **corr** [P(x), Q(x, y)] удобнее по сравнению с ранее используемым обозначением Corr(H, P, Q)(x), применяемым также при доказательстве правил корректности [7] на PVS [5].

Формулу тотальной корректности будем также представлять в виде правила **COR**:

$$\textbf{COR}:\ \frac{\forall x,y.\ P(x)\ \&\ H(x\textbf{:}\ y) \Rightarrow Q(x,\ y);\quad \forall x.\ P(x) \Rightarrow \exists y.\ H(x\textbf{:}\ y)}{H(x\textbf{:}\ y)\ \textbf{corr}\ [P(x),\ Q(x,\ y)]}$$

Формулы выше горизонтальной черты называются *посылками* правила и именуются как COR.1 и COR.2. Формула ниже горизонтальной черты называется *заключением* правила.

3. Система правил доказательства корректности

Для базисных операторов (параллельного, условного и суперпозиции) разработана универсальная система правил доказательства их корректности [8, 9], в том числе и при наличии рекурсивных вызовов. Использование правил существенно упрощает доказательство формулы тотальной корректности (1) по сравнению с доказательством формулы (1) без применения правил. Применение правил декомпозирует формулу (1) к набору более коротких и простых формул корректности.

На базе системы правил в системе предикатного программирования реализован генератор формул корректности программы для подмножества языка **P** [3]. Часть формул доказывается автоматически SMT-решателем CVC4. Оставшаяся часть формул генерируется для системы интерактивного доказательства PVS. Подсистема дедуктивной верификации применялась в рамках курса «Формальные методы в программной инженерии» в 2013-2018гг. для генерации формул корректности, которые далее доказывались студентами в системе PVS.

Корректность правил, т.е. их истинность, доказана [7] в системе PVS [5].

Наша система правил по сравнению с правилами для троек Хоара [2] порождает более простые формулы корректности. Сравнением проводилось на одном примере [8]. Наши формулы корректности проще, поскольку система правил специализирована для оператора суперпозиции и параллельного оператора, а также для разных видов оператора суперпозиции, тогда как в системе Хоара [2] имеется лишь одно правило для последовательного оператора. Следует также учитывать, что система верификации Флойда-Хоара [1, 2] реализует проверку частичной корректность, тогда как в предикатном программировании проверяется тотальная корректность.

Исторически первой была система правил для программ с однозначной спецификацией [9]. В этой системе вывод проводился в обратную сторону: из спецификации доказывалась программа. Опыт ее применения показал, что она проигрывает описываемой здесь универсальной системе правил.

Набор формул корректности программы, полученных применением правил, вместе с сопутствующими описаниями формул, типов, констант и переменных оформляются в виде теории на языке **P**. Далее теория транслируется на язык спецификаций одной из систем дедуктивной верификации: PVS или Why3.

Для аргументов типа **subtype**(T x**:** E(x)) подформула E(x) считается принадлежащей предусловию, а для результатов – подформула E(x) должна быть вынесена в постусловие. Например, программа A(**nat** x**: nat** y) **pre** pA(x) **post** qA(x, y){...}

должна быть преобразована к виду A(**nat** x**: int** y) **pre** pA(x) **post** y>=0 & qA(x, y){...} для PVS. Вынесение y>=0 необходимо. Иначе в редких случаях доказательство в PVS условия y>=0 может быть пропущено. Ослабление формул корректности критически опасно для дедуктивной верификации.

Система правил состоит из нескольких групп правил. В первой группе общие правила, не использующие спецификаций подоператоров. Во второй группе допускаются рекурсивные вызовы, причем спецификации подоператоров обязательны. Третья группа правил для раскрытия кванторов существования и упрощения вхождений семантик составных операторов. Имеется группа правил для гиперфункций.

4. Правила общего вида без использования спецификаций подоператоров

В данном разделе определяются правила доказательства корректности основных операторов (параллельного, условного и суперпозиции) сведением к доказательству для подоператоров. В данной декомпозиции доказательства спецификации (предусловия и постусловия) подоператоров не используются.

Предположим, что наборы переменных x, y, z и v не пересекаются, а x и v могут быть пустыми. Пусть B и C – произвольные операторы, используемые в качестве подоператоров. Ниже приведены правила доказательства корректности основных операторов.

QP: $\dfrac{B(x\textbf{:}\ y)\ \textbf{corr}\ [P(x), Q(x, y)];\ C(x\textbf{:}\ z)\ \textbf{corr}\ [P(x), R(x, z)];}{\{B(x\textbf{:}\ y)\ ||\ C(x\textbf{:}\ z)\}\ \textbf{corr}\ \ [P(x), Q(x, y)\ \&\ R(x, z)]}$

В правиле **QP** постусловие параллельного оператора разделяется для результатов y и z. Предполагается, что разделение постусловия на две части может быть проведено автоматически при соответствующем написании спецификаций.

QC: $\dfrac{\begin{array}{l}B(x\textbf{:}\ y)\ \textbf{corr}\ [P(x)\ \&\ E(x),\ Q(x, y)];\\ C(x\textbf{:}\ z)\ \textbf{corr}\ [P(x)\ \&\ \neg E(x),\ Q(x, y)]\end{array}}{\{\textbf{if}\ (E(x))\ B(x\textbf{:}\ y)\ \textbf{else}\ C(x\textbf{:}\ y)\}\ \textbf{corr}\ [P(x),\ Q(x, y)]}$

QS: $\dfrac{\begin{array}{l}P(x) \Rightarrow \exists z,v.\ B(x\textbf{:}\ z,\ v);\\ \forall z.\ C(x, z\textbf{:}\ y)\ \textbf{corr}\ [P(x)\ \&\ B(x\textbf{:}\ z,\ v),\ Q(x, y,v)];\end{array}}{B(x\textbf{:}\ z,\ v);\ C(x, z\textbf{:}\ y)\ \textbf{corr}\ [P(x),\ Q(x,y,v)]}$

Приведенное правило **QS** для суперпозиции наиболее общего вида [16, разд. 2.3]. Набор v может быть пустым. С учетом этого будем также использовать более простую версию данного правила.

$$\textbf{QS}:\ \frac{P(x) \Rightarrow \exists z.\ B(x\textbf{:}\ z);\quad \forall z.\ C(x, z\textbf{:}\ y)\ \textbf{corr}\ [P(x)\ \&\ B(x\textbf{:}\ z),\ Q(x, y)];}{B(x\textbf{:}\ z);\ C(x, z\textbf{:}\ y)\ \textbf{corr}\ [P(x),\ Q(x,y)]}$$

Правило **QS** определяет метод доказательства для оператора суперпозиции: достаточно доказать тотальность первого подоператора $B(x\textbf{:}\ z, v)$, после чего данный оператор подсоединяется к предусловию для второго подоператора $C(x, z\textbf{:}\ y)$.

В случае, когда программа B не является однозначной, необходимо дополнительное условие корректности, гарантирующее тотальность программы C для всех значений результатов z программы B:

$$\forall z.\ (B(x\textbf{:}\ z, v) \Rightarrow \exists y.\ C(x, z\textbf{:}\ y);$$

Обоснование см. в [4], разд. 3.7.

5. Правила для рекурсивных вызовов при наличии спецификации подоператоров

Доказательство формулы корректности $B(x\textbf{:}\ y)\ \textbf{corr}\ [P(x), Q(x, y)]$ для рекурсивной программы B проводится индукцией по мере m, заданной в заголовке программы B. Используется следующее индукционное предположение:

$$\mathrm{Induct}(B, P, Q)(x) \cong\ \forall u\ m(u) < m(x) \Rightarrow B(u\textbf{:}\ y)\ \textbf{corr}\ [P(u), Q(u, y)]$$

Доказательство формулы корректности реализуется правилом:

$$\mathrm{Induct}(B, P, Q)(x) \vdash\ B(x\textbf{:}\ y)\ \textbf{corr}\ [P(x), Q(x, y)]$$

Данное правило будет необходимо при наличии взаимной рекурсии, когда рекурсивное кольцо [4] состоит более чем из одной программы. В случае одиночной рекурсии достаточно правил приведенных ниже. Эти правила должны применяться в случае, когда один из подоператоров является рекурсивным вызовом. Однако и здесь можно будет использовать соответствующее правило предыдущего раздела, а затем для рекурсивного вызова использовать правило **RB**.

В дальнейших правилах, если подоператор $B(u\textbf{:}\ y)$ является рекурсивным вызовом в теле программы B, то посылка вида $B(x\textbf{:}\ y)\ \textbf{corr*}$ опускается, а $P^*_B(u)$ заменяется на $P_B(u)\ \&\ m(u) < m(x)$, где x обозначает аргументы предиката B. Если же подоператор $B(u\textbf{:}\ y)$ не является рекурсивным вызовом, то **corr*** и P* обозначают просто **corr** и P. Вхождения **corr*** и P* для подоператора C трактуются аналогично.

В правилах ниже подоператоры представлены своими спецификациями. Предположим, что наборы переменных x, y, z и v не пересекаются, а x и v могут быть пустыми.

RP:
$$\frac{\begin{array}{l} B(x\mathbf{:}\ y)\ \mathbf{corr*}\ [P_B(x),\ Q_B(x,\ y)];\\ C(x\mathbf{:}\ z)\ \mathbf{corr*}\ [P_C(x),\ Q_C(x,\ z)];\\ \forall y,\ z\ (P(x)\ \&\ Q_B(x,\ y)\ \&\ Q_C(x,\ z) \Rightarrow Q(x,\ y,\ z));\\ P(x) \Rightarrow P^*_B(x)\ \&\ P^*_C(x); \end{array}}{\{B(x\mathbf{:}\ y)\ ||\ C(x\mathbf{:}\ z)\}\ \mathbf{corr}\ [P(x),\ Q(x,\ y,\ z)]}$$

RS:
$$\frac{\begin{array}{l} B(x:\ z,\ v)\ \mathbf{corr}^*\ [P_B(x),\ Q_B(x,\ z,\ v)];\\ \forall z.\ C(x,\ z:\ y)\ \mathbf{corr}^*\ [P_C(x,\ z),\ Q_C(x,\ z,\ y)];\\ \forall z,\ v,\ y.\ (\ P(x)\ \&\ Q_B(x,\ z,\ v)\ \&\ Q_C(x,\ z,\ y) \Rightarrow\ Q(x,\ y,\ v)\ )\\ \forall z,\ v.\ (\ P(x)\ \&\ Q_B(x,\ z,\ v) \Rightarrow\ P^*_C(x,\ z)\ );\\ P(x) \Rightarrow P^*_B(x); \end{array}}{B(x\mathbf{:}\ z,\ v);\ C(x,\ z\mathbf{:}\ y)\ \mathbf{corr}\ [P(x),\ Q(x,\ y,\ v)]}$$

Далее следует упрощенная версия правила **RS** для случая, когда набор v пустой.

