پردیس علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

بررسی مدل‌های محاسباتی حساب لاندا و گودل- کلینی

نگارنده

کامیار میرزاوزیری

استاد راهنما: دکتر مجتبی مجتهدی

پایان‌نامه برای دریافت درجه کارشناسی
در رشته علوم کامپیوتر

چکیده

در این پروژه، ابتدا دو مدل محاسباتی لاندا و گودل ـ کلینی را معرفی می‌کنیم. سپس دو تابع برای تبدیل مسائل (توابع) هر یک به دیگری معرفی می‌کنیم. بعد از آن، ثابت می‌کنیم این دو مدل با یکدیگر معادل می‌باشند، یا به عبارت دیگر قدرت محاسباتی حساب لاندا به اندازه گودل ـ کلینی می‌باشد و بالعکس.

پیشگفتار

حساب لاندا، دستگاهی صوری در منطق ریاضی جهت بیان محاسبات بر اساس تعریف توابع و اعمال آن‌ها بر روی اشیاء می‌باشد. در این دستگاه صوری ساده چنین فرض می‌شود که همه اشیاء تابع می‌باشند، همان طور که در مبانی ریاضی معمولا فرض می‌کنیم همه چیز مجموعه می‌باشند و اعداد، توابع، روابط، ترتیب‌ها و سایر ساختارهای ریاضی را بر اساس مجموعه‌ها تعریف می‌کنیم، در حساب لاندا ساختارهای ریاضی را بر اساس توابع تعریف و پیاده‌سازی می‌کنیم.[۱] همچنین برای سادگی بیشتر این دستگاه، فرض می‌شود که توابع تنها تک‌موضعی می‌باشند و توابع چندموضعی به صورت زنجیری از توابع تک‌موضعی تعریف می‌شوند. برای مثال عملگر جمع را در نظر بگیریم. این عملگر در حساب لاندا به این صورت تعریف می‌شود.

«تابعی که اگر x را به عنوان ورودی دریافت کند، تابعی را خروجی می‌دهد که این تابع هر ورودی که بگیرد آن را با x جمع می‌کند و خروجی می‌دهد.»

هرچند این جمله تابعی شهودی را معرفی می‌کند در حالی که مفاهیم عدد و جمع را باید ابتدا به کمک λـتابع‌ها تعریف کنیم.

این دستگاه صوری، توسط آلونزو چرچ (از شاگردهای او می‌توان به آلن تورینگ، استیون کول کلینی و جان بارکلی راسر) در دهه ۱۹۳۰ میلادی، به عنوان بخشی از تحقیقات او بر روی مبانی ریاضی معرفی گشت. هرچند در سال ۱۹۳۵، هنگامی که کلینی و راسر روی پارادوکس کلینی‌ـراسر کار می‌کردند، مشخص شد که نسخه اولیه حساب لاندا که توسط چرچ معرفی شده بود به لحاظ منطقی ناسازگار است.[۲][۳] آن‌ها موفق شدند که پارادوکس ریچارد را به زبان فرمال در این دستگاه صوری بیان کنند. پارادوکس ریچارد را می‌توان به صورت‌های مختلفی بیان کرد اما یک صورت ساده آن به این صورت است.

فرض کنیم A مجموعه تمام اعداد صحیحی باشد که در زبان فارسی می‌توان هر یک را در کمتر از ۱۰۰ حرف توصیف کرد. می‌دانیم چنین مجموعه‌ای متناهی است چرا که تعداد جملاتی که کمتر از ۱۰۰ حرف دارند با توجه به متناهی بودن حروف الفبا متناهی است. حال توصیف زیر را در نظر بگیریم.

«کوچک‌ترین عدد صحیح مثبت که در A نیست.»

جمله بالا کمتر از ۱۰۰ حرف دارد و به وضوح عددی را تعریف می‌کند (چون A متناهی

است) که در A وجود ندارد و این یک تناقض است.

اما پس از آن، نسخه سازگاری از حساب لاندا معرفی شد. تا مدت‌ها، کاربرد مهمی از حساب لاندا در علوم کامپیوتر دیده نمی‌شد و تنها به عنوان یک نوع فرمال‌گرایی از حساب لاندا یاد می‌شد، اما با ظهور زبان‌های برنامه‌نویسی تابع‌گرا، که پایه و اساس‌شان بر روی حساب لاندا بنا شده، توجه به این دستگاه صوری افزایش یافت. یکی از مزایای بسیار مهم برنامه‌نویسی تابع‌گرا قابلیت کنترل عوارض جانبی اجرای قطعه‌کدها می‌باشد. به این صورت که می‌توان این محدودیت را ایجاد کرد که توابعی که در برنامه تعریف شده‌اند تنها مجاز به محاسبه و برگرداندن خروجی می‌باشد و اجازه ندارند در وضعیت برنامه تغییری ایجاد کنند. هرچند مفهوم برنامه‌نویسی تابع‌گرا نیز مفهوم به نسبت جدیدی نیست اما پس از مدت‌ها، در چند سال اخیر به شدت مورد استقبال قرار گرفته و کتابخانه‌ها و چارچوب‌های زیادی (مانند چارچوب React در زبان JavaScript برای توسعه وب و موبایل) حول تابع‌گرایی به وجود آمده و بسیاری معتقدند که روش تابع‌گرایی از بسیاری جهات (از جمله فهم راحت‌تر به دلیل نزدیکی بیشتر با شهود انسان‌ها) بهتر از رقیب اصلی خود یعنی روش شیٔ‌گرایی می‌باشد.

در نظریه محاسبه، معمولا توانایی محاسبات مدل‌های محاسباتی را بر اساس ماشین تورینگ می‌سنجند. طبق تز چرچ_تورینگ، هر تابعی که به مفهوم عام و شهودی محاسبه‌پذیر باشد، یعنی الگوریتمی برای محاسبه آن وجود داشته باشد را می‌توان با ماشین تورینگ محاسبه کرد.[۴] با قبول کردن این تز، برای آن که توان محاسباتی یک مدل محاسباتی را بسنجیم کافی است آن را با ماشین تورینگ مقایسه کنیم و بررسی کنیم که آیا هر تابعی که ماشین تورینگ بتواند آن را محاسبه کند، مدل مد نظر ما نیز می‌تواند آن را محاسبه کند. اگر چنین باشد می‌گوییم مدل محاسباتی ما تورینگ_کامل است، به این معنی که توان محاسباتی آن در حد ماشین تورینگ است. واضح است که اگر معادل بودن مدلی با ماشین تورینگ ثابت شود، می‌توانیم برای بررسی توان محاسباتی مدل‌ها از این مدل نیز استفاده کنیم، و در مواقعی ممکن است اثبات معادل بودن با این مدل کاری ساده‌تر باشد.

مدل محاسباتی گودل_کلینی در واقع به روشی استقرایی، مجموعه تمام توابع محاسبه‌پذیر را تعریف می‌کند. ثابت شده که این توابع دقیقا همان توابع تورینگ–محاسبه‌پذیر می‌باشند. یعنی هر تابع در مجموعه توابع محاسبه‌پذیر گودل کلینی را می‌توان با یک ماشین تورینگ محاسبه کرد و هر تابعی که بتوان آن را با یک ماشین تورینگ محاسبه کرد در مجموعه توابع محاسبه‌پذیر گودل_کلینی وجود دارد.

در این پروژه قصد داریم این دو مدل را معرفی کنیم و در ادامه نشان دهیم توان محاسباتی حساب لاندا با ماشین تورینگ معادل است، که برای این منظور و برای سادگی در اثبات، از مجموعه توابع محاسبه‌پذیر گودل_کلینی استفاده می‌کنیم.

اما یکی از چالش‌ها در این اثبات تفاوت دنیای حساب لاندا و سایر مدل‌های محاسباتی است. اکثر مدل‌های محاسباتی تورینگ_کامل معرفی‌شده در دنیای اعداد طبیعی یا صحیح، و یا دنیای رشته‌ها عمل می‌کنند. اما در حساب لاندا، تنها λ_تابع‌ها وجود دارند و لازم است به نوعی مسائل

را از یک دنیا به دنیای دیگر تبدیل کنیم. یعنی بتوانیم اعداد را به کمک توابع لاندا، و توابع لاندا را به کمک اعداد تعریف کنیم.

چالش دیگر پیاده‌سازی مفهوم حلقه، و به خصوص حلقه بی‌پایان در حساب لاندا است. برای این منظور عبارتی در حساب لاندا معرفی خواهیم کرد که آن را هرچقدر ساده کنیم به خودش تبدیل می‌شود و در واقع در حلقه‌ای بی‌پایان گیر می‌کند، که معادل هالت نکردن در ماشین تورینگ و یا کمینه‌سازی نامحدود تابعی که کمینه ندارد در مجموعه توابع محاسبه‌پذیر گودل_ کلینی می‌باشد.

در این پروژه ابتدا با معرفی دقیق‌تر این مدل‌های محاسباتی شروع می‌کنیم و سپس روشی برای تبدیل توابع از یک دنیا به دنیای دیگر معرفی می‌کنیم. در انتها برای اثبات هم‌ارزی این دو مدل، ابتدا نشان می‌دهیم هر تابع محاسبه‌پذیر گودل_ کلینی یک تابع محاسبه‌پذیر حساب لاندا است، و سپس عکس این موضوع را نشان می‌دهیم.

فهرست مطالب

فصل ۱

حساب لاندا

در این بخش به معرفی مدل محاسباتی[۱] حساب لاندا[۲] و تعریف λـعبارت‌ها و توابع ΛCـ محاسبه‌پذیر می‌پردازیم.

