بسم الله الرحمن الرحيم

بسمه تعالی

دانشگاه تربیت مدرس

دانشکده علوم ریاضی

تأییدیه اعضای هیأت داوران حاضر در جلسه دفاع از پایان‌نامه کارشناسی‌ارشد

اعضای هیأت داوران نسخهٔ نهایی پایان‌نامه آقای **محمد کدخدا** رشته **علوم کامپیوتر** به شماره دانشجویی ۸۷۵۷۳۱۰۰۵ تحت عنوان: «**مدل اجرایی روش برچسب‌گذاری معنایی برای اثبات خودکار پایان‌پذیری در سیستم‌های بازنویسی ترم**» را از نظر فرم و محتوا بررسی نموده و آن را برای اخذ درجهٔ کارشناسی‌ارشد مورد تأیید قرار دادند.

اعضای هیأت داوران	نام و نام‌خانوادگی	رتبه علمی	امضاء
۱- استاد راهنما	دکتر سعید جلیلی	دانشیار	
۲- استاد مشاور	دکتر محمد ایزدی	استادیار	
۳- استاد ناظر داخلی	دکتر علی رجایی	استادیار	
۴- استاد ناظر خارجی	دکتر حسن حقیقی	استادیار	
۵- نماینده تحصیلات تکمیلی	دکتر علی رجایی	استادیار	

آیین نامه چاپ پایان‌نامه (رساله)های دانشجویان دانشگاه تربیت مدرس

نظر به اینکه چاپ و انتشار پایان نامه (رساله)های تحصیلی دانشجویان دانشگاه تربیت مدرس، مبین بخشی از فعالیتهای علمی - پژوهشی دانشگاه است بنابراین به منظور آگاهی و رعایت حقوق دانشگاه، دانش آموختگان این دانشگاه نسبت به رعایت موارد ذیل متعهد می‌شوند:

ماده ۱: در صورت اقدام به چاپ پایان نامه (رساله)ی خود، مراتب را قبلاً به طور کتبی به «دفتر نشر آثارعلمی» دانشگاه اطلاع دهد.

ماده ۲: در صفحه سوم کتاب (پس از برگ شناسنامه) عبارت ذیل را چاپ کند:

«کتاب حاضر، حاصل پایان نامه کارشناسی ارشد نگارنده در رشته **علوم کامپیوتر** است که در سال **۱۳۹۰** در دانشکده **علوم ریاضی** دانشگاه تربیت مدرس به راهنمایی جناب آقای **دکتر سعید جلیلی**، مشاوره جناب آقای **دکترمحمد ایزدی** از آن دفاع شده است.»

ماده ۳: به منظور جبران بخشی از هزینه‌های انتشارات دانشگاه، تعداد یک درصد شمارگان کتاب (در هر نوبت‌چاپ) را به «دفتر نشر آثارعلمی» دانشگاه اهدا کند. دانشگاه می‌تواند مازاد نیاز خود را به نفع مرکز نشر درمعرض فروش قرار دهد.

ماده ۴: در صورت عدم رعایت ماده ۳، ۵۰٪ بهای شمارگان چاپ شده رابه عنوان خسارت به دانشگاه تربیت‌مدرس، تأدیه کند.

ماده ۵: دانشجو تعهد و قبول می کند در صورت خودداری از پرداخت بهای خسارت، دانشگاه می‌تواندخسارت مذکور را از طریق مراجع قضایی مطالبه و وصول کند؛ به علاوه به دانشگاه حق می‌دهد به منظور استیفای حقوق خود، از طریق دادگاه، معادل وجه مذکور در ماده ۴ را از محل توقیف کتابهای عرضه شده‌نگارنده برای فروش، تامین نماید.

ماده ۶: اینجانب **محمد کدخدا** دانشجوی رشته **علوم کامپیوتر** مقطع کارشناسی ارشد تعهد فوق وضمانت اجرایی آن را قبول کرده، به آن ملتزم می شوم.

نام و نام خانوادگی: محمد کدخدا

تاریخ و امضا:

دستورالعمل حق مالکیت مادی و معنوی در مورد نتایج پژوهشهای علمی دانشگاه تربیت مدرس

مقدمه: با عنایت به سیاست‌های پژوهشی دانشگاه در راستای تحقق عدالت و کرامت انسانها که لازمه شکوفایی علمی و فنی است و رعایت حقوق مادی و معنوی دانشگاه و پژوهشگران، لازم است اعضای هیات علمی، دانشجویان، دانش آموختگان و دیگر همکاران طرح، در مورد نتایج پژوهشهای علمی که تحت عناوین پایان‌نامه، رساله و طرحهای تحقیقاتی که با هماهنگی دانشگاه انجام شده است، موارد ذیل را رعایت نمایند:

ماده ۱- حقوق مادی و معنوی پایان نامه‌ها / رساله‌های مصوب دانشگاه متعلق به دانشگاه است و هرگونه بهره‌برداری از آن باید با ذکر نام دانشگاه و رعایت آیین‌نامه‌ها و دستورالعمل‌های مصوب دانشگاه باشد.

