نشریه علمی- ترویجی محاسبات نرم
شمارهٔ دوم، پاییز و زمستان ۱۳۹۱، صفحه ۱۴-۲۵

توسعهٔ روش SL با ترتیب KBO برای اثبات خودکار پایان‌پذیری سیستم بازنویسی ترم

محمد کدخدا[۱،۲]، سعید جلیلی[۳]، محمد ایزدی[٤]
[۱] دانشکده علوم ریاضی، دانشگاه تربیت مدرس، تهران، ایران
[۲] کارشناسی ارشد، گروه پژوهشی ریاضی و انفورماتیک، جهاد دانشگاهی واحد تربیت مدرس، تهران، ایران
kadkhoda@chmail.com
[۳] دانشیار، دانشکده برق و کامپیوتر، دانشگاه تربیت مدرس، تهران، ایران
sjalili@modares.ac.ir
[٤] استادیار، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران
izadi@ce.sharif.ir

چکیده: سیستم بازنویسی ترم (TRS) مدلی انتزاعی از زبان‌های تابعی ارائه می‌دهد. اثبات پایان‌پذیری یک TRS برای تأیید درستی عملکرد زبان‌های تابعی ضروری است. روش برچسب‌گذاری معنایی (SL) روشی کامل برای اثبات پایان‌پذیری به شمار می‌رود. بخش معنایی این روش توسط یک شبه- مدل از تفسیر تابع‌نشانه‌ها ایجاد می‌شود. بیشتر توان روش SL به استفاده از مدل‌های نامتناهی مربوط می‌شود که ارائهٔ آن‌ها در ابزارهای آزمون خودکار پایان‌پذیری دشوار است. در این مقاله، روش SL با دامنهٔ تفسیر اعداد طبیعی به شکلی با ترتیب کُنت- بندیکس (KBO) ترکیب شده تا بتوان اثبات پایان‌پذیری با مدل‌های نامتناهی را به طور خودکار انجام داد. ابتدا تعمیمی از KBO به نام ترتیب کُنت- بندیکس برچسب‌گذاری (ℓKBO) ارائه، سپس توانایی آن را در اثبات پایان‌پذیری TRS نشان داده‌ایم. الگوریتم جستجوی خودکار یک ℓKBO برای یک TRS معرفی شده و عملکرد آن روی کتابخانهٔ TPDB 3.1 با موفقیت، مورد آزمون قرار گرفته است.

واژه‌های کلیدی: اثبات پایان‌پذیری، برچسب‌گذاری معنایی، ترتیب کُنت- بندیکس، سیستم بازنویسی ترم.

۱. مقدمه

پایان‌پذیری[1] مهم‌ترین بحث در نظریهٔ سیستم‌های بازنویسی ترم[2](TRS) است. یک سیستم پایان‌پذیر، درستی[3] یک برنامه و ایمن[4] بودن محاسبات آن را تضمین می‌کند. آغاز تحقیق برای اثبات پایان‌پذیری با ارائهٔ یک ترتیب کاهشی[5] سازگار با سیستم بازنویسی مورد بررسی ارتباط داشته است [۱ و ۲]. برای یک دهه ترتیب‌های متنوعی معرفی شدند؛ ترتیب کُنت‌ـ بندیکس[6] (KBO)[۳]، ترتیب مسیر بازگشتی (RPO) [۱، ٤ و ٥]، ترتیب مسیر قاموسی (LPO) [٦]، ترتیب تجزیه بازگشتی (RDO) [۷]، و ترتیب مسیر چندمجموعه‌ای (MPO) [۱ و ٦]. این ترتیب‌ها بر اساس شکل و ساختار صوری سیستم‌های بازنویسی ساخته می‌شوند. بررسی کامل خواص این دسته از ترتیب‌ها توسط درشوویتز در [٥] تحت عنوان ترتیب‌های ساده‌سازی[7] انجام شده است. داشتن ویژگی زیرترمی[8]، ایدهٔ مشترک تمام این ترتیب‌هاست که زمینهٔ ساخت ترتیب‌های کاهشی متعددی را فراهم کرده است [۳، ٦ و ۷]. جستجوی خودکار یک ترتیب، مهم‌ترین مزیت آن است و ترتیب‌های ساده‌سازی این قابلیت را دارند [۸، ۹ و ۱۰]. مشکل اصلی این روش‌ها محدودیت در اثبات پایان‌پذیری بسیاری از سیستم‌هاست [۱۱].

مثال ۱. سیستم بازنویسی $\mathcal{R}_1$ را با قوانین زیر در نظر بگیرید:

f(i(x))→ f(h(x))	i(a)→b
g(h(x))→ g(i(x))	h(a)→b

این TRS روی مجموعه تابع‌نشانه‌های {f, g, h, i, a, b} ساخته شده است. ترتیب‌های ساده‌سازی در اثبات پایان‌پذیری این سیستم شکست می‌خورند. دلیل این موضوع، نقض ویژگی زیرترمی در $\mathcal{R}_1$ است. تقدم هر ترتیب، ساده‌سازی i را قبل از h قرار می‌دهد و یا برعکس؛ لذا یکی از قوانین بازنویسی نامرتب باقی خواهد بود. □

1. Termination
2. Term rewriting systems
3. Sound
4. Safety
5. Reduction order
6. Knuth-Bendix order
7. Simplification orders
8. Subterm property

روش‌های تبدیلی[9] برای افزایش توان ترتیب‌های ساده‌سازی ایجاد شدند. در این روش‌ها ابتدا TRS اصلی به یک TRS دیگر تبدیل می‌شود و سپس آزمون پایان‌پذیری روی سیستم جدید انجام می‌گردد. از بین روش‌های تبدیلی، روش برچسب‌گذاری معنایی[10](SL) زانتِما [۱۲] و جفت‌های وابسته[11](DP) آرتز و گیزل [۱۳] کارآیی بهتری داشته‌اند.

بخش معنایی در روش SL با یک شبه‌ـ مدل تأمین می‌شود. استفاده از مدل نامتناهی، مانعی برای خودکارسازی SL به حساب می‌آید. برای اینکه بتوان هم از مدل نامتناهی بهره برد و هم امکان خودکارسازی را حفظ نمود، چند راه پیشنهاد شده است. هیروکاوا و میدل‌دروپ در [١٤] برچسب‌گذاری پیش‌گویانه را معرفی کردند. این روش با پالایش قوانین بازنویسی، بخشی از محدودیت‌های شبه‌ـ مدل را حذف می‌کند تا محدودیت‌های شبه‌ـ مدل تنها برای بخشی از قانون‌ها به نام قانون‌های قابل استفاده[12](UR) کاهش یابد. ایدهٔ دیگری که برای اجرایی کردن SL با مدل نامتناهی استفاده شده است، توسعهٔ بی‌نهایتی ترتیب‌های موجود مانند RPO، LPO و KBOدر ترکیب با SL برای ساخت یک شبه‌ـ مدل است. در [۱٥] این کار توسط کُپروسکی و زانتِما برای RPO انجام شد و موفقیت‌های چشمگیری داشت [۸ و ١٦]. در این مقاله، سعی شده است تا به شکل دیگری این کار برای KBO انجام شود.

ترتیب KBO از یک تابع وزن برای ارزش‌گذاری ترم‌ها و سپس مقایسهٔ آن‌ها استفاده می‌کند.در KBO تابع وزن ارتباط دهندهٔ ساختار و معنای ترم‌ها برای استخراج اطلاعات مربوط به پایان‌پذیری یک سیستم است. این موضوع، ترکیب آن با روشSL را که در آن تفسیر ارائه شده برای یک ترم معنای آن است، توجیه می‌کند. تعیین وزن برای ترم‌ها در KBO و تفسیر آن تحت یک مدل در SL، دو روش متفاوت برای معناسازی ترم‌هاست. در این مقاله، این دو روش را با هم ترکیب کرده‌ایم

9. Transformational
10. Semantic labeling
11. Dependency pair method
12. Usable rules

تا از این طریق بتوان استفاده از مدل‌های نامتناهی در روش SL برای اثبات خودکار پایان‌پذیری را حفظ کرد.

ادامهٔ مقاله به این صورت است: در بخش ۲ تعاریف کلی، عبارات و نمادهای مورد نیاز در رابطه با TRS، KBO و SL آمده است. در بخش ۳ یک تعمیم ترتیب کُنت- بندیکس معرفی شده است. بخش ٤ به الگوریتم جستجوی ترتیب معرفی شده پرداخته است. بخش ٥ مثال‌هایی را با استفاده از این روش مورد بررسی قرار داده و نتیجه‌گیری در بخش ٦ آمده است.