تعریف ۱.۱. *مجموعه همه λـعبارت‌ها[۳] را $TERM_\lambda$ می‌نامیم و به صورت استقرایی تعریف می‌کنیم.*

$$
\begin{array}{lll}
i. & x \in \{x_0, x_1, \dots\} \Longrightarrow x \in TERM_\lambda; & (\text{متغیرهای اتمی}^{۴}) \\
ii. & A, B \in TERM_\lambda \Longrightarrow A \cdot B \in TERM_\lambda; & (\text{اعمال}^{۵}) \\
iii. & x \in \{x_0, x_1, \dots\}, A \in TERM_\lambda \Longrightarrow \lambda x : A \in TERM_\lambda. & (\text{انتزاع}^{۶})
\end{array}
$$

اما برای سادگی، قراردادهایی را تنظیم می‌کنیم.

- می‌توانیم به جای x_iها از سایر حروف کوچک لاتین به عنوان متغیرهای اتمی استفاده کنیم.
- فرض می‌کنیم عمل اعمال وابسته چپ می‌باشد، یعنی

$$A \cdot B \cdot C = (A \cdot B) \cdot C.$$

- فرض می‌کنیم عمل انتزاع نسبت به عمل اعمال مقدم است، یعنی

$$\lambda x : A \cdot B = (\lambda x : A) \cdot B.$$

[1] Model of Computation
[2] Lambda Calculus
[3] λ-term
[4] Atomic Variable
[5] Application
[6] Abstraction

تعریف ۲.۱. *دو λـ عبارت A و B را αـ هم‌ارز[۷] می‌گوییم و می‌نویسیم $A =_\alpha B$، هرگاه تنها در متغیرهای محدود تفاوت داشته باشند. همچنین مجموعه کلاس‌های αـ هم‌ارزی[۸] را $TERM_\alpha$ می‌نامیم.*

برای مثال $\lambda f : \lambda t : (f \cdot t)$ و $\lambda a : \lambda b : (a \cdot b)$ با یکدیگر αـ هم‌ارز می‌باشند.

تعریف ۳.۱. *فرض کنیم $A, B \in TERM_\lambda$ دو λـ عبارت و x یک متغیر اتمی باشد. عبارت $A[x \mapsto B]$ را جایگزین‌شده x با B در A[۹] می‌نامیم که همان عبارت A می‌باشد که در آن تمام xهایی که متغیر آزاد می‌باشند با عبارت B جایگزین شده‌اند.*

برای مثال $\lambda y : (x \cdot \lambda x : x)[x \mapsto \lambda z : z]$ برابر $\lambda y : (\lambda z : z \cdot \lambda x : x)$ می‌باشد.

تعریف ۴.۱. *تابع $\beta : TERM_\alpha \to TERM_\alpha \cup \{\Omega\}$ را گام محاسباتی[۱۰] می‌نامیم و به صورت زیر تعریف می‌کنیم.*

$$\beta([T]_\alpha) = \begin{cases} [X(A[x \mapsto B])Y]_\alpha & [T]_\alpha = [X(\lambda x : A \cdot B)Y]_\alpha \\ \Omega & otherwise. \end{cases}$$

همچنین هنگامی که می‌نویسیم $A \to_\beta B$، به این معنی است که $\beta([A]_\alpha) = [B]_\alpha$.

تعریف ۵.۱. *فرض کنیم $T \in TERM_\lambda$ یک λـ عبارت و n یک عدد صحیح مثبت باشد. تابع جزئی $\Phi_T^{(n)} : (TERM_\alpha)^n \to TERM_\alpha$ را به صورت زیر تعریف می‌کنیم.*

$$\Phi_T^{(n)}([T_1]_\alpha, \ldots, [T_n]_\alpha) = \beta^k([T \cdot T_1 \cdots T_n]_\alpha).$$

که در آن k کوچک‌ترین عددی است که $\beta^{k+1}([T \cdot T_1 \cdots T_n]_\alpha) = \Omega$. اگر چنین عددی وجود نداشته باشد، در این صورت $\Phi_T^{(n)}([T_1]_\alpha, \ldots, [T_n]_\alpha)$ را تعریف‌نشده در نظر می‌گیریم.

واضح است که توابعی که تعریف کردیم خوش‌تعریف می‌باشند. حال با داشتن این تعاریف می‌توانیم مفهوم ΛC‌ـ محاسبه‌پذیری را برای توابع تعریف کنیم.

تعریف ۶.۱. *تابع $f : (TERM_\alpha)^n \to TERM_\alpha$ را ΛCـ محاسبه‌پذیر[۱۱] می‌گوییم، اگر $T \in TERM_\lambda$ وجود داشته باشد، به طوری که $f = \Phi_T^{(n)}$. مجموعه تمام توابع ΛCـ محاسبه‌پذیر مانند $f : (TERM_\alpha)^n \to TERM_\alpha$ را ΛC_λ می‌نامیم.*

---

[7] α-equivalent

[8] α-equivalence Class

[9] A where x is replaced by B

[10] Computation Step

[11] ΛC-computable

به کمک گام محاسباتی می‌توانیم یک هم‌ارزی سطح بالاتر روی λـ عبارت‌ها تعریف کنیم.

تعریف ۷.۱. *می‌گوییم A و B ~ـ هم‌ارز*[12] *می‌باشند و می‌نویسیم $A \sim B$ هرگاه عبارات $X, X_1, \ldots, X_{n-1} \in TERM_\lambda$ و روابط $R_1, \ldots, R_n \in \{=, =_\alpha, \to_\beta, \leftarrow_\beta\}$ موجود باشند، چنان که*

$$\begin{array}{lll} A \cdot X & R_1 & X_1 \\ & R_2 & X_2 \\ & \cdots & \\ & R_{n-1} & X_{n-1} \\ & R_n & B \cdot X. \end{array}$$

همچنین اگر

$$\begin{array}{lll} A & R_1 & X_1 \\ & R_2 & X_2 \\ & \cdots & \\ & R_{n-1} & X_{n-1} \\ & R_n & B. \end{array}$$

می‌گوییم A و B اکیدا ~ـ هم‌ارز[13] *می‌باشند و می‌نویسیم $A \approx B$*

وجود X در تعریف بالا در واقع فرض کردن اصل موضوع گسترش[14] می‌باشد، که به سادگی بیان می‌کند اگر دو تابع روی تمام ورودی‌ها رفتار یکسانی نشان دهند، یکسان می‌باشند.
واضح است که اگر دو عبارت اکیدا ~ـ هم‌ارز باشند، ~ـ هم‌ارز نیز می‌باشند. کافی است X را به تمام جملات اضافه کنیم. اما عکس آن الزاما درست نیست. برای مثال $\lambda t : (A \cdot t) \sim A$ چرا که $\lambda t : (A \cdot t) \cdot X \to_\beta A \cdot X$ اما واضح است که $\lambda t : (A \cdot t) \not\approx A$ چرا که هیچ یک از طرفین را نمی‌توان بیش از این ساده کرد. (گام محاسباتی را اعمال کرد.)
برای سادگی $A \sim B$ را با $A = B$ نمایش می‌دهیم. منظور از این گزاره این است که کلاس‌های این دو عبارت با یکدیگر مساوی می‌باشند یعنی $[A]_\sim = [B]_\sim$.
این یک سطح بالاتر از تعبیر λـ عبارت‌ها است که مشابه تعبیر عبارت‌های حسابی می‌باشد. برای مثال عبارت‌های 2×2، $2 + 2$، 2^2، و 4 عبارت‌های متفاوتی می‌باشند، با طول‌های متفاوت، نمادهای متفاوت و ...، اما معمولا $4 = 2 \times 2$ را یک گزاره درست در نظر می‌گیریم، هرچند

[12] ~-equivalent
[13] Strictly ~-equivalent
[14] Axiom of Extensionality

طرف چپ تساوی یک رشته به طول ۳ و طرف راست یک رشته به طول ۱ می‌باشد و این رشته‌ها به وضوح متفاوت می‌باشند. همانطور که زبان‌های برنامه‌نویسی `"4" == "2 * 2"` را نادرست اما `4 == 2 * 2` را درست تعبیر می‌کنند. همین ایده را در مورد λـ عبارت‌ها به کار می‌بریم.

فصل ۲

گودل- کلینی

در این بخش به معرفی مدل محاسباتی گودل- کلینی[1] و تعریف توابع GK- محاسبه‌پذیر می‌پردازیم.

تعریف ۱.۲. *دو تابع $Z : \mathbb{N} \to \mathbb{N}$ و $S : \mathbb{N} \to \mathbb{N}$، و خانواده توابع $U_i^{(n)} : \mathbb{N}^n \to \mathbb{N}$ را به صورت زیر تعریف می‌کنیم.*

$$\begin{aligned} &Z(a) = 0; \\ &S(a) = a + 1; \\ &U_i^{(n)}(a_1, \ldots, a_n) = a_i. \end{aligned}$$

مجموعه این توابع را مجموعه توابع اساسی[2] می‌نامیم و با ZSU نمایش می‌دهیم.