RS:
$$\frac{\begin{array}{l} B(x:\ z)\ \mathbf{corr}^*\ [P_B(x),\ Q_B(x,\ z)];\\ \forall z.\ C(x,\ z:\ y)\ \mathbf{corr}^*\ [P_C(x,\ z),\ Q_C(x,\ z,\ y)];\\ \forall z,\ y.\ (\ P(x)\ \&\ Q_B(x,\ z)\ \&\ Q_C(x,\ z,\ y) \Rightarrow\ Q(x,\ y)\ )\\ \forall z.\ (\ P(x)\ \&\ Q_B(x,\ z) \Rightarrow\ P^*_C(x,\ z)\ );\\ P(x) \Rightarrow P^*_B(x); \end{array}}{B(x\mathbf{:}\ z);\ C(x,\ z\mathbf{:}\ y)\ \mathbf{corr}\ [P(x),\ Q(x,\ y)]}$$

RC:
$$\frac{\begin{array}{l} B(x\mathbf{:}\ y)\ \mathbf{corr*}\ [P_B(x),\ Q_B(x,\ y)];\\ C(x\mathbf{:}\ y)\ \mathbf{corr*}\ [P_C(x),\ Q_C(x,\ y)];\\ P(x)\ \&\ E(x) \Rightarrow P_B(x);\ \ P(x)\ \&\ \neg\ E(x) \Rightarrow P_C(x);\\ \forall y\ (P(x)\ \&\ E(x)\ \&\ Q_B(x,\ y) \Rightarrow Q(x,\ y));\\ \forall y\ (P(x)\ \&\ \neg E(x)\ \&\ Q_C(x,\ y) \Rightarrow Q(x,\ y)) \end{array}}{\mathbf{if}\ (E(x))\ B(x:\ y)\ \mathbf{else}\ C(x:\ y)\ \mathbf{corr}\ [P(x),\ Q(x,\ y,\ z)]}$$

Правило **QSB** является вариацией приведенного выше правила **QS** для случая, когда подоператор B, представлен своей спецификацией. Причем B может быть рекурсивным.

QSB:
$$\frac{\begin{array}{l} B(x:\ z)\ \mathbf{corr}^*\ [P_B(x),\ Q_B(x,\ z)];\ P(x) \Rightarrow P^*_B(x);\\ \forall z.\ C(x,\ z:\ y)\ \mathbf{corr}\ [P(x)\ \&\ Q_B(x,\ z),\ Q(x,\ y)]; \end{array}}{B(x:\ z);\ C(x,\ z:\ y)\ \mathbf{corr}\ [P(x),\ Q(x,y)]}$$

Следующее правило для выражений в качестве параметров в вызовах программ. Запись вида z = B(x) является эквивалентом B(x**:** z). Оператор суперпозиции B(x: z); C(x, z: y) может быть записан в виде: C(x, B(x)**:** y).

$$\textbf{RB:}\ \frac{\begin{array}{l}\forall z\ C(x, z\textbf{:}\ y)\ \textbf{corr*}\ [P_C(x, z), Q_C(x, y)];\\ SV(P_B, B)(x);\\ P(x) \Rightarrow P_B(x)\ \&\ P^*_C(x, B(x));\\ \forall y\ (\ P(x)\ \&\ Q_C(B(x), y) \Rightarrow Q(x, y)\ );\end{array}}{C(x, B(x)\textbf{:}\ y)\ \textbf{corr}\ [P(x), Q(x, y)]}$$

Предикат $SV(P_B, B)$ определяет однозначность программы B:

$$SV(P, B)(x) \cong \forall y_1, y_2.\ P(x)\ \&\ B(x\textbf{:}\ y_1)\ \&\ B(x\textbf{:}\ y_2) \Rightarrow y_1 = y_2\ .$$

Доказано [4], что предикатная программа является однозначной, поскольку все базисные предикаты (элементарные операции) являются однозначными. Поэтому посылка $SV(P_B, B)(x)$ является избыточной. Данная посылка должна проверяться в случае внесения в программу неоднозначных операций, например, датчика случайных чисел.

Истинность всех посылок правила **RB** гарантирует корректность следующей программы:

$$H(x\textbf{:}\ y)\ \textbf{pre}\ P(x)\ \textbf{post}\ Q(x, y)\ \{\ C(x, B(x)\textbf{:}\ y)\ \} \qquad (2)$$

Далее следует правило для частного случая оператора суперпозиции, соответствующего *сведению к более общей задаче* $C(x, z\textbf{:}\ y)$.

$$\textbf{RBE:}\ \frac{\begin{array}{l}\forall z\ C(x, z\textbf{:}\ y)\ \textbf{corr*}\ [P_C(x, z), Q(x, y)];\\ SV(P_B, B)(x);\\ P(x) \Rightarrow\ \exists z.\ B(x\textbf{:}\ z)\ \&\ P_C{}^*(x, B(x));\end{array}}{C(x, B(x)\textbf{:}\ y)\ \textbf{corr}\ [P(x), Q(x, y)]}$$

Истинность посылок правила **RBE** гарантирует корректность программы (2).

Отметим, что всюду определенное простое выражение $D(x)$ может быть подставлено вместо аргументов x в любое из приведенных выше правил. Не обязательно всегда использовать правило **RB**.

6. Правила для раскрытия кванторов существования и упрощения семантики составных операторов

Последовательное применение вышеописанных правил к исходной формуле тотальной корректности декомпозирует ее на множество формул вида $W(x, y) \Rightarrow R(S)(x, y)$, где $W(x, y)$ — произвольная посылка. Ниже даны правила доказательства формулы для различных видов операторов в позиции оператора $S(x\textbf{:}\ y)$:

$$\textbf{FP:}\ \frac{\begin{array}{l}W(x, y, z) \to R(B)(x, y);\\ W(x, y, z) \to R(C)(x, z)\end{array}}{W(x, y, z) \to R(B \parallel C)(x, (y, z))}$$

$$\textbf{FS:}\ \frac{\begin{array}{l}W(x, y) \to \exists z\ R(B)(x, z);\\ W(x, y)\ \&\ R(B)(x, z) \to R(C)(z, y)\end{array}}{W(x, y) \to R(B; C)(x, y)}$$

$$\textbf{FC:}\ \frac{\begin{array}{l}W(x, y)\ \&\ E \to R(B)(x, y);\\ W(x, y)\ \&\ \neg E \to R(C)(x, y)\end{array}}{W(x, y) \to R(\textbf{if}\ (E)\ B\ \textbf{else}\ C)(x, y)}$$

Ниже приведены правила для декомпозиции вхождений R(S)(x, y) в этих новых видах формул.

$$\textbf{EP:}\ \frac{\begin{array}{l}W(x) \to \exists y\ R(B)(x, y);\\ W(x) \to \exists z\ R(C)(x, z)\end{array}}{W(x) \to \exists y\ R(B \parallel C)(x, (y, z))}$$

$$\textbf{ES:}\ \frac{\begin{array}{l}W(x) \to \exists z\ R(B)(x, z);\\ W(x)\ \&\ R(B)(x, z) \to \exists y\ R(C)(z, y)\end{array}}{W(x) \to \exists y\ R(B; C)(x, y)}$$

$$\textbf{EC:}\ \frac{\begin{array}{l}W(x)\ \&\ E \to \exists y\ R(B)(x, y);\\ W(x)\ \&\ \neg E \to \exists y\ R(C)(x, y)\end{array}}{W(x) \to \exists y\ R(\textbf{if}\ (E)\ B\ \textbf{else}\ C)(x, y)}$$

Пусть A(x: y) — нерекурсивный вызов предиката, а P(x) — предусловие этого предиката. Вхождение логики A(x: y) под квантором существования декомпозируется следующим правилом:

$$\textbf{EB:}\ \frac{Corr(A, P, Q)(x);\quad \forall y\ (W(x) \to P(x))}{W(x) \to \exists y\ R(A)(x, y)}$$

Приведенная выше система правил позволяет элиминировать вхождение квантора существования. Вхождения логики оператора в левой части формулы, декомпозируются по следующим правилам:

$$\textbf{FLP:}\ \frac{W(x, y, z)\ \&\ R(B)(x, y)\ \&\ R(C)(x, z) \to H(x, y, z)}{W(x, y, z)\ \&\ R(B \parallel C)(x, (y, z)) \to H(x, y, z)}$$

$$\textbf{FLS:}\ \frac{W(x, y)\ \&\ R(B)(x, z)\ \&\ R(C)(z, y) \to H(x, y)}{W(x, y)\ \&\ R(B; C)(x, y) \to H(x, y)}$$

$$\textbf{FLC:}\ \frac{\begin{array}{l}W(x, y)\ \&\ E\ \&\ R(B)(x, y) \to H(x, y);\\ W(x, y)\ \&\ \neg E\ \&\ R(B(C))(x, y) \to H(x, y);\end{array}}{W(x, y)\ \&\ R(\textbf{if}\ (E)\ B\ \textbf{else}\ C)(x, y) \to H(x, y)}$$

$$\textbf{FLB:}\ \frac{P_A(x);\quad W(x,y)\ \&\ Q_A(x,y)\to H(x,y);}{W(x,y)\ \&\ R(A)(x,y)\to H(x,y)}$$

7. Правила для гиперфункций

Формально гиперфункция определяется через предикатную программу следующего вида:

B(x**:** y, z, e)
pre P(x) **post** e = E(x) & (E(x) ⇒ S(x, y)) & (¬E(x) ⇒ R(x, z))
{ ... };

Здесь x, y и z – непересекающиеся возможно пустые наборы переменных; P(x), E(x), S(x, y) и R(x, z) – логические утверждения. Предположим, что все присваивания вида e = **true** и e = **false** – последние исполняемые операторы в теле предиката. Программа B может быть заменена следующей программой в виде *гиперфункции*:

B(x**:** y #1**:** z #2)
pre P(x) **pre** 1**:** E(x) **post** 1: S(x, y) **post** 2: R(x, z)
{ ... };

В теле гиперфункции каждое присваивание e = **true** заменено оператором перехода #1, а e = **false** – на #2. *Метки* 1 и 2 – дополнительные параметры, определяющие два различных *выхода* гиперфункции.