۲. تعاریف TRS، KBO و SL

اصطلاحات، علائم، و نشانه‌های مورد استفاده در TRS، KBO و SL تا حدودی استاندارد شده‌اند. در این مقاله از [۱۷] پیروی کرده‌ایم.

۲-۱. سیستم بازنویسیترم

اگر $\mathcal{F}$ مجموعه متناهی غیرتهی از تابع‌نشانه‌ها که هر $f\in\mathcal{F}$، دارای موضع $n \geq 0$ و $\mathcal{V}$ مجموعه‌ای شمارش‌پذیر از متغیرها باشد که $\mathcal{F}\cap\mathcal{V}\neq\emptyset$ در این صورت $\mathcal{T}(\mathcal{F}, \mathcal{V})$ مجموعه ترم‌های تابعی[1] ساخته شده روی $\mathcal{F}$ و $\mathcal{V}$ را تشکیل می‌دهد. ترم s را زیرترم t گویند اگر یک متن[2] C وجود داشته باشد که $C[s]= t$. منظور از متن C در اینجا ترمی است که ثابت ویژه که حفره نام دارد، در آن رخ داده باشد. مجموعه تمام موقعیت‌های یک ترم t را با $\mathcal{P}os(t)$ معرفی می‌کنیم. با $p_i=0$ ریشهٔ ترم را مشخص می‌کند. اگر $P\in\mathcal{P}os(t)$ با $t|_P$ به زیرترم موقعیت P اشاره می‌شود. مجموعه متغیرهای موجود در ترم t را با $\mathcal{V}ar(t)$ و تابع‌نشانه‌های آن را با $\mathcal{F}un(t)$ نشان می‌دهیم. اندازهٔ ترم t را با عدد $|t|$ که تعداد نمادهای ظاهر شده در آن است، مشخص می‌کنند. عدد $|t|_d$ تعداد تکرارهای نشانهٔ d است.

تعریف ۱. یک سیستم بازنویسی ترم $\mathcal{R}$ روی $\mathcal{T}(\mathcal{F}, \mathcal{V})$ زیرمجموعه‌ای از $\mathcal{T}(\mathcal{F},\mathcal{V})\times\mathcal{T}(\mathcal{F},\mathcal{V})$ است که در هر $(l,r)\in\mathcal{R}$، $l\notin\mathcal{V}$ و $\mathcal{V}ar(r)\subseteq\mathcal{V}ar(l)$ باشد. یک جایگذاری[3] σ تابع $\sigma: \mathcal{V}\to\mathcal{T}(\mathcal{F}, \mathcal{V})$ است که هر متغیر را به یک ترم می‌برد. □

تعریف ۲. برای هر سیستم بازنویسی $\mathcal{R}$، رابطهٔ بازنویسی $\to_{\mathcal{R}}$[4] روی $\mathcal{T}(\mathcal{F}, \mathcal{V})$ تعریف می‌شود:

$$t \to_{\mathcal{R}} s \Leftrightarrow \exists\, (l, r)\in\mathcal{R}\wedge\sigma\wedge\ C[\,]\,) \to \quad (۱)$$
$$s\equiv C[l\sigma] \wedge t \equiv C[r\sigma]$$

یعنی برای هر $(l, r)\in\mathcal{R}$ و هر جایگذاری σ، $l\sigma\to r\sigma$ ، و برای هر متن $f(\ldots,t_k ,\ \ldots)$ با تابع‌نشانهٔ n موضعی $f \in\mathcal{F}$، در صورتی که $t_k \to_R t'_k$ آنگاه $f(\ldots,t'_k ,\ldots)f(\ldots,t_k,\ldots)\to_R$ باشد. □

ترم $t\in\mathcal{T}(\mathcal{F}, \mathcal{V})$ را در یک سیستم بازنویسی $\mathcal{R}$ نرمال نامند، اگر ترم $t'\in\mathcal{T}(\mathcal{F}, \mathcal{V})$ که $t \to_{\mathcal{R}} t'$ وجود نداشته باشد. سیستم بازنویسی $\mathcal{R}$ نرمال ضعیف (WN) است، اگر برای هر ترم یک ترم نرمال وجود داشته باشد که پس از بازنویسی متناهی روی ترم اولیه ساخته شود. یک سیستم نرمال قوی (SN) است اگر دنبالهٔ نامتناهی از ترم‌ها $t_1 \to_R t_0 \ldots t_2 \to_R$ در آن قابل ساختن نباشد. یک سیستم SN دارای ویژگی شکل نرمال منحصربه‌فرد(UN) است، اگر برای هر ترم، تنها یک شکل نرمال وجود داشته باشد [٦]. ارتباط بین انواع سیستم‌های نرمال به صورت $UN \Rightarrow SN$ $\Rightarrow WN$ است. یک سیستم پایان‌پذیر است، اگر SN باشد.

۲-۲. ترتیب کُنت- بندیکس

ترتیب کُنت- بندیکس (KBO) اولین بار در [۱۸] معرفی و بعد از آن توانست در اثبات پایان‌پذیری TRS مورد استفاده قرار گیرد [۱۹ و ۲۰]. ساخت KBO استاندارد به کمک یک تابع وزن روی $\mathcal{T}(\mathcal{F}, \mathcal{V})$ و یک ترتیب جزئی $\succ$، روی تابع‌نشانه‌های (متناهی) $\mathcal{F}$، که تقدم[5] نام دارد، انجام می‌شود. دوتایی (W,w_0) را که $W: \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathbb{N}$، $\mathbb{N}$ مجموعه اعداد حسابی است، با کمترین وزن w_0 که $w_0> 0$، را یک تابع وزن برای $\mathcal{F}$ می‌نامند، اگر برای هر ثابت $c\in\mathcal{F}^{(0)}$، $W(c) \geq w_0$ باشد. برای هر $s\in\mathcal{T}(\mathcal{F}, \mathcal{V})$، $W(s)$ مجموع وزن نشانه‌های s است؛ اگر $s\in\mathcal{V}$ باشد، $W(s) = w_0$ و اگر

---

1. Functional terms
2. Context
3. Substitution
4. Rewrite relation
5. Precedence

$s=f(s_1,\ldots ,s_n)$ باشد، $W(s)=W(f)+\Sigma_{k=1..n}W(s_k)$ است. یک تابع وزن را قابل قبول[1] گویند، اگر برای یک $f\in\mathcal{F}^{(1)}$، $W(f)=0$ باشد، آنگاه برای هر $g\in\mathcal{F}-\{f\}$، $f \succ g$ باشد. به f نشان ویژه تابع وزن W گفته و آن را با i نمایش می‌دهند.

تعریف ۳. یک KBO که به صورت $\succ_{KBO}$ نمایش می‌دهیم، بر اساس تابع وزن قابل قبول (W,w_0)، و تقدم $\succ$، روی $\mathcal{T}(\mathcal{F}, \mathcal{V})$ به این صورت تعریف می‌شود:

برای دو ترم s و t داریم $s >_{KBO} t$ اگر برای هر $x\in\mathcal{V}ar(t)$ داشته باشیم، $|s|_x \geq |t|_x$ و سپس

۱. $W(t)>W(s)$ یا

۲. $W(t)=W(s)$، $s = f^a s'$، $t = f^b t'$ که $s' = g(s_1, \ldots , s_n)$ و $t' = h(t_1, \ldots , t_m)$ بوده و یکی از وضعیت‌های زیر برقرار باشد:

الف) $a > b$، یا

ب) $a=b$ و $g \succ h$، یا

ج) $a=b$ و $g\approx h$ و $(s_1 ,\ldots,s_n) >_{KBO} (t_1 ,\ldots,t_n)$ [۲۱].□

قضیهٔ زیر رابطهٔ KBO و پایان‌پذیری را نشان می‌دهد.