ماده ۲- انتشار مقاله یا مقالات مستخرج از پایان نامه/ رساله به صورت چاپ در نشریات علمی و یا ارائه در مجامع علمی باید به نام دانشگاه بوده و استاد راهنما مسئول مکاتبات مقاله باشد.

تبصره: در مقالاتی که پس از دانش آموختگی بصورت ترکیبی از اطلاعات جدید و نتایج حاصل از پایان‌نامه/ رساله نیز منتشر می‌شود نیز باید نام دانشگاه درج شود.

ماده ۳- انتشار کتاب حاصل از نتایج پایان نامه / رساله و تمامی طرحهای تحقیقاتی دانشگاه باید با مجوز کتبی صادره از طریق حوزه پژوهشی دانشگاه و بر اساس آئین نامه های مصوب انجام می‌شود.

ماده ۴- ثبت اختراع و تدوین دانش فنی و یا ارائه در جشنواره‌های ملی، منطقه‌ای و بین‌المللی که حاصل نتایج مستخرج از پایان نامه/ رساله و تمامی طرح‌های تحقیقاتی دانشگاه باید با هماهنگی استاد راهنما یا مجری طرح از طریق حوزه پژوهشی دانشگاه انجام گیرد.

ماده ۵- این دستورالعمل در ۵ ماده و یک تبصره در تاریخ ۱۳۸۴/۴/۲۵ در شورای پژوهشی دانشگاه به تصویب رسیده و از تاریخ تصویب لازم الاجرا است و هرگونه تخلف از مفاد این دستورالعمل، از طریق مراجع قانونی قابل پیگیری می شود.

نام و نام خانوادگی: محمد کدخدا

امضاء

دانشگاه تربیت مدرس
دانشکده علوم ریاضی
گروه علوم کامپیوتر

پایان‌نامه کارشناسی ارشد علوم کامپیوتر

مدل اجرایی روش برچسب‌گذاری معنایی برای اثبات خودکار پایان‌پذیری در سیستم‌های بازنویسی ترم

نام دانشجو

محمد کدخدا

استاد راهنما:

دکتر سعید جلیلی

استاد مشاور:

دکتر محمد ایزدی

تیرماه ۱۳۹۰

این تلاش‌نامه ناچیز به پیشگاه امام صادق (ع) تقدیم می‌شود.

سپاس‌گزاری

با سپاس فراوان از راهنمایی‌های جناب آقای دکتر سعید جلیلی که با دقت مثال زدنی مراقب و پاسخگوی ذهن کنجکاو من بودند. و با تشکر از دکتر ایزدی که مشاوره علمی و عملی ایشان برایم بسیار سودمند و گرانبها بوده است. همچنین لازم می‌دانم از آقایان دکتر باقری و دکتر رجایی به خاطر زحماتی که برایم کشیده‌اند قدردانی کنم.

در پایان از مادر و پدر مهربانم که مشوق و پشتیبان من در تحصیل بوده‌اند سپاسگذاری می‌نمایم.

چکیده

سیستم بازنویسی ترم (TRS) یک مدل محاسباتی است که به دلیل قابلیت‌های خوبی که دارد در تحلیل *زبان‌های تابعی* از آن استفاده می‌شود. یک سیستم بازنویسی $\mathcal{R}$ رابطه‌ای روی ترم‌های منطقی مرتبه اول است، بطوریکه برای هر $(l, r) \in \mathcal{R}$ شرایط ، $l \notin \mathcal{V}$ و $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ برقرار باشد. مجموعه $\mathcal{V}ar(t)$ شامل متغیرهای ظاهر شده در ترم t می‌باشد.

پایان‌پذیری یک ویژگی ضروری برای زبان‌های تابعی است. سیستم $\mathcal{R}$ پایان‌پذیر است اگر دنباله‌ی نامتناهی از بازنویسی‌ها تحت آن ساخته نشده یا به عبارتی خوش‌ساخت باشد. از میان رویکردهای مختلف اثبات پایان‌پذیری، رویکردهای *صوری* و *معنایی* قابل توجه‌تر دنبال شده‌اند. روش *برچسب-گذاری معنایی* (SL) که رویکردی معنایی دارد، از قوی‌ترین روش‌های معرفی شده برای اثبات پایان-پذیری است. این روش با برچسب‌گذاری قوانین بازنویسی امکان ساخت یک ترتیب خوش‌ساخت را برای یک TRS فراهم می‌آورد. بخش معنایی SL توسط یک *شبه-مدل* تامین می‌شود. مهمترین مزیت یک روش امکان خودکارسازی آن است. روش SL با اینکه اثبات پایان‌پذیری کلاس بزرگتری از TRS را انجام می‌دهد، برای خودکارسازی چندان مناسب نیست. فقط نوع خاصی از SL که با *مدل‌های متناهی* کار می‌کند، به طور کامل در ابزارهای آزمون پایان‌پذیری مورد استفاده قرار گرفته است.