Вызов гиперфункции комбинируется с обработчиками ветвей (**case** -частей) в виде конструкции:

B(x**:** y #M1**:** z #M2) **case** M1: C(y**:** u) **case** M2: D(z**:** u) (2)

Любая композиция с вызовом гиперфункции приводится к такому виду. Приведенная комбинация (2) раскрывается в виде композиции:

B(x**:** y, z, v, e); **if** (e) C(y**:** u) **else** D(z**:** u) (3)

Здесь через v обозначены возможные дополнительные результаты гиперфункции B (кроме y и z), которые не встречаются в аргументах C и D. По каждой из ветвей возможен аналог суперпозиции наиболее общего вида B(x**:** y, v); C(x,y**:** u).

Построим правило для комбинации (2) по ее раскрытию (3). Применим **QSB**

$$\textbf{QSB:}\ \frac{\begin{array}{l}B(x{:}\ z)\ \textbf{corr}^*\ [P_B(x), Q_B(x,z)];\ P(x)\Rightarrow P^*_B(x);\\ \forall z.\ C(x,z{:}\ y)\ \textbf{corr}\ [P(x)\ \&\ Q_B(x,z), Q(x,y)];\end{array}}{B(x{:}\ z);\ C(x,z{:}\ y)\ \textbf{corr}\ [P(x), Q(x,y)]}$$

Здесь:

Q(x, y, z, v, e) = e = E(x) & (E(x) ⇒ S(x, y)) & (¬E(x) ⇒ R(x, z))

Конкретизация правила:

$$\textbf{QSB:}\ \frac{\begin{array}{l}B(x{:}\ y,z,v,e)\ \textbf{corr}^*\ [P(x), Q(x,y,z,v,e)];\ P1(x)\Rightarrow P^*(x);\\ \forall\ y,z,v,e.\ \textbf{if}\ (e)\ C(y{:}\ u)\ \textbf{else}\ D(z{:}\ u)\ \textbf{corr}\ [P1(x)\ \&\ Q(x,y,z,v,e), Q1(x,v,u)];\end{array}}{B(x{:}\ y,z,v,e);\ \textbf{if}\ (e)\ C(y{:}\ u)\ \textbf{else}\ D(z{:}\ u)\ \ \textbf{corr}\ [P1(x), Q1(x,v,u)]}$$

Применим **QC** для второй посылки.

$$\textbf{QC}:\ \frac{\begin{array}{l}B(x\textbf{:}\ y)\ \textbf{corr}\ [P(x)\ \&\ E(x),\ Q(x,\ y)];\\ C(x\textbf{:}\ z)\ \textbf{corr}\ [P(x)\ \&\ \neg E(x),\ Q(x,\ y)]\end{array}}{\{\textbf{if}\ (E(x))\ B(x\textbf{:}\ y)\ \textbf{else}\ C(x\textbf{:}\ y)\}\ \textbf{corr}\ [P(x),\ Q(x,\ y)]}$$

Конкретизация правила:

$$\textbf{QC}:\ \frac{\begin{array}{l}C(y\textbf{:}\ u)\ \textbf{corr}\ [P1(x)\ \&\ Q(x,\ y,\ z,\ v,\ e)\ \&\ e,\ Q1(x,v,u)];\\ D(z\textbf{:}\ u)\ \textbf{corr}\ [P1(x)\ \&\ Q(x,\ y,\ z,\ v,\ e)\ \&\ \neg e,\ Q1(x,v,u)]\end{array}}{\textbf{if}\ (e)\ C(y\textbf{:}\ u)\ \textbf{else}\ D(z\textbf{:}\ u)\ \textbf{corr}\ [P1(x)\ \&\ Q(x,\ y,\ z,\ v,\ e),\ Q1(x,v,u)]}$$

Проведем упрощения.

$$\textbf{QC}:\ \frac{\begin{array}{l}C(y\textbf{:}\ u)\ \textbf{corr}\ [P1(x)\ \&\ S(x,\ y)\ \&\ E(x),\ Q1(x,v,u)];\\ D(z\textbf{:}\ u)\ \textbf{corr}\ [P1(x)\ \&\ R(x,\ z)\ \&\ \neg E(x),\ Q1(x,v,u)]\end{array}}{\textbf{if}\ (e)\ C(y\textbf{:}\ u)\ \textbf{else}\ D(z\textbf{:}\ u)\ \textbf{corr}\ [P1(x)\ \&\ Q(x,\ y,\ z,\ v,\ e),\ Q1(x,v,u)]}$$

Получаем правило для конструкции (2).

$$\textbf{HSB}:\ \frac{\begin{array}{l}B(x\textbf{:}\ y,\ z,\ v,\ e)\ \textbf{corr}^{*}\ [P(x),\ Q(x,\ y,\ z,\ v,\ e)];\quad P1(x) \Rightarrow P^{*}(x);\\ C(y\textbf{:}\ u)\ \textbf{corr}\ [P1(x)\ \&\ S(x,\ y)\ \&\ E(x),\ Q1(x,v,u)];\\ D(z\textbf{:}\ u)\ \textbf{corr}\ [P1(x)\ \&\ R(x,\ z)\ \&\ \neg E(x),\ Q1(x,v,u)]\end{array}}{B(x\textbf{:}\ y\ \#M1\textbf{:}\ z\ \#M2)\ \textbf{case}\ M1:\ C(y\textbf{:}\ u)\ \textbf{case}\ M2:\ D(z\textbf{:}\ u)\ \ \textbf{corr}\ [P1(x),\ Q1(x,v,u)]}$$

Данное правило неприменимо для рекурсивного вызова гиперфункций. Рекурсивный вызов гиперфункции заменяется его раскрытием $B(x\textbf{:}\ y,\ z,\ v,\ e)$ и обрабатывается как обычный вызов.

Литература

1. Floyd R. W. Assigning meanings to programs // Proceedings Symposium in Applied Mathematics, Mathematical Aspects of Computer Science. AMS, 1967. P. 19–32.
2. Hoare C. A. R. An axiomatic basis for computer programming // Communications of the ACM. 1969. Vol. 12 (10). P. 576–585.
3. Карнаухов Н.С., Першин Д.Ю., Шелехов В.И. Язык предикатного программирования Р. Версия 0.14. Новосибирск, 2018. 45с. http://persons.iis.nsk.su/files/persons/pages/plang14.pdf
4. Шелехов В.И. Семантика языка предикатного программирования // ЗОНТ-15. Новосибирск, 2015. 13с. http://persons.iis.nsk.su/files/persons/pages/semZont1.pdf
5. PVS Specification and Verification System / SRI International. http://pvs.csl.sri.com/
6. Шелехов В.И. Классификация программ, ориентированная на технологию программирования // «Программная инженерия», Том 7, № 12, 2016. С. 531–538. http://persons.iis.nsk.su/files/persons/pages/prog.pdf
7. Доказательство правил корректности операторов предикатной программы. 2013. http://www.iis.nsk.su/persons/vshel/files/rules.zip
8. Чушкин М.С. Система дедуктивной верификации предикатных программ // «Программная инженерия». 2016. № 5. С. 202-210. http://persons.iis.nsk.su/files/persons/pages/paper.pdf
9. Шелехов В.И. Методы доказательства корректности программ с хорошей логикой // Межд. конф. "Современные проблемы математики, информатики и биоинформатики", посвященная 100-летию со дня рождения А.А. Ляпунова. — 2011. — 17с., http://conf.nsc.ru/files/conferences/Lyap-100/fulltext/74974/75473/Shelekhov_prlogic.pdf

10. Шелехов В.И. Разработка и верификация алгоритмов пирамидальной сортировки в технологии предикатного программирования. Новосибирск, 2012. 30с. (Препр. / ИСИ СО РАН. № 164).
11. Шелехов В.И. Разработка программы построения дерева суффиксов в технологии предикатного программирования. — Новосибирск, 2004. — 52с. — (Препр. / ИСИ СО РАН; N 115).
12. Шелехов В.И. Верификация и синтез эффективных программ стандартных функций в технологии предикатного программирования // Программная инженерия, 2011, № 2. С. 14-21.
13. Шелехов В.И. Дедуктивная верификация и реализация предикатной программы инвертирования односвязных списков. ИСИ СО РАН, Новосибирск, 2018. 13с. http://persons.iis.nsk.su/files/persons/pages/listinvert.pdf.
14. Шелехов В.И. Разработка и верификация алгоритмов пирамидальной сортировки в технологии предикатного программирования. Новосибирск, 2012. 30с. (Препр. / ИСИ СО РАН. № 164).
15. Шелехов В.И. Верификация и синтез программ сложения на базе правил корректности операторов // Computer Science in Russia CSR-2010. Workshop on Program Semantics and Verification: Theory and Applications. Казань, 2010. С. 150-156.
16. Шелехов В.И. Доказательное построение, верификация и синтез предикатных программ // Знания-Онтологии-Теории (ЗОНТ-2017), Том 2. Институт Математики СО РАН, Новосибирск, 2017. С. 156-165. http://persons.iis.nsk.su/files/persons/pages/lbase.pdf.
17. Методы предикатного программирования / Под ред. Шелехова В.И. Вып.1. ИСИ СО РАН. Новосибирск, 2003. 62 С.
18. Методы предикатного программирования / Под ред. Шелехова В.И. Вып.2. ИСИ СО РАН. Новосибирск, 2006. 116 С.

6. Система правил доказательства корректности операторов

Используя формулу (11) или (15) можно автоматически построить формулу корректности для программы S(x: y) при условии, что для языка программирования построена логика программы. Итоговая формула корректности будет длинной и сложной даже для коротких программ; она будет длиннее программы S(x: y). Специализация формул (11) и (15) для разных видов операторов позволяет декомпозировать длинную формулу корректности к нескольким более коротким и простым формулам.

В данном разделе представлена специализация формул (11) и (15) в виде правил доказательства корректности для оператора суперпозиции B(x: z); C(z: y), параллельного оператора B(x: y) || C(x: z) и условного оператора **if** (E) B(x: y) **else** C(x: y).

6.1. Правила для общего случая

Рассмотрим случай, когда для операторов B и C имеются спецификации и эти операторы корректны по отношению к своим спецификациям. Ниже представлены правила на базе формулы (11). Они позволяют свести доказательство корректности оператора к доказательству корректности составляющих операторов B и C.