قضیهٔ ۱. ترتیب $\succ_{KBO}$ یک ترتیب کاهشی روی $(\mathcal{F}, \mathcal{V})$ است [۵]. □

۲-۳. روش برچسب‌گذاری معنایی

پایهٔ SL را مفهوم جبر یکنواخت[2] تشکیل می‌دهد. جفت $\mathcal{A}= (A, \mathcal{F}_A)$ را که A مجموعه ناتهی و $\mathcal{F}_A =\{f_A | f\in\mathcal{F}\}$ مجموعه‌ای از تابع‌های تعریف شده روی آن است، یک $\mathcal{F}$- جبر می‌نامند. برای $\mathcal{F}$- جبر $\mathcal{A}$، یک برچسب‌گذاری ℓ مجموعه‌ای از برچسب‌های $L_f\subseteq A$ برای هر $f\in\mathcal{F}$ و نگاشت $\ell_f: A^n\rightarrow L_f$ برای هر تابع‌نشانهٔ n موضعی $f\in\mathcal{F}$ با $L_f\neq\emptyset$ است. مجموعهٔ تابع‌نشانه‌های برچسب‌گذاری شده $\mathcal{F}_{lab}$ به صورت زیر است.

$$\mathcal{F}_{lab} = \{f_a | a \in L_f \wedge f \in\mathcal{F}\} \cup \{f \mid L_f= \emptyset \wedge f\in\mathcal{F}\} \quad (۲)$$

اگر $s=f(t_1,\ldots, t_n)$ یک ترم باشد، برچسب نماد ریشه‌ای s توسط نگاشت ℓ_f و بر اساس ارزش آرگومان‌های $t_1,\ldots, t_n$ معین می‌شود.

تعریف ٤. اگر $\mathcal{V}$ مجموعه‌ای از متغیرها باشد، برای هر مشخصه $\alpha: \mathcal{V}\rightarrow A$ نگاشت $lab_\alpha: \mathcal{T}(\mathcal{F}, \mathcal{V})\rightarrow\mathcal{T}(\mathcal{F}_{lab}, \mathcal{V})$ به شکل استقرایی زیر تعریف می‌شود:

$$lab_\alpha(t)=\begin{cases} t & \text{if } t \text{ is a variable,}\\ f(lab_\alpha(t_1), \ldots, lab_\alpha(t_n)) & \text{if } t=f(t_1, \ldots, t_n) \text{ and } L_f=\emptyset\\ f_a(lab_\alpha(t_1), \ldots, lab_\alpha(t_n)) & \text{if } t=f(t_1, \ldots, t_n) \text{ and } L_f\neq\emptyset\end{cases}$$

در اینجا a به مقدار برچسب $\ell_f([\![t_1]\!]_\alpha, \ldots, [\![t_n]\!]_\alpha)$ اشاره دارد. نگاشت lab_α می‌تواند روی هر سیستم بازنویسی $\mathcal{R}$ با مجموعه تابع‌نشانه‌های $\mathcal{F}$ عمل کرده و سیستم بازنویسی برچسب‌گذاری شدهٔ $\mathcal{R}_{lab}$ با تابع‌نشانه‌های $\mathcal{F}_{lab}$ را تولید کند. اگر تابع مشخصهٔ $\alpha:\mathcal{V}\rightarrow A$ در اختیار باشد، برای $\mathcal{R}_{lab}$ داریم:

$$\mathcal{R}_{lab} = \{ lab_\alpha(l)\rightarrow lab_\alpha(r) \mid l\rightarrow r \in\mathcal{R} \} \quad (۳)$$

تعریف ٥. اگر $\mathcal{A}= (A, \mathcal{F}_A)$ یک $\mathcal{F}$- جبر ناتهی، و $>$ یک ترتیب خوش‌ساخت روی A باشد، هر تابع‌نشانه $f\in\mathcal{F}$ با موضع $n\geq1$ تحت آن یکنواخت ضعیف است، اگر برای هر $a_1,\ldots, a_n,b\in A$ و $i\in\{1,\ldots,n\}$ که $a_i> b$:

$$f_A(a_1 ,\ldots,a_i ,\ldots,a_n) \geq f_A(a_1 ,\ldots,b ,\ldots,a_n)$$

باشد. □

رابطهٔ $\geq_A$ به کمک ترتیب $>$ روی $\mathcal{T}(\mathcal{F}, \mathcal{V})$ برای هر مشخصه α به این صورت تعریف می‌شود: $s \geq_A t$ اگر $[\![s]\!]_\alpha > [\![t]\!]_\alpha$. زمانی که داشته باشیم $\rightarrow_\mathcal{R}\subseteq\geq_A$، دوتایی $(\mathcal{A}, >)$ یک شبه- مدل برای $\mathcal{R}$ نامیده می‌شود. برچسب‌گذاری ℓ برای $\mathcal{A}$ یکنواخت ضعیف است، اگر هر تابع برچسب‌دار در همه مختصات خود یکنواخت ضعیف باشند. مهم‌ترین ویژگی سیستم $\mathcal{R}_{lab}$ این است که پایان‌پذیری آن با سیستم $\mathcal{R}$ معادل است [۱۲]. این موضوع زمانی برقرار خواهد بود که ℓ یک شبه- مدل برای $\mathcal{A}$ باشد.

۳. ترتیب KBO برچسب‌گذاری

مقدار وزن ترم t، $W(t)$، در KBO و مقدار تفسیر آن، t_A، در SL دو روش مختلف معناسازی ترم‌ها را برای شناسایی

1. Admissible
2. Monotonic algebra

اطلاعات مربوط به پایان‌پذیری سیستم $\mathcal{R}$ انجام می‌دهد. در این مقاله، یک روش ترکیبی با استفاده از هر دو تابع وزن و تفسیر برای وقتی که مدل استفاده شده نامتناهی باشد، معرفی می‌کنیم. ترکیب KBO و SL به طور طبیعی از طریق برقراری ارتباط بین تابع وزن و تابع تفسیر انجام می‌شود. ارتباط بین w(f) و f_A برای هر تابع نشانه $f \in \mathcal{F}$ می‌توانند به چند شکل انجام شود. زانتِما در [۱۷] پیشنهاد کرده است تا مقدار f_A به عنوان وزن فرض شود، ترتیب جدید به دست آمده، *KBO* تعمیم‌یافته[۱] نامیده شده که با $\succ_{kbo}$ نمایش می‌دهند. برای هر $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ داریم: $s \succ_{kbo} t$ اگر

- $[\![s]\!]_\alpha > [\![t]\!]_\alpha$ برای هر مشخصه $\alpha: \mathcal{V} \to A$، و یا
- برای هر مشخصه $\alpha: \mathcal{V} \to A$ که $t = g(t_1, \dots, t_m)$ و همچنین
  - $f \succ g$، یا
  - $f \approx g$ و $[s_1, \dots, s_n] \succ^{LEX}_{kbo} [t_1, \dots, t_m]$

در ادامه برخلاف KBO تعمیم‌یافته که فقط از مقدار تفسیر تابع‌نشانه‌ها استفاده می‌کند، روش سودمند دیگری را معرفی می‌کنیم. یک KBO استاندارد روی مجموعه $\mathcal{T}(\mathcal{F}, \mathcal{V})$ به کمک یک تابع وزن w و تقدم $\succ$، روی $\mathcal{F}$ تعریف می‌شود. تعاریف مربوط به تقدم و تابع وزن را به $\mathcal{F}_{lab}$ با نامتناهی عضو تعمیم می‌دهیم. این تعمیم بر پایهٔ استفاده از برچسب تابع‌نشانه‌ها در محاسبهٔ وزن آن‌ها انجام می‌گیرد. مهم‌ترین گام در اثبات پایان‌پذیری سیستم $\mathcal{R}$ از طریق SL ساخت شبه‌ـ مدل $>_A$ است که A، دامنه $\mathcal{F}$– جبر به کار رفته $\mathcal{A} = (\mathcal{F}, A)$ می‌باشد. برای اینکه این موضوع بهتر روشن شود، مثال ۲ را ببینید.