تلاش‌های انجام گرفته نشان داد که می‌توان این مشکل را برطرف نمود. دو ایده *برچسب‌گذاری پیشگویانه* و *توسعه بینهایتی ترتیب‌های کلاسیک* که اولی با کاهش تعداد قوانین TRS برچسب‌گذاری شده و دومی با قابلیت پشتیبانی مدل‌های نامتناهی کار می‌کند، بطور مؤثری اجرایی شده‌اند.

در این پژوهش در ادامه مسیرهای طی شده دو راه حل دیگر را پیشنهاد کرده‌ایم. راه اول نوعی از توسعه بینهایتی ترتیب کُنُت-بندیکس (KBO) در ترکیب با اعداد طبیعی است که می‌تواند پایان-پذیری یک TRS داده شده را به طور خودکار انجام دهد. ابتدا تعمیمی از KBO به نام ترتیب کُنُت-بندیکس برچسب‌گذاری (ℓKBO) ارائه کرده، سپس توانایی آن را در اثبات پایان‌پذیری TRSs برچسب‌گذاری شده نشان داده‌ایم. جستجوی خودکار یک ℓKBO برای تعدادی از سیستم‌ها که پایان‌پذیری آنها به راحتی اثبات نمی‌شود، انجام گرفته است. راه دوم ارائه یک استنتاج استقرایی است که منطبق با رفتار برچسب‌گذاری معنایی می‌باشد. برای ساخت یک شبه-مدل روی سیستم برچسب-گذاری شده که تعداد نامتناهی قانون بازنویسی دارد استنتاج استقرایی شرایط یک شبه-مدل را توسط مراحل *انتزاع* و *باریک‌سازی* استخراج و برقراری آنها را آزمون می‌کند. جواب مثبت این آزمون پایان-پذیری را نتیجه می‌دهد. در انتها هنوز فرایند خودکار کاملی برای روش SL حاصل نشده و سیستم-های SL پایان‌پذیری هستند که روشهای خودکار SL اثبات پایان‌پذیری آنها را فراهم نمی‌کنند.

واژگان کلیدی: سیستم بازنویسی ترم، پایان‌پذیری، برچسب‌گذاری معنایی، ترتیب کُنُت-بندیکس، استنتاج استقرایی.

فهرست مطالب

عنوان ..... صفحه
فهرست علائم و نشانه‌ها ..... د
فهرست اختصارات ..... ز
فصل اول- کلیات ۱
۱-۱- مقدمه ۱
۱-۲- معرفی سیستم بازنویسی ترم ۱
۱-۳- پایان‌پذیری سیستم‌های بازنویسی ترم ۲
۱-۴- روش برچسب‌گذاری معنایی ۵
۱-۵- نتایج پژوهش ۷
۱-۶- ساختار پایان‌نامه ۸
فصل دوم - مفاهیم پایه ۱۰
۲-۱- مقدمه ۱۰
۲-۲- سیستم بازنویسی ترم ۱۱
۲-۳- انواع سیستم بازنویسی ۲۱
۲-۴- پایان‌پذیری ۲۲
۲-۵- نتیجه‌گیری ۳۰
فصل سوم - ترتیب‌های ساده‌سازی برای اثبات پایان‌پذیری ۳۱
۳-۱- مقدمه ۳۱
۳-۲- روش کُنُت-بندیکس ۳۲
۳-۳- روش ترتیب مسیر ۳۴
۳-۴- تفسیر چندجمله‌ای ۳۸
۳-۵- حذف توزیعی ۳۹
۳-۶- نتیجه‌گیری ۴۱