تعریف ۲.۲. *فرض کنیم $f : \mathbb{N}^k \to \mathbb{N}$ و $G = (g_1, \ldots, g_k)$ یک k- تایی از توابع $\mathbb{N}^n \to \mathbb{N}$ باشد. تابع $f \circ G : \mathbb{N}^n \to \mathbb{N}$ که آن را ترکیب[3] f و G می‌نامیم به صورت زیر تعریف می‌شود.*

$$(f \circ G)(a_1, \ldots, a_n) = f(g_1(a_1, \ldots, a_n), \ldots, g_k(a_1, \ldots, a_n)).$$

تعریف ۳.۲. *فرض کنیم $f : \mathbb{N}^n \to \mathbb{N}$ و $g : \mathbb{N}^{n+2} \to \mathbb{N}$. تابع $\eta_{f,g} : \mathbb{N}^{n+1} \to \mathbb{N}$ که آن را بازگشت مقدماتی[4] f و g می‌نامیم به صورت زیر تعریف می‌شود.*

$$\eta_{f,g}(a_1, \ldots, a_n, t) = \begin{cases} f(a_1, \ldots, a_n) & t = 0 \\ g(\eta_{f,g}(a_1, \ldots, a_n, t-1), a_1, \ldots, a_n, t-1) & otherwise. \end{cases}$$

[1] Gödel-Kleene
[2] The Set of Basic Functions
[3] Composition
[4] Primitive Recursion

تعریف ۴.۲. *فرض کنیم* $f : \mathbb{N}^{n+1} \to \mathbb{N}$ *یک تابع باشد. تابع* $\mu_f : \mathbb{N}^n \to \mathbb{N}$ *که آن را کمینه‌سازی نامحدود*[5] f *می‌نامیم به صورت زیر تعریف می‌شود.*

$$\mu_f(a_1, \ldots, a_n) = \min_t (f(a_1, \ldots, a_n, t) = 0).$$

از این به بعد، منظور از $f^{(n)}$، $f : \mathbb{N}^n \to \mathbb{N}$، و منظور از $F^{(k,n)}$ این است که F یک kـ‌تایی از توابع $\mathbb{N}^n \to \mathbb{N}$ می‌باشد.

تعریف ۵.۲. *مجموعه تمام توابع* GKـ*محاسبه‌پذیر*[6] $f : \mathbb{N}^n \to \mathbb{N}$ *را* $GK_{\mathbb{N}}$ *می‌نامیم که به صورت استقرایی تعریف می‌شود.*

$$\begin{array}{lll} \text{(توابع اساسی)} & i. & ZSU \subseteq GK_{\mathbb{N}}; \\ \text{(ترکیب)} & ii. & f^{(k)} \in GK_{\mathbb{N}} \wedge G^{(k,n)} \subseteq GK_{\mathbb{N}} \Rightarrow f \circ G \in GK_{\mathbb{N}}; \\ \text{(بازگشت مقدماتی)} & iii. & f^{(n)} \in GK_{\mathbb{N}} \wedge g^{(n+2)} \in GK_{\mathbb{N}} \Rightarrow \eta_{f,g} \in GK_{\mathbb{N}}; \\ \text{(کمینه‌سازی نامحدود)} & iv. & f^{(n+1)} \in GK_{\mathbb{N}} \Rightarrow \mu_f \in GK_{\mathbb{N}}. \end{array}$$

[5] Unbounded Minimization
[6] GK-computable

فصل ۳

توابع تبدیل

حال که دو مدل محاسباتی را معرفی کردیم، لازم است راهی برای تبدیل توابع هر یک به دیگری معرفی کنیم تا بتوانیم توان محاسباتی این دو مدل را با یکدیگر مقایسه کنیم. می‌دانیم توابع مورد بحث در حساب لاندا، روی λـ عبارت‌ها تعریف شده‌اند اما توابع مورد بحث در گودلـ کلینی روی اعداد طبیعی تعریف شده‌اند. لذا به دو تابع $\lfloor\cdot\rfloor_\lambda : \mathbb{N} \to \mathrm{TERM}_\alpha$ برای تبدیل اعداد طبیعی به λـ عبارت‌ها و $\lfloor\cdot\rfloor_\mathbb{N} : \mathrm{TERM}_\lambda \cup \{\Omega\} \to \mathbb{N}$ برای تبدیل λـ عبارت‌ها به اعداد طبیعی نیاز داریم.

تعریف ۱.۳. *توابع λـ تبدیل[1] و $\mathbb{N}$ـ تبدیل[2] را به صورت استقرایی تعریف می‌کنیم.*

$$\lfloor n\rfloor_\lambda = \begin{cases} [\lambda f : \lambda t : t]_\alpha & n = 0 \\ [\lambda f : \lambda t : (f \cdot (\lfloor n-1\rfloor_\lambda \cdot f \cdot t))]_\alpha & otherwise; \end{cases}$$

$$\lfloor T\rfloor_\mathbb{N} = \begin{cases} \langle 2, 0, 1\rangle & T = \Omega \\ \langle i+2, 0, 0\rangle & T = x_i \\ \langle 0, \lfloor A\rfloor_\mathbb{N}, \lfloor B\rfloor_\mathbb{N}\rangle & T = A \cdot B \\ \langle 1, \lfloor x\rfloor_\mathbb{N}, \lfloor A\rfloor_\mathbb{N}\rangle & T = \lambda x : A. \end{cases}$$

توابعی که تعریف کردیم به وضوح یک‌به‌یک می‌باشند اما پوشا نیستند، لذا توابع معکوس برای توابع وجود دارند اما جزئی می‌باشند. این توابع را به صورت $\lceil\cdot\rceil_\lambda : \mathrm{TERM}_\alpha \to \mathbb{N}$ و $\lceil\cdot\rceil_\mathbb{N} : \mathbb{N} \to \mathrm{TERM}_\lambda$ نمایش می‌دهیم.

توجه کنیم که تابع $\lfloor.\rfloor_\lambda$ یک کلاس هم‌ارزی برمی‌گرداند اما برای محاسبات، به یک λـ عبارت نیاز داریم.

[1] λ-coding
[2] $\mathbb{N}$-coding

تعریف ۲.۳. *تابع λـ صریح‌تبدیل[۳] را به صورت استقرایی تعریف می‌کنیم.*

$$\llcorner n \lrcorner_\lambda \;=\; \begin{cases} \lambda f : \lambda t : t & n = 0 \\ \lambda f : \lambda t : (f \cdot (\lfloor n-1 \rfloor_\lambda \cdot f \cdot t)) & otherwise. \end{cases}$$

پس $\lfloor n \rfloor_\lambda = [\llcorner n \lrcorner_\lambda]_\alpha$.
حال می‌توانیم مفاهیم ΛC‌ـ محاسبه‌پذیری و GKـ محاسبه‌پذیری را تعمیم دهیم.

تعریف ۳.۳. *تابع $f : \mathbb{N}^n \to \mathbb{N}$ را ΛCـ محاسبه‌پذیر گوییم اگر تابع ΛCـ محاسبه‌پذیری مانند $f_\lambda : (TERM_\alpha)^n \to TERM_\alpha$ وجود داشته باشد، چنان که برای هر عدد $x_1, \ldots, x_n$ داشته باشیم*

$$f_\lambda(\lfloor x_1 \rfloor_\lambda, \ldots, \lfloor x_n \rfloor_\lambda) = \lfloor f(x_1, \ldots, x_n) \rfloor_\lambda.$$

مجموعه تمام توابع ΛCـ محاسبه‌پذیر $f : \mathbb{N}^n \to \mathbb{N}$ را $\Lambda C_{\mathbb{N}}$ می‌نامیم.

به طور مشابه،

تعریف ۴.۳. *تابع $f : (TERM_\alpha)^n \to TERM_\alpha$ را GKـ محاسبه‌پذیر گوییم اگر تابع GKـ محاسبه‌پذیری مانند $f_{\mathbb{N}} : \mathbb{N}^n \to \mathbb{N}$ وجود داشته باشد، چنان که برای هر λـ عبارت $T_1, \ldots, T_n$ داشته باشیم*

$$f([T_1]_\alpha, \ldots, [T_n]_\alpha) = [\lceil f_{\mathbb{N}}(\lfloor T_1 \rfloor_{\mathbb{N}}, \ldots, \lfloor T_n \rfloor_{\mathbb{N}}) \rceil_{\mathbb{N}}]_\alpha.$$

مجموعه تمام توابع ΛCـ محاسبه‌پذیر $f : (TERM_\alpha)^n \to TERM_\alpha$ را GK_λ می‌نامیم.

این مفاهیم به وضوح خوش‌تعریف می‌باشند.

---
[3] λ-explicit-coding

فصل ۴

حساب لاندا در مقایسه با گودل ـ کلینی

در این بخش نشان می‌دهیم هر تابع GK ـ محاسبه‌پذیر $f : \mathbb{N}^n \to \mathbb{N}$ یک تابع ΛC ـ محاسبه‌پذیر می‌باشد. این موضوع را با استقرا روی ساخت f نشان می‌دهیم.

اما در ابتدا لازم است راهی ساده‌تر برای اثبات ΛC ـ محاسبه‌پذیر بودن یک تابع ارائه دهیم.