مثال ۲. سیستم $\mathcal{R}_2$ را با قوانین زیر در نظر بگیرید:

```
1: max(x, y) →  cond(x, y, ge-eq(x, y))
2: ge-eq(x, 0) → true
3: ge-eq(0, s(y)) → false
4: ge-eq(s(x), s(y)) → ge-eq(x, y)
5: cond(x,  y, false) → max(y, x)
6: cond(x, y, true)  → x
```

یک تفسیر خودکار توسط نرم‌افزار TPA برای این سیستم با فرض $A = \mathbb{N}$ به صورت زیر ایجاد می‌شود:

```
Max_ℕ = max{x, y}, true_ℕ = 1, false_
ℕ = 0, 0_ℕ = 0,s_ℕ = x+1,
```

$$\text{Cond\_}\mathbb{N} = \begin{cases} x & \text{if } z > 0 \\ \max\{x, y\} & \text{if } z = 0 \end{cases}$$

$$\text{ge-eq\_}\mathbb{N} = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{if } x < y \end{cases}$$

یک برچسب‌گذاری ℓ برای جبر

$$(A, \mathcal{F}_A) = (\mathbb{N}, \{max, true, false, 0, cond, ge\text{-}eq\}_A)$$

با انتخاب $L_{max} = \{1, 2\}$ و $L_{ge\text{-}eq} = \{1, 2, 3\}$، به صورت زیر تولید می‌شود:

$$\text{Cond\_}\ell = \begin{cases} 1 & \text{if } z > 0 \\ 2 & \text{if } z = 0 \wedge x < y \\ 3 & \text{if } z = 0 \wedge x \geq y \end{cases}$$

$$\text{max\_}\ell = \begin{cases} 1 & \text{if } x \geq y \\ 2 & \text{if } x < y \end{cases}$$

سیستم برچسب‌گذاری شده $\mathcal{R}_2$ با به شکل زیر تولید می‌شود:

```
1': max_1(x, y)→ cond_1(x, y, ge-eq(x, y))
1'':max_2(x, y)→ cond_2(x, y, ge-eq(x, y))
2:  ge-eq(x, 0) → true
3:  ge-eq(0, s(y)) → false
4:  ge-eq(s(x), s(y)) → ge-eq(x, y)
5': cond_3(x,  y, false) → max_2(y, x)
5'':cond_3(x,  y, false) → max_1(y, x)
5''': cond_2(x,  y, false) → max_1(y, x)
6:  cond_1(x, y, true)  → x
```

سیستم تبدیل یافته $\mathcal{R}_{lab}$ آماده است تا یک شبه‌ـ مدل روی آن جستجو و مراحل اثبات کامل شود. □

برای اینکه بتوانیم پایان‌پذیری سیستم‌های برچسب‌گذاری شده را توسط KBO اثبات کنیم، ابتدا آن را تعمیم می‌دهیم.

تعریف ۶. تابع وزن برچسب‌گذاری

فرض کنید w یک تابع وزن داده شده روی $\mathcal{F}$ باشد، تابع وزن برچسب‌گذاری w_ℓ روی $\mathcal{F}_{lab}$ را که $w_\ell: \mathcal{F}_{lab} \to \mathbb{N}$ تعریف می‌کنیم: برای هر $f \in \mathcal{F}$ اگر $L_f \neq \emptyset$ باشد $w_\ell(f_a) = a + w(f)$ و اگر $L_f = \emptyset$، $w_\ell(f) = w(f)$ است. تابع وزن W_ℓ نیز همانند W برای هر ترم $t \in \mathcal{T}(\mathcal{F}_{lab}, \mathcal{V})$ که $t = f_a(t_1, \dots, t_n)$ به صورت $W_\ell(t) = w_\ell(f) + \sum_{i}^{n} = 1\ W_\ell(t_i)$ تعریف می‌شود.

تعریف ۷. تقدم برچسب‌گذاری

فرض کنید $\succ$ یک تقدم روی $\mathcal{F}$ باشد، $\succ_\ell$ که تعمیم آن برای $\mathcal{F}_{lab}$ است، به این صورت تعریف می‌شود: برای هر $f_a, g_b \in \mathcal{F}_{lab}$

1. Generalized Knuth-Bendix Order

داریم: $f_a \succ_\ell g_b$ اگرو فقط اگر $f \succcurlyeq g$ و $a>b$ یا اینکه $f \approx i$ باشد و یا اینکه $f \nsucc g$ و $a>b$ به طور خلاصه داریم:

$$f_a \succ_\ell g_b \Leftrightarrow ((f \succcurlyeq g \wedge (a>b \vee f \approx i)) \vee (f \nsucc g \wedge a>b))$$

□

در روش خودکار پیشنهادی نوع برچسب‌گذاری تابع نشانه‌ها را به یکی از انواع همانی (`id`)، کمینه (`min`) و بیشینه (`max`) در کنار امکان عدم برچسب‌گذاری (`null`) محدود می‌کنیم. انتخاب نوع برچسب‌گذاری هر تابع‌نشانه f را تابع وضعیت $\pi(f)$ انجام می‌دهد. همچنین تفسیرهای f_A را به چندجمله‌ای نابدیهی ($p(x) \neq 0$) که جستجوی خودکار آن کارآیی خوبی دارد، محدود می‌کنیم. برای ترم $f(t_1,\dots,t_n)$ نمادهای زیر را تعریف می‌کنیم:

```
(f, min)_A = min{(t1)_A,…, (tn)_A }
(f, null)_A =0, (f, id)_A = f_A
```

قضیه ۲. اگر W یک تابع وزن قابل قبول برای تقدم $\succ$ روی $\mathcal{F}$ و ℓ_f برای هر $f \in \mathcal{F}$ یکنواخت ضعیف باشد، W_ℓ یک تابع وزن قابل‌قبول برای $\succ_\ell$ روی $\mathcal{F}_{lab}$ است.

اثبات: بنا بر تعریف تابع وزن قابل قبول، فرض کنید $f \in \mathcal{F}^{(1)}$ که $L_f=\emptyset$ و $W_\ell(f)=0$ باشد در این صورت لزوماً $W(f)=0$ بوده و f نماد ویژه W_ℓ خواهد بود، زیرا برای هر $g_b \in \mathcal{F}_{lab}$ داریم $f \succ_\ell g_b$ است. حال اگر برچسب $L_f \neq \emptyset$ باشد، زمانی $W_\ell(f_a)=0$ است که $\ell_f(n)=0$ یعنی $[\![(t, \mathtt{id})]\!]_\alpha=0$ یا $[\![(t, \mathtt{max})]\!]_\alpha=0$ برای هر $t=f(t_1,\dots,t_m)$ و هر مشخصه α باشد. در مورد اولی حتماً باید $f_A(x)=0$ باشد، که به این ترتیب f_0 تنها تابع‌نشانه برچسب‌گذاری شده است، و از آنجا که $f \succ g$ برای هر $g \in \mathcal{F}-\{f\}$ در نتیجه $f_0 \succ_\ell g_b$ خواهد بود. در حالت دوم، تابع `max` تنها برای $f(t_1,\dots,t_n)_A=0$ برچسب صفر را انتخاب می‌کند. چندجمله‌ای f_A با ضرایب مثبت وقتی صفر است که مقدار ثابت آن صفر باشد، در حالی که چند جمله‌ای‌های تولید شده بنا بر فرض ما اکیداً نابدیهی هستند. در نتیجه در این حالت، وجود یک تابع‌نشانه f_a با وزن $W_\ell(f_a)=0$ نقض شده، لذا W_ℓ قابل قبول است. □

حال یک تعمیم برای ترتیب KBO با استفاده از W_ℓ و تقدم $\succ_\ell$، روی مجموعه $\mathcal{T}(\mathcal{F}_{lab}, \mathcal{V})$ ارائه می‌کنیم. نام این ترتیب جدید که با استفاده از برچسب هر تابع‌نشانه وزن آن را محاسبه می‌کند، ترتیب کُنُت‌ـ بندیکس برچسب‌گذاری (ℓKBO) گذاشته‌ایم.

تعریف ۸. ترتیب کُنُت‌ـ بندیکس برچسب‌گذاری

ترتیب ℓKBO را که با $>_{\ell KBO}$ نمایش می‌دهیم، براساس تابع وزن قابل قبول (W_ℓ, w_0)، و تقدم $\succ_\ell$، روی $\mathcal{T}(\mathcal{F}_{lab}, \mathcal{V})$ به این صورت تعریف می‌کنیم: برای دو ترم $v \in \mathcal{T}(\mathcal{F}_{lab}, \mathcal{V})$ و u داریم $u >_{\ell KBO} v$ اگر برای هر $x \in \mathcal{V}ar(t)$ داشته باشیم:

$$|u(u_1, \dots, u_n)|_x \ge |v(v_{i1}, \dots, v_{im'})|_x \quad 1 \le i \le m$$

و سپس

۱. $W_\ell(v)>W_\ell(u)$ یا

۲. $W_\ell(v)=W_\ell(u)$، $u=i^a u'$، $v=i^b v'$ که $u'=g(u'_1, \dots, u'_{n'})$ و $v'=h(v'_1, \dots, v'_{m'})$ بوده و یکی از وضعیت‌های زیر برقرار باشد:

الف) $a>b$، یا

ب) $a=b$ و $hg \succ$، یا

ج) $a=b$ و $g \approx h$ و $(s_1, \dots, s_n) >^{LEX}_{\ell KBO} (t_1, \dots, t_n)$

□

برقراری یک ℓKBO برای سیستم برچسب‌گذاری شده $\mathcal{R}_{lab}$ یک شبه‌ـ مدل برای آن می‌سازد. این موضوع در قضیهٔ ۳ اثبات می‌شود.

