§1. Математическая индукция

Исходные понятия арифметики:
единица (1), *натуральное число* и *следовать за* ($\prec$).

Аксиомы арифметики:

1. $(\forall n \in \mathbb{N})\,,\ (\exists_1\, n' \in \mathbb{N}) :\ n \prec n'$;

2. $1 \in \mathbb{N}$, причём $(\nexists n \in \mathbb{N}) :\ n \prec 1$;

3. $(\forall n, k, p \in \mathbb{N})$: $(k \prec n\ \wedge\ p \prec n) \Rightarrow (k \equiv p)$;

4. Если $1 \in A$ и $(\forall k \in \mathbb{N}) : (k \in A) \Rightarrow (k' \in A)$, то $\mathbb{N} \subseteq A$
   (аксиома математической индукции).

Утверждение. (*Метод математической индукции.*) Если высказывание $P(1)$ истинно, а из истинности $P(k)$ следует истинность высказывания $P(k+1)$, то $(\forall n \in \mathbb{N}) :\ P(n)$ – истинно.

Примечание. Предположение истинности $P(k)$, из которого выводится истинность $P(k+1)$, называется *гипотезой индукции*.

Доказательство. Пусть $A = \{n \in \mathbb{N} \mid P(n) -$ истинно$\}$.

Так как $P(1)$ истинно, то $1 \in A$.

Пусть некоторое $k \in A$, то есть $P(k)$ – истинно.

Тогда $P(k+1)$ тоже истинно, откуда следует, что $k' \equiv k+1 \in A$.

Согласно аксиоме математической индукции $\mathbb{N} \subseteq A$.

Так как $A \subseteq \mathbb{N}$, то $A \equiv \mathbb{N}$, и $P(n)$ истинно для любого $n \in \mathbb{N}$. $\square$

Определение. Говорят, что функция $u : \mathbb{N} \longrightarrow X$ задана индуктивно, если вычисление $u(n)$ определяется соотношениями:
1. $u\,(1) = x$;
2. $u\,(k) = F\,(u\,(k-1))$, где $k > 1$, $F : X \longrightarrow X$.

Легко доказать, что $(\forall k \in \mathbb{N}) : \exists_1 u\,(k)$.

Пример. Функция $\gamma : \mathbb{N} \longrightarrow \mathbb{Z}$ индуктивно определяет члены геометрической прогрессии с первым членом a и знаменателем q:
1. $\gamma(1) = a$;
2. $\gamma(k) = q \cdot \gamma(k-1)$.
Здесь $F(x) = q \cdot x$, $F : \mathbb{Z} \longrightarrow \mathbb{Z}$.

Пример. Функция $\varphi : \mathbb{N} \longrightarrow \mathbb{N}$ вычисляет числа Фибоначчи:
$\varphi\,(k) = n$, где $\langle n, m\rangle = f\,(k)$.
При этом функция $f : \mathbb{N} \longrightarrow \mathbb{N}^2$ задана индуктивно:
1. $f(1) = \langle 1, 1\rangle$;
2. $f(k) = \langle b, a+b\rangle$, где $\langle a, b\rangle = f(k-1)$.
Здесь $F(x, y) = \langle y, x+y\rangle$, $F : \mathbb{N}^2 \longrightarrow \mathbb{N}^2$.

§2. Абстрактный синтаксис

Определение. Мы будем называть *абстрактным синтаксисом* упрощённую грамматику языка, в которой отсутствует информация, гарантирующая построение уникальных деревьев вывода.

Определение. Пусть $G = \langle T, N, S, P \rangle$ – грамматика.
Мы будем называть *синтаксическим доменом*, соответствующим нетерминальному символу $x \in N$, множество деревьев вывода, полученных из x по правилам P.

Замечание. Деревья вывода – конечные. Они содержат терминальные символы в качестве листовых вершин.

Абстрактный синтаксис языка While:

$n \in$ Num – числовые константы;
$x \in$ Var – переменные;
$a \in$ Aexp – арифметические выражения;
$b \in$ Bexp – логические выражения;
$S \in$ Stm – операторы.

$n ::= \mathtt{0} \mid \mathtt{1} \mid \ldots \mid \mathtt{9} \mid n\,\mathtt{0} \mid n\,\mathtt{1} \mid \ldots \mid n\,\mathtt{9}$
$x ::= \mathtt{var}\,n$
$a ::= n \mid x \mid a_1 + a_2 \mid a_1 \cdot a_2 \mid a_1 - a_2$
$b ::= \mathtt{true} \mid \mathtt{false} \mid a_1 = a_2 \mid a_1 \leq a_2 \mid \neg b \mid b_1 \wedge b_2$
$S ::= x := a \mid \mathtt{skip} \mid S_1; S_2 \mid \mathtt{if}\,b\,\mathtt{then}\,S_1\,\mathtt{else}\,S_2 \mid \mathtt{while}\,b\,\mathtt{do}\,S$

Пример. Вычисление факториала для начального значения, связанного с переменной x_2.
$x_1 := \mathtt{1};\ \mathtt{while}\,\neg\,(x_2 = \mathtt{1})\ \mathtt{do}\,(x_1 := x_1 \cdot x_2;\ x_2 := x_2 - \mathtt{1})$.

§3. Структурная индукция

Утверждение. (*Метод структурной индукции.*) Если высказывание P истинно для деревьев нулевой глубины, а из истинности P для деревьев, глубина которых меньше k, следует истинность P для деревьев, глубина которых равна k, то P истинно для любого *конечного* дерева.

Доказательство. Воспользуемся методом м.и. по глубине дерева.

Случай $n = 1$ (*глубина дерева меньше 1*):

Имеем дерево, глубина которого равна нулю, для которого P истинно согласно условию.

Случай $n = k$ (P *истинно для всех деревьев глубиной меньше* k; *надо доказать, что* P *истинно для всех деревьев глубиной меньше* $k + 1$):

Из условия имеем истинность P для деревьев глубины k.

Из гипотезы индукции имеем истинность P для деревьев, глубина которых меньше k. $\square$

Понятие индуктивно заданной функции может быть расширено для структурной индукции.