لم ۱.۰.۴ *فرض کنیم $f : \mathbb{N}^n \to \mathbb{N}$ یک تابع باشد. اگر λ ـ عبارت T وجود داشته باشند که برای هر $x_1, \ldots, x_n \in \mathbb{N}$ داشته باشیم*

$$T \cdot \llcorner x_1 \lrcorner_\lambda \cdots \llcorner x_n \lrcorner_\lambda \sim \llcorner f(x_1, \ldots, x_n) \lrcorner_\lambda.$$

در این صورت $f \in \Lambda C_{\mathbb{N}}$

اثبات. می‌دانیم $\llcorner f(x_1, \ldots, x_n) \lrcorner_\lambda$، در برد تابع λ ـ صریح‌تبدیل قرار دارد و لذا شامل عبارتی به فرم $\lambda x : A \cdot B$ نمی‌باشد، پس $\beta([\llcorner f(x_1, \ldots, x_n) \lrcorner_\lambda]_\alpha) = \Omega$ و هیچ عبارتی مانند X وجود ندارد که $\llcorner f(x_1, \ldots, x_n) \lrcorner_\lambda \to_\beta X$. پس تمام گام‌های محاسباتی در این مسیر هم‌ارزی از سمت چپ به راست می‌باشد و همچنین واضح است عددی مانند k وجود دارد به طوری که $[\llcorner f(x_1, \ldots, x_n) \lrcorner_\lambda]_\alpha = \beta^k([T_0 \cdot \llcorner x_1 \lrcorner_\lambda \cdots \llcorner x_n \lrcorner_\lambda]_\alpha)$. پس

$$\Phi^{(n)}_{T_0}(\lfloor x_1 \rfloor_\lambda, \ldots, \lfloor x_n \rfloor_\lambda) = \lfloor f(x_1, \ldots, x_n) \rfloor_\lambda.$$

لذا f تابعی ΛC ـ محاسبه‌پذیر است. □

عبارت T که در این لم به کار رفته را عبارت محاسبه‌گر[1] f می‌نامیم و معمولا با T_f نشان می‌دهیم. از این پس، برای آن که نشان دهیم f یک تابع ΛC ـ محاسبه‌پذیر است، کافی است عبارت محاسبه‌گر آن را معرفی کنیم.

[1] Computer Term

چهار لم مهم در این بخش وجود دارند که نشان می‌دهند $\mathrm{ZSU} \subseteq \Lambda\mathrm{C}_{\mathbb{N}}$، و $\Lambda\mathrm{C}_{\mathbb{N}}$ تحت ترکیب، بازگشت مقدماتی و کمینه‌سازی محدود بسته است. با اثبات اولین لم شروع می‌کنیم.

لم ۲.۴. *توابع Z، S و $U_i^{(n)}$، ΛCـ محاسبه‌پذیر می‌باشند، به عبارت دیگر*

$$ZSU \subseteq \Lambda C_{\mathbb{N}}.$$

اثبات. عبارات محاسبه‌گر زیر را تعریف می‌کنیم.

$$\begin{aligned}
T_Z &= \lambda x : \llcorner 0 \lrcorner_\lambda; \\
T_S &= \lambda x : \lambda f : \lambda t : (f \cdot (x \cdot f \cdot t)); \\
T_{U_i^{(n)}} &= \lambda x_1 : \lambda x_2 : \ldots \lambda x_n : x_i.
\end{aligned}$$

داریم

$$\begin{aligned}
T_Z \cdot \llcorner a \lrcorner_\lambda &= (\lambda x : \llcorner 0 \lrcorner_\lambda) \cdot \llcorner a \lrcorner_\lambda \\
&= \llcorner 0 \lrcorner_\lambda \\
&= \llcorner Z(a) \lrcorner_\lambda,
\end{aligned}$$

و

$$\begin{aligned}
T_S \cdot \llcorner a \lrcorner_\lambda &= (\lambda x : \lambda f : \lambda t : (f \cdot (x \cdot f \cdot t))) \cdot \llcorner a \lrcorner_\lambda \\
&= \lambda f.\lambda t.(f.(\llcorner a \lrcorner_\lambda.f.t)) \\
&= \llcorner a+1 \lrcorner_\lambda \\
&= \llcorner S(a) \lrcorner_\lambda,
\end{aligned}$$

و

$$\begin{aligned}
T_{U_i^{(n)}} \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda &= (\lambda x_1 : \lambda x_2 : \ldots \lambda x_n : x_i) \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \\
&= \llcorner a_i \lrcorner_\lambda \\
&= \llcorner U_i^{(n)}(a_1, \ldots, a_n) \lrcorner_\lambda.
\end{aligned}$$

$\square$

بدون هیچ وقفه‌ای، می‌توانیم لم مهم دوم را ثابت کنیم، $\Lambda\mathrm{C}_{\mathbb{N}}$ تحت ترکیب بسته است.

لم ۳.۴. *اگر $f^{(k)}$ یک تابع ΛCـ محاسبه‌پذیر و $G^{(k,n)}$ یک kـ تایی از توابع ΛCـ محاسبه‌پذیر باشد، در این صورت، $f \circ G$ نیز، یک تابع ΛCـ محاسبه‌پذیر می‌باشد. به عبارت دیگر*

$$(f^{(k)} \in \Lambda C_{\mathbb{N}} \text{ و } G^{(k,n)} \subseteq \Lambda C_{\mathbb{N}}) \Rightarrow f \circ G \in \Lambda C_{\mathbb{N}}.$$

اثبات. طبق فرض می‌دانیم عبارات محاسبه‌گر T_f و $T_{g_1},\ldots,T_{g_k}$ برای توابع f و $g_1,\ldots,g_k$ وجود دارند. عبارت محاسبه‌گر $T_{f\circ G}$ را به صورت زیر تعریف می‌کنیم. (می‌توانیم فرض کنیم عبارات T_f و $T_{g_1},\ldots,T_{g_k}$ متغیر آزاد مشترکی با هم و با متغیرهای $x_1,\ldots,x_n$ ندارند.)

$$T_{f\circ G}=\lambda x_1:\lambda x_2:\ldots\lambda x_n:(T_f\cdot(T_{g_1}\cdot x_1\cdots x_n)\cdots(T_{g_n}\cdot x_1\cdots x_n)).$$

داریم

$$\begin{aligned}
& T_{f\circ G}\cdot\llcorner a_1\lrcorner_\lambda\cdots\llcorner a_n\lrcorner_\lambda\\
= & \lambda x_1:\lambda x_2:\ldots\lambda x_n:(T_f\cdot(T_{g_1}\cdot x_1\cdots x_n)\cdots(T_{g_n}\cdot x_1\cdots x_n))\cdot\\
& \llcorner a_1\lrcorner_\lambda\cdots\llcorner a_n\lrcorner_\lambda\\
= & T_f\cdot(T_{g_1}\cdot\llcorner a_1\lrcorner_\lambda\cdots\llcorner a_n\lrcorner_\lambda)\cdots(T_{g_k}\cdot\llcorner a_1\lrcorner_\lambda\cdots\llcorner a_n\lrcorner_\lambda)\\
= & T_f\cdot\llcorner g_1(a_1,\ldots,a_n)\lrcorner_\lambda\cdots\llcorner g_k(a_1,\ldots,a_n)\lrcorner_\lambda\\
= & \llcorner f(g_1(a_1,\ldots,a_n),\ldots,g_k(a_1,\ldots,a_n))\lrcorner_\lambda\\
= & \llcorner f\circ G(a_1,\ldots,a_n)\lrcorner_\lambda.
\end{aligned}$$

□

برای آن که بتوانیم توابع بازگشتی را محاسبه کنیم نیاز داریم k تا λـ‌عبارت را در یک λـ‌عبارت بگنجانیم، به بیان دیگر، به عبارتی مانند K_k نیاز داریم که با داشتن عبارات $X_1,\ldots,X_k$، یک عبارت مانند $K_{(X_1,\ldots,X_k)}$ تولید کند که نماینده kـ‌تایی $(X_1,\ldots,X_k)$ باشد و تمام X_iها در آن نشانده شده باشند.

$$K_{(X_1,\ldots,X_k)}=K_k\cdot X_1\cdots X_k.$$

در تعریف زیر، فرض می‌کنیم K_k و X_iها متغیر آزاد مشترکی ندارند.

تعریف ۴.۴. *دو عبارت T_{sgn} و $T_{\overline{sgn}}$ به ترتیب λـ‌علامت*[2] *و λـ‌وارون‌علامت*[3] *می‌نامیم و به صورت زیر تعریف می‌کنیم.*

$$\begin{aligned}
T_{\overline{sgn}} &= \lambda n:(n\cdot F_Z\cdot\llcorner 1\lrcorner_\lambda);\\
T_{sgn} &= \lambda n:(T_{\overline{sgn}}\cdot(T_{\overline{sgn}}\cdot n)).
\end{aligned}$$

لم ۵.۴. *برای هر $a\in\mathbb{N}$ داریم*

$$\begin{aligned}
T_{\overline{sgn}}\cdot\llcorner a\lrcorner_\lambda &= \llcorner\overline{sgn}(a)\lrcorner_\lambda;\\
T_{sgn}\cdot\llcorner a\lrcorner_\lambda &= \llcorner sgn(a)\lrcorner_\lambda;
\end{aligned}$$

[2] λ-sign
[3] λ-sign-inverse

اثبات. برای λـوارون‌علامت اگر $a = 0$ داریم

$$\begin{aligned} T_{\overline{\text{sgn}}} \cdot \llcorner 0 \lrcorner_\lambda &= \lambda n : (n \cdot F_Z \cdot \llcorner 1 \lrcorner_\lambda) \cdot \llcorner 0 \lrcorner_\lambda \\ &= \llcorner 0 \lrcorner_\lambda \cdot F_Z \cdot \llcorner 1 \lrcorner_\lambda \\ &= \llcorner 1 \lrcorner_\lambda \\ &= \llcorner \overline{\text{sgn}}(0) \lrcorner_\lambda . \end{aligned}$$

و اگر $a > 0$ داریم

$$\begin{aligned} T_{\overline{\text{sgn}}} \cdot \llcorner a \lrcorner_\lambda &= \lambda n : (n \cdot F_Z \cdot \llcorner 1 \lrcorner_\lambda) \cdot \llcorner a \lrcorner_\lambda \\ &= \llcorner a \lrcorner_\lambda \cdot F_Z \cdot \llcorner 1 \lrcorner_\lambda \\ &= \llcorner 0 \lrcorner_\lambda \\ &= \llcorner \overline{\text{sgn}}(a) \lrcorner_\lambda . \end{aligned}$$