قضیهٔ ۳. فرض کنید W_ℓ و $\succ_\ell$ به ترتیب، مطابق تعریف ۶ و ۷ روی $\mathcal{F}_{lab}$ باشند. در این صورت، ترتیب $>_{\ell KBO}$ تعریف ۸ یک ترتیب خوش‌ساخت است.

اثبات: با برهان خلف ادعای خود را اثبات می‌کنیم. فرض کنید $>_{\ell KBO}$ خوش‌ساخت نباشد، پس یک دنباله از کاهش‌های نامتناهی از ترم‌های برچسب‌گذاری مانند زیر را ایجاد می‌کند،

$$\dots, t_i = f_a(u_1, \dots, u_n), t_{i+1} = g_b(v_1, \dots, v_n), \dots$$

که $t_i \in \mathcal{T}(\mathcal{F}_{lab}, \mathcal{V})$ برای هر $i \in \mathbb{N}$ و برچسب a برابر با $\ell_f([\![u_1]\!]_\alpha, \dots, [\![u_n]\!]_\alpha)$ است. بنا بر تعریف ۸ می‌دانیم که مقادیر $W_\ell(t_i)$ یا کاهش می‌یابد و یا مقدار خود را تحت $>_{\ell KBO}$ حفظ می‌کند. بنابراین در دنباله نامتناهی $W_\ell(t_1), W_\ell(t_2), \dots, W_\ell(t_j), \dots$ تنها $W_\ell(t_1)$ عدد مختلف ظاهر می‌شود. بنا بر خوش‌ساخت بودن $\succ_\ell$ یک $m \in \mathbb{N}$ وجود دارد که برای هر $n \ge m$ مقادیر دنبالهٔ وزن‌ها یکسان بوده و یک عدد

به طور پیوسته تکرار می‌شود؛ لذا ما برای $W_\ell(t_n), W_\ell(t_{n+1}), \ldots$ شرایط $root(t_n) > root(t_{n+1})$ (وضعیت ۲-ب) را داریم و یا برای مجموعه زیر ترم‌های t_n و t_{n+1} این امر برقرار است. با توجه به محدود بودن ساختار ترم فراخوانی بازگشتی حالت ۲- ب تعریف ۸ در تعداد متناهی گام متوقف می‌شود. در نتیجه، دنبالهٔ ساخته شده توسط $\succ_{\ell KBO}$ متوقف می‌شود. این با فرض نامتناهی بودن دنباله تناقض دارد. در نتیجه حکم برقرار است. □

اگر برای سیستم بازنویسی $\mathcal{R}$، سیستم $\mathcal{R}_{lab}$ از نگاشت lab_α روی مدل $\mathcal{A}$ تولید شده باشد، مسئلهٔ اصلی برای اثبات پایان‌پذیری $\mathcal{R}$، جستجوی یک شبه- مدل بوده که جستجوی یک ℓKBO بخشی از آن است. ایدهٔ ما این است که اگر $\mathcal{R}$ یک TRSپایان‌پذیر باشد که KBO سازگاری با آن وجود ندارد، در این صورت ممکن است سیستم برچسب‌گذاری شده $\mathcal{R}_{lab}$ با یک ℓKBO مطابق تعریف ۸ سازگار باشد. می‌توان نشان داد اگر برای هر $f \in \mathcal{F}$، $\pi(f) = \mathtt{id}$ در این صورت ℓKBO و KBO تعمیم یافته با هم معادل‌اند. این نشان می‌دهد ℓKBO می‌تواند در عمل، قابلیت‌های بیشتری داشته باشد.

قضیهٔ ٤. اگر تابع برچسب‌گذاری f_ℓ برای هر $f \in \mathcal{F}$ همانی فرض شود، KBO برچسب‌گذاریو KBO تعمیم یافته معادل- اند.

اثبات: اگر برچسب‌گذاری‌ها همانی باشد آنگاه،

$$W_\ell(f(s_1, \ldots, s_n)) = w_\ell(f) + \sum_{i=1}^{n} W_\ell(t_i) = w(f) + (f, \mathtt{id})_A + \sum_{i=1}^{n} W_\ell(t_i) = w(f) + \ell_f([\![s_1]\!]_\alpha, \ldots, [\![s_n]\!]_\alpha) + \sum_{i=1}^{n} W_\ell(t_i)$$

که $w(f)$ یک ثابت عددی است که به مقدار تابع تفسیر f_A افزوده می‌شود و تابع تفسیر جدیدی که فقط ثابت عددی آن فرق دارد را به وجود می‌آورد. □

برای کامل کردن اثبات پایان‌پذیری سیستم $\mathcal{R}_2$ و ساخت یک شبه- مدل، ترتیب ℓKBO به این صورت عمل می‌کند:

ابتدا وزن تابع نشانه‌ها را به صورت زیر محاسبه می‌کنیم:

```
w(max) = 6, w(false) = 3, w0 =1,
w(cond) = w(ge-eq) = w(s) = w(true)
=w(0) = 2,
```

حال برای تقدم $\succ_\ell$ محدودیت‌های زیر را داریم:

```
1':  w(max)>ℓKBO w(cond)+ w(ge-eq)+2w0
   ⇒ max1≻ℓ cond1
1'': w(max) >ℓKBO w(cond)+ w(ge-eq)+
2w0  ⇒ max2≻ℓ cond2
5': w(cond) + w(false) + 1 >ℓKBO
w(max)   ⇒ cond3≻ℓ max2
5''': w(cond) + w(false) +1  >ℓKBO
w(max) ⇒ cond3≻ℓ max2
```

فرمول محدودیت‌های فوق با ارائهٔ تقدم $\succ_\ell$ به صورت زیر ارضاءپذیر خواهد شد.

$cond_3 \succ_\ell max_2 \succ_\ell cond_2 \succ_\ell max_1 \succ_\ell cond_1 \succ_\ell$ ge-eq $\succ_\ell$ s $\succ_\ell$ true $\succ_\ell$ false $\succ_\ell$ 0

در نتیجه سیستم $\mathcal{R}_2$ پایان‌پذیر است.

٤. الگوریتم خودکار جستجویℓKBO

هدف اصلی این مقاله، افزایش قابلیت خودکارسازی روش SL است. با جستجوی خودکار یک ℓKBO که با $\mathcal{R}_{lab}$ سازگار باشد، می‌توان خودکارسازی SL را بهبود بخشید. اگر $\mathcal{R}_{lab}$ متناهی باشد (مانند مثال فوق) این جستجو فضای محدودی خواهد داشت که در آن، با برقراری شرایط ℓKBO و شرایط مربوط به سازگاری آن با $\mathcal{R}_{lab}$ فضای جستجوی یک تقدم خوش‌ساخت مشخص می‌شود. برای مدل‌های نامتناهی فضای جستجو نامحدود است. این مشکل را از طریق حذف یا کنار گذاشتن حالات مشابه برطرف نموده‌ایم تا کارآیی جستجو افزایش یابد.

با انتخاب نوع برچسب‌گذاری توسط تابع $\pi(f)$ برای هر $f \in \mathcal{F}$ بین تابع وزن W_ℓ و تفسیرهای معنایی $\mathcal{F}_A$ارتباط برقرار کرده و در فضای شرطهای موجود یک تقدم خوش‌ساخت را جستجو می‌کنیم. این کار به کمک گراف کُنُت- بندیکس[1] (KBG) انجام می‌شود.

در KBG گزینه‌های برچسب‌گذاری به عنوان موقعیت‌های راس گراف در نظر گرفته می‌شوند. ارتباط دو راس در صورتی که یکی نماد ریشه‌ای ترم چپ و دیگری نماد ریشه‌ای ترم راست از یک قانون بازنویسی باشند، توسط یالی که شرط مقدار وزن بین آن‌ها را بیان می‌کند، برقرار می‌شود. یافتن مسیرهایی که

1. Knuth-Bendix graph

بین شرط‌های وزنی آن سازگاری وجود دارد، امکان ساخت یک تقدم خوش‌ساخت را فراهم می‌کند.

