INSO
16301
1st. Edition

May.2013

جمهوری اسلامی ایران
Islamic Republic of Iran

سازمان ملی استاندارد ایران

Iran National Standardization Organization

استاندارد ملی ایران
۱۶۳۰۱
چاپ اول
اردیبهشت ۱۳۹۲

فناوری اطلاعات - منطق مشترک(CL): یک چارچوب کاری برای خانواده‌ای از زبان‌های مبتنی بر منطق

Information Technology – Common Logic (CL): A Framework for a Family of Logic-Based Languages

ICS: 35.060

به‌نام خدا

آشنایی با سازمان ملی استاندارد ایران

مؤسسهٔ استاندارد و تحقیقات صنعتی ایران به موجب بند یک مادهٔ۳ قانون اصلاح قوانین و مقررات مؤسسهٔ استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱ تنها مرجع رسمی کشور است که وظیفه تعیین، تدوین و نشر استانداردهای ملی (رسمی) ایران را به عهده دارد.

نام موسسه استاندارد و تحقیقات صنعتی ایران به موجب یکصد و پنجاه و دومین جلسه شورای عالی اداری مورخ ۹۰/۶/۲۹ به سازمان ملی استاندارد ایران تغییر و طی نامه شماره ۲۰۶/۳۵۸۳۸ مورخ ۹۰/۷/۲۴ جهت اجرا ابلاغ شده است .

تدوین استاندارد در حوزه های مختلف در کمیسیون های فنی مرکب از کارشناسان سازمان ، صاحب نظران مراکز و مؤسسات علمی، پژوهشی، تولیدی و اقتصادی آگاه و مرتبط انجام می شود وکوششی همگام با مصالح ملی و با توجه به شرایط تولیدی، فناوری و تجاری است که از مشارکت آگاهانه و منصفانهٔ صاحبان حق و نفع، شامل تولیدکنندگان، مصرف‌کنندگان، صادرکنندگان و وارد کنندگان، مراکز علمی و تخصصی، نهادها، سازمان‌های دولتی و غیر دولتی حاصل می شود .پیش نویس استانداردهای ملی ایران برای نظرخواهی به مراجع ذی نفع و اعضاء کمیسیون‌های فنی مربوط ارسال می‌شود و پس از دریافت نظرها و پیشنهادها در کمیتهٔ ملی مرتبط با آن رشته طرح و در صورت تصویب به عنوان استاندارد ملی (رسمی) ایران چاپ و منتشر می شود.

پیش نویس استانداردهایی که مؤسسات و سازمان های علاقه مند و ذی صلاح نیز با رعایت ضوابط تعیین شده تهیه می کنند درکمیتهٔ ملی طرح و بررسی و درصورت تصویب، به عنوان استاندارد ملی ایران چاپ و منتشر می شود .بدین ترتیب، استانداردهایی ملی تلقی می شوند که بر اساس مفاد نوشته شده در استاندارد ملی ایران شمارهٔ ۵ تدوین و در کمیتهٔ ملی استاندارد مربوط که سازمان ملی استاندارد ایران تشکیل می‌دهد به تصویب رسیده باشد.

سازمان ملی استاندارد ایران از اعضای اصلی سازمان بین المللی استاندارد (ISO)[1] ،کمیسیون بین المللی الکتروتکنیک (IEC)[2] و سازمان بین المللی اندازه شناسی قناونی (OIML)[3] است و به عنوان تنها رابط[4] کمیسیون کدکس غذایی (CAC)[5] در کشور فعالیت می کند. در تدوین استانداردهای ملی ایران ضمن توجه به شرایط کلی و نیازمندی های خاص کشور ، از آخرین پیشرفت های علمی، فنی و صنعتی جهان و استانداردهای بین‌المللی بهره‌گیری می‌شود .

سازمان ملی استاندارد ایران می تواند با رعایت موازین پیش بینی شده در قانون، برای حمایت از مصرف کنندگان، حفظ سلامت و ایمنی فردی و عمومی، حصول اطمینان از کیفیت محصولات و ملاحظات زیست محیطی و اقتصادی، اجرای بعضی از استانداردهای ملی ایران را برای محصولات تولیدی داخل کشور و/یا اقلام وارداتی، با تصویب شورای عالی استاندارد، اجباری نماید. سازمان می تواند به منظور حفظ بازارهای بین المللی برای محصولات کشور، اجرای استاندارد کالاهای صادراتی و درجه‌بندی آن را اجباری نماید. همچنین برای اطمینان بخشیدن به استفاده کنندگان از خدمات سازمان ها و مؤسسات فعال در زمینهٔ مشاوره، آموزش، بازرسی، ممیزی و صدورگواهی سیستم های مدیریت کیفیت و مدیریت زیست‌محیطی، آزمایشگاه ها و مراکز کالیبراسیون (واسنجی) وسایل سنجش، سازمان ملی استاندارد ایران این گونه سازمان ها و مؤسسات را بر اساس ضوابط نظام تأیید صلاحیت ایران ارزیابی می کند و در صورت احراز شرایط لازم، گواهینامهٔ تأیید صلاحیت به آن ها اعطا و بر عملکرد آن ها نظارت می کند. ترویج دستگاه بین المللی یکاها، کالیبراسیون (واسنجی) وسایل سنجش، تعیین عیار فلزات گرانبها و انجام تحقیقات کاربردی برای ارتقای سطح استانداردهای ملی ایران از دیگر وظایف این سازمان است.

1- International Organization for Standardization
2- International Electrotechnical Commission
3- International Organization of Legal Metrology (Organisation Internationale de Metrologie Legale)
4- Contact point
5- Codex Alimentarius Commission

کمیسیون فنی تدوین استاندارد

«فناوری اطلاعات - منطق مشترک(CL): یک چارچوب کاری برای خانواده‌ای از زبان‌های مبتنی بر منطق»

رئیس:	سمت و/ یا نمایندگی
میرزایی رضایی، طیبه (فوق لیسانس فیزیک)	رئیس اداره تدوین استاندارد ها و نظارت بر فرآیند سرویس‌ها سازمان فناوری اطلاعات
دبیر:	
میراسکندری، سید محمدرضا (لیسانس مهندسی کامپیوتر نرم افزار)	مدیر کل خدمات ارزش افزوده سازمان فناوری اطلاعات
اعضاء: (اسامی به ترتیب حروف الفبا)	
بختیاری، شیرین (لیسانس مهندسی برق کنترل)	کارشناس تدوین استاندارد سازمان فناوری اطلاعات ایران
جمیل پناه، ناصر (فوق لیسانس مدیریت)	کارشناس تدوین استاندارد سازمان فناوری اطلاعات ایران
سعیدی، عذرا (فوق لیسانس مهندسی برق-مخابرات)	کارشناس تدوین استاندارد سازمان فناوری اطلاعات ایران
سلطانی حقیقت، الهه (لیسانس مهندسی برق مخابرات)	کارشناس سازمان فناوری اطلاعات ایران
عبداللهی ازگمی، محمد (دکترای مهندسی کامپیوتر-نرم‌افزار)	استادیار دانشگاه علم و صنعت ایران
فرهاد شیخ احمد، لیلا (فوق لیسانس مهندسی کامپیوتر نرم افزار)	کارشناس تدوین استاندارد سازمان فناوری اطلاعات ایران
فولادیان، مجید (فوق لیسانس مهندسی برق-مخابرات)	مشاور سازمان فناوری اطلاعات ایران
فیاضی، مهدی (لیسانس مهندسی برق الکترونیک)	کارشناس مسؤول تدوین استاندارد و امنیت شبکه
قسمتی، سیمین (فوق لیسانس فناوری اطلاعات)	کارشناس سازمان فناوری اطلاعات ایران

کبیری، پیمان (دکترای مهندسی کامپیوتر)	استادیار دانشگاه علم و صنعت ایران
مجاهدی، الناز (لیسانس مهندسی کامپیوتر نرم‌افزار)	نماینده دانشگاه علم و صنعت ایران
معروف، سینا (لیسانس مهندسی کامپیوتر سخت‌افزار)	کارشناس سازمان فناوری اطلاعات ایران
ناظمی، اسلام (دکترای مهندسی کامپیوتر)	استادیار دانشگاه شهید بهشتی

فهرست مندرجات

عنوان **صفحه**

آشنایی با سازمان ملی استاندارد ب
کمیسیون فنی تدوین استاندارد ج
پیش‌گفتار ز
مقدمه ح
۱ هدف و دامنه کاربرد ۱
۲ مراجع الزامی ۲
۳ اصطلاحات و تعاریف ۲
۴ نمادها و کوته‌نوشت‌ها ۸
۴-۱ نمادها ۸
۴-۲ کوته‌نوشت‌ها ۹
۵ الزامات و مرور کلی طراحی ۱۰
۵-۱ الزامات ۱۰
۵-۲ خانواده‌ای از نشانه‌گذاری‌ها ۱۱
۶ نحو انتزاعی و معناشناسی منطق مشترک ۱۲
۶-۱ نحو انتزاعی منطق مشترک ۱۲
۶-۱-۱ رده‌بندی‌های نحو انتزاعی ۱۲
۶-۱-۲ مدل مفهومی نحو انتزاعی منطق مشترک ۱۴
۶-۱-۳ ساختار نحو انتزاعی گویش‌ها ۱۸
۶-۲ معناشناسی منطق مشترک ۲۰
۶-۳ واردسازی و شناسایی در یک شبکه ۲۳
۶-۳-۱ واردهها و عبارت‌های نام‌دار ۲۳
۶-۳-۲ شبکه‌های مرکب ۲۶
۶-۴ برقراری، اعتبار و استلزام ۲۷
۶-۵ نشان‌گرهای دنباله، بازگشت و لیست‌های آرگومان ۲۸
۶-۶ موارد ویژه و ترجمه‌های بین گویش‌ها ۲۹
۶-۶-۱ ترجمه بین گویش‌ها ۳۰
۷ انطباق ۳۱
۷-۱ انطباق گویشی ۳۱

۷-۱-۱ نحو ۳۱
۷-۱-۲ معناشناسی ۳۲
۷-۲ انطباق برنامه‌ای ۳۴
۷-۳ انطباق شبکه‌ای ۳۵
پیوست الف: قالب تبادل منطق مشترک (CLIF) (الزامی) ۳۶
الف-۱ مقدمه ۳۶
الف-۲ نحو CLIF ۳۸
الف-۳ معناشناسی CLIF ۴۷
الف-۴ انطباق CLIF ۴۹
پیوست ب: قالب تبادل گراف مفهومی (CGIF) (الزامی) ۵۲
ب-۱ مقدمه ۵۲
ب-۲ نحو و معناشناسی هسته‌ی CG ۶۱
ب-۲-۳ دنباله‌ی کمان (arcSequence) ۶۲
ب-۲-۱۰ دامنه‌ی کاربرد ۶۶
ب-۳ نحو CGIF گسترش‌یافته ۶۸
ب-۳-۲ کمان ۷۰
ب-۳-۸ متن ۷۶
ب-۳-۹ عبارت نوع ۷۶
ب-۴ انطباق CGIF ۷۷
پیوست پ: زبان علامت‌گذاری منطق مشترک گسترش‌پذیر (XCL) (الزامی) ۸۱
پ-۱ مقدمه ۸۱
پ-۲ نحو XCL ۸۱
پ-۳ معناشناسی XCL ۱۰۵
پ-۴ انطباق XCL ۱۰۵
کتابنامه ۱۰۶

پیش‌گفتار

استاندارد «فناوری اطلاعات - منطق مشترک(CL): یک چارچوب کاری برای خانواده‌ای از زبان‌های مبتنی بر منطق» که پیش‌نویس آن درکمیسیون‌های مربوط توسط سازمان فناوری اطلاعات تهیه و تدوین شده و در دویست و چهارمین اجلاس کمیتهٔ ملی استاندارد رایانه و فرآوری داده مورخ ۱۳۹۱/۱۰/۳مورد تصویب قرار گرفته است، اینک به استناد بند یک مادهٔ ۳ قانون اصلاح قوانین و مقررات سازمان استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱، به عنوان استاندارد ملی ایران منتشر می‌شود.

برای حفظ همگامی و هماهنگی با تحولات و پیشرفت‌های ملی و جهانی در زمینهٔ صنایع، علوم و خدمات، استانداردهای ملی ایران در مواقع لزوم تجدید نظر خواهد شد و هر پیشنهادی که برای اصلاح و تکمیل این استانداردها ارائه شود، هنگام تجدید نظر در کمیسیون فنی مربوط مورد توجه قرار خواهد گرفت. بنابراین، باید همواره از آخرین تجدیدنظر استانداردهای ملی استفاده کرد.

منبع و مأخذی که برای تهیهٔ این استاندارد مورد استفاده قرار گرفته به شرح زیر است:

ISO/IEC 24707-2007: Information Technology – Common Logic (CL): a framework for a family of logic-based languages

مقدمه

منطق مشترک[۱] یک چارچوب کاری منطقی[۲] برای تبادل و انتقال اطلاعات است. این چارچوب کاری انواع مختلف شکل‌های نحوی[۳] تحت عنوان گویش[۴] را فراهم می‌کند که تمام آن‌ها با استفاده از یک نحو مشترک مبتنی بر زبان علامت‌گذاری گسترش‌پذیر (XML)[۵] قابل بیان بوده و همه در یک معناشناسی واحد، سهیم هستند.

منطق مشترک دارای ویژگی‌های بدیعی[۶] است که از جمله مهم‌ترین آن‌ها می‌توان به یک نحو اشاره کرد که عاری از امضاء[۷] بوده و در حالی که یک نظریه‌ی مدل مرتبه اول[۸] را حفظ می‌کند، امکان استفاده از ترکیب‌های 'مرتبه بالاتر' مانند استفاده از سور[۹ ۱۰] بر روی کلاس‌ها و روابط را نیز فراهم می‌کند. همچنین معناشناسی منطق مشترک، به نظریه‌ها، اجازه توصیف هستارهای متکثر[۱۱] مانند کلاس‌ها و خصوصیت‌ها را می‌دهد. در ضمن، این منطق معانی تعدادی از قراردادها را که به صورت گسترده استفاده می‌شوند (مانند استفاده از اعداد برای مشخص کردن اعداد صحیح یا استفاده از علامت نقل قول برای مشخص کردن رشته‌های نویسه‌ای)[۱۲] اصلاح کرده و امکان استفاده از انواع داده و نیز نام‌گذاری، واردسازی[۱۳] و انتقال محتوا در تار جهان‌گستر[۱۴] را با استفاده از XML فراهم می‌کند.

1 - Common logic
2 - Logic framework
3 - Syntactic forms
4 - Dialect
5 - eXtensible Markup Language
6 - Novel
7 - Signature-free
8 - First-order model
9 - Quantification

۱۰ - منظور استفاده از سور عمومی (∀) یا سور وجودی (∃) با کلاس‌ها است.

11 - Intensional entities
12 - Character strings
13 - Importing
14 - World Wide Web

فناوری اطلاعات - منطق مشترک(CL): یک چارچوب کاری برای خانواده‌ای از زبان‌های مبتنی بر منطق

۱ هدف و دامنه کاربرد

هدف از تدوین این استاندارد، تصریح خانواده‌ای از زبان‌های منطق است که برای نمایش و تبادل داده و اطلاعات بین سامانه‌های کامپیوتری مختلف طراحی شده‌اند.

ویژگی‌های زیر الزامات طراحی برای این استاندارد به شمار می‌روند:

- زبان‌های عضو خانواده دارای معناشناسی اعلانی[۱] هستند و فهم معانی عبارت‌ها در این زبان‌ها بدون نیاز رجوع به یک مفسر برای فرابری آن‌ها امکان‌پذیر است.
- زبان‌های عضو خانواده به‌طور منطقی فراگیر هستند. این زبان‌ها به‌طور کلی امکان بیان جمله‌های منطقی مرتبه اول دلخواه را فراهم می‌کنند.
- تبادل اطلاعات بین سامانه‌های کامپیوتری نامتقارن صورت می‌گیرد.

موارد زیر در دامنه‌ی کاربرد این استاندارد ملی قرار می‌گیرند:

- نمایش اطلاعات در هستان‌شناسی‌ها[۲] و پایگاه‌های دانش،
- تصریح عبارت‌هایی که ورودی یا خروجی موتورهای استنتاج[۳] هستند،
- تفسیر صوری[۴] نمادهای زبان.

موارد زیر خارج از دامنه‌ی کاربرد این استاندارد هستند:

- تصریح نظریه‌ی اثبات[۵] و قواعد استنتاج،
- تصریح مترجم‌های بین نشانه‌گذاری‌های سامانه‌های کامپیوتری نامتقارن،
- روش‌های عملیاتی کامپیوتر - مبنا که ارتباط بین نمادها در «جهان گفتمان[۶]» استدلالی و افراد در «جهان واقعی[۷]» را برقرار می‌کنند.

این استاندارد ملی نحو و معناشناسی منطق مشترک را توصیف می‌کند.

این استاندارد یک نحو انتزاعی و یک معناشناسی مبتنی بر مدل نظری[۸] مرتبط با آن نحو را برای یک گسترش[۹] ویژه از منطق مرتبه اول[۱۰] تعریف می‌کند. انگیزه این است که محتوای هر سامانه‌ای که از منطق

---

1 - Declarative semantics
2 - Ontologies
3 - Inference engines
4 - Formal
5 - Proof theory
6 - Universe of discourse
7 - Real world
8 - Model-theoretic
9 - Extension
10 - First-order logic

مرتبه اول استفاده می‌کند قابل نمایش در این استاندارد ملی باشد. هدف، تسهیل تبادل اطلاعات مبتنی بر منطق مرتبه اول بین سامانه‌هاست.
در این استاندارد به مسائل مربوط به محاسبه‌پذیری[1] با استفاده از این استاندارد (مانند کارآمدی[2]، بهینه‌سازی و غیره) پرداخته نشده است.

۲ مراجع الزامي

مدارک الزامی زیر حاوی مقرراتی است که در متن این استاندارد ملی ایران به آن‌ها ارجاع داده شده است. بدین ترتیب آن مقررات جزئی از این استاندارد ملی ایران محسوب می‌شود.
در صورتی که به مدرکی با ذکر تاریخ انتشار ارجاع داده شده باشد، اصلاحیه‌ها و تجدید نظرهای بعدی آن مورد نظر این استاندارد ملی نیست. در مورد مدارکی که بدون ذکر تاریخ انتشار به آن‌ها ارجاع داده شده است، همواره تاریخ تجدید نظر و اصلاحیه‌های بعدی آن‌ها مورد نظر است.
استفاده از مراجع زیر برای این استاندارد الزامی است:

2-1 ISO/IEC 2382-15:1999, Information technology — Vocabulary — Part 15: Programming languages

2-2 ISO/IEC 10646:2003, Information technology — Universal Multiple-Octet Coded Character Set (UCS)

2-3 ISO/IEC 14977:1996, Information technology — Syntactic metalanguage — Extended BNF

۳ اصطلاحات و تعاریف

در این استاندارد اصطلاحات و تعاریف زیر به‌کار می‌رود:

۳-۱

تجزیه‌ناپذیر[3]

شکل جمله‌ای[4] است که هیچ زیرجمله‌ای به عنوان مؤلفه‌ی نحوی نداشته باشد.

یادآوری - تجزیه‌ناپذیر می‌تواند یک تساوی یا یک جمله‌ی تجزیه‌ناپذیر متشکل از یک مسند اعمال شده به یک دنباله‌ی آرگومان باشد.

۳-۲

اصل بدیهي[5]

هر جمله‌ای که درست فرض می‌شود و جمله‌های دیگر از آن مشتق شده یا متضمن[6] آن‌هاست.

1 - Computability
2 - Efficiency
3 - Atom
4 - Sentence form
5 - Axiom
6 - Entailed

یادآوری - در یک تنظیم محاسباتی، اصل بدیهی جمله‌ای است که هدف اثبات آن نیست، بلکه تنها برای اثبات سایر جمله‌ها استفاده می‌شود.

۳ـ۳
قالب تبادل منطق مشترک
(CLIF) [1]

نحو مبتنی بر KIF[2] در این استاندارد به منظور تشریح[3] استفاده شده است.

یادآوری - این نحو یکی از نحوهای محسوس[4] است که در پیوست الف شرح داده شده است. نام KIF در این استاندارد با هدف تمایز آن با گویش‌های رایج KIF استفاده نشده است. در این استاندارد هیچ فرضی در مورد معناشناسی KIF نشده است. در حقیقت، هیچ گونه هم‌ارزی بین CLIF و KIF مدنظر نیست.

٤ـ۳
گراف مفهومي
(CG) [5]

نمایش گرافیکی یا متنی نمادها که به سبک نظریه‌ی گراف مفهومی چیده شده‌اند.

۵ـ۳
قالب تبادل گراف مفهومي
(CGIF)[6]

نسخه‌ی متنی گراف‌های مفهومی که قواعد ساخت آن‌ها مطابق پیوست ب این استاندارد است.

یادآوری - این قالب در بعضی موارد ممکن است به نمونه‌ای از یک رشته‌ی حرفی اشاره کند که منطبق بر پیوست ب است. هدف از این قالب، انتقال ساختار و معنای به‌طور کامل یکسان با یک گراف مفهومی هم‌ارز است.

٦ـ۳
نظریه‌ي گراف مفهومي[7]

صورت‌بندی منطق مرتبه اول که سور وجودی[8] و ترکیب عطفی را به‌وسیله‌ی بیان ساختارهای منطقی که به آن‌ها مفاهیم و روابط گفته می‌شود و در یک گراف انتزاعی یا دیداری نمایش دهد، چیده شده‌اند.

یادآوری - نظریه‌ی گراف مفهومی در [1] معرفی شده است.

۷ـ۳
معناي صريح[9]

1 - Common Logic Interchange Format
2 - Knowledge Interchange Format
3 - Illustration
4 - Concrete syntaxes
5 - Conceptual graph
6 - Conceptual Graph Interchange Format
7 - Conceptual graph theory
8 - Existential quantification
9 - Denotation

رابطه‌ای که بین یک نام یا عبارت و چیزی که به آن اشاره می‌کند برقرار است.

۸-۳
گویش[1]
نمونه‌ی محسوس نحو منطق مشترک که (حداقل بخشی از) معناشناسی یکپارچه‌ی منطق مشترک را به اشتراک می‌گذارد.

یادآوری - یک گویش می‌تواند متنی یا گرافیکی و یا به احتمال به صورتی دیگر باشد. بر طبق تعریف، یک گویش همچنین یک زبان منطبق بر منطق مشترک است (مطابق بند ۷-۱).

۹-۳
نام گفتماني[2]
نامی که در جهان گفتمان تفسیر می‌شود.

یادآوری - فرض بر این نیست که نام‌های متفاوت تحت عنوان افراد متفاوت تفسیر شوند، بلکه در جهان گفتمان یک فرد ممکن است با دو یا چند نام متمایز مشخص شود.

۱۰-۳
حوزه‌ي گفتمان[3]
جهان گفتمان ملاحظه شود.

۱۱-۳
زبان علامت‌گذاري منطق مشترك گسترش‌پذير
(XCL)[4]
نحو مبتنی بر زبان علامت‌گذاری گسترش‌پذیر (XML) برای منطق مشترک.

۱۲-۳
فرد[5]
یک عنصر از جهان گفتمان.

یادآوری - جهان گفتمان مجموعه‌ی تمام افراد است.

۱۳-۳
شناسه‌ي منبع بين‌المللي شده
(IRI)[6]

1 - Dialect
2 - Discourse name
3 - Domain of discourse
4 - eXtensible Common Logic Markup Language
5 - Individual
6 - Internationalized Resource Identifier

رشته‌ی نویسه‌های یونیکد[1] که منطبق بر نحو توصیف شده در [2] بوده و به منظور استفاده به عنوان یک نحو اینترنتی برای شناسه‌های شبکه که بتواند شکل‌های مختلف نویسه‌های بین‌المللی را در خود جای دهد مطرح شده است.

یادآوری - این شناسه به منظور جایگزینی **شناسه‌ی یکسان منبع**[2] (به عنوان یک استاندارد اینترنتی برای شناسه‌های شبکه) در نظر گرفته شده است.

۱۴-۳

تفسیر[3]

تصریح صوری معانی نام‌های موجود در یک واژه‌نامه‌ی متعلق به یک گویش منطق مشترک در قالب یک جهان مرجع.

یادآوری ۱ - یک تفسیر به نوبه خود ارزش معنایی تمام عبارت‌های پیچیده‌ی گویش مورد نظر (به ویژه، ارزش درستی جمله‌های آن) را مشخص می‌کند.

یادآوری ۲ - برای توصیف دقیق‌تر نحوه‌ی تعریف یک تفسیر به بند ۶-۲ مراجعه شود.

۱۵-۳

قالب تبادل دانش

(KIF)[4]

صورت‌بندی[5] مرتبه اول مبتنی بر متن که از یک نشانه‌گذاری لیستی[6] شبه لیسپ[7] استفاده می‌کند.

یادآوری ۱ - KIF در [3] معرفی شده، نقطه شروع آن با تلاش مشاع دانش[8] پشتیبانی‌شده توسط دارپای ایالات متحده[9] بوده است.

یادآوری ۲ - KIF پایه و اساس یکی از سه گویش منطق مشترک مشمول این استاندارد ملی است.

۱۶-۳

عملگر[10]

نقش نحوی متمایز که به‌وسیله یک عنصر مشخص در یک اصطلاح تابعی[11] ایفا می‌شود.

---

1 - Unicode
2 - Uniform Resource Identifier
3 - Interpretation
4 - Knowledge Interchange Format
5- Formalism
۶ - به دلیل اینکه واژه List یا لیست در این استاندارد، معادل ساختمان داده یا ساختار گسسته List استفاده شده است، از معادل فارسی «فهرست» برای آن استفاده نشده است.
7 - LISP
8 - Knowledge Sharing Effort
9 - US Defense Advanced Research Projects Agency (DARPA)
10 - Operator
11 - Functional term

یادآوری - تشخیص[1] یک اصطلاح تابعی در یک تفسیر به‌وسیله گسترش تابعی تشخیص عملگر همراه با تشخیص مولفه‌های باقیمانده تعیین می‌شود.

۳-۱۷
مسند[2]

«منطق مشترک» نقش نحوی مشخص که به‌طور دقیق به‌وسیله یک مؤلفه در یک جمله‌ی تجزیه‌ناپذیر ایفا می‌شود.

یادآوری - ارزش درستی یک جمله‌ی تجزیه‌ناپذیر در یک تفسیر که به‌وسیله گسترش رابطه‌ای معنای صریح مسند همراه با معانی صریح مولفه‌های باقی‌مانده تعیین می‌شود.

۳-۱۸
گویش تفکیک‌شده[3]

گویشی که در آن بعضی از نام‌های غیرگفتمانی باشند.

یادآوری - در تفسیر یک گویش تفکیک‌شده معانی صریح نام‌های غیرگفتمانی در حیطه‌ی جهان مرجع قرار می‌گیرد نه جهان گفتمان.

۳-۱۹
جمله[4]

«منطق مشترک» واحد متن منطقی که درست یا غلط است، به عبارتی دیگر ارزش درستی آن در یک تفسیر مشخص می‌شود.

۳-۲۰
دسته[5]

هر زیرمجموعه از جهان گفتمان که محدوده‌ی تغییرات یک سور را مشخص می‌کند.

یادآوری - تعریف دسته مرتبط با تعریف «نوع» است (مطابق بند ۳-۲۴). اغلب منظور از دسته یک زیرمجموعه‌ی محض از افراد در جهان گفتمان است.

۳-۲۱
منطق دسته‌ای[6]

یک سامانه‌ی منطق (مرتبه اول یا از مراتب بالاتر) که در آن به تمام نمادهای غیرمنطقی باید یک دسته تخصیص داده شده باشد.

۳-۲۲
اصطلاح[1]

1 - Denotation
2 - Predicate
3 - Segregated dialect
4 - Sentence
5 - Sort
6 - Sorted logic

عبارت «منطق مشترک» که یک فرد را مشخص می‌کند. این عبارت یا متشکل از یک نام است یا بـه صـورت بازگشتی یک اصطلاح تابعی است که به دنباله‌ای از آرگومان‌ها اعمال می‌شود، به‌طوری که هـر کـدام از ایـن آرگومان‌ها خود یک اصطلاح است.

۲۳ـ۳
منطق مرتبه اول سنتي
(TFOL)[2]

فرمول‌بندی‌های سنتی ریاضی برای منطق که با اِسناد چندتایی[3]، عملگرهای بولی (از جمله نفـی) و اسـتفاده از سور سروکار دارد و در آن هر گزاره به‌صورت قطعی درست یا غلط است.

یادآوری - زبان‌هایی که برای منطق مرتبه اول سنتی ارائه شده‌اند به‌طور ویژه مانع استفاده از سورهای اسنادی[4] و استفاده از نام یکسان برای جایگاه مسند و آرگومان در جمله‌های تجزیه‌ناپذیر می‌شوند. اما هر دوی این موارد در منطق مشترک مجازند (گرچه ضروری نیستند). در منطق مشترک زبان‌های منطق مرتبه اول سنتی در رده‌ی گویش‌های تفکیک‌شده قرار می‌گیرند (مطابق بند ۶-۱-۳).

۲٤ـ۳
نوع[5]

چارچوب منطقی که در آن عبارت‌های منطق به دو دسته‌ی نحوی و لغوی رده‌بندی می‌شـوند و محـدود بـه اعمال بر روی آرگومان‌ها از یک نوع مشخص هستند.

یادآوری ۱ - در عمل، یک نوع تعریف‌کننده یک رده از افراد است. «نظریه‌ی نوع» اغلب به رده‌ی ویژه‌ای از چنین منطق‌هایی اشاره می‌کند که در آن‌ها نمادهای رابطه به مراتب مختلف تقسیم می‌شوند به گونه‌ای که روابط مرتبه n تنها قابل اعمال به روابط مرتبه پایین‌تر هستند.

یادآوری ۲ - محدودیت یک نوع بیشتر از یک دسته است چرا که قیود عمدی یا قطعی را به افرادی که عضو رده‌ی آن نوع هستند تحمیل می‌کند و این در حالی است که یک دسته تنها به هر زیرمجموعه‌ای از افراد درون یک حوزه اطلاق می‌شود که یک سور می‌تواند بر آن‌ها عمل کند.

۲۵ـ۳
جهان گفتمان[6]
حوزه‌ي گفتمان

مجموعه‌ی تمام اعضاء که در یک تفسیر قرار می‌گیرند؛ به عبارت دیگر، مجموعه‌ای که محـدوده‌ی سـورها را مشخص می‌کند.

---

1 - Term
2 - Traditional first-order logic
3 - N-ary predication
4 - Predicate quantifiers
5 - Type
6 - Universe of discourse

یادآوری - جهان گفتمان باید زیرمجموعه‌ای از جهان مرجع بوده و ممکن است با آن یکسان باشد.