فصل چهارم - روش‌های معنایی اثبات پایان‌پذیری ۴۲
۴-۱- مقدمه ۴۲
۴-۲- جبر یکنواخت ۴۲
۴-۲-۱- انواع کلاس‌های پایان‌پذیری ۵۱
۴-۳- برچسب‌گذاری معنایی ۵۴
۴-۳-۱- برچسب‌گذاری پیشگویانه ۶۲
۴-۴- روش جفت‌های وابسته ۶۴
۴-۵- جمع‌بندی و مقایسه روش‌ها ۶۸
فصل پنجم- ترتیب کُنُت-بندیکی برچسب‌گذاری ۷۰
۵-۱- مقدمه ۷۰
۵-۲- ترتیب کُنُت-بندیکس برای سیستم‌های برچسب‌گذاری ۷۱
۵-۳- روش اجرایی خودکار SL با ساخت شبه-مدل $KBO\ell$ ۷۶
۵-۴- الگوریتم جستجوی خودکار یک شبه-مدل ۸۰
۵-۵- مثال‌ها ۸۲
۵-۶- نتیجه‌گیری ۸۴
فصل ششم - اثبات استقرائی پایان‌پذیری سیستم‌های برچسب‌گذاری ۸۵
۶-۱- مقدمه ۸۵
۶-۲- استقراء برای پایان‌پذیری به کمک استراتژی محلی ۸۵
۶-۲-۱- مراحل استقراء به کمک استراتژی محلی ۸۶
۶-۳- استنتاج استقرائی برای اثبات پایان‌پذیری سیستم چسب‌گذاری ۸۷
۶-۳-۱- تعریف استنتاج استقرائی ۸۸
۶-۴- اثبات استقرائی مبتنی بر قاعده ۸۹
۶-۴-۱- انتزاع زیر ترم‌ها ۸۹
۶-۴-۲- باریک‌سازی ترم‌ها ۹۰
۶-۵- الگوریتم استقرائی مبتنی بر قاعده ۹۰
۶-۵-۱- انتزاع ۹۱

۶-۵-۲- انتزاع- I ۹۱
۶-۵-۳- انتزاع- S ۹۱
۶-۵-۴- باریکی- Y ۹۱
۶-۵-۵- باریکی- N ۹۲
۶-۵-۶- توقف استقراء ۹۲
۶-۵-۷- توقف- A ۹۲
۶-۵-۸- توقف ۹۲
۶-۶- مثال‌ها ۹۵
۶-۷- نتیجه‌گیری ۹۹
فصل هفتم - نتیجه‌گیری و آینده پژوهش ۱۰۱
۷-۱- مقدمه ۱۰۱
۷-۲- مروری بر پایان‌نامه ۱۰۱
۷-۳- آینده پژوهش ۱۰۳

فهرست علائم و نشانه‌ها

قانون بازنویسی	$\rightarrow$
رابطه بازنویسی	$\rightarrow_{\mathcal{R}}$
بستار بازتابی-تراگذری رابطه بازنویسی	$\rightarrow_{\mathcal{R}}^{*}$, $\twoheadrightarrow$
بستار نرمال رابطه بازنویسی	$\rightarrow_{\mathcal{R}}^{!}$
بستار بازتابی رابطه بازنویسی	$\rightarrow_{\mathcal{R}}^{=}$
بستار تراگذری رابطه بازنویسی	$\rightarrow_{\mathcal{R}}^{+}$
بستار تقارنی رابطه بازنویسی	$\rightarrow_{\mathcal{R}}^{-}$
بستار هم‌ارزری رابطه بازنویسی	$\rightarrow_{\mathcal{R}}^{\sim}$
رابطه بازنویسی محدود به n مرتبه	$\rightarrow_{\mathcal{R}}^{n}$
رابطه بازنویسی محدود به صفر مرحله	$\rightarrow_{\mathcal{R}}^{0}$
رابطه بازنویسی درونی	$\rightarrow_{\mathcal{R}}^{i}$
رابطه بازنویسی بیرونی	$\rightarrow_{\mathcal{R}}^{o}$
رابطه بازنویسی موازی	$\rightarrow_{\mathcal{R}}^{\parallel}$
شکل کلی قوانین بازنویسی	$(l, r), l \rightarrow r$
مجموعه متناهی از تابع نشانه‌ها	$\mathcal{F}$
تابع نشانه‌های ظاهر شده در ترم t	$\mathcal{F}\mathrm{un}(t)$
تابع نشانه‌های با موضع n	$\mathcal{F}^{(n)}$
تابع نشانه‌های برچسب‌گذاری شده	$\mathcal{F}_{lab}$
مجموعه شمارش پذیر از متغیرها	$\mathcal{V}$
متغیرهای ظاهر شده در ترم t	$\mathcal{V}\mathrm{ar}(t)$
ترم‌های ساخته شده توسط تابع نشانه‌های $\mathcal{F}$ و متغیرهای $\mathcal{V}$	$\mathcal{T}(\mathcal{F}, \mathcal{V})$
ترم‌های ساخته شده با موضع n	$\mathcal{T}_n(\mathcal{F}, \mathcal{V})$
ترم‌هایی که یا متغیرند یا ثابت	$\mathcal{T}_0(\mathcal{F}, \mathcal{V})$
ترم‌های بسته	$\mathcal{T}(\mathcal{F})$
سیستم بازنویسی ترم	$\mathcal{R}$
سیستم بازنویسی برچسب‌گذاری شده	$\mathcal{R}_{lab}$
معکوس سیستم بازنویسی ترم	$\mathcal{R}^{-}$
تابع نشانه‌های تعریف شدهٔ $\mathcal{R}$	$\mathcal{D}(\mathcal{R}), \mathcal{D}$
تابع نشانه‌های سازندهٔ $\mathcal{R}$	$\mathcal{C}(\mathcal{R})$, $\mathcal{C}$
مجموعه برچسب‌های تابع‌نشانه f	L_f
مجموعه موقعیت‌های ترم t	$\mathcal{P}\mathrm{os}(t)$
مجموعه تمام موقعیت‌های ممکن	$\mathcal{P}\mathrm{os}$