تعریف ۹. گراف کُنُت‌ـ بندیکس

فرض کنید $\succ$ و W بترتیب یک تقدم و یک تابع وزن روی $\mathcal{F}$ باشند، دوتایی $(\mathcal{F}, E)$ یک گراف کُنُت‌ـ بندیکس (KBG) گوییم اگر $\mathcal{F}$ مجموعه‌ای ناتهی متناهی از راس‌ها و رابطهٔ E به صورت زیر باشد:

$f \to g$، اگر و فقط اگر $f \succ g$ و یا $f = i$.

□

برای اینکه جستجوی یک تقدم روی $\mathcal{F}_{lab}$ با تعداد نامتناهی عضو را از طریق ساخت گراف کُنُت‌ـ بندیکس (KBG) انجام شود، باید بتوان تمام محدودیت‌های ممکن را در KBG ساخته شده ذخیره کرد. برای این منظور، تعریف ۹ را به مفهوم بزرگ‌تری به نام KBG معنایی توسعه می‌دهیم.

تعریف ۱۰. گراف کُنُت‌ـ بندیکس معنایی

فرض کنید $\succ$ و w همان تعریف قبلی را داشته باشند، دوتایی $(\mathcal{F}, E_W)$ یک گراف KBG معنایی ساخته‌شده با $\mathcal{F}$–جبر یکنواخت ضعیف $\mathcal{A}$ است، که $\mathcal{F}$ مجموعه‌ای ناتهی متناهی از راس‌ها و رابطهٔ E_W زیر مجموعهٔ $\mathcal{F}\times \{>, \geq, =\} \times\mathcal{F}$ است، به طوری که:

- $(f, \succ, g) \in E_W$ اگر و فقط اگر $f_A = g_A$، در غیر این صورت
- $(f, \approx, g) \in E_W$ اگر و فقط اگر $f_A \geq g_A$، در غیر این صورت
- $(f, \preccurlyeq, g) \in E_W$ اگر و فقط اگر $f_A > g_A$.

یک گراف KBG معنایی را برچسب‌دار نامند، اگر مجموعه راس‌های آن را $\mathcal{F}\times\{\texttt{null}, \texttt{id}, \texttt{min}, \texttt{max}\}$ در نظر بگیریم. □

برای هر راس $(f, \pi(f))$ تعریف می‌کنیم:

$$IN((f, \pi(f)))=\{ (g, \pi(g)) \mid ((f, \pi(f)),C, (g, \pi(g)))\in E_W \}$$
$$OUT((f, \pi(f))) =\{ (g, \pi(g)) \mid ((g, \pi(g)),C, (f, \pi(f)))\in E_W\}$$

که در اینجا C عضو مجموعه $\{\preccurlyeq, \approx, \succ\}$ است که هر یک از این وضعیت‌ها محدودیت تقدم و تابع وزن مناسب برای ساخت ℓKBO روی TRS برچسب‌گذاری شده مورد نظر است. یک مسیر اویلری سازگار تقدم مورد نظر را روی $\mathcal{F}_{lab}$ تضمین می‌کند. در گراف KBG برچسب‌دار معنایی این مسیرها بخش‌های همبند ضعیف[1] (WCC) بوده، که برای هر دو راس مسیری از یکی به دیگری وجود دارد. مجموعه تمام WCC ها که بین شرط‌های مسیرهای آن سازگاری وجود دارد را WCC-set می‌نامیم. مسئلهٔ جستجوی یک تقدم خوش‌ساخت روی $\mathcal{F}_{lab}$ به مسئلهٔ ناتهی بودن WCC-set یک KBG برچسب‌دار معنایی تبدیل می‌شود.

نگاشت π در گراف KBG برچسب‌دار معنایی همانی را برای قوانینی که خاصیت ناتکثیری[2]، $|(t,\pi(s))|_x \geq |(s,\pi(s))|_x$ را حفظ نکنند، در نظر نمی‌گیرد. برای این کار، $\mathcal{F}$ به دو زیرمجموعه $\mathcal{F}_{NDU}$، تابع‌نشانه‌هایی که ناتکثیری هستند و $\mathcal{F}_{DU}$ تقسیم می‌شود. نشانه‌های غیر ریشه می‌توانند برچسب‌گذاری‌های null و id داشته باشند. برای تابع‌نشانه‌های تک‌موضعی فقط دو نوع برچسب‌گذاری مختلف وجود دارد، زیرا `id`، `min{x}` و `max{x}` بر هم منطبق‌اند. ساخت یک KGB برچسب‌دار معنایی را در مثال ۳ انجام داده‌ایم.

مثال ۳. اثبات پایان‌پذیری سیستم ساده ضرب به دلیل نداشتن ویژگی ناتکثیری با روش KBO استاندارد اثبات نمی‌شود، اما ℓKBO، امکان اثبات آن را فراهم می‌کند.

```
1: add(0, y) → y
2: add(s(x), y) → s(add(x, y))
3:  mult(0, y) → 0
4:  mult(s(x), y) → add(mult( x, y), y))
```

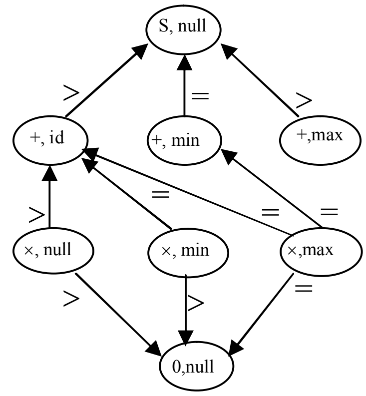
برای این سیستم $\mathcal{F}_{NDU} = \{ mult\}$ با تفسیرهای به کار برده شدهٔ زیر

```
0ℕ=1,sℕ(x)= x+1, ×ℕ(x, y)= max{x, y},
+ℕ(x, y)= min{x, y}
```

برای سیستم فوق، KBG برچسب‌دار معنایی در شکل (۴ــ۱) آمده است. در اینجا + و × به ترتیب به جای add و mult آمده‌اند.

---

1. Weakly connected components
2. Non-duplicating



شکل (۴-۱): گراف KBG برچسب‌دار معنایی سیستم ضرب

قضیهٔ ۵. اندازهٔ گراف KBG برچسب‌دار معنای برای TRS $(\mathcal{F}, \mathcal{R})$ که $|\mathcal{F}| = n$ و $|\mathcal{R}| = m$ باشد، از کلاس پیچیدگی $O(4(m + n))$ است.

اثبات: بنا بر تعریف ۵-۴ راس‌های گراف KBG برچسب‌دار معنایی مجموعه $\mathcal{F}\times\{\mathtt{null},\ \mathtt{id},\ \mathtt{min},\ \mathtt{max}\}$ است، در نتیجه تعداد راس‌های این گراف حداکثر 4n خواهد بود. همین‌طور با استفاده از هر قانون $l \to r \in\mathcal{R}$ یالی بین دو تابع‌نشانه f و g که هریک راسی از گراف هستند، برقرار می‌شود. از آنجا که هر تابع‌نشانه چهار انتخاب برای برچسب‌گذاری دارد، در نتیجه تعداد یال‌ها همان 4m خواهد بود. □

با انتخاب تابع‌های min و max برای برچسب‌گذاری، نیازمند روشی برای مقایسه وزن تابع نشانه‌های برچسب‌گذاری شده هستیم. روش مستقیم زیر با اینکه ریاضیات دشواری را به کار می‌گیرد، اما قابل برنامه‌نویسی بوده و می‌تواند به عنوان یک ماکرو فراخوانی و مقایسه‌ها را انجام دهد؛ برای مثال، اگر وضعیت برچسب‌گذاری برای یک قانون به شکل

$f_{\min\{a1,\ldots,an\}}(t_1, \ldots, t_n) \to g_{\min\{b1,\ldots,bm\}}(u_1, \ldots, u_m)$

باشد، محاسبهٔ برقراری شرط مذکور به صورت زیر فرمول‌بندی می‌شود.

$$\wedge_{i \le n} \vee_{j \le m} \left(\left(w(f) + X + \sum_i^n = 1\ w_\ell(t_i) \right) \ge \left(w(g) + Y + \sum_j^m = 1\ w_\ell(t_i) \right) \right)$$

۴-۱. الگوریتم جستجوی خودکار ℓKBO

جستجوی یک ℓKBO مناسب برای ساخت یک شبه-مدل برای سیستم برچسب‌گذاری شده $\mathcal{R}$ با ساخت KBG معنایی برچسب‌دار آن اجرا می‌شود. سپس با یافتن مسیرهای سازگار در آن، و یا با امکان برطرف کردن ناسازگاری بین شرطها، از طریق ارائه تقدم $\succ$، روی $\mathcal{F}$ انجام می‌شود. یک مسیر ناسازگار ممکن است با ایجاد یک چرخه در گراف KBG ایجاد شود.