۲۶ـ۳

جهان مرجع[1]

مجموعه‌ی تمام هستارهایی که برای تعریف معانی عبارت‌های منطقی در یک تفسیر لازم هستند.

یادآوری ۱ - جهان مرجع باید یک ابرمجموعه از جهان گفتمان باشد و می‌تواند با آن یکسان باشد.

یادآوری ۲ - گویش‌های تفکیک‌شده به‌طور کلی دارای یک جهان گفتمان هستند (بدون اشاره به جهان مرجع) و برای گویش‌های تفکیک نشده جهان گفتمان و جهان مرجع یکسان هستند. این تمایز این امکان را فراهم می‌کند که یک معناشناسی واحد بتواند هر دو سبک گویش را پوشش دهد. گویش‌های تفکیک نشده که جهان گفتمان و ارجاع را یکسان می‌بینند ممکن است در یک تفسیر برای سادگی به «جهان» ارجاع دهند.

۲۷ـ۳

شناسه‌ي منبع یکسان

(URI)[2]

دنباله‌ای از نویسه‌های آسکی[3] منطبق بر شکل‌های نحوی تعریف شده در [4].

یادآوری - در زمان نگارش این استاندارد شناسه‌ی منبع یکسان به عنوان نحو استاندارد اینترنت برای شناسه‌های شبکه مطرح است و ممکن است به‌وسیله شناسه‌ی منبع **بین‌المللی شده** منسوخ شود.

٤ نمادها و کوته‌نوشت‌ها

این نمادها و کوته‌نوشت‌ها به‌طور کلی برای قسمت‌های اصلی استاندارد است. بعضی از پیوست‌ها ممکن است نمادها و کوته‌نوشت‌ها خود را معرفی کنند که در خود آن پیوست برشمرده شده‌اند.

١ـ٤ نمادها

بعضی از این نمادها اصطلاحاتی را نشان می‌دهند که در بند ۳ تعریف شده‌اند.

fun_I نگاشتی از UR_I به توابع تعریف شده از UD_I^* به UD_I

I یک تفسیر (از حیث مدل - نظری[4])

int_I نگاشتی از نام‌های واژه‌نامه‌ی V به UR_I؛ به‌صورت غیرصوری، روشی برای ربط‌دهی نام‌های موجود در V به مراجع موجود در UR_I

rel_I نگاشتی از UR_I به زیرمجموعه‌های UD_I*

seq_I نگاشتی از نشان‌گرهای دنباله[5] در V به UD_I

1 - Universe of reference
2 - Uniform Resource Identifier
3 - ASCII
4 - Model-theoretic
5 - Sequence markers

V یک واژه‌نامه که مجموعه‌ای از نام‌ها و نشان‌گرهای دنباله است.

UD_I جهان گفتمان که در حقیقت یک مجموعه‌ی غیرتهی[1] از *اعضاء منحصر به فرد*[2] است که تفسیر I پیرامون آن‌هاست و محدوده‌ی تغییرات سورها را مشخص می‌کند.

UR_I جهان مرجع که در حقیقت مجموعه‌ی تمام مراجع نام‌ها در یک تفسیر I است.

X^* مجموعه‌ی دنباله‌های متناهی عناصر X، برای هر مجموعه‌ی X.

۲-٤ کوته‌نوشت‌ها

این کوته‌نوشت‌ها در این استاندارد استفاده شده‌اند. برای تعاریف و توضیحات بیشتر در مورد این اصطلاحات به بند ۳ مراجعه کنید.

CG	Conceptual Graph	گراف مفهومی
CGIF	Concpetual Graph Interchange Format	قالب تبادل گراف مفهومی
CL	Common Logic	منطق مشترک
CLIF	Common Logic Interchange Format	قالب تبادل منطق مشترک
DF	Display Form	شکل نمایش (مورد استفاده در پیوست ب)
EBNF	Extended Backus-Naur Format	قالب گسترش‌یافته‌ی Backus-Naur
FO	First-Order	مرتبه اول
IRI	Internationalized Resource Identifier	شناسه‌ی منبع بین‌المللی شده
KIF	Knowledge Interchange Format	قالب تبادل اطلاعات
OWL	Web Ontology Language	زبان هستان‌شناسی وب
RDF	Resource Definition Framework	چارچوب تعریف منبع
RDFS	Resource Definition Framework Schema	طرح چارچوب تعریف منبع
TFOL	Traditional First Order Logic	منطق مرتبه اول سنتی
URI	Uniform Resource Identifier	شناسه‌ی منبع یکسان
XCL	eXtensible Common Logic Markup Language	زبان علامت‌گذاری منطق مشترک گسترش‌پذیر
XML	eXtensible Markup Language	زبان علامت‌گذاری گسترش‌پذیر

1 - Non-empty
2 - Individuals

۵ الزامات و مرور کلي طراحي

این بند اطلاعاتی است و هدف از آن توصیف مختصر اهداف منطق مشترک و اصول راهنمای کلی و قیود حاکم بر محتوای آن است.

۵-۱ الزامات

منطق مشترک با توجه به الزامات متعددی طراحی و توسعه یافته است. تمام این الزامات از نقش منطق عام به عنوان ابزاری برای انتقال محتوای منطقی در یک شبکه‌ی ارتباطی باز سرچشمه گرفته است. استفاده از «توصیه می‌شود»[1] در ادامه‌ی بند ۵ به معنی یک هدف مطلوب است و از جمله الزامات منطق مشترک یا گویش منطبق بر آن به شمار نمی‌رود.

۵-۱-۱ توصیه مي‌شود منطق مشترك به صورت برابر كل منطق مرتبه اول را شامل شود

نحو و معناشناسی منطق مشترک باید تمام شکل‌های نحوی مرتبه اول با معانی متداول آن‌ها را فراهم کند.

هر نحو مرتبه اول مرسومی به صورت مستقیم و بدون از دست رفتن اطلاعات یا تغییر معنی به منطق مشترک قابل ترجمه خواهد بود.

۵-۱-۲ توصیه مي‌شود منطق مشترك يك نحو همه منظوره براي عبارت‌هاي منطقي ارتباطي فراهم كند

الف- توصیه می‌شود یک نحو منفرد مبتنی بر XML برای محتوای منطق مشترک ارتباطی وجود داشته باشد.

ب- توصیه می‌شود زبان قادر به بیان «شیرین‌کاری‌های نحوی[2]»[3] مختلف متداول برای شکل‌های منطقی یا الگوهای متداول جمله‌های منطقی باشد.

پ- توصیه می‌شود نحو با قراردادهای موجود مرتبط باشد، در حقیقت بهتر است قادر به ارائه‌ی تمام مضامین قابل بیان در RDF، RDFS یا OWL باشد.

ت- توصیه می‌شود حداقل یک نحو فشرده‌ی قابل خواندن به‌وسیله انسان که بتوان از آن برای بیان کل زبان استفاده کرد وجود داشته باشد.

۵-۱-۳ توصیه مي‌شود منطق مشترك براي استفاده در وب ساده و طبيعي باشد

الف- توصیه می‌شود نحو مبتنی بر XML با مشخصه‌های منتشر شده برای XML، نحو URI، طرح XML، یونیکد و سایر قراردادهای مربوط به انتقال اطلاعات در وب سازگار باشد.

ب- توصیه می‌شود شناسه‌های منبع یکسان و مراجع آن‌ها به صورت نام‌ها در زبان قابل استفاده باشند.

---

1 - Should

2 - Syntactic sugarings

۳- «شیرین‌کاری نحوی» یا «شکر نحوی» به یک نوع نحو اطلاق می‌شود که برای تسهیل استفاده از یک نحو پیچیده‌تر، که به اصطلاح «شور» است، تعریف و استفاده می‌شود.

پ- توصیه می‌شود شناسه‌های منبع یکسان به منظور تخصیص نام به عبارت‌ها یا مجموعه‌هایی از عبارت‌های قابل استفاده باشند، تا بدین ترتیب عملیات تحت وب مانند بازیابی، واردسازی و ارجاع متقابل تسهیل شود.

۵-۱-٤ توصیه می‌شود منطق مشترك از شبکه‌هاي باز پشتیباني کند

الف- انتقال محتوا بین عامل‌های[1] آگاه از منطق مشترک[2] نباید به مذاکره در مورد نقش نحوی نمادها یا ترجمه‌های مابین نقش‌های نحوی نیاز داشته باشد.

ب- توصیه می‌شود هر قطعه از متن منطق مشترک در هر جای شبکه معنی یکسانی داشته و از استلزامات[3] یکسانی پشتیبانی کند. توصیه می‌شود هر نام معنی منطقی یکسانی در تمام گره‌های شبکه داشته باشد.

پ- توصیه می‌شود هیچ عاملی نتواند توانایی یک عامل دیگر در مراجعه به هر هستار یا وضع حکم در مورد هر هستار را محدود کند.

ت- توصیه می‌شود زبان راه‌های ارجاع به یک جهان گفتمان محلی را پشتیبانی کرده و قادر باشد آن را به جهان‌های گفتمان دیگر ربط دهد.

ث- توصیه می‌شود استفاده‌کنندگان از منطق مشترک آزادانه بتوانند نام‌های جدید ابداع کرده و از آن‌ها در محتوای منطق مشترک منتشر شده استفاده کنند.

۵-۱-۵ توصیه می‌شود منطق مشترك هیچ فرض اختیاري در مورد معناشناسي نداشته باشد

الف- منطق مشترک هیچ فرض بی‌دلیل و دلخواهی در مورد ارتباطات منطقی بین عبارت‌های مختلف نمی‌کند.

ب- در صورت امکان عامل‌های منطق مشترک باید این فرضیات را در منطق مشترک به‌طور مستقیم بیان کنند.

۵-۲ خانواده‌اي از نشانه‌گذاري‌ها

این بند (اطلاعاتی) مفهوم «خانواده‌ای» از زبان‌ها را توصیف و بعضی اصول و مبانی توسعه‌ی منطق مشترک را بیان می‌کند.

اگر از قراردادی که به موجب آن هر زبانی یک دستور زبان دارد پیروی کنیم، آنگاه منطق مشترک به جای یک زبان، خانواده‌ای از زبان‌ها است. زبان‌های مختلف منطق مشترک که در این استاندارد به آن‌ها *گویش* گفته می‌شود ممکن است از نظر نحو سطحی بسیار متفاوت باشند، اما معناشناسی منفرد و یکپارچه‌ای دارند و همه‌ی آن‌ها قابل رونویسی به نحو انتزاعی عام هستند. شرط عضویت در این خانواده این است که گویش

1 - Agents
2 - Common Logic-aware
3 - Entailments

مورد نظر با حفظ معنی به‌طور متقابل قابل ترجمه به گویش‌های دیگر عضو خانواده باشد نه اینکه شکل نحوی خاصی داشته باشد. بنابراین، بسیاری از نشانه‌گذاری‌ها و زبان‌های موجود را می‌توان به عنوان گویش‌های منطق مشترک در نظر گرفت.

یک گویش منطق مشترک مبتنی بر KIF که به آن CLIF گفته می‌شود (به پیوست الف مراجعه شود) به منظور ارائه‌ی مثال در این استاندارد ملی استفاده شده است. CLIF را می‌توان یک فرم ساده شده و به روز رسانی شده‌ی KIF 3.0 [3] در نظر گرفت و در نتیجه خود یک زبان جدا است. بنابراین یک توصیف کامل از آن ارائه شده است که بدون نیاز به مراجعه به سایر قسمت‌های مشخصه قابل فهم است. گراف‌های مفهومی [1] نیز یک شکل شناخته شده از منطق مرتبه اول برای پردازش ماشین هستند. زبان CGIF در پیوست ب و یک گویش مبتنی بر XML که از معناشناسی منطق مشترک استفاده می‌کند در پیوست پ تصریح شده است.

٦ نحو انتزاعي و معناشناسي منطق مشترك

در این بند جنبه‌های اصلی نحو و معناشناسی منطق مشترک توصیف می‌شود.

٦-١ نحو انتزاعي منطق مشترك

در اینجا نحو منطق مشترک را با هدف عدم تعهد به قراردادهای نحوی هیچ گویش خاصی به صورت «انتزاعی» بیان می‌کنیم.

٦-١-١ رده‌بندي‌هاي[1] نحو انتزاعي

به هر کدام از درایه‌های زیر یک *رده‌بندی نحو انتزاعی* گفته می‌شود. اصطلاحات اضافی در این درایه‌ها می‌توانند تعریف‌کننده زیررده‌ها یا اجزای تشکیل‌دهنده‌ی یک رده‌بندی باشند. اصطلاحاتی که در اینجا تعریف می‌شوند برای وضوح زیرشان خط کشیده شده است. بقیه‌ی اصطلاحات ممکن است در تعاریف بند ۳ آمده باشند.

۶-۱-۱-۱ يك متن، مجموعه، ليست يا كيسه‌اى از عبارت‌ها است. هر قطعه از متن مي‌تواند به صورت اختياري با يك نام مشخص شود. يك متن منطق مشترك مي‌تواند يك دنباله، مجموعه يا كيسه‌اي از عبارت‌ها باشد. گويش‌ها مي‌توانند مشخص كنند كدام يك (مجموعه، دنباله يا كيسه) مورد نظر است يا اين را مشخص نكنند. تغيير ترتيب و تكرار عبارت‌ها در يك متن بر معناي آن تأثيري نمي‌گذارد. اما برنامه‌هايي كه متن منطق مشترك را منتقل يا دوباره منتشر مي‌كنند بايد ساختار متون را حفظ كنند، چرا كه برنامه‌هاي ديگر مجازند از اين ساختار براي اهداف ديگر مانند انديس‌گذاري استفاده كند. اگر گويشي شروطي را به متون تحميل كند، اين شروط بايد به‌وسيله برنامه‌هاي پيرو حفظ شود. يك متن مي تواند تهي باشد

۶-۱-۱-۲ يك عبارت، يك پيمانه، يك جمله، يك وارده يا يك متن است كه با يك توضيح همراه شده است.

1 - Categories

۶-۱-۱-۳ يك توضيح[1]، يك قطعه داده است. توضيحات مجازند به توضيحات ديگر يا عبارت‌هاي توضيح‌دار پيوست شده باشند. هيچ گونه محدوديت ويژه‌اي بر ماهيت توضيحات منطق مشترك وجود ندارد. در حقيقت يك توضيح مي‌تواند يك متن منطق مشترك باشد. گويش‌هاي ويژه ممكن است شروطي را بر روي شكل توضيحات اعمال كنند.

۶-۱-۱-۴ يك پيمانه از يك نام، يك مجموعه‌ي اختياري از نام‌هاي تحت عنوان *مجموعه‌ي عدم شمول* و يك متن به عنوان متن بدنه تشكيل مي‌شود. نام پيمانه جهان گفتمان «محلي» را مشخص مي‌كند كه متن در آن فهميده مي‌شود. مجموعه‌ي عدم شمول نام‌ها در متن را شامل مي‌شود كه به صراحت از اين جهان محلي بيرون نگه داشته شده‌اند. از نام يك پيمانه همچنين مي‌توان براي شناسايي آن استفاده كرد.

۶-۱-۱-۵ يك وارده دربردارنده‌ي يك نام است. هدف اين است كه اين نام بخشي از محتواي منطق مشترك را كه به صورت خارج از متن نمايش داده شده شناسايي كند و وارده مورد نظر هم آن محتوا را در متن تصديق كند. مفهوم «شناسايي» به صورت كامل تر در بند ٦-٣-١بحث شده است.

۶-۱-۱-۶ يك جمله يك جمله‌ي سوردار، يك جمله بولي، يك تجزيه‌ناپذير، يك جمله با يك توضيح پيوست شده يا يك جمله‌ي بي‌قاعده است.

۶-۱-۱-۷ يك جمله‌ي سوردار داراي (۱) يك نوع كه به آن سور گفته مي‌شود، (۲) يك دنباله‌ي متناهي و فاقد تكرار از نام‌ها و نشان‌گرهاي دنباله كه به آن دنباله‌ي انقياد و به هر عنصر آن يك انقياد جمله‌ي سوري گفته مي‌شود و (۳) يك جمله كه به آن بدنه‌ي جمله‌ي سوري گفته مي‌شود، است. هر گويش منطق مشترك بايد دو نوع وجودي و عمومي جمله‌هاي سوري را تميز دهد. به نام يا نشان‌گر دنباله‌اي كه در دنباله‌ي انقياد واقع شود مقيد در بدنه گفته مي‌شود و به هر نام يا نشان‌گر دنباله كه مقيد در بدنه نباشد آزاد در بدنه گفته مي‌شود.

۶-۱-۱-۸ يك جمله‌ي بولي داراي يك نوع، تحت عنوان كلمه‌ي ربط، و تعدادي جمله تحت عنوان مولفه‌هاي جمله‌ي بولي است. تعداد اين جمله‌ها به نوع جمله‌ي بولي بستگي دارد. هر گويش منطق مشترك بايد پنج نوع جمله‌ي بولي را تميز دهد كه عبارتند از: تركيبات عطفي و فصلي كه مي‌توانند شامل هر تعداد مولفه باشند، دلالت‌ها و دوشرطي‌ها كه داراي دو مولفه هستند و جمله‌هاي نفي كه به‌طور دقيق يك مولفه دارند.

يادآوري - مشخصه‌ی كنونی هيچ شكل ويژه‌ای از جمله‌های بی‌قاعده را به رسميت نمی‌شناسد. هدف از گنجاندن اين رده در نحو انتزاعی آماده كردن منطق مشترک برای گسترش‌های نحوی است كه معنای آن‌ها به‌طور كامل قابل تعريف در منطق مشترک نيست. به عنوان نمونه‌هايی از اين گسترش‌ها می‌توان به قيود[2]، كلمات ربط غيريكپارچه[3] و ساختارهای امری[4] اشاره كرد.

۶-۱-۱-۹ يك مولفه تجزيه‌ناپذير يا يك تساوي با دو آرگومان (كه خود اصطلاح هستند) است يا يك جمله‌ي تجزيه‌ناپذير كه خود متشكل از يك اصطلاح كه به آن مسند گفته مي‌شود و دنباله‌اي از اصطلاحات تحت عنوان دنباله‌ي آرگومان‌ها است. به عناصر اين دنباله آرگومان‌هاي تجزيه‌ناپذير گفته مي‌شود.

يادآوري - گويش‌هايی كه از يک نام برای مشخص كردن تساوی استفاده می‌كنند مجازند تجزيه‌ناپذير را به عنوان مسند در نظر بگيرند و با يک تساوی به صورت يک جمله‌ی تجزيه‌ناپذير برخورد كنند.

1 - Comment
2 - Modalities
3 - Non-monotonic connectives
4 - Imperative constructions

۶-۱-۱-۱۰ يك اصطلاح يك نام يا يك اصطلاح تابعي و يا يك اصطلاح با يك توضيح پيوست شده است.

۶-۱-۱-۱۱ يك اصطلاح تابعي متشكل از يك اصطلاح (كه به آن عملگر گفته مي‌شود) و دنباله‌اي از اصطلاحات است كه به آن دنباله‌ي آرگومان گفته مي‌شود كه عناصر آن آرگومان‌هاي آن اصطلاح تابعي هستند.

۶-۱-۱-۱۲ يك دنباله‌ي اصطلاحات، دنباله‌اي متناهي از اصطلاحات يا نشان‌گرهاي دنباله است.

یادآوری - دنباله‌های اصطلاحات می‌توانند تهی باشند، اما یک اصطلاح تابعی با دنباله‌ی تهی آرگومان‌ها نباید با استفاده از عملگرش شناسایی شود. همچنین یک جمله‌ی تجزیه‌ناپذیر با دنباله‌ی تهی آرگومان‌ها نباید با استفاده از مسند آن شناسایی شود

۶-۱-۱-۱۳ يك واژه‌نامه مجموعه‌اي از نام‌ها و نشان‌گرهاي دنباله است.

۶-۱-۱-۱۴ نام‌ها و نشان‌گرهاي دنباله دو رده‌ي نحوي متمايزند و هر كدام متمايز از ساير رده‌بندي‌هاي نحوي است.

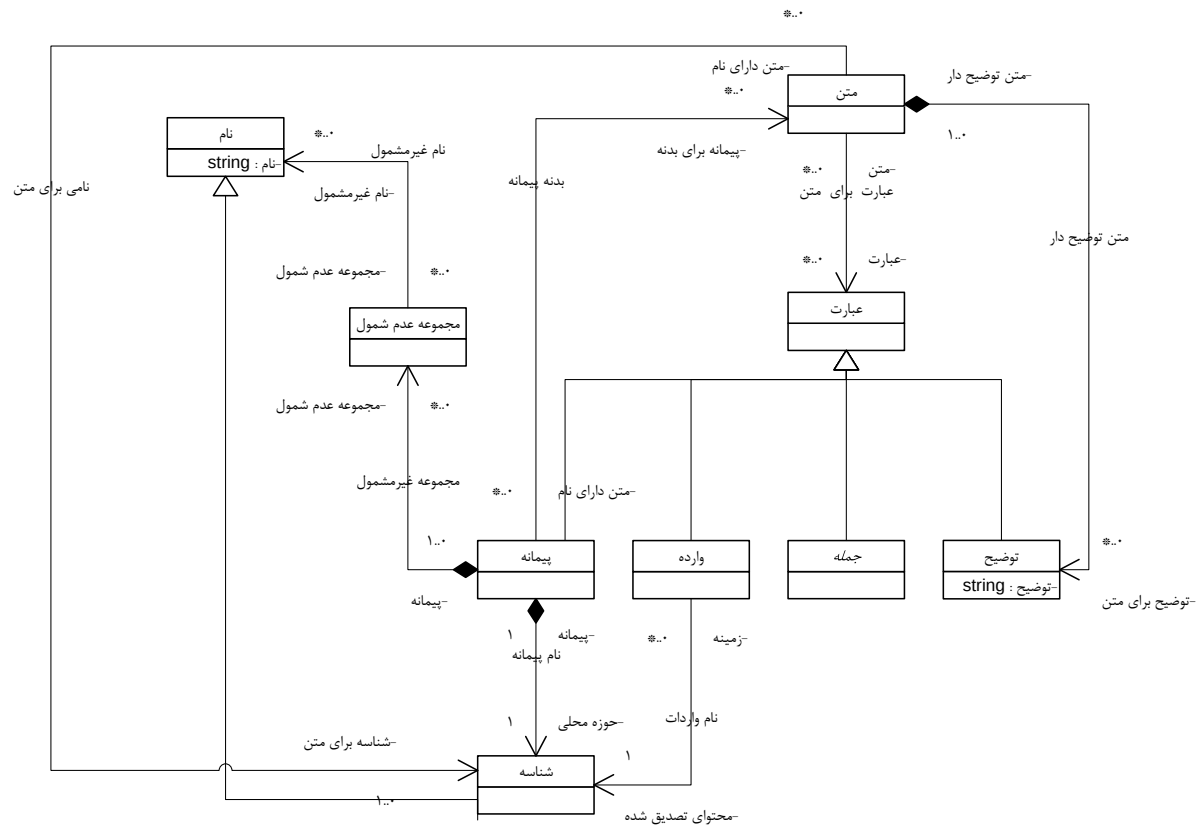
این بند ساختار نحو انتزاعی منطق مشترک را به‌طور کامل توصیف می‌کند. هر گویش که به‌طور کامل منطبق بر منطق مشترک باشد باید یک نمایش نحوی غیرمبهم برای هر کدام از انواع شناخته شده‌ی عبارت‌ها که در بالا آمده (به استثنای جمله‌های بی‌قاعده) ارائه دهد.

نمایش انواع جمله‌ها، رشته‌های متنی صریح (مانند «به ازای همه»[1] برای جمله‌ی عمومی و «و» [2] برای ترکیب عطفی) را در بر می‌گیرد. به هر حال، هیچ شرطی بر روی چگونگی نمایش رده‌بندی‌های نحوی در شکل سطحی یک گویش اعمال نشده است. در حقیقت، لازم نیست عبارت‌های یک گویش شامل رشته‌های حرفی باشند.

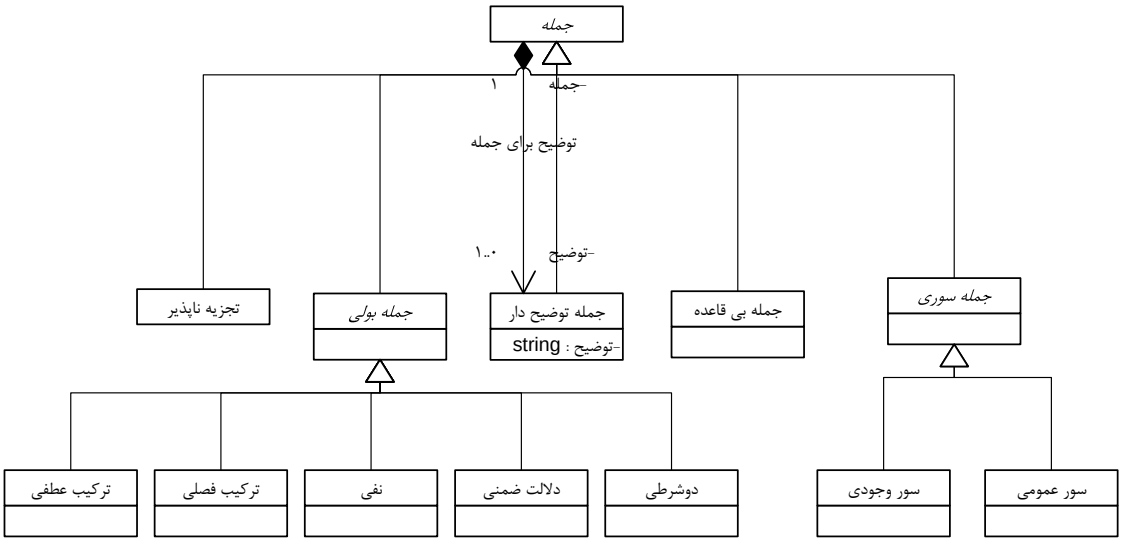
٦-١-٢ مدل مفهومي نحو انتزاعي منطق مشترك

به منظور توصیف بهتر ساختار نحو انتزاعی، در این بند یک مدل مفهومی ارائه می‌شود که رابطه‌ی بین رده‌بندی‌های نحوی را نشان داده و دلیل بعضی تصمیم‌ها را توصیف می‌کند. رده‌های نحو انتزاعی و ساختار مجاز آن‌ها با استفاده از نشانه‌گذاری نمودار کلاس در UML [5] نشان داده شده است.