Пример. Функция $\mathcal{N} : \mathsf{Num} \longrightarrow \mathbb{Z}$:

$$
\begin{array}{lcl}
\mathcal{N}[\![0]\!] & = & 0, \\
\mathcal{N}[\![1]\!] & = & 1, \\
\ldots & & \\
\mathcal{N}[\![9]\!] & = & 9, \\
\mathcal{N}[\![n\,0]\!] & = & 10 \cdot \mathcal{N}[\![n]\!], \\
\mathcal{N}[\![n\,1]\!] & = & 10 \cdot \mathcal{N}[\![n]\!] + 1, \\
\ldots & & \\
\mathcal{N}[\![n\,9]\!] & = & 10 \cdot \mathcal{N}[\![n]\!] + 9.
\end{array}
$$

Легко доказать, что $(\forall n) : \exists_1 \mathcal{N}\,(n)$.

§4. Системы переходов

Определение. *Система переходов* – это упорядоченная тройка

$$\langle \Gamma, T, \triangleright \rangle$$

где Γ – это множество *конфигураций*, $T \subseteq \Gamma$ – множество *терминальных* конфигураций, а $\triangleright \subseteq \Gamma \times \Gamma$ – *отношение переходов*.

Для отношения переходов должно выполняться
$(\forall \gamma \in T)\,(\nexists \gamma' \in \Gamma) : \gamma \triangleright \gamma'$.

Все конфигурации $\gamma \in \Gamma \setminus T$ такие, что $(\nexists \gamma' \in \Gamma) : \gamma \triangleright \gamma'$, называются *тупиковыми*.

Важные свойства систем переходов:

- *детерминизм* $((\forall \gamma \in \Gamma)\,(\exists_1\, \gamma') : \gamma \rhd \gamma')$;

- *достижимость* из определённых начальных конфигураций;

  – *безопасность* («плохие» конфигурации недостижимы);

- *правильная завершаемость* (все терминальные конфигурации – «хорошие»).

§5. Среды

Определение. *Среда* (environment) – это функция, отображающая множество переменных в множество их значений.

Пример. Множество Env сред для языка While состоит из функций вида $\mathrm{Var} \hookrightarrow \mathbb{Z}$, которые мы будем записывать в виде списка пар «переменная↦значение»:
$[x_1 \mapsto 2,\ x_2 \mapsto 3,\ x_3 \mapsto -10]$

Обозначение. Если $f : X \longrightarrow Y$, $x \in X$, $y \in Y$, то функция $f\,[x \mapsto y] : X \longrightarrow Y$ определяется как

$$f\,[x \mapsto y]\,(x') = \begin{cases} y, & \text{если } x = x'; \\ f\,(x') & \text{в противном случае.} \end{cases}$$

Пример. Если имеется среда σ, то в результате присвоения некоторого значения i переменной x получается среда $\sigma\,[x \mapsto i]$.

§6. Семантики и семантические функции

Определение. *Семантика* для синтаксического домена D – это кортеж

$\left\langle D, \Sigma, \Sigma_{start}, \Sigma_{final}, \Gamma, \triangleright \right\rangle$, где

Σ – множество состояний вычисления;
$\Sigma_{start} \subseteq \Sigma$ – множество входных состояний вычисления;
$\Sigma_{final} \subseteq \Sigma$ – множество выходных состояний вычисления;

$\left\langle \Gamma, \Sigma_{final}, \triangleright \right\rangle$ – система переходов, множество конфигураций Γ которой состоит из конфигураций вида:

$\langle d, \sigma \rangle$, означающей, что синтаксическая конструкция $d \in D$ должна быть выполнена для состояния $\sigma \in \Sigma$;

σ, представляющей одно из финальных состояний вычисления (это терминальная конфигурация, то есть $\sigma \in \Sigma_{final}$).

Пример. Семантика для арифметических выражений языка While:

$\langle$Aexp, Env $\cup\, \mathbb{Z}$, Env, $\mathbb{Z}$, Г, $\triangleright\rangle$, где

Aexp – синтаксический домен арифметических выражений;
Env $\cup\, \mathbb{Z}$ – множество состояний вычисления;
Env – множество начальных состояний вычисления;
$\mathbb{Z}$ – множество конечных состояний вычисления;
Г – множество конфигураций.

В случае естественной семантики («большие шаги») отношение переходов $\triangleright$ задаёт, например, такие переходы:
$\langle 6 \cdot (2 - x)\,, [x \mapsto 10]\rangle \triangleright -48$,
$\langle 10 \cdot 20, [x_1 \mapsto -1, x_2 \mapsto 4]\rangle \triangleright 200$.

Для редукционной семантики («малые шаги») отношение $\triangleright$ содержит переходы:
$\langle 6 \cdot (2 - x)\,, [x \mapsto 10]\rangle \triangleright \langle 6 \cdot (2 - 10)\,, [x \mapsto 10]\rangle$,
$\langle 10 \cdot 20, [x_1 \mapsto -1, x_2 \mapsto 4]\rangle \triangleright \langle 200, [x_1 \mapsto -1, x_2 \mapsto 4]\rangle$,
$\langle 200, [x_1 \mapsto -1, x_2 \mapsto 4]\rangle \triangleright 200$.

Определение. *Семантическая функция* $\mathcal{S} : D \longrightarrow \left(\Sigma_{start} \hookrightarrow \Sigma_{final}\right)$, выражающая *наблюдаемое поведение* детерминированной семантики $\left\langle D, \Sigma, \Sigma_{start}, \Sigma_{final}, \Gamma, \triangleright \right\rangle$, определяется как

$$\mathcal{S}[\![d]\!](\sigma) = \begin{cases} \sigma', & \text{если } \langle d, \sigma \rangle \triangleright^{\star} \sigma'; \\ \mathbf{undef} & \text{в противном случае.} \end{cases}$$

Определение. Говорят, что детерминированные семантики $\left\langle D, \Sigma, \Sigma_{start}, \Sigma_{final}, \Gamma, \triangleright_1 \right\rangle$ и $\left\langle D, \Sigma, \Sigma_{start}, \Sigma_{final}, \Gamma, \triangleright_2 \right\rangle$ эквивалентны,

если $(\forall d \in D)\,(\forall \sigma \in \Sigma_{start}) : (\mathcal{S}_1[\![d]\!](\sigma) = \sigma') \Leftrightarrow (\mathcal{S}_2[\![d]\!](\sigma) = \sigma')$,

где $\mathcal{S}_1$ и $\mathcal{S}_2$ – семантические функции, выражающие наблюдаемое поведение этих семантик.