تعریف ۱۱. الگوریتم جستجوی تقدم خوش‌ساخت $\succ_\ell$ روی $\mathcal{F}_{lab}$

```
-------------------------------------
Search Labelling Precedence (𝓕, 𝓡, 𝓐 = (𝓕ℕ, ℕ), answer)
        /* Compute duplicating and nonduplicating signature
        sets
              𝓕NDU = { f∈𝓕 | ∃ l → r ∈𝓡  s.t. f = root(l) ∧∀
              x∈𝒱ar(r), |l|x≥|r|x }
              𝓕DU = 𝓕 - 𝓕NDU

        /* Create labeling-semantical Knuth-Bendix graph L-
        S-KBG(G, EW)
              G = 𝓕DU×{ null, min, max } ∪𝓕NDU×{ null, id,
              min, max }
              /* Compute  EW
              for all l → r ∈𝓡 where
              f = (l) and g = 𝒟(r) and
              (f, π(f)) ∈ G and (g, π(g)) ∈G  do
              ((f, π(f)), =, (g, π(g)))∈ EWiff(f, π(f))A≥ (g,
              π(g))A،else
              ((f, π(f)), >, (g, π(g)))∈ EW iff (f, π(f))A = (g,
              π(g))A، else
              ((f, π(f)), ≤, (g, π(g)))∈ EW iff (f, π(f))A> (g,
              π(g))A }  od rof
              /* compute the initial WCC
              for i = 1 to 4n do
              if (f, z) ∈G then WCC[i] ← f,  fi
                   G= G - (f, z)

        /* Compute all weakly comone component (WCC-set )
              if  EW≠Ø
              for  all ((f, π(f)), C, (g, π(g))) ∈ EW
              for i = 1 to 4n   do
                         WCC[i] ← {g} ∪ WCC[i], if  (x, C,
              f)∈WCC[i] and  SAT( C ∧ Cond[i])
              WCC[i] ← {f} ∪ WCC[i], if  (g, C, y)∈WCC[i]
              and  SAT( C ∧ Cond[i])
              rof
              rof
              fi
              for i = 1 to 4n
              if   WCC[i] = 𝓕  then WCC[i] add to WCC-set
              fi

rof
if   WCC-set ≠Ø  then  answer ← true     fi
else    answer ← false
-------------------------------------
```

بخش‌های اول و دوم این الگوریتم با استفاده از نرم‌افزار TPA انجام‌شدنی است. برای بخش سوم از الگوریتم‌های

جدید «محاسبهٔ بخش‌های همبند گراف» ماننـد [۱۵] در زبـان Java استفاده کرده‌ایم. □

بنا بر شکل (۵ـ۱) پیش ترتیب نتیجه شده از جستجو روی مسیرهای سازگار به صورت زیر خواهد بود:

```
(×, max) ≻(+, min)≻ (s, null)≻(0, null)
```

۵. مثال‌ها

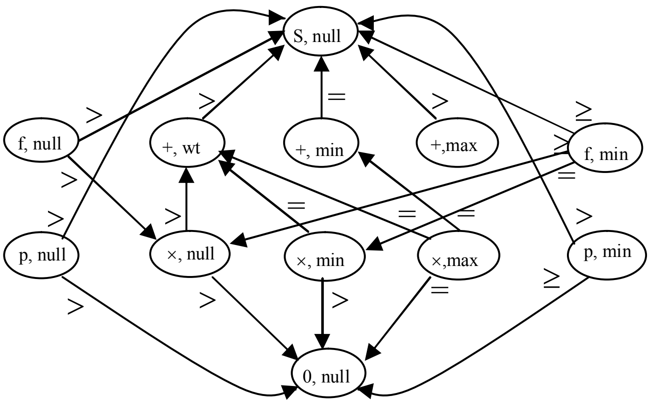
مثال ٤. یکی از TRSs مهم برای آزمون تحت مـدل‌هـای نامتناهی، سیستم فاکتوریل است که اثبات پایان‌پذیری آن بـا روش ارائه شده انجام می‌شود. این سیسـتم نیـز ویژگـی $|s|_x \geq |t|_x$ را برای هر s و t ندارد؛ لذا گزینه‌های برچسـب‌گـذاری فاقـد id خواهد بود.

```
1: fact(s(x)) → mult(fact(p(s(x)),s(x))
2: p(s(0))   →   0
3: p(s(s(x)))   → s(p(s(x)))
```

مرحلهٔ اول: $\mathcal{F}_{DU} = \{fact\}$ و $\mathcal{F}_{NDU} = \{mult, p, s, 0\}$ می‌باشد، لذا داریم:

```
V = {mult, p, s}×{id, min, max,
null}∪ {fact}×{null, min}
```

نوع برچسب‌گذاری برای تابع‌نشانه‌های تک‌پارامتری به دو حالت محدود می‌شود.



شکل (۵ـ۱): گراف KBG برچسب‌دار معنایی سیستم فاکتوریل

مرحلهٔ دوم: KBG برچسب‌دار معنـایی سیستم داده شـده مطابق شکل (۵ـ۱) ساخته می‌شود. یک تقـدم مناسـب کـه از KBG فوق نتیجه می‌شود، به این صورت است:

```
WCC[4] :=(f, min) ≻ (×, min) ≻ (+,
min)≻ (s, null) ≻(p, null)≻(0, null)
```

روشن است که شرطهای لبهٔ مسیر حاصـل شـده بـا هـم سازگارند. □

در ادامه، نحوهٔ آزمون الگوریتم تعریف ۱۱ را برای دو مثال دیگر نشان می‌دهیم.

مثال ۵. سیستم $\mathcal{R}_3$ را در نظر بگیرید:

```
1: g(s(s(x))) → true
2: g(0) → false
3: p(s(x)) → x
4: odd(s(x)) →if(g(x), x, odd( p(x)))
5: odd(0) → 0
6: if(false, x, y) → x
7: if(true, x, y)  → y
```

سیستم فوق فرد بودن یک عدد را بررسی می‌کند. رفتار آن را روی عدد ۳ مشاهده کنید:

```
odd(s(s(s (0)))) →4
if(g(s(s(0))),s(s(s(0))),odd(p(s(s(0)))))
→1   if(true,s(s(s(0))),odd(p(s(s(0))))))
→7   odd(p(s(s(0)))))
→3   odd(s(0))
→4   if(g(0), s(0), odd(p(s(0))))
→2   if(false, s(0), odd(p(s(0))))
→6   s(0)
```

بـا اسـتفاده از روش برچسـب‌گـذاری معنـایی بـرای $\mathcal{R}_3$ تفسیرهای زیر ساخته می‌شود:

```
gℕ = 2, pℕ = max{x-1, 0}, sℕ = x+1,
ifℕ = max{x, y, z}, oddℕ = 2x, trueℕ
=falseℕ = 1
```

همین‌طور با انتخاب $L_{odd} = \mathbb{N}$ تـابع برچسـب‌گـذاری بـه صورت $\ell_{odd} = n$ انتخاب می‌شود. برای اثبات پایان‌پذیری این سیستم، نیازمند تعیین هر یک از نگاشت‌های زیر هستیم:

W: {odd, if, g, s, p, true, false, 0} $\to \mathbb{N}$
π: {odd, if, g, s, p, true, false, 0} $\to$ {null, id, min, max}

الگوریتم تعریف ۱۱ با بررسـی محـدودیت‌هـای هـر یـک از قوانین سیستم فوق کـه در KBG آن اعمـال شـده، مـی‌توانـد در فضای حالت‌های ممکن، وجود یـک تقـدم مناسـب را جسـتجو کند. یک صورت ممکن در مورد سیستم فوق در زیر آمده است:

$\pi:$ odd $\to$ id, if $\to$ max, g $\to$ null, s $\to$ null, p $\to$ null, true $\to$ null, false $\to$ null, 0 $\to$ null

برای برقراری شرطهای قوانین بازنویسی بایـد نامعـادلات زیر حفظ شود:

```
Wg> 1, Wodd +2  > 1, Wp> 1,
Wodd + i+1 > Wif +max(a, i+1, i-1),
Wif +max(a, i+1, i-1) > Wodd + i-1,
Wif +max(0, i+1, i-1) > i
```

ارضاءپذیری شرایط فوق به راحتی قابل انجام بوده و سیستم ما پایان‌پذیر است. یک پاسخ ممکن به صورت زیر است:

```
(w, w0=1) odd → 3, if → 2, g → 2, s →
2, p → 2, true → 1, false →1, 0 → 1.
```

□

مثال ٦. پایان‌پذیری سیستم برچسب‌گذاری شده $\mathcal{R}_1$ را که در مقدمه معرفی شد، در ادامه به وسیلهٔ الگوریتم تعریف ۱۱ دنبال می‌کنیم.