1 - forall
2 - and

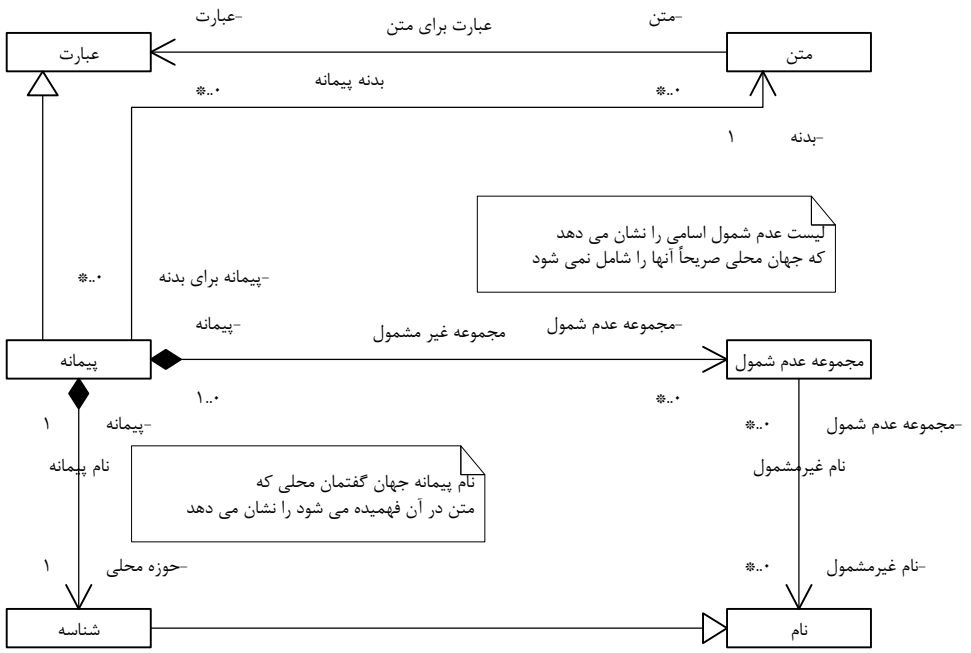


شکل ۱ - ساختار یک متن و طبقه‌بندی رده‌ی عبارت *متن*[1]

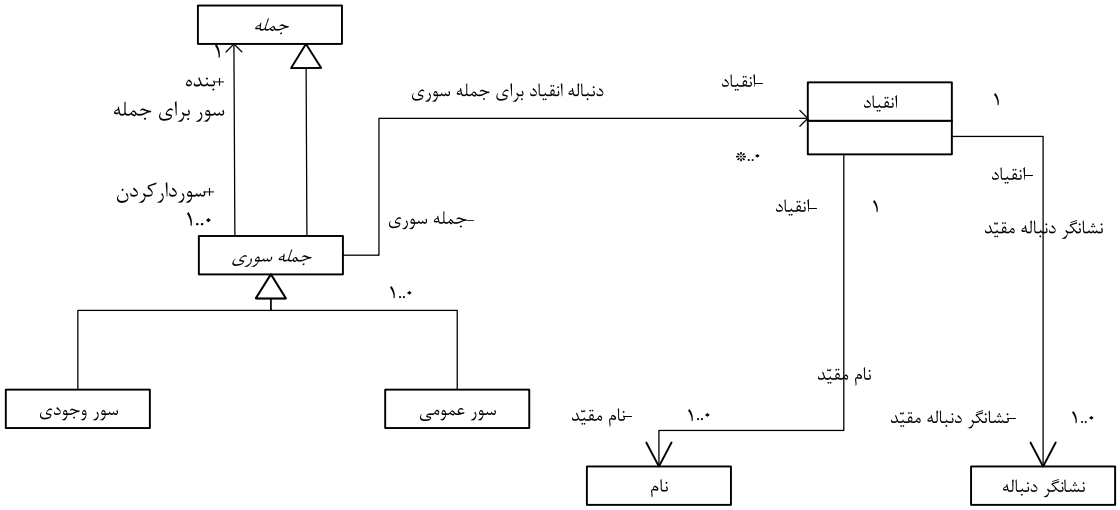


شکل ۲ - نحو انتزاعی جمله و زیررده‌های آن

1 - Phrase category *text*

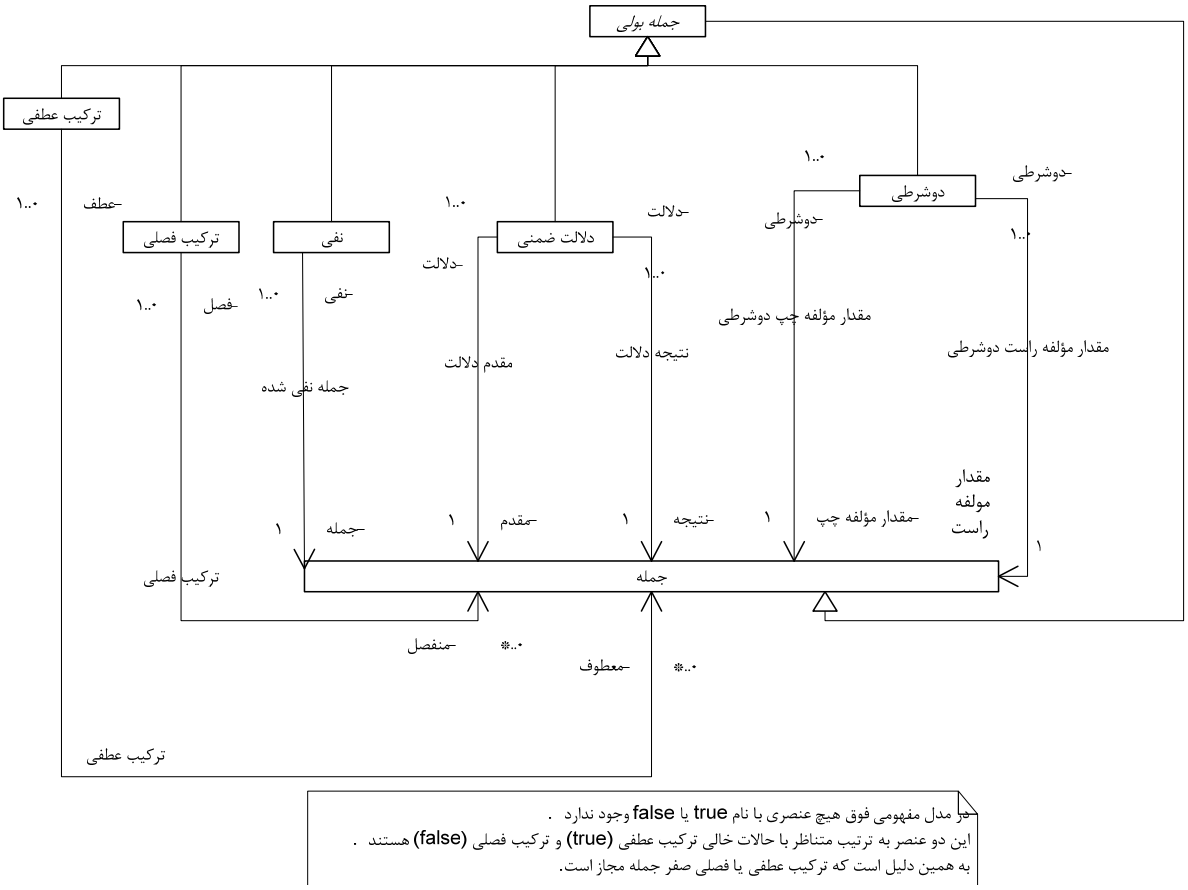


شکل ۳ - نحو انتزاعی یک پیمانه

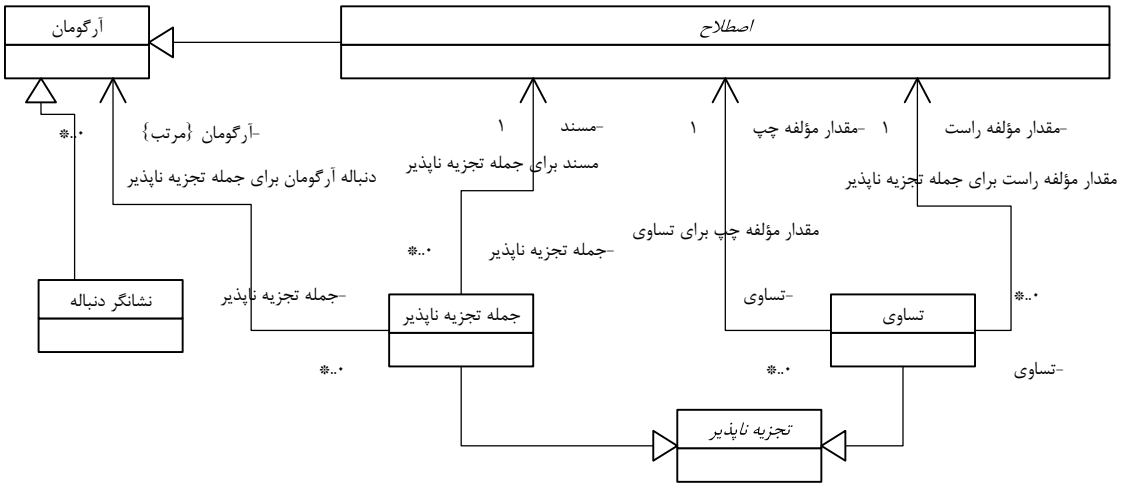


شکل ۴ - نحو انتزاعی یک جمله‌ی سوری

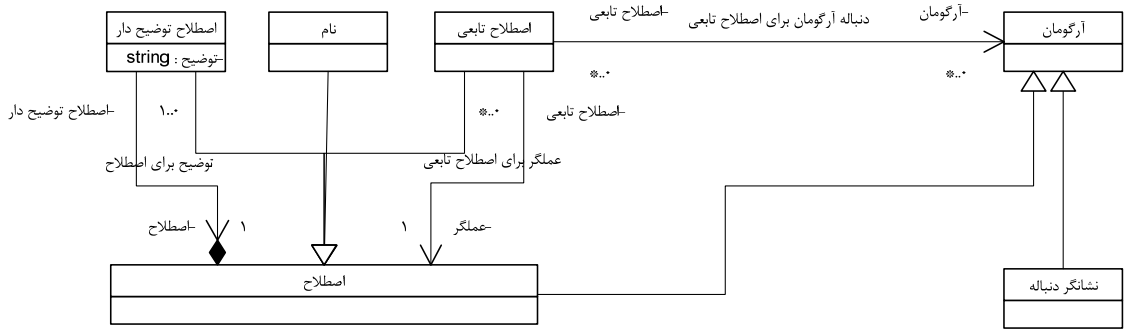
شکل ۴ نحو انتزاعی یک جمله‌ی سوری را نشان می‌دهد. یک جمله‌ی سوری عمومی جمله‌ی سوری است که سور آن عمومی باشد. یک جمله‌ی سوری وجودی جمله‌ی سوری است که سور آن وجودی باشد.



شکل ۵ - نحو انتزاعی یک جمله‌ی بولی



شکل ۶ - نحو انتزاعی یک تجزیه‌ناپذیر



شکل ۷ - نحو انتزاعی یک اصطلاح و دنباله‌ی اصطلاح

۶-۱-۳ ساختار نحو انتزاعي گویش‌ها

به گویشی که تنها بعضی از انواع عبارت‌های منطق مشترک را فراهم می‌کند گویش منطق مشترک *با نحو جزئی* یا *منطبق جزئی نحوی*[1] گفته می‌شود. در حالت خاص گویشی که نشان‌گرهای دنباله را شامل نمی‌شود اما در سایر موارد به‌طور کامل منطبق بر منطق مشترک است، به عنوان یک گویش *با نحو فشرده* شناخته می‌شود. توصیف بعضی روابط بین انطباق نحوی و انطباق معنایی در بند ۷-۱ آمده است.

گویش‌ها مجازند علاوه بر این، شکل‌های دیگر ترکیب جمله را فراهم کنند که به‌وسیله این نحو توصیف نشده است. اما به منظور انطباق کامل، چنین ترکیب‌هایی باید یا رده‌های جدیدی باشند که در قالب این رده‌ها تعریف شده‌اند یا گسترش‌هایی از این رده‌ها (برای نمونه انواع جدیدی از جمله‌های بولی یا انواع سورها) که از نظر معنی معادل ترکیبی است که با استفاده از این نحو ایجاد شده و مبتنی بر معناشناسی منطق مشترک تفسیر شده است. به عبارت دیگر، آن‌ها را می‌توان کوته‌نوشت‌های نحوی یا درشت دستور[2] در نظر گرفت که به آن‌ها «شیرینی نحوی» نیز گفته می‌شود. گویش CLIF که در پیوست الف توصیف شده دارای تعدادی شکل شیرین نحوی برای جمله‌های سوری و جمله‌های تجزیه‌ناپذیر است (گونه‌های دیگری از مقبولیت نیز به رسمیت شناخته شده‌اند: بحث کامل پیرامون انطباق در بند ۷ آمده است).

تنها اصطلاحات تعریف نشده در بند نحو انتزاعی عبارتند از: *نام* و *نشان‌گر دنباله*. تنها محدودیت نحوی بر روی رده‌های لغوی پایه‌ی *نام* و *نشان‌گر دنباله* این است که آن‌ها **باید** انحصاری باشند. **توصیه می‌شود** گویش‌هایی که هدف از آن‌ها انتقال محتوا در شبکه است محدودیت‌های دلخواه و غیرضروری بر روی شکل نام‌ها اعمال نکرده و نام‌های مشخصی را به منظور استفاده به عنوان شناسه‌ی متون منطق مشترک فراهم کنند. منظور این است که رشته‌های حرفی که به عنوان شناسه در یک گویش استفاده می‌شوند باید در آن

1 - Syntactically partially conformant
2 - Macro

گویش قابل تجزیه به عنوان نام‌های منطق مشترک باشند. **توصیه می‌شود** گویش‌هایی که قرار است در وب استفاده شوند اجازه دهند شناسه‌های منبع عمومی، شناسه‌های منبع بین‌المللی و ارجاعات URI به عنوان نام استفاده شوند [2][4]. **توصیه می‌شود** گویش‌های منطق مشترک نام‌ها را بر اساس قراردادهای یونیکد تعریف کنند.

در نحو انتزاعی منطق مشترک هیچ مفهومی برای «متغیر مقید» وجود ندارد. نام‌هایی که می‌توانند مقید واقع شوند لازم نیست از نام‌هایی که (تنها) می‌توانند مستقل واقع شوند از نظر لغوی متمایز باشند. همچنین، نام‌ها (چه مستقل و چه مقید) لازم نیست به رده‌های متمایزی چون نام رابطه، نام تابع یا نام منفرد تفکیک شوند. هیچ گونه محدودیت دسته‌ای[1] بر روی نام‌ها وجود ندارد. گویش‌های منطق مشترک ویژه‌ای مجازند تمایزهایی بین رده‌های مختلف نام‌ها قائل شده و محدودیت‌های اضافی بر وقوع انواع نام‌ها و اصطلاحات در عبارت‌ها اعمال کنند. برای نمونه ممکن است لازم بدانند نام‌هایی که می‌توانند مقید واقع شوند (همان متغیرها در زبان‌های مرتبه اول سنتی) همانند KIF با یک پیشوند خاص یا همانند Prolog به سبک خاصی نوشته شوند یا ممکن است همانند نحو مرتبه اول سنتی لازم بدانند عملگرها در رده‌ی متمایز نام‌های رابطه قرار گیرند.

یک گویش **مجاز است** شرایط معنایی ویژه‌ای را بر بعضی رده‌های نام‌ها تحمیل کند و برای محدود کردن مکان قرار گرفتن این نام‌ها در عبارت‌ها محدودیت‌های نحوی اعمال کند. برای نمونه، نحو CLIF برای اعداد مفهوم ثابتی قائل است و از استفاده از آن‌ها به عنوان شناسه جلوگیری می‌کند.

یک گویش **مجاز است** لازم بداند بعضی از نام‌هایی که برای مشخص کردن هیچ هستاری در جهان گفتمان به‌کار نمی‌روند *غیرگفتمانی* محسوب شوند. این نیازمندی یا با افراز واژه‌نامه اعمال می‌شود یا با الزام غیرگفتمانی بودن آن رده از نام‌هایی که در جایگاه نحوی خاصی قرار می‌گیرند. به گویشی که دارای نام‌های غیرگفتمانی باشد *تفکیک‌شده* گفته می‌شود. به نام‌هایی که غیرگفتمانی نباشند *نام‌های گفتمانی* گفته می‌شود.

یک گویش تفکیک‌شده **باید** محدودیت‌های نحوی کافی فراهم کند تا در هر متن این گویش که از نظر نحوی مجاز است موارد زیر تضمین شود:

- هر نام باید به یکی از دو رده‌ی گفتمانی و غیرگفتمانی تعلق داشته باشد،
- هیچ نامی نباید به هر دو رده‌ی گفتمانی و غیرگفتمانی تعلق داشته باشد،
- هیچ نام غیرگفتمانی نباید آرگومانی از یک تجزیه‌ناپذیر یا اصطلاح تابعی باشد،
- هیچ نام غیرگفتمانی به‌صورت مقید در یک جمله‌ی سوری نباشد.

1 - Sortal restriction

از آنجا که وجود نام‌های غیرگفتمانی بر معناشناسی اثر می‌گذارد شروط ویژه‌ای به گویش‌های تفکیک‌شده اعمال می‌شود.

۲-۶ معناشناسي منطق مشترك

معناشناسی منطق مشترک بر حسب یک رابطه‌ی برقراری[1] بین متن منطق مشترک و ساختارهای ریاضی تحت عنوان *تفسیر*[2] تعریف می‌شود.

واژه‌نامه‌ی یک متن منطق مشترک مجموعه‌ی نام‌ها و نشان‌گرهای دنباله‌ای است که در آن متن واقع می‌شوند. در یک گویش تفکیک‌شده نام‌های موجود در واژه‌نامه به دو رده‌ی گفتمانی و غیرگفتمانی افراز می‌شوند.

یک تفسیر I از یک واژه‌نامه‌ی V شامل یک مجموعه‌ی UR_I به عنوان جهان مرجع، با یک زیرمجموعه‌ی غیرتهی UD_I به عنوان جهان گفتمان و چهار نگاشت به صورت زیر است:

- rel_I از UR_I به زیرمجموعه‌های مجموعه‌ی $UD_I^* = \{ \langle x_1,\ldots,x_n\rangle \mid x_1,\ldots,x_n \in UD_I \}$ که همان مجموعه‌ی دنباله‌های متناهی از عناصر UD_I است. توجه کنید برای هر UD_I دنباله‌ی تهی در UD_I^* موجود است.
- fun_I از UR_I به تمام توابع تعریف شده از UD_I^* به UD_I که منظور توابعی است که هر دنباله در UD_I^* را به یک عنصر (منحصر به فرد) در UD_I نگاشت می‌دهند.
- int_I از نام‌های موجود در V به UR_I به گونه‌ای که $int_I(v)$ در UD_I است اگر و فقط اگر v یک نام گفتمانی باشد.

یادآوری - اگر گویش جمله‌های بی‌قاعده را بشناسد آنگاه با آن‌ها به عنوان نام‌های گزاره‌ها برخورد می‌شود و int_I نگاشتی از جمله‌های بی‌قاعده‌ی یک متن به ارزش‌های درستی $\{true, false\}$ را نیز شامل می‌شود.

- seq_I از نشان‌گرهای دنباله در V به UD_I^*.

UD_I به‌طور مستقیم یک جهان گفتمان یا حوزه گفتمان است که محدوده‌ی تغییرات سورها را مشخص می‌کند و شامل تمام افرادی است که تفسیر «راجع به» آن‌هاست. UR_I به صورت بالقوه مجموعه‌ی بزرگتری است و ممکن است شامل هستارهایی باشد که در جهان گفتمان نیستند. در حقیقت UR_I ممکن است شامل روابطی باشد که در UD_I نبوده و به عنوان تفسیر نام‌های غیرگفتمانی در یک گویش تفکیک‌شده به کار می‌روند. همه نام‌ها به شیوه‌ی یکسانی تفسیر می‌شوند؛ چه برای نشان دادن چیزی در جهان گفتمان به‌کار روند چه به‌کار نروند. به همین دلیل است که تنها یک نگاشت تفسیری به همه نام‌ها بدون در نظر گرفتن نقش نحوی آن‌ها اعمال می‌شود. در حقیقت $rel_I(x)$ در UD_I^* است حتی اگر x در UD_I نباشد. زمانی که

1 - Satisfaction relation
2 - Interpretation

تنها گویش‌های تفکیک‌شده مد نظر است عناصر جهان مرجع را که خارج از جهان گفتمان هستند می‌توان با استفاده از مقادیر متناظر نگاشت‌های rel_I و fun_I شناسایی کرد. این نگاشت‌ها سپس به عنوان نگاشت‌های همانی تفسیر می‌شوند. ترکیب حاصل، مسندها را به طور مستقیم به روابط و عملگرها را به توابع نگاشت می‌دهد. در نتیجه برای نحو تفکیک‌شده‌ی منطق مرتبه اول سنتی ساختار تفسیر سنتی‌تری حاصل می‌شود. از طرف دیگر، زمانی که تنها گویش‌های تفکیک نشده مد نظر باشد تمایز بین جهان گفتمان و جهان مرجع ضروری نیست چرا که می‌توان آن‌ها را یکسان در نظر گرفت. در اینجا برای اینکه با گویش‌های تفکیک‌شده و تفکیک نشده یکپارچه برخورد شود این تمایز مطرح شده است.

با جمله‌های بی‌قاعده به صورت متغیرهای گزاره‌ای دلخواه برخورد می‌شود. توجه کنید که این موضوع بر تفسیر منطق مشترک جمله‌های منطق مشترک که مؤلفه‌های نحوی یک جمله‌ی بی‌قاعده باشند تأثیر نمی‌گذارد. همچنین توجه داشته باشید که گرچه نشان‌گرهای دنباله به دنباله‌های متناهی در یک تفسیر نگاشت داده می‌شوند، این دنباله‌ها با نام مشخص نمی‌شوند، بنابراین لازم نیست در جهان مرجع باشند.

تخصیص ارزش معنایی به عبارت‌های پیچیده - به‌طور برجسته، تخصیص ارزش درستی به جمله‌ها - نیاز به تعدادی تعریف کمکی دارد.

اگر S را به عنوان زیرمجموعه‌ای از V در نظر بگیریم آنگاه یک تفسیر J از V را می‌توان یک نسخه‌ی مختصّ S از I در نظر گرفت که مشابه I بوده به جز اینکه int_J و seq_J ممکن است از نظر مقادیری که به اعضای S تخصیص می‌دهند با int_I و seq_I متفاوت باشند. به شکلی صوری‌تر J یک نسخه‌ی مختصّ S از I است اگر $UR_J = UR_I$، $UD_J = UD_I$، $rel_J = rel_I$، $fun_J = fun_I$ ، برای نام‌های $n \in S$ داشته باشیم $int_J(n) = int_I(n)$ و برای نشان‌گرهای دنباله‌ی $s \in S$ داشته باشیم $seq_J(s) = seq_I(s)$.

اگر E را به عنوان زیرمجموعه‌ای از UD_I در نظر بگیریم آنگاه محدودیت[1] I به E یک تفسیر K از واژه‌نامه‌ی یکسان و بر روی جهان یکسان و با $int_K = int_I$ و $seq_K = seq_I$ خواهد بود. اما در این حالت $UD_K = E$ و به ازای هر v در واژه‌نامه‌ی I داریم: $rel_K(v)$ انحصار $rel_I(v)$ به E^* و $fun_K(v)$ انحصار $fun_I(v)$ به $E^* \to E$ است.

اگر **N** را به عنوان مجموعه‌ای از نام‌ها در نظر بگیریم آنگاه *اختصار*[2] I از **N** که با $[I<\mathbf{N}]$ مشخص می‌شود، عبارت است از انحصار I به مجموعه‌ی $UD_I - \{int_I(v): v \text{ in } \mathbf{N}\}$.

اگر $s = <s_1,..., s_n>$ و $t = <t_1,..., t_m>$ دنباله‌های متناهی باشند آنگاه s;t دنباله‌ی الحاقی $< s_1,..., s_n, t_1,..., t_m >$ خواهد بود. در حالت خاص برای هر دنباله‌ی s داریم: $s;<> = s$.

مقدار هر عبارت E در تفسیر I مبتنی بر قواعدی که در جدول ۱ آمده، به‌دست می‌آید.

1 - Restriction
2 - Retraction

مفهوم تابع *text* در بخش‌های E17 و E20 و مفهوم مربوط به یک *مقدار متنی نام دار* در بند بعد توصیف شده است.

موارد برشمرده شده در جدول ۱ شروط معنایی منطقی پایه‌ای هستند که در تمام گویش‌های منطبق باید برقرار باشد. علاوه بر اینها یک گویش مجاز است شروط معنایی بیشتری را تحمیل کند. یک گویش با شروط معنایی اضافی یک *گسترش معنایی* است. در حالت خاص، گسترش‌های معنایی مجازند بر جمله‌های بی‌قاعده شروط نحوی و معنایی تحمیل کنند، اما نباید از جمله‌های بی‌قاعده برای نمایش محتوایی که در متن منطق مشترک قابل بیان است استفاده کنند.

جدول ۱ - تفسیرهای عبارت‌های منطق مشترک

	اگر E عبارتی به صورت زیر باشد	**آنگاه $I(E)$ برابر است با:**
E1	نام **N**	$int_I(N)$
E2	نشان‌گر دنباله‌ی **S**	$seq_I(S)$
E3	دنباله‌ی اصطلاحات $T_1,\ldots,T_n$ که در آن T_1 یک اصطلاح است	$\langle I(T_1)\rangle;I(\langle T_2 \ldots T_n \rangle)$
E4	دنباله‌ی اصطلاحات $T_1,\ldots,T_n$ که در آن T_1 یک نشان‌گر دنباله است	$I(T_1);I(\langle T_2 \ldots T_n \rangle)$
E5	اصطلاحی با عملگر **O** و دنباله‌ی آرگومان **S**	$fun_I(I(O))(I(S))$
E6	تجزیه‌ناپذیری که به صورت یک تساوی حاوی اصطلاحات T_1 و T_2 باشد	اگر $I(T_1) = I(T_2)$ آنگاه true و در غیر این صورت false
E7	جمله‌ی تجزیه‌ناپذیر با مسند **P** و دنباله‌ی آرگومان **S**	اگر $I(S)$ در $rel_I(I(P))$ باشد true و در غیر این صورت false
E8	جمله‌ی بولی از نوع نفی با مؤلفه‌ی **C**	اگر $I(C)$ = false آنگاه true و در غیر این صورت false
E9	جمله‌ی بولی از نوع ترکیب عطفی با مؤلفه‌های C_1 ... C_n	اگر $I(C_1) = \ldots = I(C_n)$ = true آنگاه true و در غیر این صورت false
E10	جمله‌ی بولی از نوع ترکیب فصلی با مؤلفه‌های C_1 ... C_n	اگر $I(C_1) = \ldots = I(C_n)$ = false آنگاه false و در غیر این صورت true
E11	جمله‌ی بولی از نوع دلالت ضمنی با مؤلفه‌های C_1 و C_2	اگر $I(C_1)$ = true و $I(C_2)$ = false آنگاه false و در غیر این صورت true
E12	جمله‌ی بولی از نوع دوشرطی با مؤلفه‌های C_1 و C_2	اگر $I(C_2) = I(C_1)$ آنگاه true و در غیر این صورت false
E13	جمله‌ی سوری از نوع عمومی با انقیادهای **N** و بدنه‌ی **B**	اگر برای هر تفسیر J که نسخه‌ی مختصّ N از I است داشته باشیم $J(B)$ = true آنگاه true و در غیر این صورت false
E14	جمله‌ی سوری از نوع وجودی با انقیادهای **N** و بدنه‌ی **B**	اگر برای هر تفسیر J که نسخه‌ی مختصّ N از I است داشته باشیم $J(B)$ = true آنگاه true و در غیر این صورت false
E15	جمله‌ی بی‌قاعده‌ی **S**	$int_I(S)$
E16	عبارت به صورت یک جمله‌ی **S**	$I(S)$
E17	عبارت به صورت یک واردهی شامل نام **N**	اگر $I(\text{text}(I(N)))$ = true آنگاه true و در غیر این صورت false

E18	پیمانه با نام **N**، مجموعه‌ی عدم شمول **L** و متن بدنه‌ی **B**	اگر $[I<\mathbf{L}](\mathbf{B}) = \text{true}$ و $\text{UD}_{[I<L]^*} = ((\mathbf{N}))I_{rel}$ آنگاه true و در غیر این صورت false
E19	متن شامل عبارت‌های $\mathbf{S}_1 \ldots \mathbf{S}_n$	اگر $I(\mathbf{S}_1) = \ldots = I(\mathbf{S}_n) = \text{true}$ آنگاه true و در غیر این صورت false
E20	یک متن **T** با یک نام **N**	UR_I دارای یک مقدار متنی نام‌دار[1] **t** است بدین صورت که $\textit{text}(\mathbf{t}) = \mathbf{T}$ و $\textit{name}(\mathbf{t}) = \mathbf{N}$

به گسترش معنایی که مفاهیم نام‌های خاص معینی (مانند انواع داده‌ای) را اصلاح می‌کند یا روابط بین منطق مشترک و سایر قراردادهای نام‌گذاری، مانند قراردادهای شناسایی شبکه، را مشخص می‌کند، گسترش *خارجی* گفته می‌شود. محدودیت‌های معنایی خارجی مجازند به قواعد و ساختارهایی اشاره کنند که خارج از خود نظریه‌ی مدل تعریف شده‌اند. برای نمونه، گویش CLIF به اعداد اشاره می‌کند. معناشناسی واردەها که در بند بعد توصیف شده خارجی و تعریفی است.

جدول ۱ هیچ تفسیری برای توضیحات[2] ارائه نمی‌دهد. عبارت‌هایی که دارای یک توضیح و یک متن تهی هستند را می‌توان درست در نظر گرفت. عبارت‌های دارای توضیحات پیوست شده باید شروط درستی یکسانی با عبارت‌های یکسان اما بدون توضیحات پیوست شده داشته باشند. بنابراین افزودن یا حذف توضیحات، شروط درستی هیچ متن منطق مشترک را تغییر نمی‌دهد. با این وجود، توضیحات بخشی از نحو صوری هستند و **توصیه می‌شود که** برنامه‌ها آن‌ها را در هنگام انتقال، ویرایش و انتشار مجدد متن منطق مشترک حفظ کنند. در حالت خاص، نامی که در منطق مشترک برای مشخص کردن یک عبارت به کار می‌رود حکم یک شناسه‌ی انعطاف‌ناپذیر سراسری را دارد (مطابق بند ۶-۳). بنابراین **نباید** از همان نام برای ارجاع به یک متن متفاوت استفاده کرد، حتی اگر دو متن معنای یکسانی داشته باشند.

۶-۳ واردسازي و شناسايي در يك شبكه

این بند تنها برای گویش‌هایی که واردەها و/یا متون دارای نام را پشتیبانی می‌کنند قابل استفاده است. (این نحوه‌ی برخورد با نام‌گذاری و شناسایی تا حدی مبتنی بر روش نام‌گذاری و شناسایی مطرح شده در [6] است).

۶-۳-۱ واردەها و عبارت‌هاي نام‌دار[3]

معنای یک عبارت وارده این است که نام آن باید برای شناسایی یک متن منطق مشترک استفاده شود و آن وارده درست است *تنها اگر* آن متن درست باشد. بنابراین، یک وارده یک «رونویسی» مجازی یک محتوای

1 - Named text value
2 - Comments
3 - Importations and named phrases

منطق مشترک از یک «مکان» به مکان دیگر است. منظور از «رونویسی» و «مکان» تنها در زمینه‌ی استقرار[1] محتوای منطقی در یک شبکه‌ی ارتباطی قابل درک است. یک *شبکه‌ی ارتباطی* یا به شکل ساده یک *شبکه*، سامانه‌ای متشکل از تعدادی عامل است که قادرند متن منطق مشترک را نگهداری، منتشر و پردازش نموده و آن را با استفاده از پروتکل‌های انتقال اطلاعات مربوط به آن شبکه بین یکدیگر ردوبدل کنند. پرکاربردترین شبکه تار جهان‌گستر است [7]، اما تعاریف این بند برای هر سامانه‌ی این‌چنینی از عامل‌های ارتباطی قابل استفاده است. به ویژه، هر زیرمجموعه از گره‌های وب را که از قراردادهای ویژه‌ای برای ارتباط استفاده می‌کند می‌توان یک شبکه‌ی منطق مشترک در نظر گرفت. فرض بر این است که یک شبکه از ارتباط و انتشار محتوای منطق مشترک در زیرمجموعه‌ی معینی از گویش‌ها پشتیبانی می‌کند. XCL[2] یک گویش همه‌منظوره برای توزیع محتوای منطق مشترک بر روی هر شبکه‌ای است که از XML پشتیبانی می‌کند.

نام‌های مورد استفاده در نام‌گذاری متون در شبکه *انعطاف‌ناپذیر*[3] بوده و دامنه‌ی کاربردشان *همگانی*[4] است. در نتیجه از یک نام می‌توان برای شناسایی چیزی که به آن نسبت داده شده - که در اینجا همان متن منطق مشترک است - در سراسر شبکه‌ی ارتباطی استفاده کرد (برای توضیحات کامل به [8] مراجعه شود). نامی که بدین شکل به صورت سراسری به معنای صریحش پیوست شده باشد، یک *شناسه* است و به سامانه‌ای از قراردادها و پروتکل‌ها مرتبط است که استفاده از چنین نامی را به منظور شناسایی، مکان‌یابی و انتقال اطلاعات در شبکه‌ای که گویش در آن مورد استفاده است مدیریت می‌کنند. در حالی که جزئیات چنین قراردادهایی فراتر از دامنه‌ی کاربرد این استاندارد است، تأثیر آن‌ها را می‌توان به صورت خلاصه چنین بیان کرد: هدف از انتشار یک متن منطق مشترک دارای نام، تعیین آن نام به عنوان یک شناسه‌ی انعطاف‌ناپذیر برای آن متن است. منطق مشترک برای تصدیق این موضوع *تمام* تفسیرها را به پیروی از چنین قراردادهایی در زمان استفاده از آن‌ها در شبکه‌ای که عمل انتشار در آن انجام می‌گیرد، ملزم می‌کند.

متن‌های دارای نام لازم نیست با مدارک، فایل‌ها یا هر واحد ذخیره‌سازی داده‌ی دیگری دارای تناظر یک به یک باشند. گویش‌ها یا پیاده‌سازی‌ها مجازند متونی را به منظور توزیع در میان واحدهای ذخیره‌سازی یا متون دارای چندین نام را برای ذخیره‌سازی در یک واحد عرضه کنند. قراردادهای نام‌گذاری متن می‌توانند به قراردادهای مورد استفاده در آدرس‌دهی واحدهای داده‌ای مرتبط باشند، اما هیچ الزامی برای این امر وجود ندارد. متون را همچنین می‌توان با استفاده از قراردادهای نام‌گذاری خارجی مشخص کرد. برای نمونه می‌توان متن را در قالب مدارک یا فایل‌هایی که شناسه‌ی شبکه دارند کدگذاری کرد. معناشناسی منطق مشترک که

1 - Context of deployment
2 - eXtended Common Logic Markup Language
3 - Rigid
4 - Global

در این بند توصیف شده **باید** به تمام نام‌هایی که به عنوان شناسه‌های شبکه در شبکه‌ای که متون منطق مشترک در آن انتشار یا انتقال یافته‌اند، قابل اعمال باشد.