برای λـعلامت داریم

$$\begin{aligned} T_{\text{sgn}} \cdot \llcorner a \lrcorner_\lambda &= \lambda n : (T_{\overline{\text{sgn}}} \cdot (T_{\overline{\text{sgn}}} \cdot n)) \cdot \llcorner a \lrcorner_\lambda \\ &= T_{\overline{\text{sgn}}} \cdot (T_{\overline{\text{sgn}}} \cdot \llcorner a \lrcorner_\lambda) \\ \text{(بخش قبل)} \quad &= T_{\overline{\text{sgn}}} \cdot \llcorner \overline{\text{sgn}}(a) \lrcorner_\lambda \\ \text{(بخش قبل)} \quad &= \llcorner \overline{\text{sgn}}(\overline{\text{sgn}}(a)) \lrcorner_\lambda \\ &= \llcorner \text{sgn}(a) \lrcorner_\lambda . \end{aligned}$$

□

تعریف ۶.۴. *λـعبارت K_k را λـkـتایی‌ـساز*[۴] *می‌نامیم و به صورت زیر تعریف می‌کنیم.*

$$K_k = \begin{cases} \lambda x_1 : \lambda x_2 : \lambda c : \lambda t : \big((T_{\overline{sgn}} \cdot c \cdot x_1) \cdot (T_{sgn} \cdot c \cdot x_2 \cdot t)\big) & k = 2 \\ \lambda x_1 : \cdots : \lambda x_k : (K_2 \cdot \big(K_{k-1} \cdot x_1 \cdots x_{k-1}) \cdot x_k\big) & otherwise. \end{cases}$$

برای بازیابی عبارات X_i از درون این عبارت، λـرشته C_i^k را چنان تعریف می‌کنیم که $K_{(X_1,\ldots,X_k)} \cdot C_i^k$ با X_i هم‌ارز باشد.

توجه کنیم، هنگامی که می‌گوییم $C = X \cdot Y$ یک λـرشته است، منظور این است که اگر K را با یک علامت «$\cdot$» (عملگر اعمال) به C بچسبانیم، یعنی $K \cdot C$، به $K \cdot X \cdot Y$ می‌رسیم که طبق قرارداد به صورت $(K \cdot X) \cdot Y$ تعبیر می‌شود.

در حالی که اگر $C = X \cdot Y$ را به عنوان یک λـعبارت، به K بچسبانیم، یعنی $K \cdot C$، به عبارت $K \cdot (X \cdot Y)$ می‌رسیم؛ چرا که C با $(X \cdot Y)$ جایگزین شده است.

[4] λ-k-tuple-generator

تعریف ۷.۴ λـرشته C_i^k را λـافکنش i از k[5] می‌نامیم و به صورت زیر تعریف می‌کنیم.

$$C_i^k = \begin{cases} \llcorner 0 \lrcorner_\lambda & i = 1 \wedge k = 2 \\ \llcorner 1 \lrcorner_\lambda & i = k \\ \llcorner 0 \lrcorner_\lambda \cdot C_i^{k-1} & otherwise. \end{cases}$$

لم ۸.۴ *برای هر $k > 1$، $1 \leq i \leq k$ و λـعبارت‌های $X_1, \ldots, X_k$ داریم*

$$K_{(X_1,\ldots,X_k)} \cdot C_i^k = X_i.$$

اثبات. ابتدا توجه کنیم $\lambda t : (A \cdot t) = A$. چرا که هر دو عبارت به عنوان تابع روی هر ورودی مانند x یکسان عمل می‌کنند. (فرض می‌کنیم اصل موضوع گسترش برای توابع حساب لاندا برقرار می‌باشد.)

$$(\lambda t : (A \cdot t)).x = (A \cdot x).$$

حال لم را به کمک استقرا و با حالت‌بندی حل می‌کنیم.

فرض کنیم $i = 1$ و $k = 2$، در این صورت

$$\begin{aligned} K_{(X_1,X_2)} \cdot C_1^2 &= K_2 \cdot X_1 \cdot X_2 \cdot \llcorner 0 \lrcorner_\lambda \\ &= (\lambda x_1 \cdot \lambda x_2 \cdot \lambda c \cdot \lambda t \cdot (T_{\overline{\text{sgn}}} \cdot c \cdot x_1) \cdot ((T_{\text{sgn}} \cdot c \cdot x_2) \cdot t)) \cdot \\ &\quad X_1 \cdot X_2 \cdot \llcorner 0 \lrcorner_\lambda \\ &= \lambda t \cdot ((T_{\overline{\text{sgn}}} \cdot \llcorner 0 \lrcorner_\lambda \cdot X_1) \cdot (T_{\text{sgn}} \cdot \llcorner 0 \lrcorner_\lambda \cdot X_2 \cdot t)) \\ &= \lambda t \cdot ((\llcorner 1 \lrcorner_\lambda \cdot X_1) \cdot (\llcorner 0 \lrcorner_\lambda \cdot X_2 \cdot t)) \\ &= \lambda t \cdot ((\llcorner 1 \lrcorner_\lambda \cdot X_1) \cdot t) \\ &= \lambda t \cdot (X_1 \cdot t) \\ &= X_1. \end{aligned}$$

برای حالت $i = 2$ و $k = 2$ داریم

$$\begin{aligned} K_{(X_1,X_2)} \cdot C_2^2 &= K_2 \cdot X_1 \cdot X_2 \cdot \llcorner 1 \lrcorner_\lambda \\ &= (\lambda x_1 : \lambda x_2 : \lambda c : \lambda t : (T_{\overline{\text{sgn}}} \cdot c \cdot x_1) \cdot ((T_{\text{sgn}} \cdot c \cdot x_2) \cdot t)) \cdot \\ &\quad X_1 \cdot X_2 \cdot \llcorner 1 \lrcorner_\lambda \\ &= \lambda t \cdot ((T_{\overline{\text{sgn}}} \cdot \llcorner 1 \lrcorner_\lambda \cdot X_1) \cdot (T_{\text{sgn}} \cdot \llcorner 1 \lrcorner_\lambda \cdot X_2 \cdot t)) \\ &= \lambda t \cdot ((\llcorner 0 \lrcorner_\lambda \cdot X_1) \cdot (\llcorner 1 \lrcorner_\lambda \cdot X_2 \cdot t)) \\ &= \lambda t \cdot ((\llcorner 1 \lrcorner_\lambda \cdot X_2) \cdot t) \\ &= \lambda t \cdot (X_2 \cdot t) \\ &= X_2. \end{aligned}$$

[5] λ-projection of i from k

برای حالت $i = k > 2$ داریم

$$
\begin{aligned}
K_{(X_1,\ldots,X_k)} \cdot C_k^k &= K_k \cdot X_1 \cdots X_k \cdot \llcorner 1 \lrcorner_\lambda \\
&= (\lambda x_1 : \cdots : \lambda x_k : (K_2 \cdot (K_{k-1} \cdot x_1 \cdots x_{k-1}) \cdot x_k)) \cdot \\
&\quad X_1 \cdots X_k \cdot \llcorner 1 \lrcorner_\lambda \\
&= (K_2 \cdot (K_{k-1} \cdot X_1 \cdots X_{k-1}) \cdot X_k) \cdot \llcorner 1 \lrcorner_\lambda \\
&= (K_2 \cdot (K_{k-1} \cdot X_1 \cdots X_{k-1}) \cdot X_k) \cdot C_2^2 \\
&= K_{(K_{k-1} \cdot X_1 \cdots X_{k-1}, X_k)} \cdot C_2^2 \\
\text{با توجه به حالت قبل} &= X_k.
\end{aligned}
$$

نهایتا برای حالت $i < k$ و $k > 2$ به کمک استقرا روی k داریم

$$
\begin{aligned}
K_{(X_1,\ldots,X_k)} \cdot C_i^k &= K_k \cdot X_1 \cdots X_k \cdot \llcorner 0 \lrcorner_\lambda \cdot C_i^{k-1} \\
&= (\lambda x_1 : \cdots : \lambda x_k : (K_2 \cdot (K_{k-1} \cdot x_1 \cdots x_{k-1}) \cdot x_k)) \cdot \\
&\quad X_1 \cdots X_k \cdot \llcorner 0 \lrcorner_\lambda \cdot C_i^{k-1} \\
&= (K_2 \cdot (K_{k-1} \cdot X_1 \cdots X_{k-1}) \cdot X_k) \cdot \llcorner 0 \lrcorner_\lambda \cdot C_i^{k-1} \\
&= (K_2 \cdot (K_{k-1} \cdot X_1 \cdots X_{k-1}) \cdot X_k) \cdot C_1^2 \cdot C_i^{k-1} \\
&= K_{(K_{k-1} \cdot X_1 \cdots X_{k-1}, X_k)} \cdot C_1^2 \cdot C_i^{k-1} \\
(\text{با توجه به حالت اول}) &= K_{k-1} \cdot X_1 \cdots X_{k-1} \cdot C_i^{k-1} \\
&= K_{(X_1,\ldots,X_{k-1})} \cdot C_i^{k-1} \\
(\text{فرض استقرا}) &= X_i.
\end{aligned}
$$

توجه کنیم پایه استقرا $k = 2$ می‌باشد که در حالات جداگانه بررسی کردیم. □

حال می‌توانیم به فرآیند ادامه دهیم و لم مهم سوم را ثابت کنیم، $\Lambda C_{\mathbb{N}}$ تحت بازگشت مقدماتی بسته می‌باشد.