6.1.1 Правило корректности для параллельного оператора

Допустим, операторы $B(x\colon y)$ и $C(x\colon z)$ корректны относительно своих спецификаций $[P_B(x), Q_B(x, y)]$ и $[P_C(x), Q_C(x, z)]$. Параллельный оператор $B(x\colon y) \,||\, C(x\colon z)$ имеет спецификацию $[P(x), Q(x, y, z)]$. Представим правило для доказательства корректности параллельного оператора.

$$RP\colon \frac{\mathrm{Corr}(B(x\colon y), P_B(x), Q_B(x, y));\ \mathrm{Corr}(C(x\colon z), P_C(x), Q_C(x, z));\quad P(x) \vdash P_B(x) \,\&\, P_C(x);\ \ Q_B(x, y) \,\&\, Q_C(x, z) \vdash Q(x, y, z)}{\mathrm{Corr}(B(x\colon y) \,||\, C(x\colon z), P(x), Q(x, y, z))}$$

Доказательство истинности правила *RP*. Необходимо доказать формулу корректности (11) для параллельного оператора. Пусть предусловие $P(x)$ истинно. Тогда в соответствии с формулой (11) следует доказать тотальность $L(B(x\colon y) \,||\, C(x\colon z))$ и выводимость постусловия $Q(x, y, z)$ из $L(B(x\colon y) \,||\, C(x\colon z))$. В соответствии с определением (7) формула $L(B(x\colon y) \,||\, C(x\colon z))$ эквивалентна $L(B(x\colon y)) \,\&\, L(C(x\colon z))$.

Из истинности предусловия $P(x)$ и третьей посылки правила *RP* следует истинность $P_B(x)$ и $P_C(x)$. Далее, из корректности операторов $B(x\colon y)$ и $C(x\colon z)$ следует истинность формул $\exists y.\, L(B(x\colon y))$ и $\exists z.\, L(C(x\colon z))$. Их конъюнкция определяет тотальность $L(B(x\colon y) \,||\, C(x\colon z))$.

Докажем выводимость постусловия $Q(x, y, z)$ из $L(B(x\colon y)) \,\&\, L(C(x\colon z))$. Допустим, истинна формула $L(B(x\colon y)) \,\&\, L(C(x\colon z))$. Из истинности $P_B(x)$ и $P_C(x)$ и корректности операторов $B(x\colon y)$ и $C(x\colon z)$ следует истинность формул $L(B(x\colon y)) \Rightarrow Q_B(x, y)$ и $L(C(x\colon z)) \Rightarrow Q_C(x, z)$. Как следствие, будут истинны $Q_B(x, y)$ и $Q_C(x, z)$. Наконец, из последней посылки правила *RP* следует истинность постусловия $Q(x, y, z)$. □

6.1.2. Правило корректности для оператора суперпозиции

Допустим, операторы $B(x\colon z)$ и $C(z\colon y)$ корректны относительно своих спецификаций $[P_B(x), Q_B(x, z)]$ и $[P_C(z), Q_C(z, y)]$. Оператор суперпозиции $B(x\colon z);\, C(z\colon y)$ имеет спецификацию $[P(x), Q(x, y)]$. Представим правило для доказательства корректности оператора суперпозиции.

$$RS\colon \frac{\mathrm{Corr}(B(x\colon z), P_B(x), Q_B(x, z));\ m(x) < m(z) \Rightarrow \mathrm{Corr}(C(z\colon y), P_C(z), Q_C(z, y));\quad P(x) \vdash P_B(x) \,\&\, \forall z.\, Q_B(x, z) \Rightarrow P_C(z) \,\&\, m(z) < m(x);\quad P(x) \,\&\, \exists z.\, Q_B(x, z) \,\&\, Q_C(z, y) \vdash Q(x, y)}{\mathrm{Corr}(B(x\colon z);\, C(z\colon y), P(x), Q(x, y))}$$

Доказательство истинности правила *RS*. Необходимо доказать формулу корректности (11) для оператора суперпозиции. Пусть предусловие $P(x)$ истинно. Тогда в соответствии с формулой (11) следует доказать тотальность $L(B(x\colon z);\, C(z\colon y))$ и выводимость постусловия $Q(x, y)$ из $L(B(x\colon z);\, C(z\colon y))$. В соответствии с определением (11) формула $L(B(x\colon z);\, C(z\colon y))$ эквивалентна $\exists z.\, L(B(x\colon z)) \,\&\, L(C(z\colon y))$.

Из истинности предусловия $P(x)$ и третьей посылки правила *RS* следует истинность формул $P_B(x)$ и $\forall z\, (Q_B(x, z) \Rightarrow P_C(z))$. Из истинности $P_B(x)$ и корректности оператора $B(x\colon z)$ следует истинность формул $\exists z.\, L(B(x\colon z))$ и $L(B(x\colon z)) \Rightarrow Q_B(x, z)$. Допустим, для некоторого z_0 формула $L(B(x\colon z_0))$ истинна. Как следствие, истинно $Q_B(x, z_0)$. Далее, из

истинности $\forall z\,(Q_B(x, z) \Rightarrow P_C(z))$ следует истинность $P_C(z_0)$. Ввиду корректности оператора $C(z{:}\ y)$ истинна формула $\exists y\ L(C(z_0{:}\ y))$. Далее, истинна конъюнкция $L(B(x{:}\ z_0))\ \&\ \exists y\ L(C(z_0{:}\ y))$, и затем – формула $\exists y.\ \exists z.\ L(B(x{:}\ z))\ \&\ L(C(z{:}\ y))$, т. е. доказана тотальность $L(B(x{:}\ z);\ C(z{:}\ y))$.

Докажем выводимость постусловия $Q(x, y)$ из $L(B(x{:}\ z);\ C(z{:}\ y))$. Пусть $L(B(x{:}\ z);\ C(z{:}\ y))$ истинно, т. е. истинна формула $\exists z.\ L(B(x{:}\ z))\ \&\ L(C(z{:}\ y))$. Пусть формула истинна для некоторого z_1. Ввиду корректности оператора $B(x{:}\ z)$ истинна формула $L(B(x{:}\ z_1)) \Rightarrow Q_B(x, z_1)$ и далее – $Q_B(x, z_1)$. Истинность $Q_B(x, z_1)$ и $\forall z\,(Q_B(x, z) \Rightarrow P_C(z))$ влечет истинность $P_C(z_1)$. Ввиду корректности оператора $C(z{:}\ y)$ истинно $L(C(z_1, y)) \Rightarrow Q_C(z_1, y)$. Поскольку $L(C(z_1, y))$ истинно, то истинно $Q_C(z_1, y)$. В итоге, истинна правая часть последней посылки правила *RS*, а значит – и левая, т. е. истинно постусловие $Q(x, y)$. □

6.1.3. Правило корректности для условного оператора

Допустим, операторы $B(x{:}\ y)$ и $C(x{:}\ y)$ корректны относительно своих спецификаций $[P_B(x), Q_B(x, y)]$ и $[P_C(x), Q_C(x, y)]$. Условный оператор **if (**E) $B(x{:}\ y)$ **else** $C(x{:}\ y)$ имеет спецификацию $[P(x), Q(x, y)]$. Представим правило для доказательства корректности условного оператора.

$$RC:\ \frac{\begin{array}{l}\mathrm{Corr}(B(x{:}\ y), P_B(x), Q_B(x, y));\ \ \mathrm{Corr}(C(x{:}\ y), P_C(x), Q_C(x, y));\\ P(x)\ \&\ E \vdash P_B(x);\ \ P(x)\ \&\ \neg E \vdash P_C(x);\\ P(x)\ \&\ E\ \&\ Q_B(x, y) \vdash Q(x, y);\ \ P(x)\ \&\ \neg E\ \&\ Q_C(x, y) \vdash Q(x, y)\end{array}}{\mathrm{Corr}(\textbf{if (}E) B(x{:}\ y) \textbf{ else } C(x{:}\ y), P(x), Q(x, y))}$$

Доказательство истинности правила *RC*. Необходимо доказать формулу корректности (11) для условного оператора. В нее дважды входит подформула $L(\textbf{if (}E) B(x{:}\ y) \textbf{ else } C(x{:}\ y))$, которая согласно определению (8) эквивалентна:

$$(E \Rightarrow L(B(x{:}\ y)))\ \&\ (\neg E \Rightarrow L(C(x{:}\ y)))\ . \qquad (16)$$

Пусть предусловие $P(x)$ истинно. В соответствии с формулой (16) следует доказать тотальность формулы (16) и выводимость из нее постусловия $Q(x, y)$.

Допустим, что условие E истинно. Из истинности предусловия $P(x)$ и третьей посылки правила *RC* следует истинность $P_B(x)$. Ввиду корректности оператора $B(x{:}\ y)$ истинна формула $\exists y.\ L(B(x{:}\ y))$. Далее будет истинной формула $\exists y.\ (E \Rightarrow L(B(x{:}\ y)))$. Из истинности E следует истинность формулы $\neg E \Rightarrow L(C(x{:}\ y))$ и, следовательно, формулы $\exists y.\ [(E \Rightarrow L(B(x{:}\ y)))\ \&\ (\neg E \Rightarrow L(C(x{:}\ y)))]$. Это доказывает тотальность формулы (16) в случае истинности E. Тотальность (16) в случае ложности E доказывается аналогичным образом.

Докажем выводимость постусловия $Q(x, y)$ из формулы (16). Допустим, истинна формула (16). Пусть E истинно. Тогда истинно $L(B(x{:}\ y))$. Из третьей посылки правила *RC* следует истинность $P_B(x)$. Ввиду корректности оператора $B(x{:}\ y)$ истинна формула $L(B(x{:}\ y)) \Rightarrow Q_B(x, y)$, а значит – и $Q_B(x, y)$. В итоге, истинна правая часть пятой посылки правила *RC*, и, следовательно, истинна левая часть посылки, т. е. истинно постусловие $Q(x, y)$. Доказательство истинности постусловия $Q(x, y)$ для случая, когда E ложно, проводится аналогично с использованием четвертой и шестой посылок правила *RC*. □

6.2. Правила для однозначной спецификации

Допустим, подоператоры B и C имеют спецификации и эти операторы корректны по отношению к своим спецификациям. Предлагаемая ниже система правил доказательства корректности операторов базируется на Теореме 1, в соответствии с которой для тотальной спецификации и при условии однозначности используемых операторов требуется доказать истинность формулы (15). Правила применимы только для однозначной спецификации.

6.2.1 Правило корректности для параллельного оператора

Допустим, операторы B(x**:** y) и C(x**:** z) корректны относительно своих спецификаций $[P_B(x), Q_B(x, y)]$ и $[P_C(x), Q_C(x, z)]$. Параллельный оператор B(x**:** y) || C(x**:** z) имеет спецификацию [P(x), Q(x, y, z)]. Представим правило для доказательства корректности параллельного оператора.

$$LP:\ \frac{\begin{array}{l} T(P(x), Q(x, y, z));\ \ Corr(B(x: y), P_B(x), Q_B(x, y));\ \ SV(P_B(x), Q_B(x, y)); \\ Corr(C(x: z), P_C(x), Q_C(x, z));\ \ SV(P_C(x), Q_C(x, z)); \\ P(x) \vdash P_B(x) \,\&\, P_C(x);\ \ P(x) \,\&\, Q(x, y, z) \vdash Q_B(x, y) \,\&\, Q_C(x, z) \end{array}}{Corr(B(x: y) \,||\, C(x: z), P(x), Q(x, y, z))}$$

Доказательство истинности правила *LP*. Поскольку спецификация [P(x), Q(x, y, z)] тотальна, в соответствии с Теоремой 1 для доказательства правила достаточно доказать истинность формулы:

$$P(x) \,\&\, Q(x, y, z) \Rightarrow L(B(x{:}\ y) \,||\, C(x{:}\ z))\ .$$

В соответствии с определением (7) формула L(B(x**:** y) || C(x**:** z)) эквивалентна L(B(x**:** y) & L(C(x**:** z)).