عمل نام‌گذاری یک متن از ادعای درستی آن متمایز است. انتشار یک متن دارای نام به خودی خود هیچ گونه اظهار قطعیتی در مورد درستی متن نمی کند، اما معنای صریح نام متن را به‌صورت قطعی مشخص می‌کند.

به منظور وضع شروط معنایی بر روی شناسه‌ها باید فرض کنیم مقادیر مناسبی در جهان گفتمان وجود دارند. به هستار معنایی معادل با یک متن دارای نام، *مقدار متنی دارای نام* گفته می‌شود. ماهیت دقیق مقادیر متنی دارای نام مهم نیست، اما معناشناسی آن‌ها را به صورت زوج‌هایی متشکل از یک نام و یک متن منطق مشترک ($\mathbf{t} = \langle name(\mathbf{t}), text(\mathbf{t})\rangle$) در نظر می‌گیرد. قرارداد شناسه‌ی انعطاف‌ناپذیر یک شرط معنایی خارجی است که *تمام* تفسیرهای متون منتشر شده در شبکه‌ی ارتباطی باید آن شرط را برقرار سازند. انعطاف‌ناپذیری سراسری این نام‌گذاری از عمومیت این نیازمندی قابل دریافت است. توجه داشته باشید که این یک شرط معنایی خارجی است چرا که به ساختاری که به‌وسیله پروتکل‌های شبکه تعریف شده اشاره می‌کند. در ضمن آن را می‌توان به عنوان یک شرط معنایی برروی شبکه در نظر گرفت.

اگر $\mathbf{t}$ یک مقدار متنی در UR_I و $name(\mathbf{t})$ در V باشد آنگاه $int_I(name(\mathbf{t})) = \mathbf{t}$.

انتشار یک متن با یک نام در یک شبکه‌ی ارتباطی تأییدی است بر وجود یک مقدار متنی دارای نام مناسب، با دامنه‌ی کاربرد سراسری، منظور از دامنه‌ی کاربرد سراسری این است که *تمام* تفسیرهای مربوط به هر متن موجود در شبکه باید آن را تصدیق کنند. این نیازمندی در درایه‌ی E20 جدول ۱ آمده است و به صورت زیر می‌توان آن را معنی کرد:

	انتشار در شبکه:	**نیازمند این است که برای هر تفسیر I از یک متن در شبکه:**
E20	یک متن T با یک نام N	UR_I حاوی یک متن دارای یک مقدار متنی دارای نام t باشد بدین صورت که $name(t) = N$ و $text(t) = T$

از آنجا که مفهوم وارده، یک رونویسی مجازی از یک قطعه متن به قطعه‌ای دیگر است (در حقیقت، منظور رونویسی مجازی *بستار واردسازی*[1] است، چرا که لازم است وضعیتی را که در آن متن وارد شده خود حاوی متن وارد شده‌ی دیگری باشد را نیز در نظر گرفت)، این موضوع یک فرض ضمنی را به وجود می‌آورد و آن این است که متون را می‌توان با هم تفسیر کرد و شروط درستی بالا نیز این امر را با اعمال مستقیم تفسیر

1 - Importation closure

عبارت وارد کننده به متن وارد شده منعکس می‌کنند. این در عمل بدین معنی است که هر گونه استفاده از این مفهوم واردسازی متنی باید مبتنی بر این فرض که متون به‌صورت متقابل قابل تفسیر هستند استوار باشد. برای نمونه، واردسازی دلالت بر این دارد که سورها در متن وارد شده باید به گونه‌ای تفسیر شوند که دامنه‌ی تغییرات آن‌ها همانند سورهای متن وارد کننده باشد. تمام متونی که در یک شبکه منتشر و شناسایی می‌شوند **باید** با تمام متون دیگری که می‌توانند آن‌ها را وارد کنند بر روی جهان مرجع و جهان گفتمان یکسان تعریف شده و در حالی که واژه‌نامه‌های آن‌ها ادغام شده، به‌صورت متقابل قابل تفسیر باشند. این شرط برای تمام متونی که ممکن است متون دیگری را وارد کنند برقرار است حتی اگر این کار را در یک وضعیت خاص شبکه انجام ندهند.

شبکه‌های واقعی که پیاده‌سازی شده‌اند در معرض خطا و خرابی هستند. قراردادهای انعطاف‌ناپذیر نام‌گذاری توصیف شده در این بند حتی در چنین شرایط خرابی قابل اجرا هستند. بنابراین، برای نمونه اگر یک URI در وب به عنوان شناسه‌ی انعطاف‌ناپذیر یک متن استفاده شود، آنگاه آن URI حتی زمانی که یک تلاش برای استفاده از آن در یک پروتکل *دریافت*[1] HTTP یک خطای 404 تولید کند، شناسه باقی خواهد ماند. برنامه‌ها **نباید** با خطاهای ارتباطی یا خرابی‌ها به عنوان نشانه‌ی بی‌معنی بودن یا غیرگفتمانی بودن یک نام برخورد کنند.

۶-۳-۲ شبکه‌هاي مرکب[2]

متن در یک شبکه ممکن است در بیش از یک گویش منتشر شود. این استاندارد ملی به چنین شرایطی به عنوان یک شبکه‌ی *مرکب* رجوع می‌کند. توصیه می‌شود تبادل و انتشار اطلاعات در یک شبکه‌ی مرکب به گونه‌ای هدایت شود که تمام عامل‌ها بتوانند محتوای نوشته شده در هر متن مورد استفاده در شبکه را نمایش دهند. یک راه برای دستیابی به این هدف استفاده از رواترین گویش برای انتقال اطلاعات و ملزم کردن عامل‌ها به بیان محتوای خود در این گویش است.

به منظور حفظ قابلیت تفسیر متقابل، هر متن در یک گویش تفکیک‌شده که در یک شبکه‌ی مرکب منتشر شود، **باید** به گونه‌ای منتشر شده باشد که هر گونه واردسازی آن متن به متن دیگری که در یک گویش تفکیک نشده نوشته شده بتواند محتوای متن وارد شده را به گونه‌ای بیان کند که قابلیت تفسیر متقابل وجود داشته باشد. این در حقیقت بدین معناست که باید برای حوزه‌ی گفتمان متن در هر متن تفکیک‌شده یک نام فراهم شده باشد و همچنین هر نام غیرگفتمانی که در چنین متنی واقع شود به شکل مؤثری به‌وسیله برنامه‌هایی که متون تفکیک نشده را پردازش می‌کنند قابل تشخیص باشد. رویه‌ی پیشنهادی در چنین مواردی این است که متن تفکیک‌شده با متن تفکیک نشده‌ای جایگزین شوند که در آن تمام سورها

1 - Get
2 - Mixed networks

به نام حوزه‌ی تفکیک‌شده محدود و یا به‌وسیله آن محافظت می‌شوند و تمام نام‌های غیرگفتمانی بـه صـورت قطعی خارج از آن حوزه باشند. پیمانه‌ها یک شیوه‌ی همه منظوره برای چنین انتشاری فراهم می‌کنند بـدین صورت که متن تفکیک‌شده می‌تواند به عنوان متن بدنه‌ی یک پیمانه به گونـه‌ای منتشـر شـود کـه نـام‌هـای غیرگفتمانی که در متن واقع می‌شوند در لیست عدم شمول آن پیمانه قرار گیرند. از نام پیمانه می‌توان بـرای مشخص کردن یک جهان گفتمان عام مرتبط با گویش یا یک جهـان گفتمـان محلـی مخـتصّ مـتن پیمانـه استفاده کرد.

شبکه‌های پشتیبانی کننده از گویش‌های تفکیک‌شده که دارای قراردادهای لغوی[1] به منظور تمـایز نـام‌هـای گفتمانی از نام‌های غیرگفتمانی هستند، ممکن است به عنوان بخشی از پروتکل انتقـال نیـاز بـه عامـل‌هـایی داشته باشند که بتوانند چنین تمایزهای لغوی را حتی در زمـان اسـتفاده از مـتن تفکیـک‌شـده تشـخیص و ترجمه‌های مناسب را هر جا که لازم باشـد بـه عنـوان بخشـی از پروتکـل انتقـال اعمـال کننـد. امـا چنـین قراردادهایی نمی‌توانند از تبادل اطلاعات در خارج از آن شبکه پشتیبانی کنند. در نتیجه نمـی‌تـوان آن‌هـا را منطبق بر منطق مشترک در نظر گرفت.

٤-٦ برقراري[2]، اعتبار و استلزام[3]

اگر T را یک مجموعه از جمله‌های منطق مشترک (یا یک متن منطق مشترک) در نظر بگیریم آنگاه تفسیر I متن T را *برقرار می‌کند* تنها زمانی که برای هر S در T داشته باشیم $I(S) = true$. یـک مـتن قابـل برقـراری است تنها اگر تفسیری وجود داشته باشد که آن را برقرار کند و در غیـر ایـن صـورت *غیـر قابـل برقـراری* یـا *متناقض* است. اگر هر تفسیری که S را برقرار می‌کند T را نیز برقرار کند آنگاه گوییم S *مستلزم* T است.

تفسیرهای منطق مشترک با جمله‌های بی‌قاعده به عنوان متغیرهای جملـه‌ای مـبهم برخـورد مـی‌کننـد. در گویشی که جمله‌های بی‌قاعده را به رسمیت می‌شناسـد، از تعـاریف فـوق بـه منظـور رجـوع بـه تفسـیرهای مشخص شده به‌وسیله معناشناسی آن گویش استفاده می‌شود. اگر تعاریف فوق با صـفت «منطـق مشـترک» همراه باشند، به عنوان «استلزام منطق مشترک»، آنگاه به تفسیرهایی اشاره می‌کنند که منطبـق بـر شـروط معنایی منطق مشترک هستند. برای نمونه، یک گویش ممکن است جمله‌هـای قیـددار[4] را پشـتیبانی کنـد و معناشناسی آن نیز استلزام «(*P لازم*) *مستلزم P است*» را پشتیبانی کند، اما این را نمـی‌تـوان یـک اسـتلزام منطق مشترک در نظر گرفت حتی اگر زبان به عنوان یک گسترش منطق مشترک، منطبق بر شروط معنایی منطق مشترک باشد. اما استلزام «(*P لازم*) *مستلزم* (*P لازم*) *است*» یک استلزام منطق مشترک است.

1 - Lexical convensions
2 - Satisfaction
3 - Entailment
4 - Modal sentences

در بسیاری از مباحث بعدی رده‌های محدودی از تفسیرها مد نظر است. تمام تعاریف فوق را می‌توان واجد شرایط اعمال در یک رده‌ی محدود معین دانست. بنابراین، گوییم S، foo - مستلزم T است تنها زمانی که برای هر تفسیر I در کلاس foo، اگر I، S را برقرار کند، آنگاه T را هم برقرار کند. استلزام (یا عدم قابلیت برقراری[1]) با توجه به یک رده‌ی تفسیرها دلالت بر استلزام (یا عدم قابلیت برقراری) با توجه به هر زیرمجموعه از آن رده دارد.

در زمان توصیف استلزام T به S، S به مقدم[2] استلزام و T به نتیجه‌ی استلزام منتسب می‌شود.

۶ـ۵ نشان‌گرهاي دنباله، بازگشت و ليست‌هاي آرگومان

نشان‌گرهای دنباله قدرت بیان منطق مشترک را از مرتبه اول فراتر می‌برد. یک نشان‌گر دنباله که در یک دنباله‌ی آرگومان واقع شود، حکم یک دنباله‌ی دلخواه و متناهی از آرگومان‌ها را دارد. یک جمله‌ی عمومی که به یک نشان‌گر دنباله مقید می‌شود با ترکیب عطفی *نامتناهی*[3] تمام عبارت‌هایی که با جایگزینی نشان‌گر دنباله با یک دنباله‌ی متناهی از نام‌ها بدست می‌آیند و همه به سور عمومی مقیدند معنای یکسانی دارد.

این قابلیت نمایش مجموعه‌های نامتناهی جمله‌ها به یک شکل متناهی بدین معنی است که منطق مشترک به همراه نشان‌گرهای دنباله، غیرفشرده بوده و در نتیجه مرتبه اول نیست. در واقع مجموعه‌ی نامتناهی از جمله‌ها که از نظر معنی متناظر یک جمله‌ی منفرد باشد که کمیت یک نشان‌گر دنباله را مشخص می‌کند، به‌طور منطقی معادل آن جمله و در نتیجه مستلزم آن است. اما این موضوع برای هیچ زیرمجموعه‌ی متناهی از آن مجموعه‌ی نامتناهی صادق نیست. به هر حال، هدف از جمله‌های دارای نشان‌گر دنباله، استفاده از آن‌ها به عنوان طرح‌های اصول بدیهی[4] است نه اینکه به عنوان نتیجه مطرح باشند و زمانی که آن‌ها محدود به چنین استفاده‌ای هستند، منطق حاصل فشرده است. بدین ترتیب، نشان‌گرهای دنباله تنها مجاز به انقیاد به سورهای عمومی در سطح عبارت اصلی متن بوده و این جمله‌ها تنها به عنوان اصول بدیهی قابل استفاده‌اند نه نتیجه. این محدودیت اغلب مناسب متونی است که به عنوان «هستان‌شناسی» در نظر گرفته می‌شوند. منظور از هستان‌شناسی‌ها منابع اطلاعات معتبری است که نمایشی مفهومی از یک حوزه‌ی کاربرد ارائه داده و به منظور اعمال بر سایر داده‌ها استفاده می‌شوند.

یک گویش فشرده که از نشان‌گرهای دنباله پشتیبانی نمی‌کند می‌تواند بخش عمده‌ای از قابلیت‌های فراهم شده به‌وسیله نشان‌گرهای دنباله را با استفاده از لیست‌های آرگومان صریح که در منطق مشترک با استفاده از اصطلاحات تولید شده به‌وسیله یک تابع مولد لیست نشان داده می‌شوند تقلید کند. یک نشان‌گر دنباله به نام یک لیست ترجمه شده و استفاده از سور بر روی لیست نام‌ها جایگزین استفاده از سور بر روی

1 - Unsatisfiability
2 - Antecedent
3 - Infinite
4 - Axiom schemata

نشان‌گرهای دنباله می‌شود. شرط متناهی بودن دنباله‌ها با فرض ممیز ثابت ضمنی بر روی تمام مدل‌های «استاندارد» اصول بدیهی لیست[1] تطبیق می‌گردد. چنین قراردادهایی به‌طور گسترده در کاربردهای برنامه-نویسی منطق، RDF و OWL استفاده می‌شوند. هزینه‌ی چنین روشی کاهش قابل توجه وضوح نحوی و خوانایی، نیاز به مجاز دانستن لیست‌ها به عنوان هستار در حوزه‌ی گفتمان و ممکن است اتکاء به نرم‌افزار خارجی برای فرابری لیست‌ها باشد. مزیت‌های این روش عبارت است از: قابلیت ارائه‌ی لیست‌های آرگومان دلخواه با استفاده از تنها تعداد اندکی از مبانی[2] و استفاده از یک منطق مبنای فشرده. پیاده‌سازی‌هایی که مبتنی بر ساخت لیست‌های آرگومان استوارند اغلب محدود به قدرت بیان متعارف مرتبه اول هستند و نمی-توانند تمام استنتاج‌هایی را که مستلزم استفاده از سور بر روی لیست‌ها هستند پشتیبانی کنند. این موضوع می‌تواند به عنوان یک مزیت یا یک عیب در نظر گرفته شود.

۶-۶ موارد ویژه و ترجمه‌هاي بين گويش‌ها

به یک گویش تفکیک‌شده که در آن تمام عملگرها و مسندها نام‌های غیرگفتمانی بوده و تمام نام‌های غیرگفتمانی نیز عملگر یا مسند هستند، گویش *رده‌ای*[3] گفته می‌شود.

یک تفسیر I *هموار*[4] است زمانی که داشته باشیم: $\mathrm{UD}_I = \mathrm{UR}_I$. اگر rel_I و fun_I همان توابع همانی تعریف شده بر UR_I - UD_I باشند آنگاه این تفسیر *وجودی* است. بدین ترتیب هستارهای درون جهان مرجع که خارج از حوزه هستند حکم گسترش‌های نام‌های غیرگفتمانی را دارند. این دو تابع به ترتیب برای یک گویش تفکیک نشده و یک گویش رده‌ای مناسب هستند. شکل کلی تفسیر که در بالا توصیف شد اجازه می‌دهد هر دو نوع گویش به همراه سایر گویش‌ها به‌وسیله یک ترکیب منفرد ترجمه شوند.

برای گویش‌های تفکیک نشده تنها لازم است تفسیرهای هموار در نظر گرفته شوند. بدین صورت که برای هر تفسیر داده شده‌ی I یک تفسیر هموار J وجود دارد که عبارت‌های یکسان هر متنی از گویش را همانند I برقرار می‌کند. J را می‌توان به سادگی با اعلان UR_I به صورت UD_I بدست آورد. برای یک گویش تفکیک نشده معانی تمام نام‌ها در UD_I قرار دارند. در نتیجه تمام عناصر خارج از UD_I به شروط درستی نامرتبط هستند.

برای گویش‌های رده‌ای تنها لازم است تفسیرهای وجودی در نظر گرفته شوند. بدین صورت که برای هر تفسیر داده شده‌ی I یک تفسیر وجودی J وجود دارد که عبارت‌های یکسان هر متنی از گویش را همانند I برقرار می‌کند. J را می‌توان با جایگزین کردن $I(\mathrm{x})$ با $fun_I(I(\mathrm{x}))$ برای هر عملگر x و با $rel_I(I(\mathrm{x}))$ برای هر مسند x در واژه‌نامه و حذف آن‌ها از حوزه در صورت وجود در آن بدست آورد. از آنجا که تمام عملگرها و

1 - List axioms
2 - Primitives
3 - Classical
4 - Flat

مسندها در یک گویش رده‌ای بر شروط درستی تنها از طریق گسترش‌های مرتبطشان تأثیر می‌گذارند، این عمل جایگزینی تأثیری بر هیچ کدام از ارزش‌های درستی ندارد. بیان صوری این مطلب بدین صورت است:

$\mathrm{UD}_J = \mathrm{UD}_I - \{I(\mathrm{v})$: v یک عملگر یا مسند در V است$\}$، برای هر نام غیرگفتمانی x داریم: $int_J(\mathrm{x})$ $= int_I(\mathrm{x})$، برای هر مسند x داریم: $int_J(\mathrm{x}) = rel_I(int_I(\mathrm{x}))$ و برای هر عملگر x داریم: $int_J(\mathrm{x}) = fun_I(int_I(\mathrm{x}))$.

۱-۶-۶ ترجمه بین گویش‌ها

یک *ترجمه* نگاشتی از عبارت‌های یک متن در گویش A به عنوان گویش مبدأ به عبارت‌های یک متن در گویش B به عنوان گویش مقصد است به گونه‌ای که برای هر تفسیر I از واژه‌نامه‌ی متن متعلق به A یک تفسیر J از واژه‌نامه‌ی متن متعلق به B وجود دارد و برای هر تفسیر J از واژه‌نامه‌ی متن متعلق به B یک تفسیر I از واژه‌نامه‌ی متن متعلق به A وجود دارد به گونه‌ای که برای هر عبارت E در متن متعلق به A داشته باشیم $I(\mathrm{E}) = J(\mathrm{tr}(\mathrm{E}))$ که در آن tr همان ترجمه‌ی مورد نظر است. از آنجا که تمام گویش‌های منطق مشترک شروط درستی یکسانی دارند ترجمه اغلب سرراست است. پیچیدگی‌ها اغلب در ترجمه‌ی بین گویش‌های تفکیک‌شده و گویش‌های تفکیک نشده رخ می‌دهد.

در ترجمه از یک گویش تفکیک‌شده‌ی A به یک گویش تفکیک نشده‌ی B لازم است ترجمه نشان دهد کدام اصطلاحات در A غیرگفتمانی هستند. از آنجا که تمام نام‌ها در گویش تفکیک نشده هستارهایی را در حوزه مشخص می‌کنند، لازم است ترجمه یک نام *گفتمانی* را معرفی کند که گسترش آن در B حوزه‌ی یک تفسیر از A است. ترجمه همچنین باید محدوده‌ی تغییرات تمام سورهای درون متن را به این حوزه محدود کرده و تصدیق کند که نام‌های غیرگفتمانی گویش تفکیک‌شده هستارهایی را در خارج از این حوزه مشخص می‌کنند. هیچ ترجمه‌ی دیگری مورد نیاز نیست. ساختار پیمانه یک روش همه منظوره برای چنین ترجمه‌هایی فراهم می‌کند، بدین صورت که متن در A با یک پیمانه در B که با نام حوزه نام‌گذاری شده باشد و نام‌های غیرگفتمانی متن در لیست عدم شمول آن پیمانه برشمرده شده باشند معنی یکسانی دارد.

برای ترجمه از یک گویش تفکیک نشده‌ی B به یک گویش تفکیک‌شده‌ی A لازم است نام‌ها به گونه‌ای استفاده شوند که به محدودیت‌های گویش احترام گذاشته شود. برای این امر ممکن است لازم باشد اصول بدیهی به ترجمه‌ها اضافه شود تا اطمینان حاصل شود حوزه‌ی تفسیر ترجمه‌ی تفکیک‌شده‌ی هر متنی متناظر با جهان مرجع تفسیر متن تفکیک نشده است. یک روش عمومی با نام *ترجمه‌ی holds-app* برای ترجمه‌ی هر گویش منطق مشترک به یک گویش رده‌ای مشابه وجود دارد. فرض می‌کنیم که یک مسند *holds* و یک عملگر *app* در اختیار داریم که در هیچ واژه‌نامه‌ای وجود ندارند. به‌طور خاص (برای گویش‌های تفکیک نشده) یک جمله‌ی تجزیه‌ناپذیر با یک مسند P و دنباله‌ی آرگومان $\mathrm{S}_1 \ldots \mathrm{S}_n$ به یک جمله‌ی

تجزیه‌ناپذیر با مسند *holds* و دنباله‌ی آرگومان $S_1 \ldots S_n$ P ترجمه می‌شود. یک اصطلاح با عملگر O و دنباله‌ی آرگومان $S_1 \ldots S_n$ به یک اصطلاح با عملگر *app* و دنباله‌ی آرگومان $S_1 \ldots S_n$ O ترجمه می‌شود. مسند و عملگر معرفی شده نیاز به هیچ اصل بدیهی دیگری ندارند. تنها نقش آن‌ها این است که به عملگرها و مسندهای گویش B اجازه دهند که هستارهایی را در حوزه‌ی ترجمه‌ی گویش A مشخص کنند (ترجمه‌ی *holds-app* برای گویش‌های تفکیک‌شده به همین اندازه واضح اما برای بیان تا حدی پیچیده‌تر است).

بعضی گویش‌ها انواع مختلف محدودیت‌های نشانه‌گذاری را تحمیل می‌کنند مانند الزام نام‌های مقید به داشتن یک شکل لغوی خاص یا الزام استفاده از عملگرها و مسندها به همراه طول مشخصی از دنباله‌ی آرگومان (طبق قرارداد به آن *arity* عملگر یا مسند گفته می‌شود). ترجمه به یک گویش با چنین محدودیت‌هایی را اغلب می‌توان با بازنویسی نام‌ها به منظور انطباق با چنین محدودیت‌هایی یا «جناس زدایی[1]» موارد وقوع یک نام که لازم است در گویش مقصد متمایز باشند (برای نمونه، با افزودن پسوند به منظور نشان دادن arity) انجام داد. در این موارد همچنین ممکن است لازم باشد برای هر arity، عملگرها و مسندهای *holds-n* و *app-n* متمایزی معرفی شود. برنامه‌هایی که باید چندین متن را صادقانه ترجمه کنند باید سازگاری مابین چنین بازنویسی‌های نام را حفظ کنند.

۷ انطباق

سه نوع انطباق را برای منطق مشترک می‌توان مشخص کرد. ممکن است شروطی بر یک گویش (که همان مشخصه‌ی یک زبان است)، بر یک برنامه (که منطبق بر استاندارد است) یا بر یک شبکه حاکم باشد.

۷-۱ انطباق گویشی

برای اینکه یک زبان یا یک نشانه‌گذاری، یک گویش منطق مشترک به حساب آید، *مشخصه‌ی* آن باید شرط‌هایی داشته باشد. انطباق به دو شیوه مشخص می‌شود: نحوی و معنایی. انطباق نحوی و معنایی یک گویش می‌توانند جداگانه مشخص شوند گر چه تمام ترکیبات ممکن مفید و معنادار نیستند.

۷-۱-۱ نحو

یک گویش بر روی مجموعه‌ای از نوشته‌های خطی تعریف می‌شود که **باید** تصریح شوند. این نوشته‌های خطی به طور کلی بهتر است رشته‌های حرفی یونیکد باشند اما سایر نوشته‌های خطی مانند نمایش‌های ترسیمی از جمله گراف‌های جهت‌دار یا عکس‌های ساخت‌یافته نیز امکان‌پذیر است. روشی **باید** برای گویش مشخص شود که هر نوشته‌ی درون مجموعه را به‌صورت غیرمبهم تجزیه یا آن را به عنوان غیرمجاز نحوی رد کند. برای نوشته‌های از نوع رشته‌ی حرفی یونیکد یک نحو در قالب گسترش‌یافته‌ی Backus-Naur

1 - De-punning

مشخصه‌ی دقیقی است. یک *تجزیه* عبارت است از تخصیص هر بخش یک نوشته‌ی مجاز به رده‌ی نحو انتزاعی منطق مشترک متناظرش در بند ۶-۱-۱، و نوشته‌ی تجزیه شده یک *عبارت* است.

یک گویش ***منطبق کامل نحوی*** است اگر تجزیه‌های آن قادر به تشخیص عبارت‌ها برای هر رده از نحو انتزاعی بند ۶-۱-۱ باشند. به منظور انطباق با منطق مشترک، گویش‌ها یا زیرگویش‌هایی که تجزیه‌های آن‌ها رده‌های دیگری از جمله‌ها را شامل می‌شوند **باید** یا (الف) آن‌ها را به عنوان جمله‌های بی‌قاعده رده‌بندی کنند یا (ب) مشخص کنند این رده‌ها چگونه به رده‌های نحو انتزاعی تعریف شده در بند ۶-۱-۱ نگاشت داده شوند. اگر گویشی مطابق حالت الف منطبق باشد، به آن گویش یا زیرگویش *گسترش معنایی* گفته می‌شود (بند ۷-۱-۲ ملاحظه شود). اگر حداقل یکی از رده‌های منطق مشترک را تشخیص دهد به آن منطبق به عنوان یک ***زیرگویش نحوی*** گفته می‌شود. اما هر گویش **باید** یک شکل از رده‌ی جمله را تشخیص دهد. یک گونه‌ی خاص از زیرگویش نحوی شناسایی شده و به آن ***زیرگویش فشرده*** گفته می‌شود. زیرگویش فشرده گویشی است که تمام رده‌ها به استثنای نشان‌گرهای دنباله را تشخیص دهد.

یک گویش ***تفکیک‌شده‌ی نحوی*** است اگر تجزیه برای بررسی مجاز بودن یک عبارت در آن گویش نیاز به تمایز بین رده‌های لغوی نام‌های منطق مشترک داشته باشد. گویش‌های تفکیک‌شده **باید** قیودی را مشخص کنند که یک برنامه را قادر سازد رده‌ی یک نام در گویش را بدون عملیات بر روی هیچ ساختار دیگری غیر از خود آن نام تشخیص دهد.

۷-۱-۲ معناشناسي

هر گویش منطق مشترک **باید** یک معناشناسی مبتنی بر نظریه‌ی مدل داشته باشد که بر روی مجموعه‌ای از تفسیرها تحت عنوان *تفسیرهای گویش* تعریف شده و به هر جمله، عبارت (به استثنای توضیح) یا متن در آن گویش یکی از دو ارزش درستی *true* یا *false* را تخصیص دهد.

یک گویش ***به‌طور دقیق منطبق معنایی*** است اگر برای هر جمله، عبارت (به جز توضیح) یا متن T در آن گویش که از نظر نحوی مجاز باشد، دو شرط انطباق زیر برقرار باشد:

- برای هر تفسیر گویشی J از T یک تفسیر منطق مشترک I از T وجود دارد بطوریکه $I(T) = J(T)$،
- برای هر تفسیر منطق مشترک I از T یک تفسیر گویشی J از T وجود دارد بطوریکه $I(T) = J(T)$.

به دنبال آن، مفاهیم قابلیت برقراری، تناقض و استلزام متناظر با تفسیرهای گویشی و تفسیرهای منطق مشترک برای یک گویشِ به‌طور دقیق منطبق یکسان خواهند بود.

گویش‌های تفکیک‌شده‌ی نحوی ممکن است ملزم به برقراری شروط اضافی باشند. توضیحات *پایین* ملاحظه شود.

آسان‌ترین راه برای دستیابی به انطباق دقیق معنایی استفاده از نظریه‌ی مدل منطق مشترک به عنوان معناشناسی مبتنی بر نظریه‌ی مدل برای گویش است. اما تعریف به نحوی عبارت‌بندی شده تا استفاده از راه‌های دیگر فرمول‌بندی نظریه‌ی معنایی تنها در صورت حفظ قابلیت برقراری، تناقض و استلزام مجاز باشد.

یک ***زیرگویش معنایی*** یک زیرگویش نحوی است (به بند ۷-۱-۱ مراجعه کنید) که در شروط معنایی صدق می‌کند بدین معنی که تنها بعضی از بخش‌های منطق مشترک کامل را به رسمیت می‌شناسد و تفسیرهای آن معادل محدودیت‌های یک تفسیر منطق مشترک به آن بخش‌ها هستند.

یک ***گسترش معنایی*** گویشی است که شرط اول را برقرار می‌کند اما شرط دوم خیر. به عبارت دیگر، یک گویش گسترش معنایی دارای بخش‌هایی است که تفسیر آن‌ها مقیدتر از یک تفسیر منطق مشترک است. از این حیث، هر گویشی که شروط معنایی غیربدیهی بر جمله‌های بی‌قاعده تحمیل کند یک گسترش معنایی است.

بدین ترتیب یک گسترش معنایی مجاز است علاوه بر شروط معنایی منطق مشترک شروط معنایی خارجی به جمله‌های بی‌قاعده اعمال کند. CLIF به خاطر شروط معنایی که بر اعداد و رشته‌های نقل قول شده اعمال می‌کند نمونه‌ای از یک گسترش معنایی است.