Пример. Семантическая функция $\mathcal{A} : \mathsf{Aexp} \longrightarrow (\mathsf{Env} \longrightarrow \mathbb{Z})$ для арифметических выражений языка While (задаётся индуктивно):

$$\begin{array}{lcl}
\mathcal{A}[\![n]\!](\sigma) & = & \mathcal{N}(n), \\
\mathcal{A}[\![x]\!](\sigma) & = & \sigma(x), \\
\mathcal{A}[\![a_1 + a_2]\!](\sigma) & = & \mathcal{A}[\![a_1]\!](\sigma) + \mathcal{A}[\![a_2]\!](\sigma), \\
\mathcal{A}[\![a_1 \cdot a_2]\!](\sigma) & = & \mathcal{A}[\![a_1]\!](\sigma) \cdot \mathcal{A}[\![a_2]\!](\sigma), \\
\mathcal{A}[\![a_1 - a_2]\!](\sigma) & = & \mathcal{A}[\![a_1]\!](\sigma) - \mathcal{A}[\![a_2]\!](\sigma).
\end{array}$$

Пример. Если добавить в язык операцию «унарный минус», то определение функции $\mathcal{A}$ нужно расширить предложением:

$$\mathcal{A}[\![-a]\!](\sigma) = -\mathcal{A}[\![a]\!](\sigma).$$

Альтернативный способ, противоречащий индуктивному заданию функции:

$$\mathcal{A}[\![-a]\!](\sigma) = \mathcal{A}[\![0 - a]\!](\sigma).$$

Пример. Семантическая функция $\mathcal{B} : \mathrm{Bexp} \longrightarrow (\mathrm{Env} \longrightarrow \mathsf{T})$ для логических выражений языка While (задаётся индуктивно):

$$\mathcal{B}[\![\mathtt{true}]\!](\sigma) = \mathbf{tt},$$

$$\mathcal{B}[\![\mathtt{false}]\!](\sigma) = \mathbf{ff},$$

$$\mathcal{B}[\![a_1 = a_2]\!](\sigma) = \begin{cases} \mathbf{tt}, & \text{если } \mathcal{A}[\![a_1]\!](\sigma) = \mathcal{A}[\![a_2]\!](\sigma), \\ \mathbf{ff}, & \text{если } \mathcal{A}[\![a_1]\!](\sigma) \neq \mathcal{A}[\![a_2]\!](\sigma), \end{cases}$$

$$\mathcal{B}[\![a_1 \leq a_2]\!](\sigma) = \begin{cases} \mathbf{tt}, & \text{если } \mathcal{A}[\![a_1]\!](\sigma) \leq \mathcal{A}[\![a_2]\!](\sigma), \\ \mathbf{ff}, & \text{если } \mathcal{A}[\![a_1]\!](\sigma) > \mathcal{A}[\![a_2]\!](\sigma), \end{cases}$$

$$\mathcal{B}[\![\neg b]\!](\sigma) = \begin{cases} \mathbf{tt}, & \text{если } \mathcal{B}[\![b]\!](\sigma) = \mathbf{ff}, \\ \mathbf{ff}, & \text{если } \mathcal{B}[\![b]\!](\sigma) = \mathbf{tt}, \end{cases}$$

$$\mathcal{B}[\![b_1 \wedge b_2]\!](\sigma) = \begin{cases} \mathbf{tt}, & \text{если } \mathcal{B}[\![b_1]\!](\sigma) = \mathbf{tt} \text{ и } \mathcal{B}[\![b_2]\!](\sigma) = \mathbf{tt}, \\ \mathbf{ff}, & \text{если } \mathcal{B}[\![b_1]\!](\sigma) = \mathbf{ff} \text{ или } \mathcal{B}[\![b_2]\!](\sigma) = \mathbf{ff}. \end{cases}$$

§7. Естественная семантика

В естественной семантике все переходы имеют вид $\langle d, \sigma \rangle \rhd \sigma'$. Это означает, что выполнение синтаксической конструкции d из состояния вычисления σ завершается, и результирующим состоянием будет σ'.

Определение. Правило естественной семантики $\left\langle D, \Sigma, \Sigma_{start}, \Sigma_{final}, \Gamma, \to \right\rangle$ записывается в виде

$$\frac{\left\langle \widehat{d_1},\, \widehat{\sigma_1} \right\rangle \to \widehat{\sigma'_1}, \cdots, \left\langle \widehat{d_n},\, \widehat{\sigma_n} \right\rangle \to \widehat{\sigma'_n}}{\left\langle \widehat{d},\, \widehat{\sigma} \right\rangle \to \widehat{\sigma'}}, \text{ если } \ldots$$

Здесь $\widehat{d}, \widehat{d_i} \subseteq D$, $\widehat{\sigma}, \widehat{\sigma_i} \subseteq \Sigma_{start}$, $\widehat{\sigma'_i} \subseteq \Sigma_{final}$ – обобщённые записи подмножеств множеств D и Σ (образцы с метапеременными).

При этом $\widehat{d_1}, \cdots, \widehat{d_n}$ – это либо непосредственные поддеревья $\widehat{d}$, либо деревья, сконструированные из непосредственных поддеревьев $\widehat{d}$.

Над чертой – *предпосылки*, под чертой – *следствия*, справа – *условия*, правила без предпосылок – *аксиомы* (записываются без гор. черты).

Пример. Естественная семантика для операторов языка While.