لم ۹.۰.۴ *اگر $f^{(n)}$ و $g^{(n+2)}$ دو تابع ΛCـ محاسبه‌پذیر باشند $\eta_{f,g}$ نیز یک تابع ΛCـ محاسبه‌پذیر می‌باشد. به عبارت دیگر*

$$(f^{(n)} \in \Lambda C_{\mathbb{N}} \text{ و } g^{(n+2)} \in \Lambda C_{\mathbb{N}}) \Rightarrow \eta_{f,g} \in \Lambda C_{\mathbb{N}}.$$

اثبات. طبق فرض می‌دانیم عبارات محاسبه‌گر T_f و T_g برای توابع f و g وجود دارند. عبارت

محاسبه‌گر $T_{\eta_{f,g}}$ را به صورت زیر تعریف می‌کنیم.

$$\begin{array}{rcl} G & = & \lambda k : \big[K_{n+2} \cdot \\ & & \big(T_g \cdot (k \cdot C_1^{n+2}) \cdots (k \cdot C_{n+2}^{n+2})\big) \cdot \\ & & (k \cdot C_2^{n+2}) \cdots (k \cdot C_{n+1}^{n+2}) \cdot \\ & & (T_S \cdot (k \cdot C_{n+2}^{n+2}))\big]; \\ H & = & \lambda x_1 : \cdots : \lambda x_n : \lambda y : \\ & & (y \cdot G \cdot \big[K_{n+2} \cdot (T_f \cdot x_1 \cdots x_n) \cdot x_1 \cdots x_n \cdot \llcorner 0 \lrcorner_\lambda\big]); \\ T_{\eta_{f,g}} & = & \lambda x_1 : \cdots : \lambda x_n : \lambda y : ((H \cdot x_1 \cdots x_n \cdot y) \cdot C_1^{n+2}). \end{array}$$

ادعا می‌کنیم

$$\begin{array}{rl} & \llcorner t \lrcorner_\lambda \cdot G \cdot \big[K_{n+2} \cdot (T_f \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda) \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 0 \lrcorner_\lambda\big] \\ = & K_{(\llcorner \eta_{f,g}(\bar{a},t) \lrcorner_\lambda, \llcorner a_1 \lrcorner_\lambda, \ldots, \llcorner a_n \lrcorner_\lambda, \llcorner t \lrcorner_\lambda)}. \end{array}$$

فرض کنیم ادعا درست باشد. در این صورت داریم

$$\begin{array}{rcl} & & T_{\eta_{f,g}} \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t \lrcorner_\lambda \\ & = & (H \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t \lrcorner_\lambda) \cdot C_1^{n+2} \\ & = & (\llcorner t \lrcorner_\lambda \cdot G \cdot \\ & & \big[K_{n+2} \cdot (T_f \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda) \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 0 \lrcorner_\lambda\big] \\ & &) \cdot C_1^{n+2} \\ (\text{طبق ادعا}) & = & K_{(\llcorner \eta_{f,g}(\bar{a},t) \lrcorner_\lambda, \llcorner a_1 \lrcorner_\lambda, \ldots, \llcorner a_n \lrcorner_\lambda, \llcorner t \lrcorner_\lambda)} \cdot C_1^{n+2} \\ & = & \llcorner \eta_{f,g}(\bar{a},t) \lrcorner_\lambda. \end{array}$$

حال کافی است ادعا را اثبات کنیم. اثبات را از طریق استقرا روی t انجام می‌دهیم.
پایه. $(t = 0)$

$$\begin{array}{rl} & \llcorner 0 \lrcorner_\lambda \cdot G \cdot \big[K_{n+2} \cdot (T_f \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda) \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 0 \lrcorner_\lambda\big] \\ = & K_{n+2} \cdot (T_f \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda) \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 0 \lrcorner_\lambda \\ = & K_{n+2} \cdot \llcorner f(\bar{a}) \lrcorner_\lambda \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 0 \lrcorner_\lambda \\ = & K_{n+2} \cdot \llcorner \eta_{f,g}(\bar{a},0) \lrcorner_\lambda \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 0 \lrcorner_\lambda \\ = & K_{(\llcorner \eta_{f,g}(\bar{a},0) \lrcorner_\lambda, \llcorner a_1 \lrcorner_\lambda, \ldots, \llcorner a_n \lrcorner_\lambda, \llcorner 0 \lrcorner_\lambda)}. \end{array}$$

گام. طبق فرض استقرا داریم

$$\begin{array}{rl} & \llcorner t-1 \lrcorner_\lambda \cdot G \cdot \big[K_{n+2} \cdot (T_f \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda) \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 0 \lrcorner_\lambda\big] \\ = & K_{(\llcorner \eta_{f,g}(\bar{a},t-1) \lrcorner_\lambda, \llcorner a_1 \lrcorner_\lambda, \ldots, \llcorner a_n \lrcorner_\lambda, \llcorner t-1 \lrcorner_\lambda)}. \end{array}$$

لذا

$$\begin{aligned}
& \llcorner t\lrcorner_\lambda \cdot G \cdot \left[K_{n+2} \cdot (T_f \cdot \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda) \cdot \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda \cdot \llcorner 0\lrcorner_\lambda\right] \\
= \; & \left[\lambda\mathfrak{f} : \lambda\mathfrak{t} : (\mathfrak{f} \cdot (\llcorner t-1\lrcorner_\lambda \cdot \mathfrak{f} \cdot \mathfrak{t}))\right] \cdot G \cdot \\
& \left[K_{n+2} \cdot (T_f \cdot \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda) \cdot \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda \cdot \llcorner 0\lrcorner_\lambda\right] \\
= \; & G \cdot (\llcorner t-1\lrcorner_\lambda \cdot G \cdot \\
& \left[K_{n+2} \cdot (T_f \cdot \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda) \cdot \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda \cdot \llcorner 0\lrcorner_\lambda\right]) \\
\text{(ف.ا.)} \quad = \; & G \cdot (K_{(\llcorner \eta_{f,g}(\bar{a},t-1)\lrcorner_\lambda, \llcorner a_1\lrcorner_\lambda, \ldots, \llcorner a_n\lrcorner_\lambda, \llcorner t-1\lrcorner_\lambda)}) \\
= \; & K_{n+2} \cdot \left[T_g \cdot \llcorner \eta_{f,g}(\bar{a},t-1)\lrcorner_\lambda \cdot \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda \cdot \llcorner t-1\lrcorner_\lambda\right] \cdot \\
& \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda \cdot \left[T_S \cdot \llcorner t-1\lrcorner_\lambda\right] \\
= \; & K_{n+2} \cdot \left[\llcorner g(\eta_{f,g}(\bar{a},t-1), \bar{a}, t-1)\lrcorner_\lambda\right] \cdot \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda \cdot \left[\llcorner t\lrcorner_\lambda\right] \\
= \; & K_{n+2} \cdot \llcorner \eta_{f,g}(\bar{a},t)\lrcorner_\lambda \cdot \llcorner a_1\lrcorner_\lambda \cdots \llcorner a_n\lrcorner_\lambda \cdot \llcorner t\lrcorner_\lambda \\
= \; & K_{(\llcorner \eta_{f,g}(\bar{a},t)\lrcorner_\lambda, \llcorner a_1\lrcorner_\lambda, \ldots, \llcorner a_n\lrcorner_\lambda, \llcorner t\lrcorner_\lambda)}.
\end{aligned}$$

□

تنها عملیات باقی‌مانده کمینه‌سازی نامحدود می‌باشد، اما می‌دانیم این عملیات است که توابع جزئی را به مجموعه $\mathrm{GK}_{\mathbb{N}}$ اضافه می‌کند. بدون کمینه‌سازی نامحدود، تنها توابع بازگشتی‌ـ مقدماتی را خواهیم داشت که می‌دانیم همگی تام می‌باشند. برای محاسبه این توابع در حساب لاندا، نیاز به عبارتی با طبیعت حلقه نامتناهی داریم، که این عبارت را در اثبات آخرین لم معرفی خواهیم کرد. اما پیش از آن، یک لم ساده را معرفی و اثبات می‌کنیم.

لم ۱۰.۴. *برای هر عدد دلخواه ناصفر a داریم*

$$K_{(X_1,X_2)} \cdot \llcorner a\lrcorner_\lambda = X_2.$$

اثبات. مشابه اثبات لم ۸.۴ داریم

$$\begin{aligned}
K_{(X_1,X_2)} \cdot \llcorner a\lrcorner_\lambda & = K_2 \cdot X_1 \cdot X_2 \cdot \llcorner a\lrcorner_\lambda \\
& = (\lambda x_1 : \lambda x_2 : \lambda c : \lambda t : (T_{\overline{\mathrm{sgn}}} \cdot c \cdot x_1) \cdot ((T_{\mathrm{sgn}} \cdot c \cdot x_2) \cdot t))) \cdot \\
& \quad X_1 \cdot X_2 \cdot \llcorner a\lrcorner_\lambda \\
& = \lambda t \cdot ((T_{\overline{\mathrm{sgn}}} \cdot \llcorner a\lrcorner_\lambda \cdot X_1) \cdot (T_{\mathrm{sgn}} \cdot \llcorner a\lrcorner_\lambda \cdot X_2 \cdot t)) \\
(a>0) \quad & = \lambda t \cdot ((\llcorner 0\lrcorner_\lambda \cdot X_1) \cdot (\llcorner 1\lrcorner_\lambda \cdot X_2 \cdot t)) \\
& = \lambda t \cdot ((\llcorner 1\lrcorner_\lambda \cdot X_2) \cdot t) \\
& = \lambda t \cdot (X_2 \cdot t) \\
& = X_2.
\end{aligned}$$

□

در نهایت، می‌توانیم چهارمین و آخرین لم مهم را ثابت کنیم. $\Lambda\mathrm{C}_{\mathbb{N}}$ تحت کمینه‌سازی نامحدود بسته است.