به گسترش‌های معنایی **باید** به جای عبارت‌های به‌طور دقیق منطبق یا به شکلی ساده‌تر «منطبق» به عبارت‌های «گسترش معنایی منطبق» یا «گسترش منطبق» منتسب شود. برای جمله‌ها، عبارت‌ها و متون یک گسترش منطبق، تناقض و استلزام با توجه به معناشناسی منطق مشترک به ترتیب دلالت بر تناقض و استلزام با توجه به معناشناسی گویش دارد، اما عکس این حالت برقرار نیست. در ضمن، برقراری با توجه به معناشناسی گویش دلالت بر برقراری با توجه به معناشناسی منطق مشترک دارد، اما عکس این حالت برقرار نیست. این بدین معنی است که موتورهای استنتاج که استنتاج‌های منطق مشترک انجام می‌دهند برای گویش درست عمل می‌کنند اما ممکن است کمتر کامل باشند.

یک ***گویش تفکیک‌شده*** یک گویش تفکیک‌شده از نظر نحوی است که لازم می‌داند نام‌ها در یک یا چند رده برای مشخص کردن هستارهایی که متعلق به مجموعه‌ی مشخص کننده‌ی محدوده‌ی تغییرات سورهای آن هستند، به کار نروند. برای نمونه، نحو سنتی مرتبه اول ممکن است به گونه‌ای تفسیر شود که لازم باشد نام‌های روابط برای مشخص کردن افراد به کار نروند. به منظور انطباق، گویش‌های تفکیک‌شده علاوه بر انطباق معنایی باید (الف) قیود نحوی را فراهم کنند که یک برنامه را قادر سازد تشخیص دهد که نام یک گویش، غیرگفتمانی است و آن را از نظر نحوی به عنوان یک نام غیرگفتمانی رده‌بندی کند. و (ب) به عنوان بخشی از انتشار هر جمله، عبارت یا متن منتشر شده‌ی گویش، نامی را فراهم کند که سایر گویش‌ها بتوانند از آن برای ارجاع به جمله، عبارت یا متن منتشر شده استفاده کنند. منظور این است که گویش باید به

عنوان یک شرط معنایی در تمام تفسیرهای گویش مشخص کند که گسترش رابطه‌ای این نام باید از حیث تمام هستارهای درون حوزه‌ی تفسیر درست باشد. ساختار پیمانه به منظور تسهیل این نیازمندی انطباقی در نحو انتزاعی تعبیه شده است.

هیچ گویشی مجاز نیست محدوده‌ی تغییرات سوری یک گویش دیگر را تغییر دهد. گویش‌های دیگر مجازند با تمام نام‌ها به عنوان نام‌های گفتمانی برخورد کنند حتی اگر این نام‌ها در گویش تفکیک‌شده به عنوان غیرگفتمانی اعلان شده باشند.

۲ـ۷ انطباق برنامه‌ای

منظور از «برنامه» هر قطعه از ماشین آلات محاسباتی (نرم‌افزار، سخت‌افزار یا یک شبکه) است که هر نوع عملیاتی را بر روی متن منطق مشترک انجام دهد (حتی جزئی‌ترین عملیات مانند ذخیره‌ی آن برای انتقال دوباره در آینده).

انطباق برنامه‌ها نسبت به مجموعه‌ای از گویش‌ها تعریف می‌شود که به آن *مجموعه‌ی انطباق* گفته می‌شود. برنامه‌هایی که منطبق با گویش XCL هستند، بدون هیچ قیدی به آن‌ها «منطبق» گفته می‌شود.

تمام برنامه‌های منطبق **باید** قادر به پردازش تمام نوشته‌های مجاز مربوط به گویش‌های عضو مجموعه‌ی انطباق باشند. برنامه‌هایی که متن منطق مشترک را وارد، خارج یا انتقال می‌دهند، حتی اگر با استفاده از سایر قراردادهای متنی پردازش شده گنجانده شده باشد، **باید** قادر به بردن و آوردن هر متن منطق مشترک باشند. منظور این است که باید نوشته‌های ورودی را بدون آنکه متن آن‌ها را تغییر دهند خارج کرده یا انتقال دهند.

برنامه‌هایی که روابط استلزام بین متون منطق مشترک عضو مجموعه‌ی انطباق را کشف می‌کنند زمانی **درست** هستند که برای متون دلخواه T و S متعلق به گویش‌های عضو مجموعه‌ی انطباق، اگر برنامه استلزام S به T را کشف کند آنگاه S مستلزم منطق مشترک T باشد (منظور این است که برای هر تفسیر منطق مشترک I، اگر $I(S) = true$ آنگاه داشته باشیم $I(T) = true$). برنامه، زمانی **کامل** است که برای متون دلخواه T و S در گویش‌های عضو مجموعه‌ی انطباق اگر S مستلزم منطق مشترک T باشد آنگاه برنامه بتواند استلزام S به T را تشخیص دهد (توجه کنید این امر نیازمند کامل بودن «سرتاسر» گویش‌های عضو مجموعه‌ی انطباق است).

کامل بودن نیازمند این نیست که برنامه بتواند استلزام را در یک گسترش معنایی که استلزام منطق مشترک نیست کشف کند. اگر یک گویش یک گسترش معنایی باشد آنگاه یک برنامه برای آن گویش **کامل گویشی**[1] است اگر برای هر تفسیر گویشی I از آن گویش هرگاه $I(S) = true$ نتیجه دهد $I(T) = true$ آنگاه

1 - Dialect complete

برنامه استلزام S به T را کشف کند. کامل بودن گویشی برای گویش D دلالت بر کامل بودن برای {D} دارد، اما نه برعکس.

۳ـ۷ انطباق شبکه‌اي

انطباق شبکه‌های ارتباطی نسبت به مجموعه‌ای از گویش‌ها تحت عنوان مجموعه‌ی انطباق تعریف می‌شود. یک شبکه منطبق است اگر تمام عبارت‌های مربوط به تمام گویش‌های عضو مجموعه‌ی انطباق را بدون انحراف از هر گره‌ای در شبکه به هر گره‌ی دیگر انتقال داده و شناسه‌های شبکه‌ای را فراهم کند که شروط معنایی E17 و E20 توصیف شده در بند ۶-۲ را برقرار کنند. خطاهای انتقال و خرابی‌های شبکه که به عنوان شرایط خطا نشان داده می‌شوند از نقطه نظر انطباق یک شبکه انحراف محسوب نمی‌شوند.

پیوست الف
قالب تبادل منطق مشترك (CLIF)
(الزامی)

الف-۱ مقدمه

از نظر تاریخی پروژه‌ی منطق مشترک از تلاش برای به روز رسانی و منطقی کردن طراحی KIF [3] سرچشمه گرفته است. KIF بیش از یک دهه قبل به عنوان یک «قالب تبادل اطلاعات» مطرح شده بود و در یک شکل ساده شده به یک نشانه‌گذاری استاندارد غیررسمی در بسیاری از کاربردهای منطق تبدیل شده است. بسیاری از ویژگی‌های منطق مشترک به خصوص استفاده‌ی آن از نشان‌گرهای دنباله به صورت آشکار از KIF اقتباس شده است. به هر حال، فلسفه‌ی طراحی منطق مشترک با KIF از جنبه‌های مختلف متفاوت است که در اینجا به مرور آن‌ها می‌پردازیم.

نخست، اهداف زبان‌ها متفاوت است. KIF قرار بود یک نشانه‌گذاری عام باشد که سایر زبان‌ها بدون از دست‌رفتن معنی به آن قابل ترجمه باشند. هدف از ارائه‌ی منطق مشترک استفاده از آن به منظور تبادل اطلاعات در یک شبکه است البته تا حد امکان بدون نیاز به هیچ ترجمه‌ای و زمانی هم که چنین ترجمه‌ای لازم باشد، منطق مشترک یک چارچوب *معنایی* منفرد عام فراهم می‌کند نه یک زبان بین‌المللی با نحو مشخص.

دوم، به KIF به عنوان یک زبان «کامل» نگاه می‌شد که دارای یک نحو نمایشی برای انواع مختلف شکل‌های عبارت‌هایی مانند مرتب‌سازی سورها، قالب‌های گوناگون تعریف و یک ابرزبان به‌طور کامل گویا باشد. هدف فراهم کردن زبانی بود که گستره‌ی وسیعی از سایر زبان‌ها به آن به‌صورت مستقیم قابل نگاشت باشند. منطق مشترک برخلاف KIF به‌طور عمدی «کوچک» نگه داشته شده است. این امر بیان یک معناشناسی دقیق و مشخص کردن محدوده‌ی دقیق قدرت بیان زیرمجموعه‌های زبان را آسان‌تر کرده و اجازه می‌دهد زبان‌های گسترش‌یافته به صورت کدگذاری‌های نظریه‌های بدیهی بیان شده در منطق مشترک تعریف شوند.

سوم، KIF آشکارا مبتنی بر لیسپ بود. نحو KIF به گونه‌ای تعریف شده بود که با عبارت‌های S در لیسپ یکی باشد. در ضمن، ایده‌های مبتنی بر لیسپ، مانند شیوه‌ی تعریف معنای متغیرهای دنباله‌ای، در معناشناسی آن به‌کار رفته بود. گرچه نحو سطحی CLIF در استفاده‌ی تو در تو از پرانتزهای بدون برچسب یک ظاهر شبیه لیسپ را حفظ کرده و همانند عبارت‌های S در لیسپ قابل تجزیه است، اما منطق مشترک مبتنی بر لیسپ نیست و هیچ فرض اولیه‌ای در مورد هیچ کدام از ساختارهای لیسپ ندارد. نشانه‌گذاری پیشنهادی برای تبادل منطق مشترک مبتنی بر XML است. XML استانداردی است که در زمان طراحی نسخه‌ی اصلی KIF در اختیار نبود.

در خاتمه، بسیاری از ویژگی‌های «جدید» منطق مشترک برگرفته از ایده‌هایی است که از کار برروی زبان‌های وب معنایی[1] [9] سرچشمه گرفته است.

نامی که برای نحو مشابه KIF منطق مشترک انتخاب شده قالب تبادل منطق مشترک (CLIF) است. هدف از انتخاب این نام مشخص کردن این قالب به عنوان نسخه‌ی تعیین شده در این استاندارد و تمایز آن از سایر گویش‌های KIF است که ممکن است به‌طور دقیق سازگار نباشند.

KIF و CLIF از جنبه‌های متعدّدی مشابهند. هر دو زبان به عنوان زیرگویش دارای نحو برای منطق مرتبه اول هستند و هر دو دارای نشانه‌گذاری برای متغیرهای دنباله‌ای هستند (که در این استاندارد ملی به آن‌ها نشان‌گرهای دنباله گفته می‌شود). هر دو زبان به‌طور انحصاری از یک قرارداد نشانه‌گذاری پیشوندی و قراردادهای نحوی به سبک عبارت‌های S استفاده می‌کنند، هر دو از پرانتز به عنوان جداکننده‌ی لغوی[2] استفاده می‌کنند و همچنین هر دو به‌طور مشابه محدودیت‌های سوری را نشان می‌دهند.

چند تفاوت شناخته شده بین KIF و CLIF عبارتند از:

۱- KIF از کدگذاری اَسکی و CLIF از کدگذاری یونیکد استفاده می‌کند.

۲- KIF دارای نشانه‌گذاری صریح برای تعریف توابع و روابط است که CLIF فاقد آن است.

۳- KIF از نشانه‌گذاری نام محصور[3] استفاده نمی‌کند درحالی که CLIF استفاده می‌کند.

۴- KIF از نماد «@» به عنوان یک پیشوند متغیر دنباله‌ای استفاده می‌کند درحالی که CLIF از دنباله‌ی سه نقطه برای نشان‌گرهای دنباله استفاده می‌کند.

۵- KIF توضیحات را به شیوه‌ای متفاوت با CLIF اداره می‌کند و از ساختار «حصر»[4] برخوردار نیست.

۶-KIF ساختار نقش-زوج[5] را ندارد در حالی که CLIF دارد.

۷- KIF دارای مفاهیم واردسازی، متن، عبارت و پیمانه نیست در حالی که CLIF هست.

۸- KIF متغیرها را از نام‌ها تمیز می‌دهد و لازم می‌داند که سورها تنها به متغیرها مقید باشند در حالی که CLIF چنین تمایزی را قائل نیست.

۹- با متغیرهای مستقل در KIF به صورت متغیرهای با سور عمومی برخورد می‌شود. این در حالی است که نام‌های مستقل در CLIF فقط نام هستند و دلالت بر هیچ گونه سوری ندارند.

۱۰- عملگرها و مسندها در KIF باید نام باشند. اما CLIF استفاده از اصطلاحات عمومی را مجاز می‌داند و نیز اجازه می‌دهد این نام‌ها به سورها مقید باشند.

---

1 - Semantic web
2 - Lexical delimiter
3 - Enclosed name
4 - ‘Enclosing’ construction
5 - Role-pair

۱۱- KIF ساختار سور محافظت شده را پشتیبانی نمی‌کند.

الف-۲ نحو CLIF

نحو زیر با استفاده از قالب توسعه‌یافته‌ی Backus-Naur (EBNF) نوشته شده است. نویسه‌های لفظی[1] در میان دو علامت نقل قول منفرد قرار می‌گیرند، دنباله‌های اقلام با استفاده از کاما از هم جدا می‌شوند، | برای تفکیک بدل‌ها[2] به کار می‌رود، {} دنباله‌ای از صفر یا تعداد بیشتر عبارت‌های متعلق به رده‌ی محصور در آن را نشان می‌دهد، - یک استثناء را نشان می‌دهد، [] یک قلم اختیاری را نشان می‌دهد و () برای گروه‌بندی نویسه‌ها استفاده می‌شود. هر ساخت[3] به «;» ختم می‌شود.

نحو به منظور اعمال به کدگذاری‌های اَسکی نوشته شده است. این نحو همچنین به تمام کدگذاری‌های حرفی یونیکد قابل اعمال است، البته با در نظر گرفتن تغییری که باید در رده‌ی غیراسکی اعمال شود و در زیر بدان اشاره شده است.

نحو در اینجا در دو بخش ارائه شده است. بخش اول به تجزیه‌ی جریان‌های حرفی به اقلام لغوی می‌پردازد و بخش دوم به نحو منطقی CLIF اختصاص دارد و با این فرض که اقلام لغوی به‌وسیله یک تحلیلگر لغوی از هم جدا شده‌اند نوشته شده است. این شیوه‌ی ارائه‌ی نحو به نحو عبارت اجازه می‌دهد از پیچیدگی‌های ناشی از اداره‌ی نویسه‌های فاصله[4] صرف نظر کند.

الف-۲-۱ نویسه‌ها

هر عبارت CLIF به صورت دنباله‌ای از نویسه‌های یونیکد تعریف شده در استاندارد ISO/IEC 10646:2003 کدگذاری می‌شود. هر کدگذاری حرفی که از مجموعه‌ی آورده شده در استاندارد ISO/IEC 10646:2003 پشتیبانی کند برای استفاده مجاز است، اما UTF-8 [5] (پیوست د ISO/IEC 10646:2003) ارجحیت دارد. تنها نویسه‌های عضو زیرمجموعه‌ی US-ASCII برای استفاده‌ی خاص در خود CLIF رزرو شده‌اند. بنابراین زبان را در صورت لزوم می‌توان به صورت یک رشته‌ی متنی اَسکی کدگذاری کرد. این استاندارد ملی از نویسه‌های اَسکی استفاده می‌کند. نویسه‌های یونیکد خارج از محدوده‌ی اَسکی در متن اَسکی CLIF به صورت دنباله‌ای از کدهای حرفی به شکل \u*nnnn* یا \U*nnnnnn* نمایش داده می‌شوند که در آن *n* یک عدد مبنای شانزده است. زمان تبدیل یک رشته‌ی متنی اَسکی به یک مجموعه‌ی کامل کدگذاری حرفی یا زمان چاپ یا ارائه‌ی متن به منظور حداکثر دسترس‌پذیری برای خواننده‌های انسانی چنین دنباله‌ای **مجاز است** با

1 - Literal characters
2 - Alternatives
3 - Production
4 - Whitespace characters
5 - UCS (Universal Character Set) Transformation Format

کدگذاری متناظر با حرف یا یک نشان تصویری مناسب جایگزین شود. علاوه بر این، این دنباله‌های کدگذاری اگر در رشته‌های نقل قول شده واقع شوند به مفهوم حرف یونیکد متناظرشان هستند.

نحو در قالب بلوک‌های منفصلی از نویسه‌ها تحت عنوان *نشانه‌های لغوی* (به صورتی که در استاندارد ISO/IEC 2382-15:1999، بند ۱۵-۱ در مورد نشانه‌های لغوی آمده است) تعریف می‌شود. یک جریان حرفی را می‌توان با یک فرآیند ساده‌ی واژه‌سازی به جریانی از نشانه‌های لغوی تبدیل کرد. این فرآیند در جریان ورودی به دنبال تعداد کمی نویسه‌های *جداکننده* می‌گردد. هر حرف جداکننده نشان‌دهنده‌ی خاتمه‌ی یک نشانه‌ی لغوی و شروع نشانه‌ی بعدی است. هر دنباله‌ی متوالی از نویسه‌های فاصله به عنوان یک *تفکیک کننده*[1] بین نشانه‌های لغوی عمل می‌کند (به استثنای درون رشته‌های نقل قول شده و نام‌ها). نویسه‌های معینی به منظور استفاده‌ی ویژه به عنوان حرف اول یک قلم لغوی رزرو شده‌اند. از حرف نقل قول دوتایی[2] «"» (U+0022) به منظور شروع و خاتمه‌ی نام‌های حاوی جداکننده‌های حرفی استفاده می‌شود. حرف نقل قول منفرد «'» (U+002C) برای شروع و خاتمه‌ی رشته‌های نقل قول شده استفاده می‌شود. این رشته‌ها همچنین اقلام لغوی هستند که ممکن است دارای نویسه‌های جداکننده باشند. علامت تساوی زمانی که حرف اول یک قلم لغوی است باید خود یک قلم لغوی منفرد باشد.

خط کج برعکس[3] «\» (U+005C) برای استفاده‌ی ویژه رزرو شده است. اگر به دنبال این حرف، به ترتیب حرف u (یا U) و یک کد چهار (یا شش) رقمی در مبنای شانزده آمده باشد آنگاه همانگونه که در بالا توضیح داده شد، از آن به منظور رونویسی نویسه‌های یونیکد غیرآسکی در یک جریان آسکی استفاده می‌شود. در یک رشته‌ی آسکی هر رشته‌ای به این شکل با یک حرف عادی، نقش نحوی منطق مشترک یکسانی دارد. ترکیب «\'» (U+005C, U+002C) به منظور کدگذاری یک علامت نقل قول منفرد درون یک متن نقل قول شده‌ی منطق مشترک استفاده می‌شود. به‌طور مشابه، ترکیب «\"» (U+005C, U+0022) یک علامت نقل قول دوتایی را درون یک رشته‌ی نامی که در بین دو علامت نقل قول دوتایی قرار گرفته باشد نشان می‌دهد. در هر دو مورد فوق، برای نشان دادن خود خط کج برعکس از دو خط کج برعکس «\\» (U+005C, U+005C) استفاده می‌شود. خط کج برعکس به هر شیوه‌ی دیگری که قرار گیرد خطا است. این قراردادها برای درون علامت‌های نقل قول در هر دو نمایش آسکی و یونیکد به کار می‌رود.

الف-۲-۲ نحو لغوي

1 - Separator
2 - Double-quote character
3 - Backslash

به منظور تقسیم ارائه به دو بخش، مابین ساختارهای لغوی و نحوی تمایز قائل می‌شویم. این زیربخش می‌تواند به اجراکنندگان این استاندارد در تشخیص نشانه‌های منطقی که عبارت‌های نحوی را به گونه‌ای می‌سازند که در زیربند الف-۲-۳ نشان داده شده، کمک کند. پیاده‌سازی‌ها لازم نیست این تمایز را پشتیبانی کنند.

الف-۲-۲-۱ فاصله[1]

whitechar = space U+0020 | tab U+0009 | line U+000A | page U+000C | return U+000D

white = whitechar | '/*' , {char - '*' | '*' , char - '/' | open | close | namequote | stringquote | backslash | whitechar }, ['*'] , '*/' | '//' {char | open | close | namequote | stringquote | backslash | space | tab }, (page | line | return) ;

بدین ترتیب می‌توان توضیحات موقت را در متن CLIF مطابق قواعد جاوا و C++ درج کرد. هر تجزیه‌گر CLIF با متن یک خطی بعد از // و متن چند خطی که با /* ... */ احاطه شده باشد به عنوان فاصله برخورد می‌کند.

دنباله‌های // و /* ... */ تنها زمانی موجب راه اندازی این ساخت[2] می‌شوند که خارج از رشته‌های نقل قول شده یا نام‌های محصور قرار گرفته باشند. بنابر این نام‌ها در متن CLIF که دارای دنباله‌های حرفی //، */ یا /* باشند باید به صورت نام‌های محصور نوشته شوند.

توضیحات موقت از توضیحات منطق مشترک که بخش‌های دایم متن CLIF تجزیه شده هستند متمایزند. توضیحات موقت از آنجا که فاصله محسوب می‌شوند به عنوان نویسه‌های تفکیک لغوی[3] عمل می‌کنند.

الف-۲-۲-۲ جداکننده‌ها

علامت نقل قول منفرد برای جداسازی رشته‌های نقل قول شده و علامت نقل قول دوتایی برای جداسازی نام‌های محصوری که از قواعد واژه‌سازی خاصی پیروی می‌کنند به‌کار می‌روند. رشته‌های نقل قول شده و نام‌های محصور تنها اقلام لغوی CLIF هستند که می‌توانند فاصله و پرانتز داشته باشند. پرانتزها در جاهای دیگر حکم خودجداکننده را دارند و به خودی خود نشانه‌های لغوی محسوب می‌شوند. پرانتزها ابزار اصلی گروه‌بندی در نحو CLIF هستند.

---

open = '(' ;

close = ')' ;

stringquote = '"' ;

namequote = '"' ;

---

1 - White space
2 - Production
3 - Lexical break characters

backslash = '\' ;

الف-۲-۲-۳ نویسه‌ها

char برای مشخص کردن باقی نویسه‌های غیرکنترلی اسکی که از آن‌ها می‌توان برای ساختن نشانه‌های لغوی استفاده کرد به‌کار می‌رود (با بعضی محدودیت‌ها بر روی حرف اول نشانه‌ی لغوی). این نویسه‌ها شامل تمام نویسه‌های الفباعددی هستند.

char = digit | '~' | '!' | '#' | '$' | '%' | '^' | '&' | '*' | '_' | '+' | '{' | '}' | '|' | ':' | '<' | '>' | '?' | '`' | '-' | '=' | '[' | ']' | ';'| ',' | '.' | '/' | 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z' | 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u' | 'v' | 'w' | 'x' | 'y' | 'z' ;

digit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' ;

hexa = digit | 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'a' | 'b' | 'c' | 'd' | 'e' | 'f' ;

الف-۲-۲-٤ نقل قول درون رشته‌ها

از دنباله‌های حرفی معینی برای نشان دادن وجود یک حرف منفرد استفاده می‌شود. *nonascii* مجموعه‌ی تمام نویسه‌های یا دنباله‌های حرفی است که یک حرف یونیکد در خارج از محدوده‌ی اَسکی را نشان می‌دهند.

یادآوری - از این ساخت نباید برای ورودی گرفتن با استفاده از کدگذاری کامل یونیکد استفاده کرد و به جای آن باید nonascii را به عنوان مجموعه‌ی تمام نویسه‌های یونیکد خارج از محدوده‌ی اَسکی در نظر گرفت. استفاده از دنباله‌های \u*nnnn* یا \U*nnnnnn* در متنی که با استفاده از یک مجموعه‌ی کامل نویسه‌های یونیکد کدگذاری شده شایسته نیست.

از *innerstringquote* برای نشان دادن وجود یک علامت نقل قول منفرد در یک رشته‌ی نقل قول شده استفاده می‌شود. یک رشته‌ی نقل قول شده می‌تواند دارای هر حرفی از جمله فاصله باشد. اما علامت نقل قول منفرد تنها می‌تواند به عنوان بخشی از *innerstringquote* یعنی بلافاصله پس از یک خط کج برعکس در یک رشته‌ی نقل قول شده ظاهر شود. وجود یک علامت نقل قول منفرد در جریان حرفی یک رشته‌ی نقل قول شده انتهای نشانه‌ی لغوی رشته‌ی نقل قول شده را مشخص می‌کند مگر اینکه بلافاصله پس از یک خط کج برعکس قرار گرفته باشد. با علامت نقل قول دوتایی در نام‌های محصور به‌طور دقیق به شیوه‌ای مشابه برخورد می‌شود. از *innernamequote* برای نشان دادن وجود یک علامت نقل قول دوتایی در یک نام محصور استفاده می‌شود.

nonascii = '\u' , hexa, hexa, hexa, hexa | '\U' , hexa, hexa, hexa, hexa, hexa, hexa ;

innerstringquote = '\'' ;

innernamequote = '\"' ;

innerbackslash = '\\'

numeral = digit , { digit } ;

نشان‌گرهای دنباله یک شکل نحوی متمایز با یک معنـای خـاص در منطـق مشـترک هسـتند. توجـه کنیـد انداختن لغات بدون هیچ متنی (یعنی "...") خود یک نشان‌گر دنباله است.

seqmark = '...' , { char } ;

علامت‌های نقل قول منفرد برای رشته‌های نقل قول شده و علامت‌های نقل قول دوتایی برای نام‌های محصور جداکننده هستند.

یک نام محصور، نامی است که ممکن است دارای نویسه‌هایی باشد که موجب تفکیک واژه‌سازی می‌شوند مانند «Mrs Norah Jone» یا «Girl(interrupted)» و همانند هر نام دیگری ممکن است هر معنایی داشته باشد. علامت‌های نقل قولی که نام گفتمانی را احاطه می‌کنند بخشی از آن به حساب نمی‌آیند. نام گفتمانی از حذف علامت‌های نقل قول و جایگزین کردن هر innernamequote با یک علامت نقل قول دوتایی بدست می‌آید. پیشنهاد می‌شود از نحو نام محصور در زمان نوشتن شناسه‌های منبع عمومی،ارجاعات آن‌ها و شناسه‌های منبع بین‌المللی شده به عنوان نام، استفاده شود چرا که این شناسه‌ها ممکن است دارای نویسه‌هایی باشند که در غیر این صورت موجب تفکیک واژه‌سازی CLIF می‌شوند. در حالت خاص، ارجاعات URI مطیع Xpath اغلب به یک پرانتز بسته ختم می‌شوند.

در مقابل، یک رشته‌ی نقل قول شده عبارتی است دارای یک مفهوم معناشناسی معین، بدین صورت که یک رشته‌ی متنی مشابه رشته‌ی مابین علامت‌های نقل قول را *مشخص می‌کند*.

الف-۲-۲-۵ رشته‌هاي نقل قول شده

رشته‌های نقل قول شده و نام‌های محصور، الگوریتم واژه‌سازی متفاوتی با سایر بخش‌های متن CLIF لازم دارند چرا که پرانتزها و جاهای خالی یک جریان رشته‌ی نقل قول شده را به نشانه‌های لغوی تفکیک نمی‌کنند.

زمانی که متن CLIF درون متن یا مدرکی قرار گرفته باشد که از قراردادهای گریز نویسه‌ای[1] استفاده می‌کند، قراردادهای منطق مشترک مربوط به رشته‌های نقل قول شده که در اینجا توصیف شده‌اند به متن توصیف شده با استفاده از قواعد گریز حرفی اعمال می‌شوند. بنابراین، برای نمونه، محتوای عنصر XML <cl-text>'a\'b<c'</cl-text> طبق نحو CLIF رشته‌ی نقل قول شده‌ی 'a\'b<c' به معنی رشته‌ی متنی پنج حرفی a'b<c است. اما اگر این عنصر XML را متن ساده‌ی CLIF در نظر بگیریم آنگاه 'a\'b<c&apos تنها یک نام طولانی خواهد بود.

quotedstring = stringquote, { white | open | close | char | nonascii | namequote |innerstringquote | innerbackslash }, stringquote ;

1 - Character escaping conventions

enclosedname = namequote, { white | open | close | char | nonascii | stringquote | innernamequote }, namequote ;

الف-۲-۲-٦ نشانه‌هاي رزرو شده

reservedelement متشکل از نشانه‌های لغوی است که به منظور نمایش ساختار نحوی عبارت‌های منطق مشترک استفاده می‌شوند. استفاده از این نشانه‌ها به عنوان نام در متن CLIF مجاز نیست.

reservedelement = '=' | 'and' | 'or' | 'iff' | 'if' | 'forall' | 'exists' | 'not' | 'roleset:' | ‘cl:text‘ | 'cl:imports' | 'cl:excludes' | 'cl:module' | 'cl:comment' ;

الف-۲-۲-۷ دنباله‌ي نويسه‌اي نام[1]

یک *namecharsequence* یک نشانه لغوی است که با هیچ کدام از نویسه‌های خاص شروع نمی‌شود. توجه داشته باشید یک namecharsequence مجاز نیست فاصله یا پرانتز داشته باشد. در ضمن مجاز نیست با علامت نقل قول شروع شود گرچه می‌تواند شامل آن باشد. اعداد و نشان‌گرهای دنباله، namecharsequence به شمار نمی‌روند.

namecharsequence = (char , { char | stringquote | namequote | backslash }) - (reservedelement | numeral | seqmark) ;

الف-۲-۲-۸ رده‌هاي لغوي

وظیفه‌ی یک تحلیلگر لغوی تجزیه‌ی جریان حرفی به رشته‌های متوالی و غیرهمپوشان و تحویل نشانه‌های لغوی پیدا شده در قالب جریانی از نشانه‌ها به مرحله‌ی بعد پردازش نحوی است. نشانه‌های لغوی به هشت رده‌ی متمایز تقسیم می‌شوند که عبارتند از: پرانتزهای باز و بسته، اعداد، رشته‌های نقل قول شده (که با ‘ شروع و ختم می‌شوند)، نشان‌گرهای دنباله (که با ... شروع می‌شوند)، نام‌های محصور (که با " شروع و ختم می‌شوند)، دنباله‌های نامی و عناصر رزرو شده.

lexbreak = open | close | white , { white } ;

nonlexbreak = numeral | quotedstring | seqmark | reservedelement | namecharsequence | enclosedname ;

lexicaltoken = open | close | nonlexbreak ;

charstream = { white } , { lexicaltoken, lexbreak } ;

الف-۲-۳ نحو عبارت[2]

این بخش از نحو به منظور اعمال بر دنباله‌ای از نشانه‌های لغوی منطق مشترک و نه یک جریان حرفی نوشته شده است.