$[\text{assign}_{ns}]$ $\quad \langle x := a,\ \sigma\rangle \to \sigma\,[x \mapsto \mathcal{A}[\![a]\!](\sigma)]$

$[\text{skip}_{ns}]$ $\quad \langle \texttt{skip},\ \sigma\rangle \to \sigma$

$[\text{comp}_{ns}]$ $\quad \dfrac{\langle S_1,\ \sigma\rangle \to \sigma', \quad \langle S_2,\ \sigma'\rangle \to \sigma''}{\langle S_1; S_2,\ \sigma\rangle \to \sigma''}$

$\left[\text{if}^{\mathbf{tt}}_{ns}\right]$ $\quad \dfrac{\langle S_1,\ \sigma\rangle \to \sigma'}{\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ \sigma\rangle \to \sigma'}$, если $\mathcal{B}[\![b]\!](\sigma) = \mathbf{tt}$

$\left[\text{if}^{\mathbf{ff}}_{ns}\right]$ $\quad \dfrac{\langle S_2,\ \sigma\rangle \to \sigma'}{\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ \sigma\rangle \to \sigma'}$, если $\mathcal{B}[\![b]\!](\sigma) = \mathbf{ff}$

$\left[\text{while}^{\mathbf{tt}}_{ns}\right]$ $\quad \dfrac{\langle S,\ \sigma\rangle \to \sigma', \quad \langle \texttt{while}\ b\ \texttt{do}\ S,\ \sigma'\rangle \to \sigma''}{\langle \texttt{while}\ b\ \texttt{do}\ S,\ \sigma\rangle \to \sigma''}$, если $\mathcal{B}[\![b]\!](\sigma) = \mathbf{tt}$

$\left[\text{while}^{\mathbf{ff}}_{ns}\right]$ $\quad \langle \texttt{while}\ b\ \texttt{do}\ S,\ \sigma\rangle \to \sigma$, если $\mathcal{B}[\![b]\!](\sigma) = \mathbf{ff}$

Каждое правило семантики является схемой для порождения некоторого множества переходов.

Определение. Мы будем называть переходы, порождаемые правилами семантики, *экземплярами* этих правил.

Пример. Одним из экземпляров аксиомы [assign$_{ns}$] является переход $\langle x_1 := 1,\ [x_1 \mapsto 0,\ x_2 \mapsto 0]\rangle \to [x_1 \mapsto 1,\ x_2 \mapsto 0]$

Пример. Одним из экземпляров правила [comp$_{ns}$] является переход $\langle x_1 := 5;\ x_2 := 3,\ [x_1 \mapsto 4]\rangle \to [x_1 \mapsto 5,\ x_2 \mapsto 3]$

Вывод перехода $\langle d, \sigma\rangle \to \sigma'$ в некоторой семантике связан с построением *дерева вывода*, корнем которого является выводимый переход, листьями – экземпляры аксиом семантики.

Пример. Дерево вывода для перехода

$$\langle (x_3 := x_2;\ x_2 := x_1)\,;\ x_1 := x_3,\ \sigma_1\rangle \to \sigma_4$$

записывается так:

$$\dfrac{\dfrac{\langle x_3 := x_2,\ \sigma_1\rangle \to \sigma_2, \quad \langle x_2 := x_1,\ \sigma_2\rangle \to \sigma_3}{\langle x_3 := x_2;\ x_2 := x_1,\ \sigma_1\rangle \to \sigma_3}, \quad \langle x_1 := x_3,\ \sigma_3\rangle \to \sigma_4}{\langle (x_3 := x_2;\ x_2 := x_1)\,;\ x_1 := x_3,\ \sigma_1\rangle \to \sigma_4},$$ где

$\sigma_1 = [x_1 \mapsto 4,\ x_2 \mapsto 5]$,
$\sigma_2 = [x_1 \mapsto 4,\ x_2 \mapsto 5,\ x_3 \mapsto 5]$,
$\sigma_3 = [x_1 \mapsto 4,\ x_2 \mapsto 4,\ x_3 \mapsto 5]$,
$\sigma_4 = [x_1 \mapsto 5,\ x_2 \mapsto 4,\ x_3 \mapsto 5]$.

Пусть требуется построить дерево вывода для некоторой синтаксической конструкции d из состояния вычисления σ. Для этого требуется найти правило семантики, левую часть следствия которого можно отождествить с конфигурацией $\langle d, \sigma \rangle$. При этом возможны два случая:

1. Если найденное правило является аксиомой, и условия этой аксиомы выполняются, то мы можем сразу же определить выходное состояние вычисления. Тем самым построение дерева вывода завершается.

2. Если найденное правило содержит предпосылки, то мы пытаемся построить деревья вывода для каждой предпосылки. В случае успешного построения этих деревьев мы обязаны проверить условия, связанные с правилом, и, если эти условия выполняются, мы можем определить выходное состояние вычисления.

Пример. Покажем процесс построения дерева вывода T для оператора

$$\texttt{while}\,\neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)$$

из состояния вычисления $\sigma_1 = [x_1 \mapsto 2,\ x_2 \mapsto 2]$.

1. Согласно правилу $\left[\text{while}^{\mathbf{tt}}_{ns}\right]$:

$$T = \frac{T_1, \quad T_2}{\langle \texttt{while}\,\neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\ \sigma_1\rangle \to \sigma_6}, \text{ где}$$

$$T_1 = \frac{\ldots}{\langle x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1,\ \sigma_1\rangle \to \sigma_3},$$

$$T_2 = \frac{\ldots}{\langle \texttt{while}\,\neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\ \sigma_3\rangle \to \sigma_6}.$$

2. Согласно правилу [comp$_{ns}$]:

$$T_1 = \frac{T_3, \quad T_4}{\langle x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1,\ \sigma_1\rangle \to \sigma_3}, \text{ где}$$

$T_3 = \langle x_2 := x_2 \cdot 2,\ \sigma_1\rangle \to \sigma_2$ (аксиома [assign$_{ns}$]),
$T_4 = \langle x_1 := x_1 - 1,\ \sigma_2\rangle \to \sigma_3$ (аксиома [assign$_{ns}$]).

Учитывая, что $\sigma_1 = [x_1 \mapsto 2,\ x_2 \mapsto 2]$, получаем
$\sigma_2 = \sigma_1\,[x_2 \mapsto 4] = [x_1 \mapsto 2,\ x_2 \mapsto 4]$,
$\sigma_3 = \sigma_2\,[x_1 \mapsto 1] = [x_1 \mapsto 1,\ x_2 \mapsto 4]$.