لم ۱۱.۴. *اگر $f^{(n+1)}$ یک تابع ΛCـ محاسبه‌پذیر باشد μ_f نیز یک تابع ΛCـ محاسبه‌پذیر می‌باشد. به عبارت دیگر*

$$f^{(n+1)} \in \Lambda C_{\mathbb{N}} \Rightarrow \mu_f \in \Lambda C_{\mathbb{N}}.$$

اثبات. طبق فرض می‌دانیم عبارت محاسبه‌گر T_f برای تابغ f وجود دارد. عبارت محاسبه‌گر T_{μ_f} را به صورت زیر تعریف می‌کنیم.

$$\begin{array}{rcl} F & = & \lambda g : \lambda x_1 : \cdots : \lambda x_n : \lambda y : \\ & & (K_2 \cdot y \cdot (g \cdot x_1 \cdots x_n \cdot (T_S \cdot y)) \cdot (F \cdot x_1 \cdots x_n \cdot y)); \\ G & = & (\lambda x : (F \cdot (x \cdot x))) \cdot (\lambda x : (F \cdot (x \cdot x))); \\ T_{\mu_f} & = & \lambda x_1 : \cdots : \lambda x_n : (G \cdot x_1 \cdots x_n \cdot \llcorner 0 \lrcorner_\lambda). \end{array}$$

توجه کنیم $G = F \cdot G$، که همان حلقه نامتناهی است که به دنبال آن بودیم.

$$\begin{array}{rcl} G & = & (\lambda x : (F \cdot (x \cdot x))) \cdot (\lambda x : (F \cdot (x \cdot x))) \\ & = & F \cdot [(\lambda x : (F \cdot (x \cdot x))) \cdot (\lambda x : (F \cdot (x \cdot x)))] \\ & = & F \cdot G. \end{array}$$

حال ادعا می‌کنیم برای هر $a_1, \ldots, a_n, t$ داریم

$$G \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t \lrcorner_\lambda = \begin{cases} \llcorner t \lrcorner_\lambda & f(\bar{a}, t) = 0 \\ G \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t+1 \lrcorner_\lambda & \text{otherwise.} \end{cases}$$

برای اثبات به سادگی داریم

$$\begin{array}{rl} & G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t \lrcorner_\lambda \\ = & F \cdot G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t \lrcorner_\lambda \\ = & K_2 \cdot \llcorner t \lrcorner_\lambda \cdot (G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot (T_S \cdot \llcorner t \lrcorner_\lambda)) \cdot \\ & (T_f \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t \lrcorner_\lambda) \\ = & K_2 \cdot \llcorner t \lrcorner_\lambda \cdot (G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t+1 \lrcorner_\lambda) \cdot \\ & (T_f \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t \lrcorner_\lambda) \\ = & K_{(\llcorner t \lrcorner_\lambda, G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t+1 \lrcorner_\lambda)} \cdot \llcorner f(\bar{a}, t) \lrcorner_\lambda \\ = & \begin{cases} \llcorner t \lrcorner_\lambda & f(\bar{a}, t) = 0 \\ G \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \cdot \llcorner t+1 \lrcorner_\lambda & \text{otherwise. (طبق لم ۱۰.۴)} \end{cases} \end{array}$$

نهایتا نشان می‌دهیم T_{μ_f} یک عبارت محاسبه‌گر برای μ_f می‌باشد. دو حالت وجود دارد که این دو حالت را جداگانه بررسی می‌کنیم.

در حالت اول فرض کنیم $\mu_f(\bar{a}) = m$. پس برای هر $t < m$، داریم $f(\bar{a}, t) \neq 0$ در حالی که $f(\bar{a}, m) = 0$. (در نوشتار زیر فرض شده $m > 2$ اما واضح است که چنین فرضی ضروری نیست.)

$$
\begin{array}{rcl}
T_{\mu_f} \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda & = & (\lambda x_1 : \cdots : \lambda x_n : (G \cdot x_1 \cdots x_n \cdot \llcorner 0 \lrcorner_\lambda)) \cdot \\
& & \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \\
& = & G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 0 \lrcorner_\lambda \\
(0 < m \Rightarrow f(\bar{a}, 0) \neq 0) & = & G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 1 \lrcorner_\lambda \\
(1 < m \Rightarrow f(\bar{a}, 1) \neq 0) & = & G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 2 \lrcorner_\lambda \\
& \vdots & \\
(m-1 < m \Rightarrow f(\bar{a}, m-1) \neq 0) & = & G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner m \lrcorner_\lambda \\
(f(\bar{a}, m) = 0) & = & \llcorner m \lrcorner_\lambda \\
& = & \llcorner \mu_f(\bar{a}) \lrcorner_\lambda .
\end{array}
$$

در حالت دوم فرض کنیم $\mu_f(\bar{a})$ تعریف‌نشده است. پس برای هر $t \in \mathbb{N}$، داریم $f(\bar{a}, t) \neq 0$.

$$
\begin{array}{rcl}
T_{\mu_f} \cdot \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda & = & (\lambda x_1 : \cdots : \lambda x_n : (G \cdot x_1 \cdots x_n \cdot \llcorner 0 \lrcorner_\lambda)) \cdot \\
& & \llcorner a_1 \lrcorner_\lambda \cdots \llcorner a_n \lrcorner_\lambda \\
& = & G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 0 \lrcorner_\lambda \\
(f(\bar{a}, 0) \neq 0) & = & G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 1 \lrcorner_\lambda \\
(f(\bar{a}, 1) \neq 0) & = & G \cdot \llcorner a_1 \lrcorner_\lambda \ldots \llcorner a_n \lrcorner_\lambda \cdot \llcorner 2 \lrcorner_\lambda \\
& \vdots &
\end{array}
$$

و محاسبه هرگز متوقف نمی‌شود. □

قضیه ۱۲.۴. $GK_{\mathbb{N}} \subseteq \Lambda C_{\mathbb{N}}$.

اثبات. طبق لم‌های ۲.۴، ۳.۴، ۹.۴ و ۱۱.۴، مجموعه $\Lambda \mathrm{C}_{\mathbb{N}}$ هر چهار شرط مذکور در تعریف ۵.۲ را ارضا می‌کند. لذا از آنجا که $\mathrm{GK}_{\mathbb{N}}$ کوچک‌ترین مجموعه با این خواص می‌باشد، پس $\mathrm{GK}_{\mathbb{N}} \subseteq \Lambda \mathrm{C}_{\mathbb{N}}$. □

فصل ۵

گودل- کلینی در مقایسه با حساب لاندا

در این بخش نشان می‌دهیم هر تابع ΛC ـ محاسبه‌پذیر $f : \mathrm{TERM}_\alpha^n \to \mathrm{TERM}_\alpha$ یک تابع GK ـ محاسبه‌پذیر می‌باشد.

می‌دانیم توابع ΛC ـ محاسبه‌پذیر روی TERM_α تعریف می‌شوند و نه TERM_λ. اما تابع $\mathbb{N}$ ـ تبدیل یک عبارت در TERM_λ را به عنوان ورودی دریافت می‌کند، لذا بسته به این که چه عضوی از یک کلاس α ـ هم‌ارزی را انتخاب می‌کنیم ممکن است عدد طبیعی متفاوتی بگیریم و تابع نتیجه متفاوتی بدهد. اما نشان می‌دهیم تمام نتایج با یک‌دیگر α ـ هم‌ارز می‌باشند.

می‌دانیم توابع تبدیل یک k ـ تایی از اعداد طبیعی به یک عدد طبیعی و معکوس آن، توابع ساده حسابی، توابع تعریف‌شده بر اساس چند حالت، و سایر عملگرهای ساده، همگی GK ـ محاسبه‌پذیر می‌باشند.

ابتدا تابع جایگذاری $\rho : \mathbb{N}^3 \to \mathbb{N}$ را طوری تعریف می‌کنیم که

$$\rho(\lfloor A \rfloor_{\mathbb{N}}, \lfloor x \rfloor_{\mathbb{N}}, \lfloor B \rfloor_{\mathbb{N}}) = \lfloor A[x \mapsto B] \rfloor_{\mathbb{N}}.$$

توجه کنیم هنگامی که $n = \langle a, b, c \rangle$، می‌گوییم $l(n) = a$، $c(n) = b$ و $r(n) = c$.

تعریف ۱.۵. *تابع ρ را به صورت زیر تعریف می‌کنیم.*

$$\rho(a, v, b) = \begin{cases} b & l(a) > 1 \wedge a = v \\ a & l(a) > 1 \wedge a \neq v \\ \langle 0, \rho(c(a), v, b), \rho(r(a), v, b) \rangle & l(a) = 0 \\ a & l(a) = 1 \wedge c(a) = v \\ \langle 1, c(a), \rho(r(a), v, b) \rangle & l(a) = 1 \wedge c(a) \neq v. \end{cases}$$

این تابع به وضوح عملکر مورد انتظار را دارد. حال می‌توانیم گام محاسباتی β را روی اعداد طبیعی تعریف کنیم.