1 - Name character sequence
2 - Expression syntax

الف-۲-۳-۱ دنباله‌ي اصطلاح

هم اصطلاحات و هم جمله‌های تجزیه‌ناپذیر از مفهوم دنباله‌ای از اصطلاحات که بردار آرگومان‌های یک تابع یا رابطه را نشان می‌دهد استفاده می‌کنند. از نشان‌گرهای دنباله برای نشان دادن زیردنباله‌ای از یک دنباله‌ی اصطلاح استفاده می‌شود. هر اصطلاح یک عنصر منفرد را نشان می‌دهد.

```
termseq = { term | seqmark } ;
```

الف-۲-۳-۲ نام

به هر نشانه‌ی لغوی معنی‌دار نام گفته می‌شود. مابین نام‌هایی که دارای معنی معینی هستند و نام‌هایی که معنی آن‌ها به‌وسیله یک تفسیر مشخص می‌شود تمایز قائل می‌شویم.

```
interpretedname = numeral | quotedstring ;

interpretablename = namecharsequence | enclosedname ;

name = interpretedname | interpretablename ;
```

الف-۲-۳-۳ اصطلاح

نام‌ها، اصطلاح محسوب می‌شوند. یک اصطلاح پیچیده از یک عملگر که خود یک اصطلاح است به همراه برداری از آرگومان‌ها تشکیل می‌شود. اصطلاحات مجازند یک توضیح[1] نمایش داده شده به صورت یک رشته-ی نقل قول شده داشته باشند. پوشش‌های توضیحی از نظر نحوی اصطلاحی را که در مورد آن توضیح می‌دهند دربرمی‌گیرند.

```
term = name | ( open, operator, termseq, close ) | ( open, 'cl:comment', quotedstring, term, close ) ;

operator = term ;
```

الف-۲-۳-٤ تساوي

تساوی‌ها به دلیل نقش معنایی ویژه و همچنین شیوه‌ی اداره‌ی ویژه‌ی آن‌ها به‌وسیله برنامه‌های متعدد به عنوان یک رده‌ی ویژه تمیز داده شده‌اند. علامت تساوی یک نام نیست.

```
equation = open, '=', term, term, close ;
```

الف-۲-۳-٥ جمله

1 - Comment

جمله‌ها مجازند همانند اصطلاحات، توضیحات دربرگیرنده[1] داشته باشند. توجه کنید که توضیحات مجازند به جمله‌هایی که زیرعبارت‌های جمله‌های بزرگتر هستند اعمال شوند.

```
sentence = atomsent | boolsent | quantsent | commentsent ;
```

الف-۲-۳-۶ جمله‌ي تجزیه‌ناپذیر

جمله‌های تجزیه‌ناپذیر از نظر ساختار مشابه اصطلاحات هستند، اما علاوه بر این، آرگومان‌های یک جمله‌ی تجزیه‌ناپذیر را می‌توان با استفاده از زوج‌های نقشی متشکل از یک نام نقش و یک اصطلاح نمایش داد. تساوی‌ها جمله‌های تجزیه‌ناپذیر محسوب می‌شوند و یک جمله‌ی تجزیه‌ناپذیر را می‌توان با استفاده از زوج-های نقشی متشکل از یک نام نقش و یک اصطلاح نمایش داد.

```
atomsent = equation | atom ;

atom = ( open, predicate , termseq, close ) | ( open, term, open, 'roleset:' , { open, name, term, close }, close, close ) ;

predicate = term ;
```

الف-۲-۳-۷ جمله‌ي بولي

در جمله‌های بولی لازم است دلالت‌های ضمنی و دوشرطی‌ها دوتایی باشند اما ترکیبات عطفی و فصلی می‌توانند هر تعداد آرگومان (حتی صفر) داشته باشند. از جمله‌های (and) و (or) می‌توان به ترتیب به عنوان ارزش‌های درستی true و false استفاده کرد.

```
boolsent = ( open, ('and' | 'or') , { sentence }, close ) | ( open, ('if' | 'iff') , sentence , sentence, close ) | ( open, 'not' , sentence, close ;
```

الف-۲-۳-۸ جمله‌ي سوري

سورها مجازند به هر تعداد متغیر مقید شوند و همچنین مجازند محافظ داشته باشند. متغیرهای مقید مجازند محدود به رده‌ای باشند که به‌وسیله یک اصطلاح مشخص شده است.

```
quantsent = open, ('forall' | 'exists') , [ interpretablename ] , boundlist, sentence, close ;

boundlist = open, { interpretablename | seqmark | ( open, (interpretablename | seqmark), term, close )} , close ;
```

الف-۲-۳-۹ جمله‌ي توضیح‌دار

1 - Enclosing comments

یک توضیح مجاز است به هر جمله‌ای اعمال شود. بنابراین توضیحات مجازند به جمله‌هایی که خود زیرعبارت‌های جمله‌های بزرگتر هستند نیز مقید شوند.

```
commentsent = open, 'cl:comment', quotedstring , sentence , close ;
```

الف-۲-۳-۱۰ پیمانه

پیمانه‌ها قطعات متن دارای نام هستند که متنی را که باید در یک زمینه‌ی «محلی» قابل فهم باشد را نمایش می‌دهند. نام یک پیمانه حوزه‌ی تعریف سورها در متن را نشان می‌دهد. این نام نباید یک عدد یا رشته‌ی نقل قول شده باشد. یک پیمانه مجاز است به‌صورت اختیاری یک لیست عدم شمول از نام‌هایی که مفاهیم آن‌ها خارج از حوزه است را داشته باشد. توجه کنید متن و پیمانه رده‌های بازگشتی هستند. بنابراین پیمانه‌ها مجازند تو در تو باشند.

```
module = open, 'cl:module' , interpretablename , [open, 'cl:excludes' , {name} , close ] , cltext, close;
```

یک پیمانه که لیست عدم شمول نداشته باشد با یک متن دارای نام یکسان نیست.

الف-۲-۳-۱۱ عبارت[1]

متن دنباله‌ای از عبارت‌ها است که هر کدام از آن‌ها یک جمله، یک پیمانه، یک وارده یا یک متن ساده‌ی توضیح‌دار است. متن توضیح‌دار مجاز است تهی بوده و یا یک جمله‌ی منفرد باشد. تخصیص نام به یک متن به شیوه‌ای یکسان با یک پیمانه صورت می‌گیرد. اما در این صورت از آن نام تنها برای شناسایی آن متن استفاده می‌شود و محدود به نام جهان گفتمان نیست. با یک پیمانه‌ی منفرد نیز می‌توان همانند یک متن برخورد کرد. هر نام تخصیصی به یک متن دارای نام یا یک پیمانه بوده و هر نام که درون یک وارده واقع شود باید یک شناسه‌ی شبکه باشد. در زمان نوشتن این استاندارد، برای برنامه‌های تحت وب این نام باید یک IRI [2] باشد. برنامه‌های ویژه‌ای مجازند شروط اضافی را بر نام‌هایی که به عنوان شناسه استفاده می‌شوند تحمیل کنند. تنها حرف غیرپایانه برای این نحو <code>cltext</code> است.

```
phrase = sentence | module | (open, 'cl:imports' , interpretablename , close) | (open, 'cl:comment', quotedstring, cltext, close);

cltext = { phrase } ;

namedtext = open, 'cl:text' interpretablename, text, close ;

cltext = module | namedtext | text ;
```

1 - Phrase

الف-۳ معناشناسي CLIF

ما از بعضی مفاهیم و نشانه‌گذاری‌های تعریف شده در بند ۶-۲، به خصوص نشانه‌گذاری <..>;<..> استفاده خواهیم کرد.

فرض کنید IN مجموعه‌ی تمام نام‌های تفسیر شده‌ی CLIF (که همان اعداد دهدهی و رشته‌های نقل قول شده هستند) و N مجموعه‌ی اعداد طبیعی و تمام رشته‌های متناهی از نویسه‌های یونیکد است. $V = VN \cup VS$ یک واژه‌نامه‌ی CLIF است که به صورت اجتماع گسسته‌ی یک مجموعه‌ی VN از نام‌های قابل تفسیر و یک مجموعه‌ی VS از نشان‌گرهای دنباله تعریف می‌شود.

در CLIF یک تفسیر I از یک واژه‌نامه‌ی V ساختاری متشکل از یک مجموعه‌ی U_I تحت عنوان *جهان* که ابرمجموعه‌ای از N است و دو نگاشت rel_I از U_I به زیرمجموعه‌های U_I^*، fun_I از U_I به توابعی از U_I^* به U_I و یک نگاشت int_I بر روی V از VN به U_I و از VS به U_I^* است. همانند بند ۶-۲ برای هر زیرمجموعه‌ی S از V یک تفسیر J از V یک نسخه‌ی مختصّ S از I است اگر J درست مشابه I باشد به استثنای اینکه int_I و int_J ممکن است از نظر مقادیری که به اعضای S تخصیص می‌دهند متفاوت باشند.

یادآوری - CLIF بین جهان گفتمان و جهان مرجع تمایزی قائل نیست چرا که تمام نام‌ها ارجاع دارند.

تفسیر هر عبارت CLIF با استفاده از درایه‌های جدول الف-۱ تعیین می‌شود. نشانه‌گذاری $<T_1,\ldots,T_n>$ زمان رجوع به نحو یک دنباله‌ی اصطلاح را نشان می‌دهد و زمان رجوع به معناشناسی تعریف‌کننده یک دنباله یا به عبارتی یک عنصر از U_I^* است.

ستون اول پیوندهایی را به سطرهای معناشناسی منطق مشترک که در جدول ۱ در بند ۶-۲ آمده نشان می‌دهد.

جدول الف-۱ معناشناسی CLIF

	اگر E عبارتی بدین شکل باشد:	**آنگاه I(E) برابر خواهد بود با:**
E1	یک عدد دهدهی	عدد طبیعی مشخص شده به‌وسیله آن عدد دهدهی
E1	یک رشته‌ی نقل قول شده به شکل **'s'**	رشته نویسه یونیکد که با حذف علامت‌های نقل قول دربرگیرنده‌ی آن و جایگزینی زیررشته‌های مستقل درون آن با معادل یونیکد آن‌ها، بدست می‌آید
E1,E2	یک نام قابل تفسیر	int_I(**E**)
E3	دنباله‌ی اصطلاح <**T**$_1$,…,**T**$_n$> که با اصطلاح **T**$_1$ شروع می‌شود	<I(**T**$_1$)>;I(<**T**$_2$,…,**T**$_n$>)
E4	دنباله‌ی اصطلاح <**T**$_1$,…,**T**$_n$> که با نشان‌گر دنباله‌ی **T**$_1$ شروع می‌شود	I(**T**$_1$);I(<**T**$_2$,…,**T**$_n$>)
E5	یک اصطلاح (**O T**$_1$,…,**T**$_n$)	fun_I(I(O))(I(<**T**$_2$,…,**T**$_n$>)
	یک اصطلاح (cl:comment 'string' **T**)	I(**T**)
E6	یک تساوی (= **T**$_1$**T**$_2$)	اگر I(**T**$_1$) = I(**T**$_2$) آنگاه true و در غیر این صورت false
E7	یک جمله‌ی تجزیه‌ناپذیر (**P T**$_1$,…,**T**$_n$)	اگر I(<**T**$_2$,…,**T**$_n$>) در rel_I(I(**P**)) باشد آنگاه true و در غیر این صورت false
E8	یک جمله‌ی بولی (not **P**)	اگر I(**P**) = false آنگاه true و در غیر این صورت false
E9	یک جمله‌ی بولی (and **P**$_1$,…,**P**$_n$)	اگر I(**P**$_1$) = … I(**P**$_n$) = true آنگاه true و در غیر این صورت false
E10	یک جمله‌ی بولی (or **P**$_1$,…,**P**$_n$)	اگر I(**P**$_1$) = … I(**P**$_n$) = false آنگاه false و در غیر این صورت true
E11	یک جمله‌ی بولی (if **P Q**)	اگر I(**P**) = true و I(**Q**) = false آنگاه false و در غیر این صورت true
E12	یک جمله‌ی بولی (iff **P Q**)	اگر I(**P**) = I(**Q**) آنگاه true و در غیر این صورت false
	یک جمله‌ی (cl:comment 'string' **P**)	I(**P**)
E13	یک جمله‌ی سوری (forall (**N**$_1$ … **N**$_n$) **B**) **N** = { **N**$_1$ … **N**$_n$ } که در آن **N**$_n$ مجموعه‌ی انقیادهای جمله است	اگر برای هر تفسیر J که نسخه‌ای مختصّ N از I است داشته باشیم J(**B**) = true آنگاه true و در غیر این صورت false
E14	یک جمله‌ی سوری (exists (**N**$_1$ … **N**$_n$) **B**) **N** = { **N**$_1$ … **N**$_n$ } که در آن **N**$_n$ مجموعه‌ی انقیادهای جمله است	اگر برای هر تفسیر J که نسخه‌ای مختصّ N از I است داشته باشیم J(**B**) = true آنگاه true و در غیر این صورت false
	یک عبارت (cl:comment 'string')	true
E17	یک عبارت (cl:imports **N**)	اگر I(text(I(**N**))) = true آنگاه true و در غیر این صورت false
E19	یک عبارت (cl:text **T**$_1$ … **T**$_n$)	اگر I(**T**$_1$) = … = I(**T**$_n$) = true آنگاه true و در غیر این صورت false
E20	(cl:text **T**$_1$ … **T**$_n$)	اگر یک مقدار متنی دارای نام **t** در U وجود داشته باشد که text(**t**) = I(**N**) = **t** و name(**t**) = **N**، (cl:text **T**$_1$ … **T**$_n$) آنگاه true و در غیر این صورت false

تمام شکل‌های نحوی CLIF در این جدول پوشش داده نشده‌اند. تفسیر باقی موارد نحوی با استفاده از نگاشت آن‌ها به دیگر عبارت‌های CLIF تعریف می‌شود که تفسیر آن‌ها در جدول فوق تعریف شده است. این ترجمه در جدول الف-۲ تعریف شده است.

جدول الف-۲ نگاشت شکل‌های اضافی CLIF به شکل‌های هسته‌ی آن

اگر E به صورت زیر باشد:	**آنگاه E به T[E] ترجمه می‌شود:**
یک جمله‌ی تجزیه‌ناپذیر به صورت زیر: (**T**$_0$ (roleset: (**N**$_1$ **T**$_1$) … (**N**$_n$ **T**$_n$)))	جمله‌ی (exists (**X**)(and (**T**$_0$ **X**)(**N**$_1$ **X T**$_1$)…(**N**$_n$ **X T**$_n$))) که در آن **X** نام جدیدی است که نه در جمله‌ی تجزیه‌ناپذیر مورد نظر و نه هیچ جمله‌ای که شامل آن باشد واقع نمی‌شود
یک جمله‌ی سوری به صورت زیر: (forall (**N**$_1$ … **N**$_n$) **B**)	جمله‌ی سوری زیر: (forall (**N**$_1$) **T[**(forall (…)(if (**T**$_1$ **N**$_1$) **B**)**]**
یک جمله‌ی سوری به صورت زیر: (exists (**N**$_1$ … **N**$_n$) **B**)	جمله‌ی سوری زیر: (exists (**N**$_1$) **T[**(exists (…)(and (**T**$_1$ **N**$_1$) **B**)**]**
یک جمله‌ی سوری به صورت زیر: (forall **G** (…) **B**)	جمله‌ی سوری **[**(forall (…)(if (G X$_1$ … **X**$_n$) **B**)**]T** که در آن X$_1$ … **X**$_n$ تمام نام‌های مستقل در **B** هستند
یک جمله‌ی سوری به صورت زیر: (exists **G** (…) **B**)	جمله‌ی سوری **[**(exists (…)(and (G X$_1$ … **X**$_n$) **B**)**]T** که در آن X$_1$ … **X**$_n$ تمام نام‌های مستقل در **B** هستند
یک پیمانه به صورت زیر: (cl:module **N** (cl:excludes **N**$_1$ … **N**$_n$) **T**)	متن **T[T']** (not (**N N**$_1$)) … (not (**N N**$_n$)) که در آن **T'** متن **T** است که در آن هر نام یا نشان‌گر دنباله‌ی **X** در لیست انقیاد یک سور با (**X N**) جایگزین شده است

شکل‌های نحوی سمت راست جدول الف-۲ را می‌توان «شیرین‌کاری نحوی» برای ترجمه‌ی آن‌ها در سمت چپ در نظر گرفت. به این ترجمه‌ها معادل نحوی *ترش*[1] شکل‌های نحوی سمت راست گفته می‌شود. به زیرگویش CLIF که فاقد این عبارت‌ها، باشد *CLIF ترش* گفته می‌شود.

الف-٤ انطباق CLIF

انطباق CLIF بر منطق مشترک از حیث نحوی و معناشناسی نشان داده می‌شود. این بخش نه تنها انطباق CLIF را تصریح می‌کند بلکه تصریح کنندگان سایر گویش‌ها را راهنمایی می‌کند که چگونه می‌توان انطباق را نشان داد.

الف-٤-١ انطباق نحوي

تناظر نحو CLIF با نحو انتزاعی منطق مشترک با استفاده از درایه‌های ستون سمت راست جدول الف-۱ که به درایه‌های جدول ۱ ارجاع می‌دهند نشان داده شده و با بازبینی آن‌ها می‌توان انطباق کامل نحوی CLIF ترش را بدست آورد. توجه کنید هر دو رده‌ی interpretednames و interpretablenames در منطق مشترک

1 - Sour

نام محسوب می‌شوند. انطباق نحوی CLIF بر اساس نگاشت تعریف شده در جدول الف-۲ ادامه می‌یابد. توجه کنید در نحو توضیحات CLIF یک عبارت توضیح‌دار با یک عبارت بدون توضیح معنی یکسانی دارد. بنابراین توضیح را می‌توان پیوست عبارت بدون توضیح در نظر گرفت.

نحو CLIF به موجب محدودیت‌هایی که بر مکان مجاز قرار گرفتن نام‌های تفسیر شده در عبارت‌ها، تحمیل می‌کند، تفکیک‌شده است. اما آن را از نقطه نظر مباحث بند ۷-۱ نمی‌توان گویش تفکیک‌شده به حساب آورد.

الف-٤-٢ انطباق معنايي

CLIF یک گسترش معنایی منطق مشترک است. برای نشان دادن این موضوع باید نشان دهیم اگر I یک تفسیر CLIF باشد آنگاه یک تفسیر منطق مشترک J باید وجود داشته باشد که ارزش درستی یکسانی به هر جمله می‌دهد. برای نشان دادن این موضوع، J را از روی I بدین صورت می‌سازیم که در زمان توصیف I از نشانه‌گذاری و قراردادهای فوق و در زمان توصیف J از نشانه‌گذاری و قراردادهای بند ۶-۲ استفاده می‌کنیم.

واژه‌نامه‌ی J با واژه‌نامه‌ی I یکسان است، همچنین $UD_J = UR_J = U_I$، $rel_J = rel_I$ و $fun_J = fun_I$. تفسیر نام‌های قابل تفسیر برای هر نام قابل تفسیر x به روش بدیهی $int_J(x) = int_I(x)$ صورت می‌گیرد. از آنجا که نام‌های تفسیر شده در واژه‌نامه‌ی CLIF به عنوان نام‌هایی در منطق مشترک رده‌بندی می‌شوند، ما باید $int_J(x)$ را برای حالتی که x یک نام تفسیر شده است نیز تعریف کنیم. واضح است که این کار به منظور پیروی از دو درایه‌ی اول جدول معناشناسی CLIF انجام می‌شود. طبق این درایه‌ها اگر x یک عدد دهدهی باشد آنگاه $int_I(x)$ برابر است با عدد صحیح مشخص شده به‌وسیله x و اگر x یک رشته‌ی نقل قول شده باشد آنگاه $int_I(x)$ برابر با رشته‌ی نویسه‌های یونیکدی است که به‌وسیله x مشخص شده است. این موضوع سپس با مقایسه‌ی مواردی که برای هر جمله‌ی s متعلق به CLIF داریم J(s) = I(s) به راحتی قابل مشاهده خواهد بود. اگر s پیمانه‌ای با نام N، لیست عدم شمول L و بدنه‌ی B باشد آنگاه باید نشان دهیم J(s) = true تنها زمانی که داشته باشیم [J<L](B) = true و $rel(J(N)) = UR_{[J<L]}^*$ (چرا که $UD_J = UR_J$). همانطور که در بند ۶-۲ توصیف شده، به راحتی قابل مشاهده است که این معادل درستی[1] در تفسیر I جمله‌های موجود در ترجمه‌ی ترش متن بدنه‌ی پیمانه که در جدول الف-۲ تعریف شده خواهد بود. (اثبات صوری این موضوع با استقرای ساختاری بر روی جمله‌های متن بدنه انجام می‌شود) بنابراین برای هر متن t در CLIF داریم J(t) := I(t).

این طور نیست که برای هر I به عنوان یک تفسیر منطق مشترک از متن t درCLIF باید یک تفسیر J در CLIF وجود داشته باشد که مقدار یکسانی را برای t مشخص کند. از آنجا که در منطق مشترک با نام‌های تفسیر شده در CLIF به صورت نام‌ها برخورد می‌شود، J مجاز است مقداری را به آن‌ها تخصیص دهد که منطبق بر تفسیر ثابت آن‌ها در CLIF نیست. برای نمونه، J('a string') = 3 مبتنی بر قواعد معناشناسی منطق مشترک ممنوع نیست. این یک پدیده‌ی عمومی برای هر گویشی است که معانی از پیش تعیین شده و تعریف شده به صورت خارجی را بر یک رده از نام‌ها مانند اعداد و عبارت‌های دارای نوع داده‌ای[2] تحمیل

1 - Truth
2 - Data typed expressions

می‌کند. چنین گویش‌هایی مجازند از استنتاج‌هایی پشتیبانی کنند که قابل بیان به صورت اصول بدیهی منطق مشترک نبوده و باید به عنوان گسترش‌های معنایی خارجی منطق مشترک رده‌بندی شوند. زیرگویش CLIF که از اعداد و رشته‌های نقل قول شده استفاده نمی‌کند به‌طور دقیق از نظر معنایی منطبق بر منطق مشترک است. این موضوع را با تولید ساختار J از I به شیوه‌ای عکس روشی که در بالا توصیف شد می توان نشان داد.

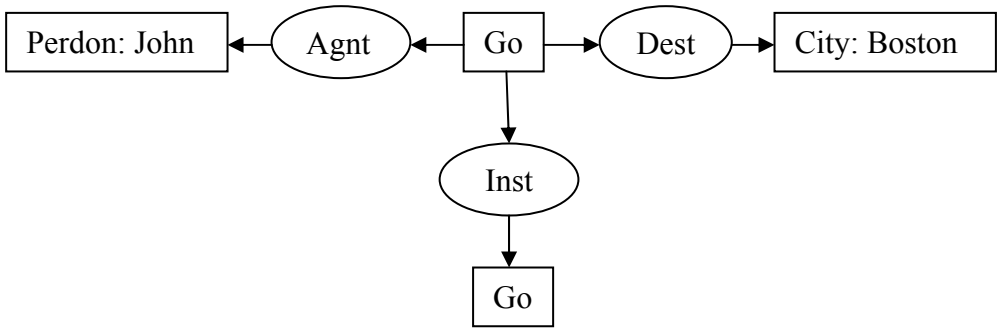
پیوست ب
قالب تبادل گراف مفهومي (CGIF)[1]
(الزامي)

ب-۱ مقدمه

این زیربند گراف‌های مفهومی را به‌طور مختصر شرح داده و سپس مجموعه‌ای از قواعد تبدیل (بازنویسی) را توصیف می‌کند که در ادامه‌ی این پیوست به منظور توصیف قواعد نحوی برای CGIF استفاده می‌شوند.

ب-۱-۱ گراف‌هاي مفهومي

یک گراف مفهومی نمایشی برای منطق در قالب یک گراف دو بخشی با دو نوع گره‌ی *مفهوم* و *رابطه‌ی مفهومی* است. *قالب تبادل گراف مفهومی* (CGIF) یک گویش منطبق بر منطق مشترک است که حکم یک نمایش سریال برای گراف‌های مفهومی را دارد. این پیوست نحو CGIF و نگاشت آن به معناشناسی منطق مشترک را مشخص می‌کند. در این استاندارد از یک نشانه‌گذاری گرافیکی غیراصولی تحت عنوان *شکل نمایش CG* تنها در مثال‌هایی که ساختارهای CG را نشان می‌دهند استفاده شده است. مثال اول (شکل ب-۱) شکل نمایش جمله‌ی *John goes to Boston by bus* را نشان می‌دهد.



شکل ب-۱ شکل نمایش CG برای جمله‌ی John goes to Boston by bus

در این شکل نمایش مستطیل‌ها یا جعبه‌ها نشان‌دهنده‌ی مفاهیم و بیضی‌ها یا دایره‌ها نشان‌دهنده‌ی روابط مفهومی است. کمان نوک تیزی که به سمت یک دایره اشاره می‌کند اولین *آرگومان* رابطه و کمانی که از سمت یک دایره به دور اشاره می‌کند آخرین آرگومان را هدف می‌گیرد. اگر رابطه‌ای تنها یک آرگومان داشته باشد نوک تیز کمان حذف می‌شود و اگر رابطه‌ای بیش از دو آرگومان داشته باشد نوک‌های تیز کمان‌ها با اعداد صحیح $1,\ldots,n$ جایگزین می‌شوند.

گراف مفهومی (CG) در شکل ب-۱ دارای چهار مفهوم است که هر کدام یک *برچسب نوع* دارد که نوع هستاری که آن مفهوم به آن اشاره می‌کند را نشان می‌دهد (در شکل فوق: Person، Go، Boston و Bus).

1 - Conceptual Graph Interchange Format

دو مورد از مفاهیم دارای *ثابت‌هایی* هستند که افراد (یعنی John و Boston) را مشخص می‌کنند. هر کدام از سه رابطه‌ی مفهومی دارای یک برچسب نوع است که نوع رابطه را نشان می‌دهد. این برچسب‌ها عبارتند از Agnt به عنوان عامل، Inst: به عنوان وسیله و Dest به عنوان مقصد رفتن. گراف مفهومی فوق به‌طور کلی نشان می‌دهد که شخص John عامل نمونه‌ای از رفتن (going) است که Boston مقصد و Bus وسیله‌ی آن است. نمایش CGIF این شکل نمایش به صورت زیر است:

[Go: *x] [Person: John] [City: Boston] [Bus: *y]
(Agnt ?x John) (Dest ?x Boston) (Inst ?x ?y)

در CGIF مفاهیم با استفاده از کروشه و روابط مفهومی با استفاده از پرانتز نشان داده می‌شوند. یک رشته‌ی حرفی که پیشوند آن یک ستاره باشد (مانند x*) یک *برچسب تعریف‌کننده* است. این برچسب ممکن است به‌وسیله *برچسب مقید* x? که پیشوند آن یک علامت سؤال است مورد ارجاع قرار گیرد. این رشته‌ها که در CGIF به آن‌ها *برچسب‌های هم‌مرجع*[1] گفته می‌شود متناظر با متغیرها در CLIF هستند. اگر پیشوند یک برچسب تعریف‌کننده every@ نباشد به یک سور وجودی ترجمه می‌شود. نمایش CLIF معادل با شکل ب-۱ به این صورت است:

(exists ((x Go) (y Bus))
(and (Person John) (city Boston)
(Agnt x John) (Dest x Boston) (Inst x y)))

همانگونه که این مثال نشان می‌دهد تفاوت بین CLIF و CGIF از ساختار گراف نتیجه می‌شود. گره‌های گراف هیچ ترتیب ضمنی ندارند و برچسب‌های هم‌مرجع مانند x* و x? ارتباط بین گره‌ها را نشان می‌دهند نه متغیرها. توجه کنید CGIF از پیشوندهای * و ? به منظور تمایز برچسب‌های هم‌مرجع از ثابت‌ها استفاده می‌کند، اما CLIF از هیچ قرارداد نحوی به منظور تمایز بین متغیرها و ثابت‌ها استفاده نمی‌کند.

شکل ب-۱ و نمایش آن در CGIF *نحو گسترش‌یافته*[2] CGIF را نشان می‌دهند که برچسب‌های نوع برروی مفاهیم و تعدادی گسترش نحوی دیگر را به *نحو هسته*[3] اضافه می‌کند. به منظور تبدیل گسترش‌های شکل ب-۱ به CGIF اصلی، برچسب‌های نوع در گره‌های مفهوم با روابط پیوسته به گره‌ها جایگزین می‌شوند. برای نمونه، مفهوم [Go: *x] به مفهوم بدون نوع [*x] و رابطه‌ی مفهومی (Go ?x) تبدیل می‌شود. مفهوم [Person: John] نیز به (Person John) [:John] تبدیل می‌شود که می‌توان آن را به رابطه‌ی (Person John) ساده کرد. CGIF اصلی برای مثال فوق و معادل CLIF آن به‌صورت زیر است:

[*x] [*y]
(Go ?x) (Person John) (City Boston) (Bus ?y)
(Agnt ?x John) (Dest ?x Boston) (Inst ?x ?y)

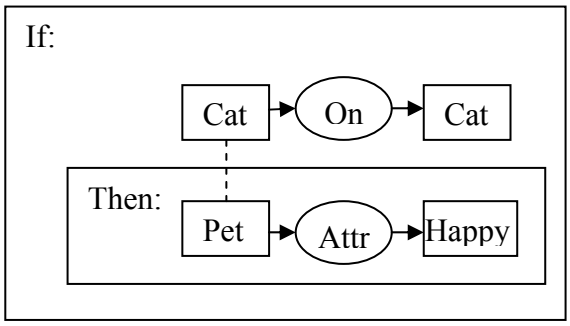
CLIF:

(exists (x y)
(and (Go x) (Person John) (City Boston) (Bus y)

1 - Coreference labels
2 - Extended syntax
3 - Core syntax

(Agnt x John) (Dest x Boston) (Inst x y)))

به منظور نشان دادن *زمینه‌ها*[1] و عملگرهای منطقی شکل ب-۲ شکل نمایش جمله‌ی *If a cat is on a mat then it is a happy pet* را نشان می‌دهد. همانند شکل ب-۱، مستطیل‌ها گره‌های مفهوم را نشان می‌دهند. اما دو مستطیل بزرگ حاوی گراف‌های مفهومی تو در تو هستند. به هر مفهومی که دارای یک CG تو در تو باشد یک *زمینه* گفته می‌شود. در این مثال برچسب‌های نوع If و Then نشان می‌دهند، گزاره‌ای که به‌وسیله CG در زمینه‌ی If بیان شده دلالت بر گزاره‌ای دارد که به‌وسیله CG در زمینه‌ی Then بیان شده است. رابطه‌ی Attr نشان می‌دهد که cat (که به آن pet نیز گفته می‌شود) دارای صفتی است که نمونه‌ای از شادی (happiness) است.