3. Согласно правилу $\left[\text{while}^{\mathbf{tt}}_{ns}\right]$:

$$T_2 = \frac{T_5, \quad T_6}{\langle \texttt{while}\, \neg\, (x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\ \sigma_3 \rangle \to \sigma_6}, \text{ где}$$

$$T_5 = \frac{\ldots}{\langle x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1,\ \sigma_3 \rangle \to \sigma_5},$$

$$T_6 = \frac{\ldots}{\langle \texttt{while}\, \neg\, (x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\ \sigma_5 \rangle \to \sigma_6}.$$

4. Согласно правилу [comp$_{ns}$]:

$$T_5 = \frac{T_7, \quad T_8}{\langle x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1,\ \sigma_3\rangle \to \sigma_5}, \text{ где}$$

$T_7 = \langle x_2 := x_2 \cdot 2,\ \sigma_3\rangle \to \sigma_4$ (аксиома [assign$_{ns}$]),
$T_8 = \langle x_1 := x_1 - 1,\ \sigma_4\rangle \to \sigma_5$ (аксиома [assign$_{ns}$]).

Учитывая, что $\sigma_3 = [x_1 \mapsto 1,\ x_2 \mapsto 4]$, получаем
$\sigma_4 = \sigma_3\,[x_2 \mapsto 8] = [x_1 \mapsto 1,\ x_2 \mapsto 8]$,
$\sigma_5 = \sigma_4\,[x_1 \mapsto 0] = [x_1 \mapsto 0,\ x_2 \mapsto 8]$.

5. Согласно аксиоме $\left[\text{while}^{\mathbf{ff}}_{ns}\right]$:

$T_6 = \langle \texttt{while}\ \neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\ \sigma_5\rangle \to \sigma_6$, где

$\sigma_6 = \sigma_5 = [x_1 \mapsto 0,\ x_2 \mapsto 8]$.

§8. Проведение доказательств по форме дерева вывода

Если отношение переходов семантики не задано индуктивно, то структурная индукция по деревьям абстрактного синтаксиса невозможна, и приходится проводить доказательства по форме дерева вывода.

Пример. Естественная семантика языка While содержит неиндуктивное правило $\left[\text{while}^{\mathbf{tt}}_{ns}\right]$, поэтому доказательство свойств семантики языка While с помощью структурной индукции по деревьям абстрактного синтаксиса невозможно.

Схема доказательства по форме дерева вывода:

1. Необходимо доказать, что некое свойство семантики выполняется для всех элементарных деревьев вывода, то есть для аксиом семантики.

2. Для каждого правила, не являющегося аксиомой, предполагаем, что нужное свойство выполняется для всех его предпосылок. Базируясь на этом предположении, доказываем свойство для следствия.

Пример. Докажем, что естественная семантика языка While является детерминированной.

Доказательство. Предполагая $\langle S, \sigma \rangle \to \sigma'$, мы докажем, что из $\langle S, \sigma \rangle \to \sigma''$ следует $\sigma' = \sigma''$.

Для доказательства используем индукцию по форме дерева вывода для $\langle S, \sigma \rangle \to \sigma'$.

Случай [**assign**$_{ns}$]:
$S = x := a$ и $\sigma' = \sigma\,[x \mapsto \mathcal{A}[\![a]\!]\,(\sigma)]$.
Единственное правило, которое может быть применено для вывода $\langle S, \sigma \rangle \to \sigma''$, – это [assign$_{ns}$]. Поэтому $\sigma'' = \sigma\,[x \mapsto \mathcal{A}[\![a]\!]\,(\sigma)] = \sigma'$.

Случай [**skip**$_{ns}$]:
аналогично.

Случай [**comp**$_{ns}$]:
Предположим, что $\langle S_1; S_2,\, \sigma\rangle \to \sigma'$.
Значит $\exists \sigma_0 : \langle S_1,\, \sigma\rangle \to \sigma_0, \quad \langle S_2,\, \sigma_0\rangle \to \sigma'$.

Единственное правило, которое может быть применено для вывода $\langle S_1; S_2,\, \sigma\rangle \to \sigma''$, – это [comp$_{ns}$].
Отсюда $\exists \sigma_1 : \langle S_1,\, \sigma\rangle \to \sigma_1, \quad \langle S_2,\, \sigma_1\rangle \to \sigma''$.

Применяя гипотезу индукции к предпосылке $\langle S_1,\, \sigma\rangle \to \sigma_0$, получаем, что $\langle S_1,\, \sigma\rangle \to \sigma_1 \Rightarrow \sigma_0 = \sigma_1$.
Применяя гипотезу индукции к предпосылке $\langle S_2,\, \sigma_0\rangle \to \sigma'$, получаем, что $\langle S_2,\, \sigma_0\rangle \to \sigma'' \Rightarrow \sigma' = \sigma''$.

Случай $\left[\mathbf{if}^{\mathbf{tt}}_{ns}\right]$:
Предположим, что $\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ \sigma\rangle \to \sigma'$.
То есть $\mathcal{B}[\![b]\!]\,(\sigma) = \mathbf{tt}$ и $\langle S_1,\ \sigma\rangle \to \sigma'$.

Так как $\mathcal{B}[\![b]\!]\,(\sigma) = \mathbf{tt}$, то единственное правило, которое может быть применено для вывода $\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ \sigma\rangle \to \sigma''$ – это $\left[\mathbf{if}^{\mathbf{tt}}_{ns}\right]$.
Отсюда $\langle S_1,\ \sigma\rangle \to \sigma''$.

Применяя гипотезу индукции к предпосылке $\langle S_1,\ \sigma\rangle \to \sigma'$, получаем, что $\langle S_1,\ \sigma\rangle \to \sigma'' \Rightarrow \sigma' = \sigma''$.

Случай $\left[\mathbf{if}^{\mathbf{ff}}_{ns}\right]$:
аналогично.

Случай $\left[\textbf{while}^{\mathbf{tt}}_{ns}\right]$:
Предположим, что $\langle \texttt{while}\, b\, \texttt{do}\, S,\, \sigma\rangle \to \sigma'$.
То есть $\mathcal{B}[\![b]\!]\,(\sigma) = \mathbf{tt}$ и $\exists \sigma_0 : \langle S,\, \sigma\rangle \to \sigma_0,\, \langle \texttt{while}\, b\, \texttt{do}\, S,\, \sigma_0\rangle \to \sigma'$.