Пусть истинны P(x) и Q(x, y, z). Докажем истинность L(B(x**:** y) и L(C(x**:** z)). Из истинности предусловия P(x) и посылки $P(x) \vdash P_B(x)$ & $P_C(x)$ следует истинность $P_B(x)$ и $P_C(x)$. Из последней посылки правила *LP* становятся истинными $Q_B(x, y)$ и $Q_C(x, z)$. Для предикатов B и C выполняются условия Леммы 4. Поэтому истинны формулы:

$$P_B(x) \,\&\, Q_B(x, y) \Rightarrow L(B(x{:}\ y);$$
$$P_C(x) \,\&\, Q_C(x, z) \Rightarrow L(C(x{:}\ z)).$$

Поскольку посылки этих формул истинны, то истинны L(B(x**:** y) и L(C(x**:** z)). □

6.2.2 Правило корректности для оператора суперпозиции

Допустим, операторы B(x**:** z) и C(z**:** y) корректны относительно своих спецификаций $[P_B(x), Q_B(x, z)]$ и $[P_C(z), Q_C(z, y)]$. Оператор суперпозиции B(x**:** z); C(z**:** y) имеет спецификацию [P(x), Q(x, y)]. Представим правило для доказательства корректности оператора суперпозиции.

$$LS:\ \frac{\begin{array}{l} T(P(x), Q(x, y));\ \ Corr(B(x: z), P_B(x), Q_B(x, z)); \\ Corr(C(z: y), P_C(z), Q_C(z, y));\ \ SV(P_C(x), Q_C(z, y)); \\ P(x) \vdash P_B(x);\ \ P(x) \,\&\, Q(x, y) \,\&\, Q_B(x, z) \vdash P_C(x) \,\&\, Q_C(z, y) \end{array}}{Corr(B(x: z);\ \ C(z: y), P(x), Q(x, y))}$$

Доказательство истинности правила *LS*. В соответствии с Теоремой 1 достаточно доказать истинность формулы:

$$P(x)\ \&\ Q(x, y) \Rightarrow L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))\ .$$

В соответствии с определением (11) формула $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$ эквивалентна $\exists z.(L(B(x\mathbf{:}\ z))\ \&\ L(C(z\mathbf{:}\ y))$.

Пусть истинны $P(x)$ и $Q(x, y)$. Докажем истинность $\exists z.(L(B(x\mathbf{:}\ z))\ \&\ L(C(z\mathbf{:}\ y))$. Из истинности предусловия $P(x)$ и посылки $P(x) \vdash P_B(x)$ следует истинность $P_B(x)$. Из корректности оператора B следует истинность формул $\exists z.\ L(B(x\mathbf{:}\ z))$ и $L(B(x\mathbf{:}\ z)) \Rightarrow Q_B(x, z)$. Допустим для некоторого z_0 истинно $L(B(x\mathbf{:}\ z_0))$. Тогда истинно $Q_B(x, z_0)$. В соответствии с последней посылкой правила *LS* истинна формула $P_C(z_0)\ \&\ Q_C(z_0, y)$. В соответствии с Леммой 4 истинна формула

$$P_C(z)\ \&\ Q_C(z, y) \Rightarrow L(C(z\mathbf{:}\ y))\ .$$

Тогда истинна $L(C(z_0\mathbf{:}\ y))$. В итоге, будет истинна формула $\exists z.(L(B(x\mathbf{:}\ z))\ \&\ L(C(z\mathbf{:}\ y))$. □

6.2.3. Правило корректности для условного оператора

Допустим, операторы $B(x\mathbf{:}\ y)$ и $C(x\mathbf{:}\ y)$ корректны относительно своих спецификаций $[P_B(x), Q_B(x, y)]$ и $[P_C(x), Q_C(x, y)]$. Условный оператор **if (**E) $B(x\mathbf{:}\ y)$ **else** $C(x\mathbf{:}\ y)$ имеет спецификацию $[P(x), Q(x, y)]$. Представим правило для доказательства корректности условного оператора.

$$LC\colon\ \frac{\begin{array}{l} T(P(x), Q(x, y));\ \ Corr(B(x\colon y), P_B(x), Q_B(x, y));\ \ SV(P_B(x), Q_B(x, y)); \\ Corr(C(x\colon y), P_C(x), Q_C(x, y));\ \ SV(P_C(x), Q_C(x, y)); \\ P(x)\ \&\ Q(x, y)\ \&\ E \vdash P_B(x)\ \&\ Q_B(x, y);\ \ P(x)\ \&\ Q(x, y)\ \&\ \neg E \vdash P_C(x)\ \&\ Q_C(x, y) \end{array}}{Corr(\mathbf{if\ (}E) B(x\mathbf{:}\ y)\ \mathbf{else}\ C(x\mathbf{:}\ y), P(x), Q(x, y))}$$

Доказательство истинности правила *LC*. В соответствии с Теоремой 1 достаточно доказать истинность формулы:

$$P(x)\ \&\ Q(x, y) \Rightarrow LS(\mathbf{if\ (}E) B\ \mathbf{else}\ C)(x, y)\ .$$

Согласно определению (8) формула $L(\mathbf{if\ (}E) B(x\mathbf{:}\ y)\ \mathbf{else}\ C(x\mathbf{:}\ y))$ эквивалентна:

$$(E \Rightarrow L(B(x\mathbf{:}\ y)))\ \&\ (\neg E \Rightarrow L(C(x\mathbf{:}\ y)))\ .$$

Пусть истинны $P(x)$ и $Q(x, y)$. Докажем истинность формулы $E \Rightarrow L(B(x\mathbf{:}\ y))$. Пусть истинно E. Докажем истинность $L(B(x\mathbf{:}\ y))$. Можно применить правило LC1, Поскольку истинны $P(x)$, $Q(x, y)$ и E в правой части предпоследней посылки правила *LC*, то истинна левая часть, т.е. формула $P_B(x)\ \&\ Q_B(x, y)$. В соответствии с Леммой 4 истинна формула

$$P_B(x)\ \&\ Q_B(x, y) \Rightarrow L(B(x\mathbf{:}\ y))\ .$$

Поскольку истинна посылка, то истинно $L(B(x\mathbf{:}\ y))$. Следовательно, доказана истинность формулы $E \Rightarrow L(B(x\mathbf{:}\ y))$. Истинность формулы $\neg E \Rightarrow L(C(x\mathbf{:}\ y))$ доказывается аналогично. □

В приведенных правилах посылка вида $SV(...)$ для операторов B и C может быть опущена, если корректность этих операторов доказывается по основе формулы (15). Это вытекает из Следствия Леммы 3.

6.3. Система правил декомпозиции доказательства корректности для произвольной спецификации

Спецификации подоператоров как правило отсутствуют. Ниже представлены правила на базе формулы (11). Они позволяют свести доказательство корректности оператора к доказательству корректности составляющих операторов B и C.

6.3.1. Правило корректности для параллельного оператора

Предположим, что для параллельного оператора L(A(x**:** y) || B(x**:** z)) постусловие представимо в виде конъюнкции предикатов Q(x, y) & R(x, z).

$$QP:\ \frac{\mathrm{Corr}(B(x{:}\ y),\ P(x),\ Q(x,\ y));\ \ \mathrm{Corr}(C(x{:}\ z),\ P(x),\ R(x,\ z))}{\mathrm{Corr}(B(x{:}\ y)\ ||\ C(x{:}\ z),\ P(x),\ Q(x,\ y)\ \&\ R(x,\ z))}$$

Доказательство истинности правила *QP*. Допустим, истинно предусловие P(x). Необходимо доказать тотальность формулы L(A(x**:** y) || B(x**:** z)) и выводимость из этой формулы постусловия Q(x, y) & R(x, z). В соответствии с определением (7) формула L(B(x**:** y) || C(x**:** z)) эквивалентна L(B(x**:** y) & L(C(x**:** z)). Из истинности P(x) и корректности операторов B(x**:** y) и C(x**:** z) следует истинность следующих формул: L(B(x**:** y)) ⇒ Q(x, y), ∃y. L(B(x**:** y)), L(C(x**:** z)) ⇒ R(x, z) и ∃z. L(C(x**:** z)). Конъюнкция ∃y. L(B(x**:** y)) и ∃z. L(C(x**:** z)) дает тотальность формулы L(A(x**:** y) || B(x**:** z)). Выводимость Q(x, y) & R(x, z) также очевидна. □

6.3.2. Правило корректности для условного оператора

$$QC:\ \frac{\mathrm{Corr}(B(x{:}\ y),\ P(x)\ \&\ E,\ Q(x,\ y));\ \ \mathrm{Corr}(C(x{:}\ y),\ P(x)\ \&\ \neg E,\ Q(x,\ y))}{\mathrm{Corr}(\textbf{if (}E)\ B(x{:}\ y)\ \textbf{else}\ C(x{:}\ y),\ P(x),\ Q(x,\ y))}$$

Доказательство истинности правила *QC*. Допустим, истинно предусловие P(x). Необходимо доказать тотальность формулы L(**if (**E) B(x**:** y) **else** C(x**:** y)) и выводимость из нее постусловия Q(x, y). В соответствии с определением (8) формула L(**if (**E) B(x**:** y) **else** C(x**:** y)) эквивалентна:

$$(E \Rightarrow L(B(x{:}\ y))) \ \&\ (\neg E \Rightarrow L(C(x{:}\ y)))\ .$$

Допустим, что условие E истинно. Из истинности P(x) & E и корректности оператора B(x**:** y) по первой посылке правила *QC* следует истинность формулы ∃y. L(B(x**:** y)). Далее будет истинной формула ∃y. (E ⇒ L(B(x**:** y))). Из истинности E следует истинность формулы ¬E ⇒ L(C(x**:** y)) и, следовательно, формулы

$$\exists y.\ [(E \Rightarrow L(B(x{:}\ y))) \ \&\ (\neg E \Rightarrow L(C(x{:}\ y)))]\ .$$

Это доказывает тотальность L(**if (**E) B(x**:** y) **else** C(x**:** y)) для истинного E. Тотальность в случае ложного E доказывается аналогичным образом.