شکل ب-۲ شکل نمایش CG برای جمله‌ی "If a cat is on a mat then it is a happy pet"

خط چینی که مفاهیم [Cat] و [Pet] را به هم متصل می‌کند یک *پیوند هم‌مرجع*[2] است و نشان دهنده‌ی این است که هر دو به یک هستار اشاره می‌کنند. این ارتباط در CGIF با استفاده از برچسب تعریف‌کننده x* در مفهوم [Cat: *x] و برچسب مقید x? در مفهوم [Pet: ?x] نشان داده می‌شود:

[If: [Cat: *x] [Mat: *y] (On ?x ?y)
[Then: [Pet: ?x] [Happy: *z] (Attr ?x ?z)]]

در CGIF اصلی برچسب‌های نوع If و Then با نماد نفی ~ قبل از کروشه باز و سایر برچسب‌های نوع با روابط تکین[3] به صورت زیر جایگزین می‌شوند:

~[[*x] [*y] (Cat ?x) (Mat ?y) (On ?x ?y)
~[[*z] (Pet ?x) (Happy ?z) (Attr ?x ?z)]]

CLIF:

(not (exists (x y) (and (Cat x) (Mat y) (On x y)
(not (exists (z) (and (Pet x) (Happy z) (Attr x z)))))))

1 - Contexts
2 - Coreference link
3 - Monadic relations

در CGIF اصلی تنها سور وجودی وجود دارد. در CGIF گسترش‌یافته از سور عمومی می‌توان به منظور نشان دادن جمله‌ی *For every cat and every mat, if the cat is on the mat, then it is a happy pet* که معادل منطقی جمله‌ی فوق است استفاده کرد. در CGIF گسترش‌یافته سور عمومی به صورت @every نشان داده می‌شود:

[Cat: @every *x] [Mat: @every *y]
[If: (On ?x ?y) [Then: [Pet: ?x] [Happy: *z] (Attr ?x ?z)]]

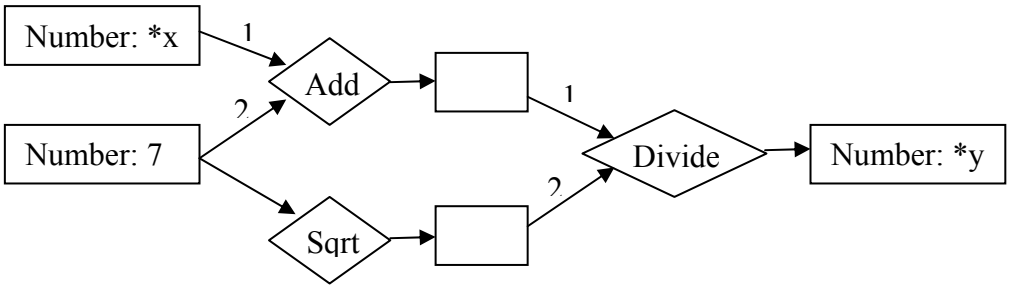
CLIF:

(forall ((x Cat) (y Mat))
(if (On x y) (and (Pet x) (exists ((z Happy)) (Attr x z)))))

در گراف‌های مفهومی توابع با استفاده از روابط مفهومی تحت عنوان *کنش‌گر*[1] نشان داده می‌شوند. شکل ب-۳ شکل نمایش CG برای تساوی زیر است که با استفاده از نشانه‌گذاری متداول جبری نوشته شده است.

y = (x + 7)/sqrt(7)

سه تابع موجود در این تساوی با استفاده از سه کنش‌گر نشان داده می‌شوند که در شکل ب-۳ با لورزی‌هایی با برچسب‌های نوع Add، Sqrt و Divide مشخص شده‌اند. جعبه‌ها گره‌های مفهوم[2] حاوی مقادیر ورودی و خروجی کنش‌گرها را نمایش می‌دهند. دو جعبه‌ی خالی به مقادیر خروجی کنش‌گرهای Add و Sqrt اختصاص دارند.



شکل ب-۳ توابع CL نمایش داده شده با استفاده از گره‌های کنش‌گر

در CGIF کنش‌گرها با استفاده از روابطی با دو نوع کمان نشان داده می‌شوند: کمان‌های دنباله‌ی ورودی و کمان‌های دنباله‌ی خروجی که با استفاده از یک خط عمودی به صورتی که در زیر نشان داده شده از هم جدا شده‌اند.

[Number: *x] [Number: *y] [Number: 7]
(Add ?x 7 | [*u]) (Sqrt 7 | [*v]) (Divide ?u ?v | ?y)

1 - Actor
2 - Concept nodes

در شکل نمایش (شکل ب-۳) کمان‌های ورودی کنش‌گرهای Add و Divide با استفاده از اعداد 1 و 2 شماره‌گذاری شده‌اند تا بدین ترتیب ترتیب نوشته شدن کمان‌ها در CGIF را نشان دهند. عبارت CLIF متناظر به صورت زیر است:

```
(exists ((x Number) (y Number))
(and (Number 7) (= y (Divide (Add x 7) (Sqrt 7)))))
```

در CLIF استفاده از هیچ متغیری برای نشان دادن برچسب‌های هم‌مرجع u* و v* لازم نیست چرا که نشانه‌گذاری تابعی مورد استفاده در CLIF ارتباطات را به‌صورت مستقیم نشان می‌دهد.

در CGIF تمام قابلیت‌های معنایی CL از جمله قابلیت استفاده از سور برروی کلاس‌ها و روابط را پشتیبانی می‌کند. برای نمونه، شخصی ممکن است بگوید: «Bob and Sue are related»، اما مشخص نکند این ارتباط چگونه است. جمله‌های زیر به ترتیب در CGIF و CLIF وجود یک ارتباط خانوادگی *r* که Bob و Sue را به هم ربط می‌دهد را بیان می‌کنند:

```
[Relation: *r] (Familial ?r) (#?r Bob Sue)
(exists ((r Relation)) (and (Familial r) (r Bob Sue)))
```

مفهوم [Relation: *r] بیان می‌کند که یک رابطه‌ی r وجود دارد. دو مفهوم بعدی بیان می‌کنند که این رابطه خانوادگی است و Bob و Sue را به هم ربط می‌دهد. در CGIF، پیشوند # یک برچسب هم‌مرجع مقید را نشان می‌دهد که به عنوان یک برچسب نوع استفاده شده است.

ب-۱-۲ قواعد نحوي EBNF براي CGIF (اطلاعاتي)

به منظور توصیف نحو CGIF نشانه‌گذاری EBNF (مرجع: ISO/IEC 14977:1996) استفاده شده است. در این پیوست تنها از زیرمجموعه‌ی زیر از ویژگی‌هایی که در ISO/IEC 14977:1996 آمده استفاده شده است. این بند تنها یک بخش آموزنده است چرا که ISO/IEC 14977:1996 باید به عنوان مرجع اصلی در نظر گرفته شود.

نماد پایانه[1]: هر رشته‌ای که در میان علامت‌های نقل قول منفرد یا دوتایی قرار گرفته باشد. برای نمونه:

```
"This is a quoted string." 'and so is this'
```

نماد غیرپایانه: نام یک رده در یک قاعده نحوی. برای نمونه، قاعده نحوی زیر دارای دو نماد غیرپایانه، یک نماد پایانه "';'" ، یک نماد تعریف "="، یک نماد الحاق "," و یک نماد ختم کننده ";" است.

```
syntaxRule = expression, ";";
```

گزینه: هر عبارتی که درون کروشه قرار گرفته باشد. هر رشته‌ای که به‌وسیله این عبارت مشخص شود صفر یا یک بار ظاهر خواهد شد. مثال:

```
["This string may or may not occur."]
```

تکرار: هر عبارتی که درون آکولاد قرار گرفته باشد. هر رشته‌ای که به‌وسیله این عبارت مشخص شود صفر یا چند بار ظاهر خواهد شد. برای نمونه:

```
{"This string may occur many times"}
```

الحاق: دو یا چند اصطلاح که با کاما جدا شده باشند.

---

1 - Terminal symbol

"Two kinds of quotes: ", """, " and ", "", "."

استثناء: دو اصطلاح که با علامت منها جدا شده باشند و مبین هر رشته‌ای است که با استفاده از اصطلاح اول و نه اصطلاح دوم مشخص شود. مثال زیر دنباله‌ای از صفر یا چند رقم را نشان می‌دهد که شامل ۶ نباشد.

{digit} – 6

گروه: هر عبارت درون پرانتز که به عنوان یک اصطلاح منفرد با آن برخورد شود. گروه زیر عبارتی را دربرمی‌گیرد که دنباله‌ای از یک یا چند رقم را به جز اصطلاح تهی مشخص می‌کند.

({digit} -)

بدل‌ها: دو یا چند الحاق که با خطوط عمودی جدا شده باشند. برای نمونه:

"cat", "dog" | "cow", "horse", "sheep" | wildAnimal

دنباله‌ی ویژه: هر رشته‌ای که بین دو علامت سؤال قرار گرفته باشد. این دنباله‌ها نباید بر نحوی که به‌وسیله قواعد نحوی مشخص شده تأثیر بگذارند، اما استفاده از آن‌ها به منظور کپی برداری از رشته‌های تحلیل شده به‌وسیله یک قاعده نحوی برای استفاده‌ی مجدد طبق قواعد بازنویسی مشخص شده در بند ب-۶-۲ مجاز است. برای نمونه:

?sqn?

قاعده نحوی: یک نماد غیرپایانه که به دنبال آن یک "=" و یک عبارت منتهی به «";» آمده باشد. قواعد نحوی زیر نحو قواعد نحوی مورد استفاده در پیوست ب را تعریف می‌کنند.

```
syntaxRule = expression, ";";
expression = alternative, {"|" alternative} | term, "-", term;
alternative = term [variable], {"," term [variable]};
term = terminal | nonterminal | "[", expression, "]"
| "{", expression, "}" | "(", expression, ")" | empty;
terminal = """, ({character - ""} - empty), """
| "", ({character - """} - empty), "";
nonterminal = identifier;
variable = "?", identifier, "?";
identifier = letter, {letter | digit | "_"};
empty = ;
```

این قواعد زیرمجموعه‌ای از قواعد نحوی مشخص شده در بند ۸-۱ ISO/IEC 14977:1996 را مشخص می‌کنند. این قواعد دلالت بر این دارند که «,» تقدم[1] بالاتری نسبت به « | » و « | » تقدم بالاتری نسبت به "=" دارد. از پرانتز می‌توان به منظور باطل کردن تقدم یا وضوح‌بخشی گروه‌بندی استفاده کرد.

ب-۱-۳ نشانه‌گذاري براي قواعد بازنويسي

نحو CGIF اصلی (بند ب-۲) و نحو CGIF گسترش‌یافته (بند ب-۳) با استفاده از قواعد EBNF مطابق استاندارد ISO/IEC 14977:1996 بیان شده است. به منظور تصریح ترجمه از CGIF اصلی به منطق

1 - Precedence

مشترک، بند ب-۲ از ترکیبی از قواعد EBNF و نشانه‌گذاری ریاضی تکمیل شده با زبان انگلیسی استفاده می‌کند. به منظور تصریح ترجمه از CGIF گسترش‌یافته به CGIF اصلی بند ب-۳ از ترکیبی از قواعد EBNF در این بند و قواعد بازنویسی تعریف شده در بند ب-۱-۳-۲ استفاده می‌کند. در قواعد نحوی پیوست ب فرض بر این است که متن در یک مرحله‌ی تحلیل لغوی به *نشانه‌هایی* مطابق استاندارد ISO/IEC 2382-15:1999 (بند ۱۵-۰۱ در مورد نشانه‌های لغوی) تقسیم شده است. بنابراین در یک قاعده EBNF[1] هر جا کاما قرار گیرد صفر یا چند حرف فاصله می‌توانند در متن ورودی وجود داشته باشند.

ب-۱-۳-۱ قواعد تبدیل

هر قاعده تبدیل باید تابعی را تعریف کند که یک رشته‌ی ورودی را تحلیل و دنباله‌ای از یک یا چند رشته‌ی خروجی را برگرداند. یک قاعده تبدیل باید دارای سه بخش باشد: یک *سرایند*[2]، یک قاعده نحوی به گونه‌ای که در بند ب-۱-۲ تعریف شده و صفر یا چند *قاعده بازنویسی*. رشته‌ی اول در سرایند باید نام تابع را مشخص کند. این نام باید با نماد غیرپایانه تعریف شده به‌وسیله آن قاعده نحوی یکسان باشد. سرایند باید همچنین یک متغیر را مشخص کند که مقدار آن باید *رشته‌ی ورودی* باشد که به‌وسیله قاعده نحوی تحلیل می‌شود. سرایند همچنین باید دنباله‌ای از یک یا چند متغیر خروجی را مشخص کند. اگر قاعده نحوی رشته-ی ورودی را با موفقیت از ابتدا تا انتها تحلیل کند آنگاه قواعد بازنویسی در صورت وجود اجرا خواهند شد. قواعد نحوی که نحو قواعد تبدیل را تعریف می‌کنند عبارتند از:

```
transRule = header, syntaxRule, {rewriteRule}, "end", ";";
header = nonterminal, "(", variable, ")", "->",
variable, {"," variable};
rewriteRule = assignment | conditional;
assignment = variable, "=", rewriteExpr, ";";
conditional = "if", condition, ({rewrite rule} - empty),
{"elif", condition, ({rewrite rule} - empty)},
["else", ({rewrite rule} - empty)], "end;"
condition = "(", test, {"&", test}, ")";
test = rewriteTerm, ["~"], "=", rewriteTerm;
test = rewriteTerm, ["~"], "=", rewriteTerm;
rewriteExpr = rewriteTerm {"," rewriteTerm};
rewriteTerm = terminal | variable | funTerm;
funTerm = identifier, "(", [funTerm, {"," funTerm}, ")";
```

نمادهای غیرپایانه زیر که از ISO/IEC 14977:1996 استخراج شده‌اند باید به گونه‌ای که در بند ب-۱-۲ مشخص شده تعریف شوند: syntaxRule، terminal، nonterminal، variable، identifier و empty.

تابعی که با استفاده از یک قاعده تبدیل تعریف می‌شود باید رشته‌ی ورودی را به دنباله‌ی مقادیر متغیرهای خروجی تبدیل کند بدین صورت که زیررشته‌هایی از رشته‌ی ورودی را کپی برداری و قواعد بازنویسی را به منظور تبدیل آن‌ها اجرا کند. اجرا باید به وسیله روال زیر تعیین شود:

1 - Extended BNF
2 - Header

استفاده از هر الگوریتم تجزیه‌ای به منظور تحلیل رشته‌ی ورودی مبتنی بر مشخصه‌های قاعده نحو مجاز است. در ابتدای تحلیل تمام متغیرهایی که در قاعده تبدیل واقع می‌شوند باید به رشته‌ی تهی مقداردهی اولیه شوند. گرچه بعضی الگوریتم‌های تجزیه ممکن است مقادیری را به متغیرها در حین تجزیه تخصیص دهند، اما معناشناسی نباید تا اتمام کل تجزیه این مقادیر را برای اجرای هیچ قاعده بازنویسی لازم بداند.

هر متغیر x در قاعده نحو باید بلافاصله پس از یک واژه t در آن قاعده قرار گیرد. نباید کاما یا هیچ نماد دیگری x و t را از هم جدا کند. مقدار تخصیصی به x باید زیررشته‌ای مانند s از رشته‌ی ورودی باشد که با الگوی مشخص شده به‌وسیله t مطابق باشد. اگر بدلی که t در آن واقع شده انتخاب نشده بود یا اگر t یک رشته‌ی تهی بود آنگاه مقدار x باید رشته‌ی تهی باشد.

پس از اتمام تجزیه، قواعد بازنویسی پیرو آن قاعده نحوی به ترتیب اجرا می‌شوند مگر اینکه یک یا چند قاعده بازنویسی در گزینه‌های یک گزاره‌ی شرطی حذف شوند.

وقتی یک تخصیص اجرا شود مقادیر نمادهای پایانه، متغیرها و اصطلاحات تابعی در سمت راست آن قاعده باید به ترتیبی که نوشته شده‌اند به هم الحاق شوند. رشته‌ی حاصل باید به عنوان مقدار به متغیر سمت چپ قاعده تخصیص داده شود.

شرطی که در یک گزاره‌ی شرطی واقع می‌شود ترکیب عطفی یک یا چند آزمون تساوی یا عدم تساوی مقادیر دو اصطلاح است. مقدار یک اصطلاح تهی که به صورت یک فاصله نوشته می‌شود یک رشته‌ی تهی است. بنابراین شرط (=~?y? & =?x?) باید تنها زمانی درست باشد که ?x? تهی و ?y? غیرتهی است.

وقتی یک گزاره‌ی شرطی اجرا می‌شود شروط مربوط به if، elif و else باید به ترتیب ارزیابی شوند. (شرط مربوط به else باید همیشه درست باشد.) وقتی اولین شرط درست پیدا شد تمام قواعد بازنویسی پیرو آن شرط باید به ترتیب و تا رسیدن به elif، else یا end اجرا شوند. سپس، اگر قاعده بازنویسی بعد از نشان‌گر خاتمه‌ی آن قاعده وجود داشته باشد اجرا خواهد شد.

هنگام رسیدن به نشان‌گر خاتمه‌ی قاعده تبدیل، اجرا باید تمام شود. سپس مقدار تابعی که نام آن در سرایند مشخص شده باید دنباله‌ای از مقادیر تمام متغیرهای خروجی باشد. هر متغیر خروجی که مقداری به آن تخصیص داده نشده بود باید به رشته‌ی تهی مقداردهی شود. هر متغیر خروجی که با یک متغیر در قاعده نحو شناسه‌ی یکسانی داشته باشد باید دارای مقدار تخصیصی به آن از رشته‌ی ورودی باشد. هیچ تخصیص مقداری نباید مقدار یک متغیر را پس از تخصیص یک مقدار به آن تغییر دهد.

بر اساس این مشخصه، بعضی قواعد تبدیل مجازند هیچ قاعده بازنویسی نداشته باشند. برای نمونه، قواعد زیر یک تابع همانی تعریف می‌کند که خروجی آن با ورودی آن یکسان است.

```
identity(?s?) -> ?t?;
identity = {character} ?t?;
end;
```

رشته‌ی ورودی s به‌وسیله قاعده نحوی به صورت رشته‌ای از صفر یا تعداد بیشتر نویسه‌ها تجزیه می‌شود. این رشته به t تخصیص داده می‌شود که خروجی تابع خواهد بود.

مقدار تخصیص داده شده به یک متغیر به عنوان نتیجه‌ی تجزیه همواره زیررشته‌ای از ورودی است. به استثنای تابع همانی، مقادیر خروجی تولید شده به‌وسیله قواعد بازنویسی برای هر رده‌ی نحوی اغلب بسیار متفاوت با هر زیررشته‌ای از ورودی هستند. برای نمونه، قاعده تبدیل زیر یک جمله‌ی نفی را از CGIF گسترش‌یافته به CGIF اصلی ترجمه می‌کند.

```
negation(?b?) -> ?ng?;
negation = "~[", [comment] ?cm?, CG ?x?, [endComment] ?ecm?, "]";
?ng? = "~[", ?cm?, CG(?x?), ?ecm?, "]";
end;
```

رشته‌های مربوط به توضیح شروع *cm* و توضیح انتهایی *ecm* بدون هیچ تغییری از ورودی به خروجی کپی می‌شوند. اما CG تودرتو با رشته‌ی ورودی *x* در CGIF گسترش‌یافته با خروجی CG(x) که در CGIF اصلی است بسیار متفاوت است. قواعد تبدیل برای رده‌بندی‌های نحوی CGIF گسترش‌یافته همانند کامپایلرهایی رفتار می‌کنند که رشته‌های ورودی در CGIF گسترش‌یافته را به رشته‌های خروجی در CGIF اصلی ترجمه می‌کنند.

ب-۱-۳-۲ توابع مورد استفاده در قواعد بازنویسي

هر تابعی که به‌وسیله یک قاعده تبدیل تعریف شود را می‌توان در یک قاعده بازنویسی استفاده کرد. این تابع می‌تواند حتی به صورت بازگشتی در خود قاعده تبدیلی که آن را تولید کرده استفاده شود. علاوه بر توابع تعریف شده به‌وسیله قواعد تبدیل، هفت تابع زیر باید به منظور استفاده در پردازش رشته‌ها و دنباله‌ها در هر قاعده بازنویسی در دسترس باشند:

- first(s) باید اولین یا تنها عنصر یک دنباله‌ی *s* را برگرداند. اگر length(s) = "0" آنگاه first(s) باید تهی باشد.
- gensym() باید رشته‌ای را برگرداند که نمایانگر یک CGname است که باید با هر CGname دیگر موجود در متن جاری متفاوت باشد. هر زمان gensym() فراخوانی شود رشته‌ای که برمی‌گرداند باید با تمام رشته‌هایی که برگردانده متفاوت باشد.
- length(s) باید طول دنباله‌ی *s* را به صورت رشته‌ای از یک یا چند حرف که ارقام دهدهی طول را نشان می‌دهند برگرداند. اگر *s* تهی باشد آنگاه length(s) باید "0" باشد. اگر *s* یک عنصر منفرد باشد آنگاه length(s) باید "1" باشد.
- map(*f*,s) باید یک تابع *f* را به هر عنصر یک دنباله‌ی *s* اعمال کند تا برای هر *x* در *s* دنباله‌ی مقادیر *f*(*x*) را برگرداند.
- second(s) باید دومین عنصر یک دنباله‌ی *s* را برگرداند. اگر length(s) < "2" آنگاه second(s) باید تهی باشد.

- substitiute(s,t,x) باید نتیجه‌ی جایگزینی رشته‌ی s به جای هر مورد از وقوع رشته‌ی t در رشته‌ی x را برگرداند. اگر t در x نباشد آنگاه substitiute(s,t,x) باید خود x را برگرداند.
- third(s) باید سومین عنصر یک دنباله‌ی s را برگرداند. اگر "3" < length(s) آنگاه third(s) باید تهی باشد.

عبارت انگلیسی "CG name" به هر نشانه‌ی نحوی که به رده‌ی "CGname" تعلق داشته باشد اشاره می‌کند.

ب-۲ نحو و معناشناسي هسته‌ي CG

نحو انتزاعی CG یک مشخصه‌ی مستقل از نشانه‌گذاری برای عبارت‌ها و مؤلفه‌های *هسته‌ی گراف مفهومی* است. هسته‌ی گراف مفهومی مجموعه‌ی کمینه‌ی CG است که قادر به بیان معناشناسی کامل CL است. معنی هر عبارت x در نحو اصلی CG با استفاده از تابع *cg2cl*(x) مشخص می‌شود. این تابع x را به یک عبارت در نحو انتزاعی CL که معادل منطقی x است نگاشت می‌دهد. از آنجایی که یک CG یا مؤلفه‌های آن می‌توانند درون مؤلفه‌های دیگر باشند، تابع *cg2cl* بازگشتی است.

بخش‌های ۲-۱ تا ۲-۱۱ نحو انتزاعی CG، نگاشت نحو انتزاعی CG به نحو انتزاعی CL و نحو محسوس متناظر برای هسته‌ی CGIF را تعریف می‌کنند. هر بخش یک تعریف صوری، یک نگاشت به CL، یک قاعده نحوی برای نحو محسوس CGIF و یک توضیح با شرح و مثال است. قواعد نحوی به صورت قواعد EBNF، که در ISO/IEC 14977:1996 تصریح و در بند ب-۱-۲ خلاصه شده، نوشته شده‌اند. برای هر قاعده نحو CGIF رده‌های لغوی بند الف-۲-۲ باید مد نظر باشند. در بند الف-۲-۳-۲ رده‌ی *name* شامل یک رده‌ی *encosedname* از رشته‌های محصور بین علامت‌های نقل قول و یک رده‌ی *namesequence* از رشته‌های غیر محصور است. به منظور اجتناب از ابهامات احتمالی، در رده‌ی *CGname* لازم است تمام دنباله‌های نامی CLIF به استثنای موارد متعلق به رده‌ی *identifier* میان علامت‌های نقل قول قرار گیرند:

```
CGname = identifier | "", (namesequence - identifier), "" | numeral | enclosedname | quotedstring;
identifier = letter, {letter | digit | "_"};
```

زمانی که CGIF به CL ترجمه می‌شود هر CGname باید با حذف علامت‌های نقل قول دو طرف دنباله‌ی نامی به یک نام در CLIF ترجمه شود. CLIF بین متغیرها و ثابت‌ها از نظر نحوی تمایزی قائل نیست. اما در CGIF به هر CGname که به عنوان یک برچسب تعریف‌کننده یا برچسب مقید مورد استفاده نیست یک *ثابت* گفته می‌شود.

نماد شروع برای نحوCGIF باید رده‌ی text باشد. اگر ورودی یک متن کامل یا رده‌ی CG باشد آنگاه ورودی رشته‌ای است که یک گراف مفهومی را نمایش می‌دهد.

ب-۲-۱ کنش‌گر

تعریف: یک رابطه‌ی مفهومی $ac = (r,s)$ که در آن r باید یک مرجع تحت عنوان *برچسب نوع* ac و دنباله‌ی کمان $s = s_1, s_2$ باید متشکل از یک دنباله‌ی کمان s_1 تحت عنوان *کمان‌های ورودی* و یک کمان منفرد s_2 تحت عنوان *کمان خروجی* باشد.

CL: $cg2cl(ac)$ باید یک تساوی :eq باشد که اصطلاح اول آن باید نام $cg2cl(s_2)$ و اصطلاح دوم آن باید اصطلاح تابعی با عملگر $cg2cl(r)$ و دنباله‌ی اصطلاح $cg2cl(s_1)$ با یک نشان‌گر دنباله‌ی اختیاری sqn باشد.

CGIF:

```
actor = "(", [comment], [ "#", "?"], CGname, arcSequence, "|", arc,
[endComment], ")";
```

همانند دیگر روابط مفهومی، یک گره‌ی کنش‌گر بین دو پرانتز قرار می‌گیرد. نماد # باید یک برچسب هم‌مرجع مقید را که به عنوان یک برچسب نوع استفاده می‌شود نشانگذاری کند.

توضیح - گرچه یک کنش‌گر به عنوان یک مورد خاص از یک رابطه‌ی مفهومی تعریف می‌شود اما نحو اصلی CG یک کنش‌گر را به‌طور دقیق به یک کمان خروجی محدود می‌کند تا بدین ترتیب نگاشت آن به یک تابع CL امکان‌پذیر باشد. نمونه‌هایی از کنش‌گرها و نگاشت آن‌ها به CGIF و CLIF در شکل ب-۳ نشان داده شده است. کمان‌های ورودی می‌توانند یک نشان‌گر دنباله در انتهای خود داشته باشند اما برای کمان‌های خروجی نباید از هیچ نشان‌گر دنباله‌ای استفاده کرد. نحو CGIF گسترش‌یافته به کنش‌گرها اجازه می‌دهد هر تعداد کمان خروجی داشته باشند.

ب-۲-۲ کمان

تعریف: یک مرجع ar که در دنباله‌ی کمان یک رابطه‌ی مفهومی قرار می‌گیرد.

CL: $cg2cl(ar)$ باید نام n بدون نشان‌گر مرجع ar باشد.

CGIF:

```
arc = [comment], reference;
```

توضیح - تابع $cg2cl$ یک کمان را به نام مرجع آن نگاشت داده و هر نشان‌گری که یک برچسب مقید را مشخص کند حذف می‌کند.

ب-۲-۳ دنباله‌ي کمان (arcSequence)

تعریف: یک زوج $as = (s , sqn)$ متشکل از یک دنباله‌ی s از صفر یا چند کمان که به دنبال آن‌ها یک نشان‌گر دنباله‌ی اختیاری sqn آمده باشد.

CL: $cg2cl(ar)$ باید یک دنباله‌ی اصطلاح $ts = cg2cl(s)$ و نشان‌گر دنباله‌ی sqn (در صورت وجود در as) باشد. دنباله‌ی اصطلاح ts باید $map(cg2cl,s)$ باشد که در آن map تابعی است که $cg2cl$ را به هر کمان دنباله‌ی s اعمال می‌کند تا نامی را استخراج کند که عنصر متناظر دنباله‌ی ts شود:

CGIF:

```
arcSequence = {arc}, [[comment], "?", seqmark];
```

هر نشان‌گر دنباله در یک دنباله‌ی کمان *as* باید با نشان‌گر دنباله‌ای در یک مفهوم وجودی یکسان باشد که به‌طور مستقیم مشمول زمینه‌ای است که کنش‌گر یا رابطه‌ی مفهومی دارای دنباله‌ی کمان *as* را شامل می‌شود.

توضیح - اختیارِ داشتن یک نشان‌گر دنباله در یک دنباله‌ی کمان دلالت بر این دارد که یک رابطه‌ی مفهومی می‌تواند تعداد متغیری کمان داشته باشد.

ب-۲-٤ توضیح

تعریف: یک رشته‌ی *cm* که نباید هیچ تأثیری بر معنای هیچ عبارت *x* در CLIF که *s* در آن قرار می‌گیرد داشته باشد.

CL: *cg2cl*(*cm*) باید زیررشته‌ی *s* از *cm* باشد که جداکننده‌های "/*" و "*/" از یک توضیح یا سرآغاز ";" از یک توضیح انتهایی را شامل نشود. رشته‌ی *s* باید در نمایش CL یک توضیح آورده شده باشد. در ضمن باید به عبارت نحوی منطق مشترک که عبارت *x* به آن ترجمه می‌شود مرتبط باشد. قواعد نحوی برای توضیح و توضیح انتهایی در CGIF اصلی و CGIF گسترش‌یافته یکسان است.

CGIF:

```
comment = "/*", {(character-"*") | [ "*", (character-"/")]}, [ "*"], "*/";
endComment = ";", {character - ("]" | ")")};
```

رشته‌ای که مابین جداکننده‌های "/*" و "*/" قرار می‌گیرد نباید دارای زیررشته‌ی "*/" باشد. رشته‌ی یک توضیح انتهایی می‌تواند هر تعداد ";" داشته باشد اما نباید دارای "]" یا ")" باشد.

توضیح - یک توضیح می‌تواند بلافاصله پس از کروشه باز هر مفهوم، پرانتز باز هر کنش‌گر یا رابطه‌ی مفهومی یا بلافاصله قبل از هر کمان قرار گیرد. همچنین یک توضیح می‌تواند به صورت در هم آمیخته با مفاهیم و روابط مفهومی هر گراف مفهومی ظاهر شود. یک توضیح انتهایی می‌تواند بلافاصله قبل از کروشه بسته‌ی هر مفهوم یا پرانتز بسته‌ی هر کنش‌گر یا رابطه‌ی مفهومی قرار گیرد. از آنجا که نحو توضیحات در CGIF اصلی و گسترش‌یافته یکسان است هیچ گونه قواعد نحوی اضافی برای توضیحات نباید در بند ب-۳ آورده شود.

ب-۲-٥ مفهوم

تعریف: یک زوج $c = (R,g)$ که در آن *R* باید یک برچسب تعریف‌کننده یا مجموعه‌ای از صفر یا چند مرجع و *g* باید یک گراف مفهومی باشد که *به طور مستقیم تحت‌شمول c* است.