Так как $\mathcal{B}[\![b]\!]\,(\sigma) = \mathbf{tt}$, то единственное правило, которое может быть применено для вывода $\langle \texttt{while}\, b\, \texttt{do}\, S,\, \sigma\rangle \to \sigma''$ – это $\left[\textbf{while}^{\mathbf{tt}}_{ns}\right]$.
Отсюда $\exists \sigma_1 : \langle S,\, \sigma\rangle \to \sigma_1,\, \langle \texttt{while}\, b\, \texttt{do}\, S,\, \sigma_1\rangle \to \sigma''$.

Применяя гипотезу индукции к предпосылке $\langle S,\, \sigma\rangle \to \sigma_0$, получаем, что $\langle S,\, \sigma\rangle \to \sigma_1 \Rightarrow \sigma_0 = \sigma_1$.
Применяя гипотезу индукции к предпосылке $\langle \texttt{while}\, b\, \texttt{do}\, S,\, \sigma_0\rangle \to \sigma'$, получаем, что $\langle \texttt{while}\, b\, \texttt{do}\, S,\, \sigma_0\rangle \to \sigma'' \Rightarrow \sigma' = \sigma''$.

Случай $\left[\textbf{while}^{\mathbf{ff}}_{ns}\right]$:
элементарно.$\square$

§9. Редукционная семантика

В редукционной семантике используются два вида переходов:

$\langle d, \sigma \rangle \triangleright \langle d', \sigma' \rangle$ – частичное выполнение синтаксической конструкции d из состояния вычисления σ (оставшаяся часть вычислений выражается промежуточной конфигурацией $\langle d', \sigma' \rangle$);

$\langle d, \sigma \rangle \triangleright \sigma'$ – выполнение синтаксической конструкции d из состояния вычисления σ завершается, и результирующим состоянием становится σ'.

Правила редукционной семантики записываются в том же виде, что и правила естественной семантики.

Из естественной семантики в редукционную также переходит понятие *дерева вывода*.

Пример. Редукционная семантика для операторов языка While.

$[\text{assign}_{rs}]$ $\quad \langle x := a,\ \sigma\rangle \Rrightarrow \sigma\,[x \mapsto \mathcal{A}[\![a]\!](\sigma)]$

$[\text{skip}_{rs}]$ $\quad \langle \texttt{skip},\ \sigma\rangle \Rrightarrow \sigma$

$$\left[\text{comp}^1_{rs}\right] \quad \frac{\langle S_1,\ \sigma\rangle \Rrightarrow \left\langle S_1',\ \sigma'\right\rangle}{\langle S_1; S_2,\ \sigma\rangle \Rrightarrow \left\langle S_1'; S_2,\ \sigma'\right\rangle}$$

$$\left[\text{comp}^2_{rs}\right] \quad \frac{\langle S_1,\ \sigma\rangle \Rrightarrow \sigma'}{\langle S_1; S_2,\ \sigma\rangle \Rrightarrow \langle S_2,\ \sigma'\rangle}$$

$\left[\text{if}^{\mathbf{tt}}_{rs}\right]$ $\quad \langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ \sigma\rangle \Rrightarrow \langle S_1,\ \sigma\rangle$, если $\mathcal{B}[\![b]\!](\sigma) = \mathbf{tt}$

$\left[\text{if}^{\mathbf{ff}}_{rs}\right]$ $\quad \langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ \sigma\rangle \Rrightarrow \langle S_2,\ \sigma\rangle$, если $\mathcal{B}[\![b]\!](\sigma) = \mathbf{ff}$

$[\text{while}_{rs}]$ $\quad \langle \texttt{while}\ b\ \texttt{do}\ S,\ \sigma\rangle \Rrightarrow \langle \texttt{if}\ b\ \texttt{then}\ (S; \texttt{while}\ b\ \texttt{do}\ S)\ \texttt{else}\ \texttt{skip},\ \sigma\rangle$

Пример. Дерево вывода для перехода

$$\langle (x_3 := x_2;\ x_2 := x_1)\,;\ x_1 := x_3,\ \sigma\rangle \Rrightarrow \langle x_2 := x_1;\ x_1 := x_3,\ \sigma'\rangle$$

записывается так:

$$\dfrac{\dfrac{\langle x_3 := x_2,\ \sigma\rangle \Rrightarrow \sigma'}{\langle x_3 := x_2;\ x_2 := x_1,\ \sigma\rangle \Rrightarrow \langle x_2 := x_1,\ \sigma'\rangle}}{\langle (x_3 := x_2;\ x_2 := x_1)\,;\ x_1 := x_3,\ \sigma\rangle \Rrightarrow \langle x_2 := x_1;\ x_1 := x_3,\ \sigma'\rangle}$$, где

$\sigma = [x_1 \mapsto 4,\ x_2 \mapsto 5]$,
$\sigma' = [x_1 \mapsto 4,\ x_2 \mapsto 5,\ x_3 \mapsto 5]$.

Определение. *Последовательность вывода* для синтаксической конструкции d из состояния σ – это либо конечная последовательность конфигураций $\gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_k$, либо бесконечная последовательность конфигураций $\gamma_0, \gamma_1, \gamma_2, \ldots$.
При этом $\gamma_0 = \langle d, \sigma \rangle$, $\gamma_i \Rrightarrow \gamma_{i+1}$ и γ_k – это либо терминальная, либо тупиковая конфигурация.

Обозначение. Мы будем использовать запись $\gamma_0 \Rrightarrow^i \gamma_i$, чтобы показать, что γ_0 переходит в γ_i за i шагов. Запись $\gamma_0 \Rrightarrow^\star \gamma_i$ будет означать, что между γ_0 и γ_i – конечное число шагов. При этом надо понимать, что $\gamma_0 \Rrightarrow^i \gamma_i$ и $\gamma_0 \Rrightarrow^\star \gamma_i$ являются последовательностями вывода только в том случае, если γ_i – терминальная или тупиковая конфигурация.

Определение. Мы будем говорить, что выполнение синтаксической конструкции d из состояния вычисления σ *завершается*, если существует конечная последовательность вывода, начинающаяся с $\langle d, \sigma \rangle$.
При этом выполнение *успешно*, если $\langle d, \sigma \rangle \Rrightarrow^{\star} \sigma'$.