Пусть формула L(**if (**E) B(x**:** y) **else** C(x**:** y)) истинна. Докажем истинность Q(x, y). Допустим, что условие E истинно. Из истинности P(x) & E и корректности оператора B(x**:** y) следует истинность формулы L(B(x, y)) ⇒ Q(x, y). Из истинности (E ⇒ L(B(x**:** y)) следует истинность L(B(x**:** y) и далее – Q(x, y). Истинность Q(x, y) при ложном E доказывается аналогично. □

6.3.3. Правило корректности для оператора суперпозиции

$$QS:\ \frac{P(x) \vdash \exists z.\ L(B(x: z));\ \ L(B(x: z)) \vdash \exists y.\ L(C(z: y));\quad \exists z.\ L(B(x: z)) \ \&\ L(C(z: y)) \vdash Q(x, y)}{Corr(B(x: z);\ \ C(z: y), P(x), Q(x, y))}$$

Доказательство истинности правила *QS*. Необходимо доказать формулу корректности (11) для оператора суперпозиции. Пусть предусловие $P(x)$ истинно. Тогда в соответствии с формулой (11) следует доказать тотальность $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$ и выводимость постусловия $Q(x, y)$ из $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$. В соответствии с определением (11) формула $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$ эквивалентна $\exists z.\ L(B(x\mathbf{:}\ z)) \ \&\ L(C(z\mathbf{:}\ y))$.

Из истинности предусловия $P(x)$ и первой посылки правила *QS* следует истинность формулы $\exists z.\ L(B(x\mathbf{:}\ z))$. Допустим, для некоторого z_0 формула $L(B(x\mathbf{:}\ z_0))$ истинна. Из второй посылки правила *QS* следует истинность формулы $\exists y.\ L(C(z_0\mathbf{:}\ y))$. Далее, истинна $L(B(x\mathbf{:}\ z_0)) \ \&\ \exists y\ L(C(z_0\mathbf{:}\ y))$, и затем – формула $\exists y.\ \exists z.\ L(B(x\mathbf{:}\ z)) \ \&\ L(C(z\mathbf{:}\ y))$, т. е. доказана тотальность $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$.

Докажем выводимость постусловия $Q(x, y)$ из $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$. Пусть $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$ истинно, т. е. истинна формула $\exists z.\ L(B(x\mathbf{:}\ z)) \ \&\ L(C(z\mathbf{:}\ y))$. Истинность $Q(x, y)$ следует из третьей посылки правила *QS*. □

6.3.4. Правило корректности для оператора суперпозиции с первым корректным оператором

$$QS1:\ \frac{Corr(B(x: z), P_B(x), Q_B(x, z));\quad P(x) \vdash P_B(x) \ \&\ \forall z.\ (Q_B(x, z) \Rightarrow \exists y.\ L(C(z: y)));\quad P(x) \ \&\ \exists z.\ Q_B(x, z) \ \&\ L(C(z: y)) \vdash Q(x, y)}{Corr(B(x: z);\ \ C(z: y), P(x), Q(x, y))}$$

Доказательство истинности правила *QS1*. Необходимо доказать формулу корректности (11) для оператора суперпозиции. Пусть предусловие $P(x)$ истинно. Тогда в соответствии с формулой (11) следует доказать тотальность $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$ и выводимость постусловия $Q(x, y)$ из $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$. В соответствии с определением (11) формула $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$ эквивалентна $\exists z.\ L(B(x\mathbf{:}\ z)) \ \&\ L(C(z\mathbf{:}\ y))$.

Из истинности предусловия $P(x)$ и второй посылки правила *QS1* следует истинность формул $P_B(x)$ и $\forall z\ (Q_B(x, z) \Rightarrow \exists y.\ L(C(z\mathbf{:}\ y)))$. Из истинности $P_B(x)$ и корректности оператора $B(x\mathbf{:}\ z)$ следует истинность формул $\exists z.\ L(B(x\mathbf{:}\ z))$ и $L(B(x\mathbf{:}\ z)) \Rightarrow Q_B(x, z)$. Допустим, для некоторого z_0 формула $L(B(x\mathbf{:}\ z_0))$ истинна. Как следствие, истинно $Q_B(x, z_0)$. Далее, из истинности $\forall z\ (Q_B(x, z) \Rightarrow \exists y.\ L(C(z\mathbf{:}\ y)))$ следует истинность $\exists y.\ L(C(z_0\mathbf{:}\ y)))$. Далее, истинна $L(B(x\mathbf{:}\ z_0)) \ \&\ \exists y\ L(C(z_0\mathbf{:}\ y))$, и затем – формула $\exists y.\ \exists z.\ L(B(x\mathbf{:}\ z)) \ \&\ L(C(z\mathbf{:}\ y))$, т. е. доказана тотальность $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$.

Докажем выводимость постусловия $Q(x, y)$ из $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$. Пусть $L(B(x\mathbf{:}\ z);\ C(z\mathbf{:}\ y))$ истинно, т. е. истинна формула $\exists z.\ L(B(x\mathbf{:}\ z)) \ \&\ L(C(z\mathbf{:}\ y))$. Из истинности $P_B(x)$ и корректности оператора $B(x\mathbf{:}\ z)$ следует истинность формулы и $L(B(x\mathbf{:}\ z)) \Rightarrow Q_B(x, z)$. Как следствие, истинна формула $\exists z.\ Q_B(x, z) \ \&\ L(C(z\mathbf{:}\ y))$. Из

истинности правой части третьей посылки правила *QS1* следует истинность левой части, т.е. Q(x, y). □

6.4. Система правил декомпозиции доказательства корректности для однозначной спецификации

Теорема 1 сводит доказательство корректности оператора к формуле (15): P(x) & Q(x, y) ⇒ L(S(x**:** y)). Декомпозиция доказательства формулы (15) реализуется для вхождения L(S(x**:** y)). Таким образом, имеется задача доказательства формулы вида R(x, y) ⇒ L(S(x**:** y)), где R(x, y) – произвольная посылка. Решением задачи являются правила доказательства формулы для различных видов операторов в позиции оператора S(x**:** y).

6.4.1. Правило корректности для параллельного оператора

$$FP:\ \frac{R(x, y, z) \vdash L(B(x: y));\ \ R(x, y, z) \vdash L(C(x: z))}{R(x, y, z) \vdash L(B(x: y) \parallel C(x: z))}$$

Доказательство истинности правила *FP*. В соответствии с определением (7) формула L(B(x**:** y) || C(x**:** z)) эквивалентна L(B(x**:** y)) & L(C(x**:** z)). Поэтому достаточно доказать истинность двух формул:

$$R(x, y, z) \Rightarrow L(B(x: y))$$
$$R(x, y, z) \Rightarrow L(C(x: z))$$

Эти формулы представлены посылками правила *FP*. □

6.4.2. Правило корректности для условного оператора

$$FC:\ \frac{R(x, y) \,\&\, E \vdash L(B(x: y));\ \ R(x, y) \,\&\, \neg E \vdash L(C(x: y))}{R(x, y) \vdash L(\textbf{if } (E)\ B(x: y)\ \textbf{else}\ C(x: y))}$$

Доказательство истинности правила *FC*. В соответствии с определением (8) формула L(**if (**E) B(x**:** y) **else** C(x**:** y)) эквивалентна

$$(E \Rightarrow L(B(x: y))) \,\&\, (\neg E \Rightarrow L(C(x: y)))$$

Таким образом, требуется доказать истинность:

$$R(x, y) \Rightarrow (E \Rightarrow L(B(x: y))) \,\&\, (\neg E \Rightarrow L(C(x: y)))$$

Последняя формула эквивалентна конъюнкции формул:

$$R(x, y) \Rightarrow (E \Rightarrow L(B(x: y)))$$
$$R(x, y) \Rightarrow (\neg E \Rightarrow L(C(x: y)))$$

А эти формулы представлены посылками правила *FC*. □

6.4.3. Правило корректности для оператора суперпозиции

$$FS:\ \frac{R(x, y) \vdash \exists z.\ L(B(x: z));\ \ R(x, y) \,\&\, L(B(x: z)) \vdash L(C(z: y))}{R(x, y) \vdash L(B(x: z);\ \ C(z: y))}$$

Доказательство истинности правила *FS*. В соответствии с определением (6) формула L(B(x**:** z); C(z**:** y)) эквивалентна ∃z. L(B(x**:** z)) & L(C(z**:** y)). Пусть истинно R(x, y). Докажем истинность ∃z. L(B(x**:** z)) & L(C(z**:** y)). Из первой посылки правила *FS*

истинна формула $\exists z.\ L(B(x{:}\ z))$. Допустим для некоторого z_0 истинно $L(B(x{:}\ z_0))$. Из второй посылки истинна формула $L(C(z_0{:}\ y))$. В итоге, будет истинна формула $\exists z.\ L(B(x{:}\ z)) \ \&\ L(C(z{:}\ y))$. □

6.4.4. Правило для нерекурсивного вызова предиката

$$FB:\ \frac{Corr(A(x{:}\ y), P(x), Q(x, y));\ SV(P(x), Q(x, y));\ R(x, y) \vdash P(x) \ \&\ Q(x, y)}{R(x, y) \vdash L(A(x{:}\ y))}$$

Доказательство истинности правила *FB*. Пусть истинно $R(x, y)$. Докажем истинность $L(A(x{:}\ y))$. Из третьей посылки правила *FB* истинна формула $P(x) \ \&\ Q(x, y)$. В соответствии с Леммой 4 истинна формула $P(x) \ \&\ Q(x, y) \Rightarrow L(A(x{:}\ y))$ и, следовательно, $L(A(x{:}\ y))$. □

Правило *FB* используется для доказательства формулы $R(x, y) \Rightarrow L(A(x{:}\ y))$ при условии, что оператор $A(x{:}\ y)$ корректен относительно спецификации $[P(x), Q(x, y)]$. Если оператор $A(x{:}\ y)$ является рекурсивным вызовом процедуры A, правило *FB* (точнее его модификация *FB3*) должно включать дополнительную посылку $m(x) < m(z)$, где z обозначает аргументы процедуры A, а m – функция меры, определенная на аргументах. Заметим, что посылка $SV(P(x), Q(x, y))$ может быть опущена, если корректность вызова доказывается на базе формулы (15).