تفسیرهای طبیعی ما:

$$f_{\mathbb{N}} = g_{\mathbb{N}} = a_{\mathbb{N}} = b_{\mathbb{N}} = n, i_{\mathbb{N}}(n) = n+2$$
$$h_{\mathbb{N}}(n) = n+1$$

برای این سیستم $\mathcal{F}_{NDU} = \{f, g, h, i, a, b\}$ می‌باشد؛ لذا $V = \{f, g, h, i\} \times \{id, null\}$ است.

از آنجا که تابع‌نشانه‌ها یک پارامتر دارند، نوع برچسب‌گذاری به دوحالت محدود می‌شود. با جستجوی $L_f = \mathbb{N}$ و $L_g = L_i = L_h = \emptyset$ برچسب‌گذاری را تنها برای f خواهیم داشت. سیستم برچسب‌گذاری شده به صورت زیر خواهد بود:

$$f_{n+1}(i(x)) \rightarrow f_n(h(x)) \qquad n \in \mathbb{N}$$
$$g(h(x)) \rightarrow g(i(x))$$
$$i(a) \rightarrow b,\ h(a) \rightarrow b$$

مقایسهٔ محدودیت‌های این سیستم به شکل زیر نتیجه می‌شود.

$$w(f) + n+1 + w(i) + w_0 >_{\ell KBO} w(f) + n + w(h) + w_0$$
$$\Rightarrow 2 + w(i) >_{\ell KBO} w(h)$$
$$w(g) + w(h) + w_0 >_{\ell KBO} w(g) + w(i) + w_0 \Rightarrow$$
$$w(h) >_{\ell KBO} w(i)$$

ارضاءپذیری این محدودیت‌ها، پایان‌پذیری را نتیجه می‌دهد. □

٦. نتیجه‌گیری

در این مقاله، KBO را برای $\mathcal{R}_{lab}$ نامتناهی تعمیم داده و ترتیب کُنُت- بندیکس برچسب‌گذاری (ℓKBO) را معرفی کردیم. سیستم‌های پایان‌پذیر زیادی یک KBO سازگار ندارد، اما در موارد زیادی با تولید سیستم برچسب‌گذاری شده یک ℓKBO سازگار قابل ساخت است. برای ساخت یک ℓKBO ابتدا باید $\mathcal{R}_{lab}$ را ایجاد کنیم. برای تولید تفسیرهای طبیعی و اجرای تبدیل برچسب‌گذاری از نرم افزار TPA استفاده کرده‌ایم. همچنین TPA برای ساخت تابع‌های W_ℓ و π به کار رفته است. مقایسهٔ تفسیرهای چندجمله‌ای تحت زبان TPA انجام می‌شود. پیاده‌سازی یک مسیر کاملاً نرم‌افزاری برای مراحل بیان شده از کارهای آیندهٔ این تحقیق است؛ هر چند مراحل هر قسمت از الگوریتم ۱۱ قابلیت انجام نرم‌افزاری دارد. محاسبهٔ بخش‌های همبند ضعیف نیز توسط الگوریتم‌های جستجوی WCC، گراف‌های جهت‌دار استفاده شده است. کدبندی محدودیت‌ها در این تحقیق از [۲۲] اقتباس شده که می‌توانند توسط TPA اجرا شود.

روشن است که محدود کردن برچسب‌گذاری به چهار حالت، قدرت آن را کاهش می‌دهد. اگر تابع π به دامنهٔ بزرگ‌تری از تابع‌های مجاز تعمیم پیدا کند، قدرت $>_{\ell KBO}$ افزایش خواهد یافت؛ اگرچه تعمیم π زمان اجرای خودکار آن را افزایش خواهد داد. روش SL روی مدل‌های متناهی توسط نرم‌افزارهای زیادی مانند AProVE، Jambox، TORPA، TPA و TEPARLA پشتیبانی می‌شود، اما مدل‌های نامتناهی را ابزارهای کمی مانند TPA فراهم می‌آورند. مراحل الگوریتم ۱۱ روی مجموعه TRSsTPDB 3.1 [۲۳] با موفقیت اجرا شده است.

مراجع

[1] N. Dershowitz, *Ordering for term rewriting systems*, Theo. Com. Sci., 17, pp 279-301, 1982.

[2] A. Middeldorp and H. Zantema, *Simple Termination Revisited*, LNAI, 814, pp 451-465, 1994.

[3] D. E. Knuth, and P. Bendix, *Simple word problems in universal algebras, Computational Problems in Abstract Algebra*, pp. 263-297. Pergamon Press, 1970.

[4] N. Dershowitz., *termination ofrewriting*, journal of symbolic computation, volume 3, pp 69-116, 1987.

[5] N. Dershowitz, *A Note on Simplification Orderings*, Info. Proc. Let, 9, pp. 212-215, 1979.

[6] S. Kamin, J.L´evy, *Two Generalizations of the Recursive Path Ordering*, University of Illinois at Urbana-Champaign, Unpublished manuscript, 1980.

[7] J.-P.Jouannaud, P.Lescanne, F.Reinig, Recursive *Decomposition Ordering*, In Working Conference on Formal Description of Programming Concepts II (IFIP), D.BjÁrner, Ed., North-Holland Publishing Company, pp. 331–353, 1982.

[8] X. Urbain. *Modular & incremental automated termination proofs*. Journal of Automated Reasoning, 32:315–355, 2004.
[9] A. Koprowski, *TPA: Termination proved automatically*, In Proc of the 17th Inter. Conf. on RTA, LNCS, 2006.
[10] J. Dick, J. Kalmus, and U.Martin. *Automating the Knuth-Bendix ordering*, Acta Infomatica, 28:95–119, 1990.
[11] Y. Toyama. *Counterexamples to the termination for the direct sum of term rewriting systems*, Info. Proc. Let. 25, pp. 141–143, 1987.
[12] H. Zantema, *Termination of term rewriting by semantic labeling*, Fund. Info. 24(1/2), pp. 89–105, 1995.
[13] T. Arts and J. Giesl, *Termination of term rewriting using dependency pairs*, Theo. Com. Sci., 236, pages 133-178, 2000.
[14] N. Hirokawa, A. Middeldorp, *Predictive labeling*, In Proc. 17th RTA, volume 4098 of LNCS, pp. 313–327, 2006.
[15] A. Koprowski and H. Zantema., *Recursive path ordering for infinite labelled rewritesystems*. In Proc. 3rd IJCAR, volume 4130 of LNAI, pp. 332–346, 2006.
[16] A. Koprowski, *TPA: Termination proved automatically*, In Proceedings of the 17thInte. Conf. on RTA, LNCS, volume 4098, 2006.
[17] TeReSe, *Term Rewriting Systems*, volume 55 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2003.
[18] A. Koprowski and A. Middeldorp , *Predictive Labeling with Dependency Pairs using SAT*, LNCS, volume 4603, pp. 410-425, 2007.
[19] A. Middeldorp. H. Ohsaki, and H. Zantema., *Transforming termination by self-labelling*, LNAI, pp. 373-386, 1996.
[20] G. Moser, *Derivational Complexity of Knuth-Bendix Orders revisited*, LNCS, volume 4246 ,pp.75-89 , 2007.
[21] I. Lepper, The Computational Strength of Knuth–Bendix Orders,PHD dissertation at the University of M¨unster, Germany 2000.
[22] H. Zankl and A. Middeldorp, KBO as a Satisfaction Problem, Eighth International Workshop on Termination Seattle, WA, USA, 1516 August 2006.
[23] http://www.lri.fr/~marche/tpdb.