Определение. Мы будем говорить, что выполнение синтаксической конструкции d из состояния вычисления σ *зацикливается*, если существует бесконечная последовательность вывода, начинающаяся с $\langle d, \sigma \rangle$.

Пример. Покажем процесс построения последовательности вывода для оператора

$$\texttt{while}\,\neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)$$

из состояния вычисления $\sigma_1 = [x_1 \mapsto 2,\ x_2 \mapsto 2]$.

1. Согласно правилу [while$_{rs}$]:

$$\begin{array}{l}\langle \texttt{while}\,\neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\ \sigma_1\rangle \Rrightarrow \\ \quad \langle \texttt{if}\,\neg\,(x_1 = 0)\ \texttt{then}\,(\\ \qquad (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,; \\ \qquad \texttt{while}\,\neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1) \\ \qquad) \\ \quad \texttt{else} \\ \qquad \texttt{skip}, \\ \quad \sigma_1\rangle\end{array}$$

2. Согласно правилу $\left[\mathsf{if}^{\mathbf{tt}}_{rs}\right]$:

$$
\begin{array}{l}
\langle \texttt{if} \neg (x_1 = 0) \texttt{ then} (\\
\quad (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,; \\
\quad \texttt{while} \neg (x_1 = 0) \texttt{ do } (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1) \\
\quad) \\
\texttt{else} \\
\quad \texttt{skip}, \\
\sigma_1 \rangle \Rrightarrow \\
\quad \langle (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,; \\
\quad \texttt{while} \neg (x_1 = 0) \texttt{ do } (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,, \\
\quad \sigma_1 \rangle
\end{array}
$$

3. Согласно правилу $\left[\mathrm{comp}^1_{rs}\right]$:

$$\dfrac{\dfrac{\langle x_2 := x_2 \cdot 2,\, \sigma_1\rangle \Rrightarrow \sigma_2}{\langle x_2 := x_2 \cdot 2;\, x_1 := x_1 - 1,\, \sigma_1\rangle \Rrightarrow \langle x_1 := x_1 - 1,\, \sigma_2\rangle}}{\begin{array}{l}\langle (x_2 := x_2 \cdot 2;\, x_1 := x_1 - 1)\,; \\ \ \texttt{while}\, \neg (x_1 = 0)\, \texttt{do}\, (x_2 := x_2 \cdot 2;\, x_1 := x_1 - 1)\,, \\ \ \sigma_1\rangle \Rrightarrow \\ \quad \langle x_1 := x_1 - 1; \\ \quad\ \texttt{while}\, \neg (x_1 = 0)\, \texttt{do}\, (x_2 := x_2 \cdot 2;\, x_1 := x_1 - 1)\,, \\ \quad\ \sigma_2\rangle\end{array}}$$

Учитывая, что $\sigma_1 = [x_1 \mapsto 2,\, x_2 \mapsto 2]$, получаем $\sigma_2 = \sigma_1\,[x_2 \mapsto 4] = [x_1 \mapsto 2,\, x_2 \mapsto 4]$.

4. Согласно правилу $\left[\mathsf{comp}^1_{rs}\right]$:

$$\frac{\langle x_1 := x_1 - 1,\, \sigma_2\rangle \Rrightarrow \sigma_3}{\begin{array}{l}\langle x_1 := x_1 - 1;\\ \ \mathtt{while}\, \neg (x_1 = 0)\ \mathtt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\\ \ \sigma_2\rangle \Rrightarrow\\ \quad \langle \mathtt{while}\, \neg (x_1 = 0)\ \mathtt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\, \sigma_3\rangle\end{array}}$$

Учитывая, что $\sigma_2 = [x_1 \mapsto 2,\, x_2 \mapsto 4]$, получаем $\sigma_3 = \sigma_2\,[x_1 \mapsto 1] = [x_1 \mapsto 1,\, x_2 \mapsto 4]$.

5. Согласно правилу [while$_{rs}$]:

$$\begin{array}{l}\langle \texttt{while}\ \neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\ \sigma_3\rangle \Rrightarrow \\ \quad \langle \texttt{if}\ \neg\,(x_1 = 0)\ \texttt{then}\,(\\ \qquad (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,; \\ \qquad \texttt{while}\ \neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1) \\ \qquad) \\ \quad \texttt{else} \\ \qquad \texttt{skip}, \\ \quad \sigma_3\rangle\end{array}$$

6. Согласно правилу $\left[\mathsf{if}^{\mathbf{tt}}_{rs}\right]$:

$$
\begin{array}{l}
\langle \texttt{if} \neg (x_1 = 0) \texttt{ then} (\\
\quad (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,; \\
\quad \texttt{while} \neg (x_1 = 0) \texttt{ do } (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1) \\
\quad) \\
\texttt{else} \\
\quad \texttt{skip}, \\
\sigma_3 \rangle \Rrightarrow \\
\quad \langle (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,; \\
\quad \texttt{while} \neg (x_1 = 0) \texttt{ do } (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,, \\
\quad \sigma_3 \rangle
\end{array}
$$

7. Согласно правилу $\left[\mathrm{comp}^1_{rs}\right]$:

$$\dfrac{\dfrac{\langle x_2 := x_2 \cdot 2,\, \sigma_3\rangle \Rrightarrow \sigma_4}{\langle x_2 := x_2 \cdot 2;\, x_1 := x_1 - 1,\, \sigma_3\rangle \Rrightarrow \langle x_1 := x_1 - 1,\, \sigma_4\rangle}}{\begin{array}{l}\langle (x_2 := x_2 \cdot 2;\, x_1 := x_1 - 1)\,; \\ \texttt{while}\, \neg (x_1 = 0)\, \texttt{do}\, (x_2 := x_2 \cdot 2;\, x_1 := x_1 - 1)\,, \\ \sigma_3\rangle \Rrightarrow \\ \quad \langle x_1 := x_1 - 1; \\ \quad \texttt{while}\, \neg (x_1 = 0)\, \texttt{do}\, (x_2 := x_2 \cdot 2;\, x_1 := x_1 - 1)\,, \\ \quad \sigma_4\rangle\end{array}}$$

Учитывая, что $\sigma_3 = [x_1 \mapsto 1,\, x_2 \mapsto 4]$, получаем $\sigma_4 = \sigma_3\,[x_2 \mapsto 8] = [x_1 \mapsto 1,\, x_2 \mapsto 8]$.