6.4.5. Правила для оператора суперпозиции в позиции квантора существования и в левой части

Приведенные выше правила достаточно просты. Их можно применять многократно для декомпозиции вхождений $L(S(x{:}\ y))$ при доказательстве формул вида: $R(x, y) \Rightarrow L(S(x{:}\ y))$. Таких формул большинство среди посылок в приведенных выше правилах. Однако правило *FS* использует посылки двух других видов: $R(x, y) \Rightarrow \exists y.\ L(S(x{:}\ y))$ и $R(x, y) \ \&\ L(S(x{:}\ y)) \Rightarrow H(x, y)$, где $R(x, y)$ и $H(x, y)$ – произвольные предикаты. Необходимо разработать правила для декомпозиции вхождений $L(S(x{:}\ y))$ в этих новых видах формул. Ограничимся правилами для оператора суперпозиции. Правила для параллельного и условного операторов значительно проще.

$$FE:\ \frac{R(x) \vdash \exists z.\ L(B(x{:}\ z));\ R(x) \ \&\ L(B(x{:}\ z)) \vdash \exists y.\ L(C(z{:}\ y))}{R(x) \vdash \exists y.\ L(B(x{:}\ z);\ C(z{:}\ y))}$$

Доказательство истинности правила *FE*. В соответствии с определением (6) формула $\exists y.\ L(B(x{:}\ z);\ C(z{:}\ y))$ эквивалентна $\exists y.\ \exists z.(L(B(x{:}\ z)) \ \&\ L(C(z{:}\ y)))$. Пусть истинно $R(x)$. Из первой посылки правила *FE* истинна $\exists z.\ L(B(x{:}\ z))$. Допустим, для некоторого z_0 истинна $L(B(x{:}\ z_0))$. Из второй посылки правила *FE* истинна формула $\exists y.\ L(C(z_0{:}\ y))$. В итоге будет истинна $\exists y.\ \exists z.(L(B(x{:}\ z)) \ \&\ L(C(z{:}\ y)))$. □

$$FL:\ \frac{R(x) \ \&\ L(B(x{:}\ z)) \ \&\ L(C(z{:}\ y)) \vdash H(x, y)}{R(x) \ \&\ L(B(x{:}\ z);\ C(z{:}\ y)) \vdash H(x, y)}$$

Доказательство истинности правила *FL*. Пусть истинно $R(x) \ \&\ L(B(x{:}\ z);\ C(z{:}\ y))$. Докажем истинность $H(x, y)$. В соответствии с определением (6) формула $L(B(x{:}\ z);\ C(z{:}\ y))$ эквивалентна $\exists z.\ L(B(x{:}\ z)) \ \&\ L(C(z{:}\ y))$. Допустим, формула истинна

при некотором z_0. Тогда истинны L(B(x: z_0)) и L(C(z_0: y)). Применение посылки правила *FL* доказывает истинность H(x, y). □

7. Пример применения правил доказательства корректности

Дадим иллюстрацию применения некоторых из приведенных правил для доказательства корректности программы умножения mult1, представленной во Введении. Программа mult1(a, b, d: c) имеет спецификацию

$$[a \ge 0 \,\&\, b \ge 0 \,\&\, d \ge 0,\ c = a * b + d] .$$

Перепишем программу (5) в новых обозначениях.

```
proc mult1(nat a, b, d: nat c)                                    ( 17)
    { if (a = 0) c = d else mult1(a – 1, b, d + b: c) }
    measure a;
```

Для удобства генерации формул корректности программы введем обозначения для формул предусловия, постусловия и функции меры:

```
formula P_mult1(nat a, b, d) = a ≥ 0 & b ≥ 0 & d ≥ 0;
formula Q_mult1(nat a, b, d, c) = c = a * b + d;
function m(nat a: nat) = a;
```

Поскольку спецификация однозначна, к программе (17) применяется правило *T1* (Теорема 1). Первая посылка правила *T1* порождает лемму тотальности спецификации:

```
lemma     P_mult1(a, b, d) ⇒ ∃c. Q_mult1(a, b, d, c) .
```

На базе второй посылки правила *T1* генерируется цель для доказательства:

```
P_mult1(a, b, d) & Q_mult1(a, b, d, c) ⇒                          (18)
        L(if (a = 0) c = d else mult1(a – 1, b, d + b: c));
```

Формулу (18) следует декомпозировать для вхождения функции логики L для условного оператора. Применяется правило декомпозиции *FC*. Первая посылка правила порождает лемму:

```
lemma     P_mult1(a, b, d) & Q_mult1(a, b, d, c) & a = 0 ⇒ c = d .
```

На базе второй посылки правила *FC* генерируется следующая цель для доказательства:

```
P_mult1(a, b, d) & Q_mult1(a, b, d, c) & ¬ a = 0 ⇒ L(mult1(a – 1, b, d + b: c)) .  (19)
```

Для рекурсивного вызова процедуры mult1 в формуле (19) применяется правило *FB3* (модификация *FB*), включающее дополнительную посылку, гарантирующую завершение рекурсии. Это правило порождает лемму:

```
lemma     P_mult1(a, b, d) & Q_mult1(a, b, d, c) & ¬ a = 0 ⇒
        m(a - 1) < m(a) & P_mult1(a – 1, b, d + b) & Q_mult1(a – 1, b, d + b, c) .
```

Данные три леммы определяют набор формул для доказательства тотальной корректности программы (17).

Описанную верификацию программы (17) можно сравнить с классической верификацией по Хоару для программы, полученной из (17) заменой хвостовой рекурсии на цикл типа **while**:

```
c = d; while a != 0 do  a = a – 1; c = c + b end
```

Заключение

Для программы, построенной с использованием оператора суперпозиции, параллельного и условного операторов, имеющих простую логику, представлена система правил доказательства корректности программы. На базе данных трех видов операторов

разработан язык предикатного программирования P [3], находящийся на границе меду функциональными и логическими языками. Для полного языка P разработана формальная операционная и логическая семантика, доказана согласованность операционной и логической семантики [4].

Множество правил доказательства корректности программы делится на две части в зависимости от того, является ли спецификация однозначной или нет. Каждая из двух частей, в свою очередь, делится на две части в зависимости от того, имеется ли спецификация для подоператоров. Из этих четырех частей правила для однозначных спецификаций на базе формулы (15) при отсутствии спецификации для подоператоров оказались наиболее простыми и удобными для верификации. Эта часть правил разработана для полного языка P. На базе этой части правил для программы на языке P разработан детальный алгоритм генерации формул корректности на язык системы автоматического доказательства PVS [5].

Алгоритм генерации формул корректности опробован примерно для 20 небольших программ. Наиболее значимыми являются программа сумматора Линь (Ling adder) [6] и эффективные программы для стандартных функций floor, isqrt, и ilog2 [7]. Сгенерированные формулы корректности для всех программ были доказаны в системе PVS. Результаты данного эксперимента следующие. Задача автоматической генерации формул корректности решена успешно. Генерируемые формулы корректности декомпозированы хорошо: они достаточно короткие и вполне понятные. Доказательство на PVS оказалось нетривиальным и трудоемким процессом. Обнаружено 15 случайных ошибок (описок) в программе или спецификации. Лишь одна обнаруженная при верификации ошибка в программе функции floor оказалась нетривиальной и опасной.

Правила для доказательства корректности программы достаточно просты. Они годятся также для программного синтеза [6, 7]. Причем при разработке программы в стиле синтеза доказательство формул корректности проводится существенно легче.

Из-за ограничений на циклы типа **while** и указатели предлагаемый подход к дедуктивной верификации и программному синтезу напрямую применим лишь к чистым языкам функционального программирования. Тем не менее, подход может быть применен к частям императивных программ с простой операторной структурой, сходной с представленной в данной работе. Подобные смешанные модели программ становятся популярными в дедуктивной верификации. Например, дедуктивная верификация для частей программ без указателей переменных проводится более простым способом [8]. В другом подходе для упрощения верификации вместо произвольных указателей в программе допускаются лишь двухсвязные списки или структуры типа «лес деревьев» [9, 10].

$$RB: \quad \frac{\mathrm{Corr}(B(x{:}\ z), P_B(x), Q_B(x, z));\ \mathrm{Corr}(C(z{:}\ y), P_C(z), Q_C(z, y));\ \mathrm{SV}(P_B(x), Q_B(x, z));\ P(x) \vdash P_B(x) \,\&\, P_C(B(x));\ P(x) \,\&\, Q_C(B(x), y) \vdash Q(x, y)}{\mathrm{Corr}(C(B(x){:}\ y), P(x), Q(x, y))}$$

10. Правила для рекурсивного вызова предиката

Далее ограничимся случаем, когда рекурсивное кольцо состоит из единственного определения предиката:

$$A(x{:}\ y) \equiv P(x)\ \{\ K(x{:}\ y)\ \}\ Q(x, y) \qquad (5.20)$$

Здесь $K(x{:}\ y)$ — оператор, в котором имеются рекурсивные вызовы предиката A. Формула корректности определения (5.20) принимает вид:

$$\mathrm{Corr}(K(x{:}\ y), P(x), Q(x, y)) \equiv W(x) \equiv P(x) \Rightarrow (\exists y.\ LS(K(x{:}\ y))) \,\&\, (LS(K(x{:}\ y)) \Rightarrow Q(x, y))$$

Правило RR для кольца (5.20) принимают следующий вид:

Правило RR1*. $\mathrm{Induct}(x, W) \,\&\, P(x) \vdash (\exists y.\ LS(K(x{:}\ y))) \,\&\, (LS(K(x{:}\ y)) \Rightarrow Q(x, y))$

$\mathrm{Induct}(t, W) \cong \forall u.\ m(u) < m(t) \Rightarrow W(u).$

Лемма 13. Если правило RR1* истинно, то рекурсивное определение (5.20) корректно.

$$RS: \quad \frac{\mathrm{Corr}(B(x{:}\ z), P_B(x), Q_B(x, z));\ \mathrm{Corr}(C(z{:}\ y), P_C(z), Q_C(z, y));\ P(x) \vdash P_B(x) \,\&\, \forall z.\ Q_B(x, z) \Rightarrow P_C(z);\ P(x) \,\&\, \exists z.\ Q_B(x, z) \,\&\, Q_C(z, y) \vdash Q(x, y)}{\mathrm{Corr}(B(x{:}\ z);\ C(z{:}\ y), P(x), Q(x, y))}$$

Предикаты логики программы не могут быть слабее[1] соответствующих формул, извлекаемых из программы с помощью формальной семантики языка программирования в целях дедуктивной верификации программы. В общем случае неочевидно, что предикаты логики программы представимы, например, в языке исчисления предикатов.

Итак, введенное понятие логики программы отражает особенности построения и анализа программы программистом.

Простота логики позволяет сравнительно легко разработать методы дедуктивной верификации и синтеза программ.

Итак, истинность формулы (11) определяет корректность программы $S(x{:}\ y)$ относительно спецификации $[P(x), Q(x, y)]$. При наличии ошибки в программе формула (11) становится недоказуемой. Определим условия, при которых доказательство истинности формулы (11) дает абсолютную гарантию корректности программы.