زیر ترم موقعیت p	$t\vert_p$
مجموعه تمام نشانه‌های ظاهر شده در t	$\mathcal{S}ym(t)$
زمانیکه $\to_R$ یک قانون در سیستم بازنویسی $\mathcal{R}$ باشد	$R \in \mathcal{R}$
اندازه ترم t	$\vert t\vert$
تعداد تکرار متغیر x در ترم t	$\vert t\vert_x$
مجموعه تهی	$\emptyset$
زیر مجموعه	$\subseteq$
تابع جفت‌ساز	o
اجتماع مجزا	$\oplus$
تساوی دو ترم	$\equiv$
اختلاف دو ترم	$\not\equiv$
ثابت حفره	$\Box$
ترم متن	$C[]$
تابع زیرترم موقعیت p	$\vert_p$
تابع نشانه ریشه زیرترم موقعیت p	$()_p$
تابع موضع تابع نشانه‌ها	$ar()$
تابع مشخصه	α
نگاشت جایگذاری	σ
دنباله تهی از اعداد طبیعی	ε
جایگذاری متغیر x_1 با ترم t_1	x_1/t_1
بازنویسی در موقعیت p ترم داده شده توسط قانون $l \to r$ و جایگذاری σ	$\to_{p,\, l \to r,\, \sigma}$
تابع تعیین استراتژی محلی	LS
لیست‌های متناهی از اعداد طبیعی	$L(\mathbb{N})$
تکرار تابع نشانه f به تعداد a روی ترم s یعنی $f(f(\ldots(f(s))\ldots))$	$f^a s$
نماد ریشه ترم t	$root(t)$
ترتیب زیر ترمی و زیر ترمی اکید	$\trianglerighteq, \triangleright$
تقدم و ترتیب پیشوند	$\prec, \succ$
$\approx \wedge \prec$	$\preccurlyeq$
ترتیب	$>$
ترتیب جزئی	$\geq$
تقدم روی $\mathcal{F}_{lab}$	$\succ_\ell$
شبه-ترتیب	$\gtrsim$
تابع وزن	w
تابع وزن برچسب‌گذاری	w_ℓ
شکل نرمال منحصربفرد	UN

نرمال قوی	SN
نرمال ضعیف	WN
مجموعه تابع‌های چندمجموعه‌ای ساخته شده روی A	$M(A)$
استلزام	$\Rightarrow$
یک $\mathcal{F}$-جبر روی مجموعه ناتهی A	$\mathcal{A} = (\mathcal{F}, A)$
تابع تفسیر	$[\![-]\!]_-$
تفسیر ترم t در مدل A با تابع مشخصه α	$[\alpha]_{\mathcal{A}}(t)$
تابع وضعیت برای انتخاب ترتیب تعمیم دهنده برای مقایسه ترم‌های با ریشه f	$\tau(f)$
تابع انتخاب نوع برچسب‌گذاری تابع‌نشانه f	$\pi(f)$
یک برچسب‌گذاری برای جبر داده شده	ℓ
تابع برچسب‌گذاری تابع‌نشانه f	f_ℓ
مجموعه برچسب‌های تابع نشانه f	L_f
تابع نشانه برچسب‌خورده f توسط یک برچسب $a \in L_f$	f_a
قوانین قابل استفاده سیستم بازنویسی $\mathcal{R}$	$\mathcal{U}(\mathcal{R})$
یک رابطه روی $\mathcal{F}$ تحت سیستم $\mathcal{R}$ بطوریکه $f \rhd_d g$ زمانیکه f ریشه سمت یک قانون $\mathcal{R}$ باشد و g یک نماد تعریف شده در سمت راست آن قانون باشد.	$\rhd_d$
مجموعه تابع نشانه‌هایی که با تابع نشانه‌های $\mathcal{F}un(t)$ تحت $\rhd_d^*$ ارتباط دارند	$\mathcal{G}_\ell(t)$
اجتماع $\mathcal{G}_\ell(l) \cup \mathcal{G}_\ell(r)$ برای هر قانون $l \to r \in \mathcal{R}$	$\mathcal{G}_\ell(\mathcal{R})$
مجموعه قوانین قابل استفاده سیستم $\mathcal{R}$ تحت برچسب‌گزاری ℓ	$\mathcal{U}(\ell)$