تعریف ۲.۵. *تابع $\beta_{\mathbb{N}}$ را به صورت زیر تعریف می‌کنیم.*

$$\beta_{\mathbb{N}}(a) = \begin{cases} \lfloor\Omega\rfloor_{\mathbb{N}} & l(a) > 1 \\ \langle 0, c(a), \beta_{\mathbb{N}}(r(a))\rangle & l(a) = 0 \wedge l(c(a)) > 1 \wedge \beta_{\mathbb{N}}(r(a)) \neq \lfloor\Omega\rfloor_{\mathbb{N}} \\ \lfloor\Omega\rfloor_{\mathbb{N}} & l(a) = 0 \wedge l(c(a)) > 1 \wedge \beta_{\mathbb{N}}(r(a)) = \lfloor\Omega\rfloor_{\mathbb{N}} \\ \langle 0, \beta_{\mathbb{N}}(c(a)), r(a)\rangle & l(a) = 0 \wedge l(c(a)) = 0 \wedge \beta_{\mathbb{N}}(c(a)) \neq \lfloor\Omega\rfloor_{\mathbb{N}} \\ \langle 0, c(a), \beta_{\mathbb{N}}(r(a))\rangle & l(a) = 0 \wedge l(c(a)) = 0 \wedge \\ & \beta_{\mathbb{N}}(c(a)) = \lfloor\Omega\rfloor_{\mathbb{N}} \wedge \beta_{\mathbb{N}}(r(a)) \neq \lfloor\Omega\rfloor_{\mathbb{N}} \\ \lfloor\Omega\rfloor_{\mathbb{N}} & l(a) = 0 \wedge l(c(a)) = 0 \wedge \\ & \beta_{\mathbb{N}}(c(a)) = \lfloor\Omega\rfloor_{\mathbb{N}} \wedge \beta_{\mathbb{N}}(r(a)) = \lfloor\Omega\rfloor_{\mathbb{N}} \\ \rho(r(c(a)), c(c(a)), r(a)) & l(a) = 0 \wedge l(c(a)) = 1 \\ \langle 1, c(a), \beta_{\mathbb{N}}(r(a))\rangle & l(a) = 1 \wedge \beta_{\mathbb{N}}(r(a)) \neq \lfloor\Omega\rfloor_{\mathbb{N}} \\ \lfloor\Omega\rfloor_{\mathbb{N}} & l(a) = 1 \wedge \beta_{\mathbb{N}}(r(a)) = \lfloor\Omega\rfloor_{\mathbb{N}}. \end{cases}$$

باز هم واضح است که این تعریف انتظار ما را برآورده می‌کند، به عبارت دیگر

$$[\lceil\beta_{\mathbb{N}}(\lfloor A\rfloor_{\mathbb{N}})\rceil_{\mathbb{N}}]_{\alpha} = \beta(A).$$

تعریف ۳.۵. *برای تابع GKـ محاسبه‌پذیر f، تابع GKـ محاسبه‌پذیر $\iota_f(a,k)$ را برابر $f^k(a)$ تعریف می‌کنیم.*

$$\iota_f(k,a) = \begin{cases} a & k = 0 \\ f(\iota_f(a, k-1)) & k > 0. \end{cases}$$

همچنین $\iota_{\beta_{\mathbb{N}}}$ را با γ نمایش می‌دهیم.

حال می‌توانیم قضیه اصلی این بخش را بیان کنیم.

قضیه ۴.۵. $\Lambda C_\lambda \subseteq GK_\lambda$.

اثبات. فرض کنیم $f : \mathrm{TERM}_{\alpha}^{n} \to \mathrm{TERM}_{\alpha}$ یک تابع $\Lambda\mathrm{C}$ـ محاسبه‌پذیر باشد. باید یک تابع GKـ محاسبه‌پذیر مانند $f_{\mathbb{N}} : \mathbb{N}^n \to \mathbb{N}$ معرفی کنیم به طوری که برای هر $T_1, \ldots, T_n \in \mathrm{TERM}_{\lambda}$، دشته باشیم

$$f([T_1]_{\alpha}, \ldots, [T_n]_{\alpha}) = [\lceil f_{\mathbb{N}}(\lfloor T_1\rfloor_{\mathbb{N}}, \ldots, \lfloor T_n\rfloor_{\mathbb{N}})\rceil_{\mathbb{N}}]_{\alpha}.$$

از آنجا که f ΛC ـ محاسبه‌پذیر است، پس عبارت محاسبه‌گری مانند $T_f \in \mathrm{TERM}_\lambda$ وجود دارد که $f = \Phi_T^{(n)}$. تابع $f_{\mathbb{N}}$ را به صورت زیر تعریف می‌کنیم.

$$\begin{array}{rcl} \sigma(\lfloor T \rfloor_{\mathbb{N}}) & = & \min_t(\gamma(\lfloor T \rfloor_{\mathbb{N}}, t+1) = \lfloor \Omega \rfloor_{\mathbb{N}}); \\ f_{\mathbb{N}}(\lfloor T_1 \rfloor_{\mathbb{N}}, \ldots, \lfloor T_n \rfloor_{\mathbb{N}}) & = & \gamma(\lfloor T \cdot T_1 \cdots T_n \rfloor_{\mathbb{N}}, \sigma(\lfloor T \cdot T_1 \cdots T_n \rfloor_{\mathbb{N}})). \end{array}$$

فرض کنیم $T_1, \ldots, T_n \in \mathrm{TERM}_\lambda$ دلخواه باشند.

$$\begin{array}{rcl} [\lceil f_{\mathbb{N}}(\lfloor T_1 \rfloor_{\mathbb{N}}, \ldots, \lfloor T_n \rfloor_{\mathbb{N}}) \rceil_{\mathbb{N}}]_\alpha & = & [\lceil \gamma(\lfloor T \cdot T_1 \cdots T_n \rfloor_{\mathbb{N}}, \sigma(\lfloor T \cdot T_1 \cdots T_n \rfloor_{\mathbb{N}})) \rceil_{\mathbb{N}}]_\alpha \\ & = & [\lceil \beta_{\mathbb{N}}^{\sigma(\lfloor T \cdot T_1 \cdots T_n \rfloor_{\mathbb{N}})}(\lfloor T \cdot T_1 \cdots T_n \rfloor_{\mathbb{N}}) \rceil_{\mathbb{N}}]_\alpha \\ & = & \beta^{\sigma(\lfloor T \cdot T_1 \cdots T_n \rfloor_{\mathbb{N}})}(T \cdot T_1 \cdots T_n) \\ & = & \Phi_T^{(n)}([T_1]_\alpha, \ldots, [T_n]_\alpha) \\ & = & f([T_1]_\alpha, \ldots, [T_n]_\alpha). \end{array}$$

$\square$

فصل ۶

نتیجه‌گیری

در قضیه ۱۲.۴ نشان دادیم هر تابع GK ـ محاسبه‌پذیر یک تابع ΛC ـ محاسبه‌پذیر می‌باشد و در قضیه ۴.۵ نشان دادیم که هر تابع ΛC ـ محاسبه‌پذیر نیز یک تابع GK ـ محاسبه‌پذیر می‌باشد. لذا در نهایت پس از اثبات این دو قضیه می‌توانیم نتیجه بگیریم که دو مدل محاسباتی حساب لاندا و گودل ـ کلینی به لحاظ توان محاسباتی با یکدیگر معادل می‌باشند. با توجه به این موضوع، با نشان دادن معادل بودن هر یک از این مدل‌های محاسباتی با ماشین تورینگ، معادل بودن مدل دیگر نیز ثابت می‌شود. قضیه زیر نتیجه اصلی این پروژه را بیان می‌کند.

قضیه ۱.۶. *دو مدل محاسباتی حساب لاندا و گودل ـ کلینی به لحاظ توان محاسباتی با یکدیگر معادل می‌باشند.*

اثبات. با توجه به قضایای ۱۲.۴ و ۴.۵ واضح است. □

واژه‌نامه فارسی به انگلیسی

ا

اصل موضوع گسترش .... Axiom of Extensionality
اعمال .... Application
λـ افکنش i از k .... λ-projection of i from k
اکیدا $\sim$ـ هم‌ارز .... Strictly $\sim$-equivalent
انتزاع .... Abstraction

ب

بازگشت مقدماتی .... Primitive Recursion

ت

λـ تبدیل .... λ-coding
ترکیب .... Composition

ج

جایگزین‌شده x با B در A .... A where x is replaced by B

ح

حساب لاندا .... Lambda Calculus

ص

ع

ک

گ

م

و

ه

K

N

مراجع

[۱] H.P. Barendregt, *The Lambda Calculus: Its Syntax and Semantics*, North-Holland, 1984

[۲] Kurt Gödel, *On Undecidable Propositions of Formal Mathematical Systems*, Princeton University, 1934

[۳] Stephen Kleene, Barkley Rosser, *The Inconsistency of Certain Formal Logics*, The Annals of Mathematics, 1942

[۴] Stephen Kleene, *Introduction to Metamathematics*, North-Holland, 1971

Abstract

In this project, we introduce two models of computation, known as λ-Calculus and Gödel-Kleene, and introduce two coding functions to convert problems (functions) from one to another, after that, we prove these two models are equivalent, i.e. the λ-Calculus is as capable as Gödel-Kleene and vice versa.

College of Science
School of Mathematics, Statistics, and Computer Science

λ-Calculus & Gödel-Kleene Models of Computation

Kamyar Mirzavaziri

Supervisor: Dr. Mojtaba Mojtahedi

A thesis submitted to Graduate Studies Office
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

2022