Доказательство должно быть правильным. Ошибка в доказательстве может повлечь ложное доказательство недоказуемой формулы, появившейся из-за ошибки в программе. Как следствие, ошибка в программе обнаружена не будет. Традиционный метод

[1] Предикат P сильнее предиката Q, если истинно $P \Rightarrow Q$

математического доказательства[2] признается ненадежным. Поэтому доказательство формул корректности должно проводиться в рамках некоторой системы автоматического доказательства (например, HOL [ ] или PVS [ ]), позволяющей автоматически контролировать правильность доказательства. Тем не менее, постоянно дискутируется вопрос о степени надежности системы автоматического доказательства, поскольку ошибка в программе автоматического доказательства потенциально может стать причиной ложного доказательства недоказуемой формулы.

Использование формулы (11) предполагает доказательство согласованности логики программы с формальной операционной семантикой. Следующий шаг – доказательство правильности реализации языка программирования. Для этого необходима формализация задачи трансляции. Необходимо доказать правильность кодирования формальной операционной семантики в исходном языке и объектном языке, на который транслируется программа. Необходимо доказать, что транслятор реализует корректное отображение исходного кодирования операционной семантики в объектное кодирование, а применяемые оптимизирующие преобразования сохраняют функциональную эквивалентность программы. Наконец, требуется доказательство правильности реализации объектного языка, т.е. правильности кодирования его операционной семантики и правильности программы интерпретатора языка, реализованной на интегральных схемах. Разумеется, абсолютная гарантия корректности программы при доказательстве формулы (11) возможна лишь при проведении всех перечисленных доказательств в рамках некоторой системы автоматического доказательства. Пока этого не сделано, все еще остается, хотя и крайне малая, вероятность необнаружения ошибки в верифицируемой программе. Отметим, что экспериментальные проекты по дедуктивной верификации трансляторов появились лишь в последнее время.

Язык P2: другое обобщение оператора суперпозиции

B(x**:** z); C(x, z**:** y) — обобщение оператора суперпозиции.

P(x) {B1(x**:** z1); B2(x, z1**:** z2);...; Bj(x, zj-1**:** zj);...; Bn(x, zn-1**:** y)} Q(x, y) (5.10)

Частный случай: B(x**:** z); C(u, z**:** y), набор u — часть набора x

Наиболее общая форма суперпозиции:

A(x**:** t, y) ≡ P(x) {B(x**:** z, t); C(x, z**:** y)} Q(x, t, y) (5.11)

наборы x и t могут быть пустыми

Спецификации: PB(x), QB(x, z, t), PC(x, z), QC(x, z, y).

B1(x**:** x1, z, t1) ≡ PB(x) {B(x**:** z, t1) || x1 = x} QB(x, z, t1) & x1 = x

C1(x1,z,t1**:** y, t) ≡ PC(x1, z) {C(x1, z**:** y) || t = t1} QC(x1,z,y) & t = t1

[2] издевательски называемый доказательством с помощью карандаша и бумаги

Поскольку B1(x**:** x1, z, t1); C1(x1, z, t1**:** t, y) ≡ B(x**:** z, t); C(x, z**:** y), то справедливо другое определение предиката A:

A(x**:** t, y) ≡ P(x) {B1(x**:** x1, z, t1); C1(x1, z, t1**:** t, y)} Q(x, t, y) (5.12)

LS(B(x**:** z, t); C(x, z**:** y)) ≡ ∃z. (LS(B(x**:** z, t)) & LS(C(z, y))) (5.13)

runCallBlock(s, B(x**:** z, t)); (5.14)

runCallBlock (s, C(x, z**:** y))

Правило RS1'. P(x) ├

PB(x) & ∀x1, z, t1 ((QB(x, z, t1) & x1 = x) ⇒ PC(x1, z))

Правило RS2'. P(x) &

∃x1,z,t1 (QB(x1, z, t1) & x1 = x & QC(x1, z, y) & t = t1) ├ Q(x, t, y)

A(x**:** t, y) ≡ P(x) {B(x**:** z, t); C(x, z**:** y)} Q(x, t, y) (5.11)

Правило RS5. P(x) ├ PB(x) & ∀z, t (QB(x, z, t) ⇒ PC(x, z))

Правило RS6. P(x) & ∃z (QB(x, z, t) & (QC(x, z, y)) ├ Q(x, t, y)

Лемма 5.5. Пусть предусловие P(x) истинно. Допустим, операторы B и C корректны. Если правила **RS5** и **RS6** истинны, то оператор суперпозиции (5.11) является корректным.

Правило LS1'. P(x) ├ PB(x)

Правило LS2'. P(x) & Q(x, t, y) & QB(x, z, t1) & x1=x ├

PC(x1, z) & QC(x1, z, y) & t = t1

Правило LS6. P(x) ├ PB(x)

Правило LS7. P(x) & Q(x, t, y) & QB(x, z, t1) ├

PC(x, z) & QC(x, z, y) & t = t1

Лемма 5.6. Допустим, спецификация оператора суперпозиции (5.11) реализуема, операторы B и C однозначны в области предусловий и корректны, а их спецификации — однозначны. Если правила **LS6** и **LS7** истинны, то оператор суперпозиции (5.11) является корректным.

Метод индукции, использующий меру

∀t ∈ X. [(∀y ∈ X. m(y) < m(t) ⇒ W(y)) ⇒ W(t)]

$$\Rightarrow \forall u \in X.\ W(u) \qquad (3.5)$$

Функция m: X → **nat** называется ***мерой***

Вместо типа **nat** может использоваться ЧУМ со свойством обрыва бесконечно убывающих цепей (well-founded partial order)

5. Построение языка предикатного программирования. Методы доказательства корректности предикатных программ (продолжение)

Методы доказательства корректности рекурсивных программ

Определения предикатов рекурсивного кольца A1, A2,…, An:

$$Aj(xj{:}\ yj) \equiv Pj(xj)\ \{Kj(xj{:}\ yj)\}\ Qj(xj, yj);\ j=1\ldots n;\ n > 0 \qquad (3.36')$$

Здесь Pj и Qj — предусловие и постусловие предиката Aj; xj, yj — различающиеся наборы переменных.

Индукционные переменные

$$Aj(x{:}\ yj) \equiv Pj(x)\ \{Kj(x{:}\ yj)\}\ Qj(x, yj);\ j=1\ldots n;\ n > 0 \qquad (5.17)$$

Набор x объединяет все наборы xj; j=1…n.

Корректность определений предикатов кольца (5.17):

$$W(x) \equiv \forall j=1..n.\{\ Pj(x) \Rightarrow (\exists yj.\ LS(Kj(x{:}\ yj))) \ \&$$

$$(LS(Ki(x{:}\ yj)) \Rightarrow Qj(x, yj))\ \} \qquad (5.18)$$

$$V(x) \equiv \forall j=1..n.\{Pj(x)\ \&\ Qj(x, yj) \Rightarrow LS(Kj(x{:}\ yj)\} \qquad (5.19)$$

Правило RR. $Induct(t, W) \vdash W(t)$.

Правило LR. $Induct(t, V) \vdash V(t)$.

Индукционное предположение $Induct(t, W) \cong \forall u.\ u \sqsubset t \Rightarrow W(u)$

или $Induct(t, W) \cong \forall u.\ m(u) < m(t) \Rightarrow W(u)$.

На значениях набора t определен строгий частичный порядок ⊏, удовлетворяющий свойству обрыва бесконечных убывающих цепей. Либо определена мера m(t)

Рекурсивное кольцо состоит из единственного определения предиката:

$$A(x{:}\ y) \equiv P(x)\ \{\ K(x{:}\ y)\ \}\ Q(x, y) \qquad (5.20)$$

Здесь K(x: y) — оператор, в котором имеются рекурсивные вызовы предиката A. Формула корректности определения (5.20):

$$W(x) \equiv P(x) \Rightarrow (\exists y.\ LS(K(x\mathbf{:}\ y))) \ \&\ (LS(K(x\mathbf{:}\ y)) \Rightarrow Q(x, y))$$

Правило RR для (5.20) принимают следующий вид:

Правило RR1*.

$$Induct(t,W)\ \&\ P(x) \vdash (\exists y.\ LS(K(x\mathbf{:}\ y))) \ \&\ (LS(K(x\mathbf{:}\ y)) \Rightarrow Q(x, y))$$

Лемма 13. Если правило **RR1*** истинно, то рекурсивное определение (5.20) корректно.

Формула корректности определения (5.20) для серии **L** принимает вид:

$$V(x) \equiv P(x)\ \&\ Q(x, y) \Rightarrow LS(K(x\mathbf{:}\ y)) \qquad (2)$$

Правила корректности **LR** для (5.20) принимает вид:

Правило LR1. $Induct(t, V)\ \&\ P(x)\ \&\ Q(x, y) \vdash LS(K(x\mathbf{:}\ y))$.

Лемма 14. Допустим, спецификация [P(x), Q(x, y)] является тотальной, а оператор K(x**:** y) — однозначный. Если правило **LR1** истинно, то определение (5.20) корректно.

Правило для рекурсивного вызова предиката

Пусть имеется рекурсивный вызов предиката A(u**:** y1) в составе оператора K(x**:** y) в определении (5.20) предиката A. Для рекурсивного вызова A(u**:** y1) определим правило:

Правило FB3. $R(u, x, y1) \vdash Less(u, x)\ \&\ P(u)\ \&\ Q(u, y1)$

Предикат Less(u, x) обозначает отношение $u \sqsubset x$ или $m(u) < m(x)$, находящееся в составе предиката Induct(x, V).

Лемма 17. Если истинно правило **FB3**, то истинна следующая формула:

$$Induct(x, V)\ \&\ R(u, x, y1) \Rightarrow LS(A(u\mathbf{:}\ y1))$$

Доказательство. Пусть истинно Induct(x, V) & R(u, x, y1). Докажем истинность LS(A(u**:** y1)). Из истинности предиката R по правилу **FB3** следует истинность отношения Less(u, x) и формулы P(u) & Q(u, y1). Из истинности Induct(x, V) и Less(u, x) следует истинность формулы V(u) (см. (2)), т.е.:

$$P(u)\ \&\ Q(u, y) \Rightarrow LS(K(u\mathbf{:}\ y))$$

Из истинности формулы P(u) & Q(u, y1) следует истинность LS(K(u**:** y1)), т.е. истинна правая часть определения. Следовательно, истинна и LS(A(u**:** y1)). □

$$A(x: y) \equiv P(x) \{ K(x\mathbf{:}\ y) \} Q(x, y) \qquad (5.20)$$

$$\text{Induct}(t, V) \cong \forall u.\ m(u) < m(t) \Rightarrow V(u)$$