فهرست اختصارات

ARS	Abstract Reduction System
CL	Combinatorial logic
CR	Church-russel
DP	Dependency pair method
ES	Equational system
KBO	Knuth-Bendix order
KBG	Knuth-Bendix graph
ℓKBO	Labeling Knuth-Bendix order
LPO	Lexicographic path order
MA	Monotonic Algebra
MPO	Multiset path order
PI	Polynomial Interpretation
RDO	Recursive decomposition order
RPO	Recursive path order
RTA	Rewriting technique and application
SL	Semantic labeling method
SN	Strong normalizing
SPO	Semantic path order
TRS	Term rewriting system
UN	Unique normal form
UR	Usable rule
WCC	Weakly Common Component
WN	Weakly normalizing

فصل اول

کلیات

۱-۱- مقدمه

این فصل در پنج بخش تنظیم شده است. در بخش دوم تصویری کلی از سیستم‌های بازنویسی ارائه می‌شود. بخش سوم مساله پایان‌پذیری سیستم‌های بازنویسی و روش‌های حل آن را مرور می‌کند. بخش چهارم به شناسایی روش برچسب‌گذاری معنایی در اثبات پایان‌پذیری می‌پردازد؛ در این بخش مشکلات این روش برای خودکارسازی بیان و راه‌حل‌هایی که تاکنون ارائه شده معرفی می‌شود. بخش پنجم نیز به شرح خلاصه‌ای از کارها و نتایج این پژوهش و جمع‌بندی آنها می‌پردازد. بخش ششم نیز ساختار پایان‌نامه را مرور کرده است.

۱-۲- معرفی سیستم بازنویسی ترم

سیستم *بازنویسی ترم*[۱] (TRS) یک مدل محاسباتی غیرقطعی است که *برنامه نویسی توصیفی*[۲] را پشتیبانی می‌کند. بیشتر توانایی یک TRS به ارائه‌ی نمایش انتزاعی از زبان‌های تابعی و سیستم *دستکاری صوری*[۳] ساختارها مربوط است. شاید *حساب* λ[۴] مشهورترین نوع یک سیستم بازنویسی ترم باشد، که نقش مهمی در منطق ریاضی برای رسمی ساختن مفهوم محاسبه دارد. سیستم *منطق ترکیبیاتی*[۵] (CL) و انواع دیگر آن مانند CCL[۶] هر یک خانواده دیگری از TRS بوده که برای ساخت ابزاری که بتواند نمایش کاملی

---

1 Term rewriting system
2 Declarative programming
3 Formal manipulation system
4 Calculus
5 Combinatorial logic
6 Categorical combinatory logic

از زبان‌های تابعی ارائه دهد بکار می‌روند [۳۶]. نظریه سیستم‌های بازنویسی به دلیل داشتن زبان و معنای ساده و استفاده از متغیرها بطور آزاد برخلاف حساب λ که متغیر وابسته نیز دارد، کارایی محاسباتی بالایی را در منطق مرتبه اول ایجاد کرده است.

امروزه TRS بعنوان ابزاری برای تحلیل و بهینه‌سازی و نیز *وارسی*[۱] *برنامه‌های منطقی*[۲] استفاده می‌شود. بطور خاص سیستم بازنویسی در زمینه‌هایی مانند بررسی ویژگی *انوع داده مجرد*[۳]، استنباط خودکار، *اثبات خودران قضایا*[۴]، معناشناسی *زبان‌های برنامه‌نویسی*[۵] و وارسی *زبانهای تابعی*[۶] نقش مهمی دارد [۵۲،۳۶،۱۱،۳]. این سیستم‌ها به دلیل ویژگی‌های بارز منطقی خود کاربردهای متنوع دیگری نیز یافته‌اند [۳].

شاید بتوان آغاز رشد و تکامل نظریه بازنویسی را به کارهای انجام شده توسط *کُنُت و بندیکس*[۷] [۳۷]، برای حل مساله معروف Word در جبر جهانی مربوط دانست. روش ارائه شده توسط آنها که فرایند کُنُت-بندیکس نام گرفته با گرفتن یک سیستم معادلات ترم؛ مجموعه‌ای متناهی از *اصل‌ها*[۸]، به همراه یک ترتیب روی ترم‌ها، بعنوان ورودی، می‌تواند به طورخودکار مساله Word را تحت شرایط همگرایی و پایان-پذیری حل کند. اگر چه این فرایند نمی‌تواند بطور مثال سیستم‌های شامل معادله جابجایی ($x+y = y+x$)، را بپذیرد، و سیستم پایان‌ناپذیر $y+x \rightarrow x+y$ را شامل شود، اما همچنان نتایج نظری و استفاده‌های عملی این فرایند در زمینه اثبات خودران قضایا بسیار با اهمیت است [۵۳].