8. Согласно правилу $\left[\mathsf{comp}^1_{rs}\right]$:

$$\frac{\langle x_1 := x_1 - 1,\, \sigma_4\rangle \Rrightarrow \sigma_5}{\begin{array}{l}\langle x_1 := x_1 - 1;\\ \ \texttt{while}\, \neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\\ \ \sigma_4\rangle \Rrightarrow\\ \quad \langle \texttt{while}\, \neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\, \sigma_5\rangle\end{array}}$$

Учитывая, что $\sigma_4 = [x_1 \mapsto 1,\, x_2 \mapsto 8]$, получаем $\sigma_5 = \sigma_4\,[x_1 \mapsto 0] = [x_1 \mapsto 0,\, x_2 \mapsto 8]$.

9. Согласно правилу [while$_{rs}$]:

$$
\begin{array}{l}
\langle \texttt{while}\ \neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,,\ \sigma_5\rangle \Rrightarrow \\
\quad \langle \texttt{if}\ \neg\,(x_1 = 0)\ \texttt{then}\ (\\
\qquad (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,; \\
\qquad \texttt{while}\ \neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1) \\
\qquad) \\
\quad \texttt{else} \\
\qquad \texttt{skip}, \\
\quad \sigma_5\rangle
\end{array}
$$

10. Согласно правилу $\left[\mathsf{if}^{\mathbf{ff}}_{rs}\right]$:

$$\begin{array}{l}\langle \texttt{if}\ \neg\,(x_1 = 0)\ \texttt{then}\,(\\ \quad (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\,;\\ \quad \texttt{while}\ \neg\,(x_1 = 0)\ \texttt{do}\ (x_2 := x_2 \cdot 2;\ x_1 := x_1 - 1)\\ \quad)\\ \texttt{else}\\ \quad \texttt{skip},\\ \sigma_5\rangle \Rrightarrow\\ \quad \langle \texttt{skip},\ \sigma_5\rangle\end{array}$$

11. И, наконец, по правилу $[\mathsf{skip}_{rs}]$:

$\langle \texttt{skip},\ \sigma_5\rangle \Rrightarrow \sigma_5$.

§10. Проведение доказательств по длине последовательности вывода

Доказательство свойств редукционных семантик удобно проводить индукцией по длине последовательности вывода.

Схема доказательства по длине последовательности вывода:

1. Необходимо доказать, что нужное свойство выполняется для всех последовательностей вывода нулевой длины.

2. На базе предположения, что нужное свойство выполняется для всех последовательностей вывода, длина которых не превышает k, доказываем, что оно также верно и для последовательностей длины $k+1$.

Пример. Докажем для семантики языка While, что если $\langle S_1; S_2, \sigma\rangle \Rrightarrow^k \sigma''$, то существует состояние σ' и натуральные числа k_1 и k_2 такие, что $\langle S_1, \sigma\rangle \Rrightarrow^{k_1} \sigma'$ и $\langle S_2, \sigma'\rangle \Rrightarrow^{k_2} \sigma''$, причём $k_1 + k_2 = k$.

Доказательство. Проведём доказательство индукцией по числу k, то есть индукцией по длине последовательности вывода $\langle S_1; S_2, \sigma\rangle \Rrightarrow^k \sigma''$.
Случай $k = 0$ (*последовательности нулевой длины*):
Существование последовательности $\langle S_1; S_2, \sigma\rangle \Rrightarrow^0 \sigma''$ означало бы, что $\langle S_1; S_2, \sigma\rangle \equiv \sigma''$. Это невозможно, поэтому доказываемое свойство автоматически верно для $k = 0$.

Случай $k = k_0$ (*доказываемое свойство истинно для всех последовательностей, длина которых меньше или равна* k_0; *надо доказать, что* оно *истинно для всех последовательностей длины* $k_0 + 1$):
Пусть
$\langle S_1; S_2, \sigma\rangle \Rrightarrow^{k_0+1} \sigma''$.
Мы можем записать эту последовательность вывода в виде
$\langle S_1; S_2, \sigma\rangle \Rrightarrow \gamma \Rrightarrow^{k_0} \sigma''$.

Конфигурация γ может быть получена двумя способами:
по правилу $\left[\text{comp}^1_{rs}\right]$ или по правилу $\left[\text{comp}^2_{rs}\right]$.

1. В первом случае $\langle S_1; S_2, \sigma\rangle \Rrightarrow \left\langle S_1'; S_2, \sigma_0\right\rangle$, так как $\langle S_1, \sigma\rangle \Rrightarrow \left\langle S_1', \sigma_0\right\rangle$.

Тем самым мы имеем последовательность вывода $\left\langle S_1'; S_2, \sigma_0\right\rangle \Rrightarrow^{k_0} \sigma''$, к которой может быть применена гипотеза индукции. Это означает, что существует состояние σ' и натуральные числа k_1 и k_2 такие, что $\left\langle S_1', \sigma_0\right\rangle \Rrightarrow^{k_1} \sigma'$ и $\langle S_2, \sigma'\rangle \Rrightarrow^{k_2} \sigma''$, причём $k_1 + k_2 = k_0$.

Учитывая, что $\langle S_1, \sigma\rangle \Rrightarrow \left\langle S_1', \sigma_0\right\rangle$ и $\left\langle S_1', \sigma_0\right\rangle \Rrightarrow^{k_1} \sigma'$, получаем
$\langle S_1, \sigma\rangle \Rrightarrow^{k_1+1} \sigma'$.
Кроме того, мы знаем, что $\langle S_2, \sigma'\rangle \Rrightarrow^{k_2} \sigma''$.
А так как $(k_1 + 1) + k_2 = k_0 + 1$, то доказываемое свойство выполняется.

2. Во втором случае $\langle S_1; S_2, \sigma\rangle \Rrightarrow \langle S_2, \sigma_0\rangle$, так как $\langle S_1, \sigma\rangle \Rrightarrow \sigma_0$.

Тем самым $\langle S_2, \sigma_0\rangle \Rrightarrow^{k_0} \sigma''$, и доказываемое свойство выполняется, если мы выберем $k_1 = 1$ и $k_2 = k_0$. $\square$