۱-۳- پایان‌پذیری سیستم‌های بازنویسی ترم

سیستم بازنویسی ترم علاوه بر ارائه کامل یک زبان تابعی، در تحلیل مسایل زیادی پیرامون زبان‌های برنامه‌نویسی استفاده شده است. یکی از مهمترین این مسایل *پایان‌پذیری*[۹] است. مساله پایان‌پذیری به

1 Verification
2 Logical programming
3 Abstract data type
4 Automated deduction
5 Programming language semantics
6 Functional language
7 Knuth & Bendix
8 Axiom
9 Termination

تایید درستی عملکرد برنامه‌ها ارتباط پیدا می‌کند و در حالت کلی تصمیم‌ناپذیر است [۴۲،۳۲]. با این وجود برای سیستم‌های مختلفی قابل بررسی و اثبات می‌باشد [۱۰].

پایان‌پذیری مهمترین بحث در تئوری سیستم‌های بازنویسی است. آغاز تحقیق پیرامون آن با ایجاد روش‌های متعددی برای اثبات این ویژگی همراه بوده است. اولین تلاش‌ها ارائه‌ی یک ترتیب کاهشی سازگار با سیستم بازنویسی مورد بررسی بوده است [۱۱]. تلاش برای اثبات این ویژگی روی یک سیستم بازنویسی $\mathcal{R}$ با قوانین ثابت و مشخص با ارائه یک ترتیب کاهشی خوش‌ساخت که با آن سازگار باشد انجام می‌شود. تحت ترتیب کاهشی سازگار هر گام بازنویسی یک ترم را به ترمی کوچکتر از ترم اولیه بازنویسی می‌کند. در نتیجه $\mathcal{R}$ پایان‌پذیر خواهد بود. ترتیب‌های متنوعی در طول چندین سال معرفی شدند؛ *ترتیب کُنُت-بندیکس*[۱] (KBO)[۳۷]، *ترتیب مسیر بازگشتی*[۲] (RPO) [۹،۷]، *ترتیب مسیر قاموسی*[۳] (LPO) [۳۴]، *ترتیب تجزیه بازگشتی*[۴] (RDO) [۳۳]، و *ترتیب مسیر چندمجموعه‌ای*[۵] (MPO) [۸] از آن جمله-اند. تمام این ترتیب‌ها بر اساس شکل و ساختار صوری سیستم‌های بازنویسی ساخته می‌شوند. بررسی کامل خواص این دسته از ترتیب‌ها توسط درشوویتز [۱۰،۷] تحت عنوان *ترتیب‌های ساده‌سازی*[۶] انجام شده است. ایده ترتیب‌های ساده‌سازی توانست تا مدت‌ها زمینه‌ای را برای ساخت ترتیب‌های کاهشی فراهم کند [۴۹]. امکان جستجوی خودکار یک ترتیب، مهمترین مزیت آن است. و تمام ترتیب‌های ساده-سازی این قابلیت را دارند [۶۸،۲۲،۱۳]. با این وجود در اثبات پایان‌پذیری انواع زیادی از سیستم‌های بازنویسی پایان‌پذیر محدودیت دارند[۶۰،۲]. برای مثال سیستم بازنویسی $\mathcal{R}_1$ را با قوانین زیر در نظر بگیرید:

$$\begin{array}{ll} f(i(x)) \to f(h(x)) & i(a) \to b \\ g(h(x)) \to g(i(x)) & h(a) \to b \end{array}$$

تمام ترتیب‌های ساده‌سازی در اثبات پایان‌پذیری این سیستم شکست می‌خورند [۹]. تقدم هر ترتیب ساده‌سازی یا i را قبل از h قرار می‌دهد یا برعکس. در هر دوحالت یکی از قوانین بازنویسی نامرتب باقی

1 Knuth-Bendix order
2 Recursive path order
3 Lexicographic path order
4 Recursive decomposition order
5 Multiset path order
6 Simplification order

خواهد ماند. توان پایین ترتیب‌های ساده‌سازی همواره آنها را هدف توسعه و تعمیم قرار داده است. روش-های *تبدیلی*[1] پس از مدتی برای این افزایش توان ارائه شدند. این روش‌ها رویکردی معنایی داشته و از مدلی برای تعبیر یا تفسیر سیستم بازنویسی استفاده می‌کنند. دراین روش‌ها ابتدا TRS اصلی به یک TRS دیگر تبدیل، و آزمون پایان‌پذیری روی سیستم جدید انجام می‌گیرد. از بین روش‌های تبدیلی، *برچسب‌گذاری معنایی*[2] (SL) [۶۷] و *جفت‌های وابسته*[3] (DP) [۱] از دیگران مشهورترند. روش DP از پیشرفته ترین روش‌هاست که بطور موفق خودکارسازی شده است [۲۹،۲۱]. کامل بودن این روش هنوز اثبات نشده است، علاوه بر این روش DP ظرفیت استفاده از جنبه‌های معنایی مختلف یک سیستم بازنویسی را ندارد. با این وجود راهی برای استفاده از تکنیک‌های گراف و حل مساله ارضاءپذیری در اثبات پایان‌پذیری باز می‌کند.

اغلب روش‌های تبدیلی رویکردی *معنایی*[4] داشته و علاوه‌بر اطلاعات صوری مربوط به پایان‌پذیری، به جنبه‌های تفسیری مرتبط با پایان‌پذیری نیز می‌پردازند. بطور کلی مقایسه رویکردهای *صوری* و *معنایی* نشان می‌دهد روش‌های صوری عموماً ساده، *مستقیم*، قابل خودکارسازی کامل و دارای توان پایین در اثبات پایان‌پذیری انواع سیستم‌ها بوده و روش‌های معنایی شامل پیچیدگی‌های خاص، *تبدیلی* و عدم امکان کامل خودکارسازی اما دارای توان بسیار بالاتری هستند. ارزیابی و مقایسه روش‌های اثبات پایان-پذیری بر پایه ملاک‌هایی همچون قدرت اثبات، سادگی، ایستا یا پویا بودن، قابلیت خودکارسازی، سرعت اثبات خودکار و پیچیدگی محاسباتی روش انجام می‌شود.

از سال ۲۰۰۴ میلادی در کنار پانزدهمین کنفرانس سالانه سیستم‌های بازنویسی[5]RTA، یک رقابت جهانی بین نرم‌افزارهای آزمون پایان‌پذیری آغاز شده است. کمیته برگزاری این رقابت‌ها پروژه CoLoR را با هدف تولید ابزاری برای تایید نتیجه‌های اثبات خودکار نرم‌افزارهای شرکت کننده انجام داده است[6]. همچنین پایگاه داده‌ای از انواع سیستم‌های بازنویسی که مورد آزمون نرم‌افزارهای پایان‌پذیری قرار می-

1 transformational
2 semantic labeling method
3 dependency pair method
4 semantical
5 Rewriting technique and application
6 http://color.loria.fr/

گیرند، در دسترس قرار گرفته است[1]. تحلیل نتایج این رقابت هر ساله به بهبود ابزارهای آزمون خودکار پایان‌پذیری کمک زیادی می‌کند.

۱-۴- روش برچسب‌گذاری معنایی

برچسب‌گذاری معنایی یا به اختصار SL، یک روش تبدیلی در اثبات پایان‌پذیری سیستم‌های بازنویسی ترم است. سیستم تبدیل یافته و سیستم اصلی با یکدیگر هم‌ارزی پایان‌پذیری دارند. این روش ابتدا هر تابع‌نشانه $f \in \mathcal{F}$ تحت جبر $\mathcal{A} = (\mathcal{F}, A)$ به تابع f_A تفسیر شده سپس از مقادیر تفسیر برای برچسب-گذاری تابع‌نشانه f استفاده می‌شود. بخش معنایی SL با یک *شبه-مدل*[2] تامین می‌شود. شبه-مدل یک دامنه برای تفسیر تابع‌نشانه‌هاست، که با یک ترتیب خوش‌ساخت مرتب شده و با سیستم بازنویسی مورد بررسی سازگاری دارد. این سازگاری به معنای شامل بودن رابطه بازنویسی در ترتیب خوش ساخت است. روش SL از جنبه نظری پایان‌پذیری کلاس بزرگتری از سیستم‌های بازنویسی را اثبات می‌کند [۶۷]. پیاده‌سازی خودکار SL در عمل مشکلات زیادی دارد که مهمترین آن استفاده از مدل‌های نامتناهی برای انجام برچسب‌گذاری است. بطور مثال اگر در سیستم $\mathcal{R}_1$ تابع‌نشانه‌های i و h به توابع طبیعی $i_{\mathbb{N}}(n) = n+1$ و $h_{\mathbb{N}}(n) = n$ تفسیر شوند و تابع برچسب‌گذاری f، *همانی*[3] در نظر گرفته شود، تبدیل برچسب‌گذاری، سیستم $\mathcal{R}_1$ که تنها چهار قانون دارد را به سیستمی با بینهایت قانون به شکل زیر نگاشت می‌کند.

$$n \in \mathbb{N} \text{ زمانیکه } f_{n+1}(i(x)) \to f_n(h(x))$$

برای اثبات کننده‌های خودکار زمانیکه یک دامنه نامتناهی مانند *اعداد طبیعی* برای برچسب‌گذاری استفاده شود کنترل جنبه‌های پایان‌پذیری سیستم تولید شده کار بسیار سختی است. فقط نوع خاصی از SL که از دامنه متناهی برای برچسب‌گذاری استفاده می‌کند بطور کامل در ابزارهای اثبات خودکار پایان-پذیری مورد استفاده قرار گرفته است. تلاش‌هایی برای ارائه مدل اجرایی کاملی از SL انجام شده تا بتوان SL را در اثبات پایان‌پذیری بطور خودکار و کامل پیاده‌سازی نمود.

---

1 http://www.lri.fr/~marche/tpdb.
2 Quasi-model
3 Identity