CL: *cg2cl*(*c*) باید جمله‌ی *s* تعیین شده با استفاده از یکی از سه گزینه‌ی زیر باشد:

زمینه: اگر *R* پوچ[1] باشد، آنگاه $s = cg2cl(g)$. در این مورد باید به *c* یک *زمینه* گفت.

وجودی: اگر *g* خالی[1] و *R* یک برچسب تعریف‌کننده باشد آنگاه *s* باید یک جمله‌ی سوری از نوع وجودی با مجموعه‌ای از نام‌های {*cg2cl*(*R*)}، یک بدنه متشکل از یک جمله‌ی بولی از نوع ترکیب عطفی و بدون مؤلفه باشد. در این مورد *c* را باید یک *مفهوم وجودی* نامید.

1 - Empty

هم‌مرجع: اگر g خالی و R مجموعه‌ای از یک یا چند مرجع باشد آنگاه اگر هر مرجع درون R را با r مشخص کنیم جمله‌ی s باید یک جمله‌ی بولی از نوع ترکیب عطفی باشد که مؤلفه‌های آن باید مجموعه‌ای از تساوی‌ها باشد که اصطلاح اول آن‌ها $cg2cl(r)$ و اصطلاح دوم آن‌ها $cg2cl(t)$ باشد (برای هر t عضو $R - \{r\}$). در این مورد c را باید یک *مفهوم هم‌مرجع* نامید.

غیر معتبر نحوی: در CGIF اصلی موردی که در آن g غیرخالی و R غیرپوچ باشد مجاز نیست و برای آن هیچ ترجمه‌ای در CL تعریف نشده است.

CGIF:

concept = context | existentialConcept | coreferenceConcept;
context = "[", [comment], CG, [endComment], "]";
existentialConcept = "[", [comment], "*", (CGname | seqmark),
[endComment], "]";
coreferenceConcept = "[", [comment], ":", {reference}-,
[endComment], "]";

یک زمینه باید مفهومی باشد که دارای یک CG است. اگر CG پوچ باشد زمینه *تهی* است حتی اگر دارای یک یا چند توضیح[2] باشد. هر توضیحی که بلافاصله پس از کروشه باز قرار گیرد باید بخشی از مفهوم و هر توضیح دیگری باید بخشی از CG درون آن باشد. یک مفهوم هم‌مرجع باید دارای یک یا چند ثابت یا برچسب هم‌مرجع مقید باشد. در EBNF، یک تکرار که پس از آن علامت منها قرار داشته باشد بطوری که پس از آن علامت چیز دیگری قرار نداشته باشد، حداقل یک تکرار را نشان می‌دهد.

توضیح - یک زمینه با استفاده از یک جفت کروشه نشان داده می‌شود که محدوده‌ی حوزه‌ی کاربرد سورهای CG درون آن را مشخص می‌کنند. یک زمینه‌ی تهی [ ] به (and) در CLIF ترجمه می‌شود که مقدار آن طبق تعریف true است. یک مفهوم وجودی با استفاده از یک مفهوم مانند [x*] نمایش داده می‌شود. این مفهوم به ;((exists (x) (and)) در CLIF ترجمه می‌شود. این جمله وجود یک x را تصدیق می‌کند. یک مفهوم هم‌مرجع با استفاده از یک مفهوم مانند [: ?x Cicero Tully] [: ?x Cicero abcd?] که دارای مجموعه‌ای از ثابت‌ها یا برچسب‌های هم‌مرجع مقید است نمایش داده می‌شود. مفهوم [: ?x Cicero Tully ?abcd] در CLIF به یک ترکیب عطفی تساوی‌ها ترجمه می‌شود:

(and (= x Cicero) (= x Tully) (= x abcd))

یک مفهوم هم‌مرجع با تنها یک مرجع، مانند [:?x] به یک ترکیب عطفی تهی (and) تبدیل می‌شود. چنین مفهومی را از آنجا که تأثیر معنایی ندارد می‌توان حذف نمود.

ب-۲-۶ گراف مفهومي (CG)

تعریف: یک سه‌تایی $g = (C,R,A)$ که در آن C مجموعه‌ای از مفاهیم، R مجموعه‌ای از روابط مفهومی و A مجموعه‌ی تمام کمان‌هایی است که در دنباله‌ی کمان یک رابطه‌ی مفهومی عضو R وجود داشته باشند. اگر C و R هر دو تهی باشند آنگاه A نیز تهی خواهد بود و به g نیز یک گراف مفهومی *پوچ* گفته می‌شود.

---

1 - Blank
2 - Comment

CL: فرض کنید L زیرمجموعه‌ای از C باشد که شامل مفاهیم وجودی است. همچنین فرض کنید X مجموعه‌ی تمام مفاهیم، روابط مفهومی و گزاره‌های نفی g به استثنای موارد عضو E است.
فرض کنید B یک جمله‌ی بولی از نوع ترکیب عطفی است که مؤلفه‌های آن متشکل از تمام جمله‌های $cg2cl(x)$ به ازای هر x عضو X است.
اگر E تهی باشد آنگاه $cg2cl(g)$ برابر با B است.
اگر E غیرتهی باشد آنگاه $cg2cl(g)$ یک جمله‌ی سوری از نوع وجودی است که مجموعه‌ی نام‌های آن از CGname برچسب هم‌مرجع تعریف‌کننده هر e در E تشکیل شده و بدنه‌ی آن نیز B است.

CGIF:

CG = {concept | conceptualRelation | negation | comment};

یک گراف مفهومی از یک مجموعه‌ی نامرتب از مفاهیم، روابط مفهومی، گزاره‌های نفی و توضیحات تشکیل می‌شود. به طور صوری، یک گزاره‌ی نفی، زوجی است تشکیل شده از یک مفهوم و یک رابطه‌ی مفهومی که در CGIF هیچ‌گاه جدا از هم نیستند.

توضیح - بر اساس این مشخصه، هر CG یا به یک جمله‌ی سوری از نوع وجودی و یا به یک جمله‌ی بولی از نوع ترکیب عطفی نگاشت داده می‌شود. اگر ترکیب عطفی تنها یک مؤلفه داشته باشد آنگاه جمله را می‌توان به یک تساوی، جمله‌ی تجزیه‌ناپذیر یا یک جمله‌ی بولی از نوع نفی ساده کرد. g پوچ در CLIF متناظر است با (and) که طبق تعریف مقدار آن true است. اگر چه لازم نیست گره‌های یک CG مرتب باشند اما یک نرم‌افزار که CGIF را پردازش می‌کند ممکن است در صورتی که برچسب‌های هم‌مرجع تعریف‌کننده قبل از برچسب‌های مقید متناظرشان واقع شوند کاراتر اجرا شود. ساده‌ترین راه برای تضمین چنین شرایطی انتقال مفاهیم وجودی به جلوی هر زمینه است.

ب-۲-۷ رابطه‌ي مفهومي

تعریف: یک زوج $cr = (r,s)$ که در آن r باید یک مرجع تحت عنوان *برچسب نوع* cr و s باید یک دنباله‌ی کمان باشد.

CL: $cg2cl(ac)$ باید یک جمله‌ی تجزیه‌ناپذیر باشد که مسند آن $cg2cl(r)$ و دنباله‌ی اصطلاح آن $cg2cl(s)$ است.

CGIF:

conceptualRelation = ordinaryRelation | actor;
ordinaryRelation = "(", [comment], ["#", "?"], CGname, arcSequence,
[endComment], ")";

یک رابطه‌ی مفهومی معمولی تنها یک دنباله‌ی کمان دارد. یک کنش‌گر دنباله‌ی کمان را به دو زیردنباله تقسیم می‌کند. یک برچسب هم‌مرجع مقید که به عنوان یک برچسب نوع استفاده می‌شود باید با رشته‌ی "#?" شروع شود.

توضیح - با اجازه دادن به برچسب نوع یک رابطه‌ی مفهومی که یک برچسب مقید باشد، CGIF از قابلیت CL برای استفاده از سور برروی روابط و توابع پشتیبانی می‌کند. برای نمونه عبارت CGIF انتهای بند ب-۱-۱ را که جمله‌ی "Bob and Sue are related" را نمایش می‌دهد ببینید.

ب-۲-۸ نفي

تعريف: يک زوج $ng = (c,cr)$ که در آن c بايد يک مفهوم و cr بايد يک رابطه‌ی مفهومی باشد. r به عنوان برچسب نوع cr بايد يک ثابت باشد که CGname آن Neg است. با زوج (c,cr) بايد به عنوان يک واحد منفرد برخورد کرد.

CL: $cg2cl(ng)$ بايد يک جمله‌ی بولی از نوع نفی با مؤلفه‌ی $cg2cl(g)$ باشد.

CGIF:

negation = "~", context;

يک نفی بايد با نماد ~ شروع شود. گرچه يک نفی به طور صوری به صورت يک زوج متشکل از يک زمينه و يک رابطه‌ی مفهومی تعريف می‌شود اما دو عنصر اين زوج نبايد به صورت دو گره‌ی جداگانه در CGIF بيان شوند.

توضيح - يک نفی گزاره‌ای که به‌وسيله گراف مفهومی g بيان شده را نفی می‌کند. برای نمونه به CGIF مربوط به شکل ب-۲ مراجعه شود. نفی CG پوچ که به صورت [ ] ~ نوشته می‌شود همواره false و CLIF متناظر آن عبارت است از (not (and)).

ب-۲-۹ مرجع

تعريف: يک زوج $r = (m,n)$ که n يک نام CG و m يک نشان‌گر است که بايد يک *ثابت* يا يک برچسب *مقيد* را مشخص کند.

CL: $cg2cl(r)$ بايد نام n باشد. نشان‌گر m بايد برای يک برچسب مقيد يک ? و برای يک ثابت، يک رشته‌ی تهی "" باشد.

CGIF:

reference = ["?"], CGname;

نحو مراجع در CGIF اصلی و CGIF گسترش‌يافته يکسان است. هيچ گونه قواعد نحو اضافی برای مراجع در بند ب-۳ نيامده است.

ب-۲-۱۰ دامنه‌ي کاربرد

تعريف: يک مجموعه‌ی S از زمينه‌ها که به يک مفهوم x با يک برچسب تعريف‌کننده با نام x در CG، مرتبط است.

اصطلاحات زير در تعريف محدوديت‌ها برروی برچسب‌های تعريف‌کننده در CGIF اصلی و CGIF گسترش‌يافته به‌کار می‌روند.

- *ثابت*، يک نام CG بدون هيچ پيشوندی.
- *برچسب هم‌مرجع مقيد*، يک نام CG با پيشوند «?».

• *برچسب دنباله‌ی مقید*، یک نشان‌گر دنباله‌ی با پیشوند «?».

• *برچسب مقید*، یک برچسب هم‌مرجع مقید یا یک برچسب دنباله‌ی مقید.

• *برچسب هم‌مرجع تعریف‌کننده*، یک نام CG با پیشوند «*».

• *برچسب دنباله‌ی تعریف‌کننده*[1]، یک نشان‌گر دنباله‌ی CG با پیشوند «*».

• *برچسب تعریف‌کننده*، یک برچسب هم‌مرجع تعریف‌کننده یا یک برچسب دنباله‌ی تعریف‌کننده.

بر اساس این تعریف، یک برچسب دنباله‌ی تعریف‌کننده باید با رشته‌ی «...*» و یک برچسب دنباله‌ی مقید باید با رشته‌ی «...?» شروع شود.

محدودیت‌ها: فعل *دربردارد* باید به عنوان بستار گذرای *رابطه‌ی به طور مستقیم دربردارد* تعریف شود و چه در CGIF اصلی و چه در CGIF گسترش‌یافته محدودیت‌های زیر باید در مورد آن برقرار باشد:

ب-۲-۱۰-۱ اگر یک زمینه‌ی c به طور مستقیم دربردارنده‌ی یک گراف مفهومی g باشد آنگاه c به طور مستقیم دربردارنده‌ی تمام گره‌های این گراف و تمام مؤلفه‌های هرکدام از این گره‌هاست به استثنای مؤلفه‌هایی که یک زمینه ی g آن‌ها را در بردارد.

ب-۲-۱۰-۲ اگر یک زمینه‌ی c یک زمینه‌ی d را به طور مستقیم دربرداشته باشد آنگاه c تمام آنچه را که d به طور مستقیم در بردارد، به‌طور غیرمستقیم دربردارد.

ب-۲-۱۰-۳ عبارت «c دربردارنده‌ی x است» با عبارت «c به‌طور مستقیم یا غیرمستقیم x را دربردارد» هم‌معنی است.

ب-۲-۱۰-۴ اگر یک زمینه‌ی c به‌طور مستقیم یک مفهوم x با برچسب تعریف‌کننده با نام n را دربرداشته باشد آنگاه c نباید هیچ مفهوم دیگری غیر از x با برچسب تعریف‌کننده با نام n را دربرداشته باشد. در ضمن c باید متعلق به دامنه‌ی کاربرد S مربوط به x باشد.

ب-۲-۱۰-۵ اگر یک زمینه‌ی c در دامنه‌ی کاربرد S مربوط به یک مفهوم x باشد آنگاه هر زمینه‌ی d که c به طور مستقیم آن را دربرداشته باشد باید در دامنه‌ی کاربرد S وجود داشته باشد مگر اینکه d به طور مستقیم دربردارنده‌ی یک مفهوم y با برچسب تعریف‌کننده‌ی هم‌نام با برچسب تعریف‌کننده x باشد.

ب-۲-۱۰-۶ هر برچسب مقید که نام CG آن n باشد باید در محدوده‌ی کاربرد مربوط به یک مفهوم که نام CG برچسب تعریف‌کننده آن n است باشد.

ب-۲-۱۰-۷ هیچ ثابتی که نام CG آن n است نباید در دامنه‌ی کاربرد مربوط به مفهومی با برچسب تعریف‌کننده با نام CG n وجود داشته باشد.

یادآوری - این محدودیت‌ها تضمین می‌کند که برای هر جمله‌ی *s* در CGIF، ترجمه‌ی *cg2cl*(s) از محدودیت‌های CL برروی دامنه‌ی کاربرد سورها پیروی کند. از آنجا که محدودیت‌های دامنه‌ی کاربرد در CGIF اصلی و CGIF گسترش‌یافته یکسان است هیچ محدودیت اضافی نباید در بند ب-۳ آمده باشد.

1 - Defining sequence label

ب-۲-۱۱ متن

تعریف: یک مفهوم c که هیچ مفهوم دیگری به‌طور مستقیم یا غیرمستقیم دربردارنده‌ی آن نباشد

CL: $cg2cl(c)$ باید متنی متشکل از جمله‌ی $cg2cl(g)$ باشد که در آن g گراف مفهومی است که c به طور مستقیم دربردارنده‌ی آن است. اگر n به عنوان یک نام CG بلافاصله قبل از g در مشخصه‌ی CGIF زمینه‌ی c قرار گیرد آنگاه c باید نام متن CL باشد.

CGIF:
text = "[", [comment], "Proposition", ":", [CGname], CG,
[endComment], "]";

از آنجا که هیچ زمینه‌ای دربردارنده‌ی یک متن نیست آن را باید *دورترین زمینه*[1] نامید.

توضیح - این قاعده نحوی از نحو CGIF گسترش‌یافته که اجازه می‌دهد یک زمینه دارای یک برچسب نوع و یک نام CG باشد استفاده می‌کند. از آنجا که نحو CGIF زیرمجموعه‌ای از نحو CGIF گسترش‌یافته است، هر پردازنده‌ای که CGIF گسترش‌یافته را می پذیرد می‌تواند از متنی که در قالب CGIF اصلی نوشته شده باشد استفاده کند. از کروشه‌های زمینه می-توان به منظور گروه‌بندی مفاهیم و روابط یک متن در قالب واحدهایی متناظر با جمله‌های CLIF استفاده کرد. این گروه‌بندی هیچ تأثیری بر معناشناسی ندارد.

ب-۳ نحو CGIF گسترش‌یافته

CGIF گسترش‌یافته[2] یک ابرمجموعه‌ی CGIF اصلی است و هر جمله‌ای که از نظر نحو CGIF اصلی درست باشد از نظر نحو CGIF گسترش‌یافته نیز درست است. برجسته‌ترین ویژگی CGIF گسترش‌یافته امکان انتخاب یک *برچسب نوع* یا یک *عبارت نوع*[3] در سمت چپ هر مفهوم است. علاوه بر انواع، CGIF گسترش‌یافته قابلیت‌های زیر را به CGIF اصلی اضافه می‌کند:

- انتخاب‌های بیشتر در مفاهیم از جمله سورهای عمومی؛
- زمینه‌های بولی برای نمایش عملگرهای or، if و iff؛
- اجازه‌ی قرار دادن گره‌های مفهوم در دنباله‌ی کمان روابط مفهومی؛
- قابلیت واردسازی یک متن به متن دیگر.

هدف از طراحی این گسترش‌ها افزایش دقت و خوانایی جمله‌ها و تبدیل آن‌ها به زبان مقصد مناسب‌تر برای ترجمه از زبان‌های طبیعی و سایر گویش‌های CL از جمله CLIF است. اما هیچ کدام از این موارد قدرت بیان CGIF را فراتر از هسته‌ی CG نمی‌برند چرا که معناشناسی هر ویژگی افزونه به واسطه‌ی ترجمه‌ی آن به CGIF اصلی تعریف می‌شود. معناشناسی CGIF اصلی نیز به واسطه‌ی ترجمه‌ی آن به CL تعریف می‌شود.
این بند نحو محسوس CGIF گسترش‌یافته و ترجمه‌ی هر ویژگی افزونه به CGIF اصلی را تعریف می‌کند. در اثر این ترجمه یک تابع CG مشخص می‌شود که هر جمله‌ی s از CGIF گسترش‌یافته را به جمله‌ی

1 - Outermost context
2 - Extended
3 - Type expression

$CG(s)$ در CGIF اصلی که از نظر معنی معادل آن است ترجمه می‌کند. تابع ترکیبی $cg2cl(CG(s))$ جمله‌ی s را به یک جمله در نحو انتزاعی CL که به‌طور منطقی معادل آن است ترجمه می‌کند.

تابع CG و توابع دیگر برای سایر رده‌های CGIF با استفاده از *قواعد تبدیلی* که نشانه‌گذاری آن‌ها در بند ب-۱-۳-۱ مشخص شده تعریف می‌شوند. دو رده‌ی comment و reference نحو یکسانی در CGIF اصلی و گسترش‌یافته دارند. بدین صورت که برای هر توضیح cm در CGIF گسترش‌یافته داریم comment(cm) = cm و برای هر مرجع r در CGIF گسترش‌یافته داریم reference(r) = r. برای هر رده‌ی دیگر X از CGIF اصلی، رشته‌های رده‌ی X زیرمجموعه‌ی محض رشته‌های همان رده در CGIF گسترش‌یافته هستند.

از آنجا که از تعاریف بند ب-۲ نحو انتزاعی گراف مفهومی و نگاشت آن به نحو انتزاعی منطق مشترک را مشخص کرد، در این تعاریف از ساختارهای مستقل از نشانه‌گذاری، مانند مجموعه‌ها، استفاده شد. تعاریف زیر نحوه‌ی نگاشت از نحو محسوس CGIF گسترش‌یافته به نحو محسوس CGIF اصلی را مشخص می‌کنند. در نتیجه، این تعاریف در قالب رشته‌ها و توابع تبدیل آن‌ها تعریف شده‌اند.

ب-۳-۱ کنش‌گر

تعریف: یک رشته‌ی ac که باید حاوی یک توضیح cm، یک مرجع r تحت عنوان *برچسب نوع*، یک دنباله‌ی کمان s_1 تحت عنوان *کمان‌های ورودی*، یک دنباله‌ی کمان s_2 تحت عنوان *کمان‌های خروجی* و یک توضیح انتهایی اختیاری ecm باشد. کمان‌های خروجی s_2 نباید دارای یک نشان‌گر دنباله باشند.

ترجمه: یک گراف مفهومی g.

```
actor(?ac?) -> ?g?;
    actor   = "(", [comment] ?cm?, (["#", "?"], CGname) ?r?,
    arcSequence ?s1?, "|", {arc} ?s2?, [endComment] ?ecm?, ")";
    ?z1?  = first(arcSequence(?s1?));
    ?z2?   = first(arcSequence(?s2?));
    ?sqn?  = third(arcsequence(?s1?));
    if (length(?s2?)="0")
    ?cr?  = "(", ?cm?, ?r?, ?z1?, ?sqn?, ?ecm?, ";0-output actor", ")";
    elif (length(?s2?)="1")
    ?cr?   = "(", ?cm?, ?r?, ?z1?, ?sqn?, "|", ?z2?, ?ecm?, ")";
    else ?cr?  =  "(", ?cm?, ?r?, ?z1?, ?sqn?, "/*|*/", ?z2?, ?ecm?, ")";
    end;
    ?g? = second(arcSequence(?s1?)), second(arcSequence(?s2?)), ?cr?;
    end;
```

اگر s_2 هیچ کمان خروجی نداشته باشد آنگاه cr باید یک رابطه‌ی مفهومی معمولی به صورتی که در بند ب-۳-۷ تعریف شده باشد. اما به منظور نشان دادن اینکه cr از یک کنش‌گر مشتق شده است یک توضیح انتهایی "0-output actor" درج شده است. اگر s_2 یک کمان خروجی داشته باشد آنگاه ac باید یک کنش‌گر باشد. اما cr با ac متفاوت است چرا که کمان‌ها به CGIF اصلی ترجمه می‌شوند. اگر s_2 دو یا چند کمان خروجی داشته باشد آنگاه cr باید یک رابطه‌ی مفهومی معمولی باشد و هدف از درج توضیح "/* | */" تمایز

بین کمان‌های ورودی و خروجی است. آخرین قاعده بازنویسی، *cr* را پس از هر گراف مفهومی که از دنباله‌های کمان مشتق شود قرار می‌دهد.

توضیح - برای مثال، اثر ترکیب‌شده قواعد تبدیل کنش‌گرها، کمان‌ها، دنباله‌های کمان و مفاهیم، گره‌ی کنش‌گر زیر را:

(IntegerDivide [Integer: *x] [Integer: 7] | *u *v)

به شش گره، متشکل از سه مفهوم و سه رابطه‌ی مفهومی زیر ترجمه می‌کند:

[*x] (Integer ?x) (Integer 7) [*u] [*v]

(IntegerDivide ?x 7 /*|*/ ?u ?v)

توضیح /* | */ هیچ اثر معنایی بر CGIF اصلی یا CL ندارد. اما در صورت حفظ، یک نگاشت معکوس به CGIF گسترش‌یافته را به منظور تمایز کمان‌های ورودی از کمان‌های خروجی ممکن می‌سازد. اگر این تمایز برای برنامه‌ای مهم باشد می‌توان از اصول بدیهی برای بیان وابستگی‌های تابعی خروجی‌ها به ورودی‌ها استفاده کرد. برای نمونه، رابطه‌ی منطق مشترک که از ترجمه‌ی یک کنش‌گر از نوع IntegerDivide بدست می‌آید در محدودیت زیر که در CLIF بیان شده صدق می‌کند:

```
(exists (Quotient Remainder) (forall (x1 x2 x3 x4)
  (iff (IntegerDivide x1 x2 x3 x4)
    (and (= x3 (Quotient x1 x2)) (= x4 (Remainder x1 x2))))))
```

این جمله تصدیق می‌کند توابع Quotient و Remainder وجود دارند که مقادیر آرگومان‌های سوم و چهارم رابطه‌ی IntegerDivide را تعیین می‌کنند. قواعد ترجمه آن اصل بدیهی را به صورت خودکار تولید نمی‌کنند اما آن اصل بدیهی را می‌توان با استفاده از یک جمله‌ی CGIF که به جمله‌ی CLIF زیر ترجمه می‌شود بیان کرد.

```
[*Quotient] [*Remainder]
[[@every*x1] [@every*x2] [@every*x3] [@every*x4]
[Equiv: [Iff: (IntegerDivide ?x1 ?x2 | ?x3 ?x4)]
  [Iff: (#?Quotient ?x1 ?x2 | ?x3) (#?Remainder ?x1 ?x2 | ?x4)]]]
```

برای نشان دادن اینکه سورهای وجودی برای [Quotient*] و [Remainder*] بر سورهای عمومی برای چهار آرگومان تقدم دارند از یک جفت کروشه‌ی زمینه[1] به منظور حصر گره‌های مفهوم با سورهای عمومی استفاده شده است.

ب-۳-۲ کمان

تعریف: یک رشته‌ی *ar* که باید دارای یک توضیح اختیاری *cm* به علاوه‌ی یک مرجع *r* یا یک برچسب تعریف‌کننده که نام CG آن *n* باشد یا یک مفهوم *c* باشد.

ترجمه: یک زوج (x,g) متشکل از یک کمان x و یک گراف مفهومی g.

1 - Context brackets

```
arc(?ar?) -> ?x?, ?g?;
arc = [comment] ?cm?, (reference ?r? | "*", CGname ?n? | concept ?c?);
if  (?r?~= )?x?=  ?ar; ?g? = ;
elif  (?n?~= ) ?x?  =  ?cm?, "?", ?n?; ?g? = "[*", ?n?, "]";
else      ?x?=  ?cm?, first(concept(?c?));
?g? =  third(concept(?c?));
end; end;
```

اگر *ar* یک مرجع باشد آنگاه *x* باید همان *ar* بدون تغییر و *g* باید پوچ باشد. اگر *ar* حاوی یک برچسب تعریف‌کننده باشد آنگاه *x* باید نتیجه‌ی جایگزینی نشان‌گر * در *ar* با ? و *g* باید یک مفهوم [*n] باشد. اگر *ar* حاوی یک مفهوم *c* باشد آنگاه *x* باید نتیجه‌ی جایگزینی مفهوم *c* در *ar* با یک مرجع *r* و *g* باید third(concept(*c*)) باشد.

توضیح: برای نمونه، اگر کمان *ar* به صورت [Integer] باشد آنگاه مقدار concept([Integer]) یک نام CG مانند g0023 و arc([Integer]) زوج متشکل از مرجع ?g0023 و گراف مفهومی [*g0023] (Integer ?g0023) خواهد بود.

ب-۳-۳ arcSequence

تعریف: یک رشته‌ی *as* که باید دربردارنده‌ی یک دنباله‌ی *s* از صفر یا چند کمان باشد که به دنبال آن یک نشان‌گر دنباله‌ی اختیاری *sqn* آمده است.

ترجمه: یک سه‌تایی (*rs,g,swqn*) متشکل از یک دنباله‌ی مراجع *rs*، یک گراف مفهومی *g* و نشان‌گر دنباله‌ی *sqn*.

```
arcSequence(?as?) -> ?rs?, ?g?, ?sqn?;
arcSequence = {arc} ?s?, [[comment], "?", seqmark] ?sqn?;
?rs? = map(first, map(arc,?s?));
?g? = map(second ,map(arc,?s?));
end;
```

توضیح - تابع map(arc, ?s?) کمان arc را به هر کمان متعلق به *s* اعمال تا دنباله‌ای از زوج‌های متشکل از یک مرجع و یک مفهوم تولید کند. سپس، تابع map(first, map(arc, ?s?)) دنباله‌ی مراجع را از عنصر اول هر زوج استخراج می‌کند. در نهایت map(second, map(arc, ?s?)) دنباله‌ی مفاهیم را از عنصر دوم هر زوج استخراج می‌کند. اختیارِ داشتن یک نشان‌گر دنباله در یک دنباله‌ی کمان دلالت بر این دارد که یک رابطه‌ی مفهومی می‌تواند تعداد متغیری کمان داشته باشد. یک کنش‌گر می‌تواند تعداد متغیری کمان ورودی داشته باشد، اما تعداد کمان‌های خروجی باید ثابت باشد. بنابر این کمان‌های خروجی نمی‌توانند یک نشان‌گر دنباله داشته باشند.

ب-۳-٤ بولي

تعریف: یک رشته‌ی *b* که باید دربردارنده‌ی یک زمینه‌ی *bc* باشد. این زمینه نباید به طور مستقیم دربردارنده‌ی یک مرجع یا یک برچسب تعریف‌کننده باشد. زمینه‌ی *bc* باید یا یک پیشوند "~" داشته و هیچ

برچسب نوعی نداشته باشد یا هیچ پیشوندی نداشته و یکی از ثابت‌ها Either، Equiv، Equivalence، If، Iff و Then را به عنوان برچسب نوع داشته باشد.

ترجمه: یک نفی *ng* که باید به صورت *negation*(*b*)، *eitherOr*(*b*)، *ifThen*(*b*) یا *equiv*(*b*) باشد.

```
boolean = negation | eitherOr | ifThen | equiv;
negation(?b?) -> ?ng?;
negation = "~[", [comment] ?cm?, CG ?x?, [endComment] ?ecm?, "]";
?ng? = "~[", ?cm?, CG(?x?), ?ecm?, "]";
end;

ifThen(?b?) -> ?ng?;
ifThen = "[", [comment] ?cm1?, "If", [":"], CG ?ante?,
"[", [comment] ?cm2?, "Then", [":"], CG ?conse?,
[endComment] ?ecm1?, "]", [endComment] ?ecm2?, "]";
?ng?   = "~[", ?cm1?, CG(?ante?),
"~[", ?cm2?, CG(?conse?), ?ecm1?, "]", ?ecm2?, "]";
end;

equiv(?b?) -> ?ng?;
equiv = "[", [comment] ?cm1?, ("Equiv" | "Equivalence"), [":"],
 "[", [comment] ?cm2?, "Iff", [":"], CG ?g1?,
  [endComment] ?ecm2? "]",
"[", [comment] ?cm3?, "Iff", [":"], CG ?g2?,
                [endComment] ?ecm3? "]", [endComment] ?ecm1? "]";
?ng? = "[", ?cm1?, "~[", ?cm2?, CG(?g1?),
"~[", CG(?g2?), "]", ?ecm2?, "]",
?cm2?, "~[", ?cm3?, CG(?g2?),
"~[", CG(?g1?), "]", ?ecm3?, "]", ?ecm1?, "]";
end;

eitherOr(?b?) -> ?ng?;
eitherOr = "[", [comment] ?cm?, "Either", [":"],
{[comment], nestedOrs} ?ors?, [endComment] ?ecm?, "]";
?ng? = "~[", ?cm?, nestedOrs(?ors?), ?ecm?, "]";
end;

nestedOrs(?ors?) -> ?g?;
nestedOrs = ( "[", [comment] ?cm?, "Or" ?first?, [":"], CG ?ng?,
[endComment] ?ecm?, "]", nestedOrs ?more?
| );
if (?first?= ) ?g? = ;
```

```
else ?g? = "~[", ?cm?, CG(?ng?), ?ecm?, "]", nestedOrs(?more?);
end; end;
```

قاعده مربوط به nestedOrs به صورت بازگشتی دنباله‌ای از صفر یا چند زمینه‌ی بولی از نوع or را پردازش می‌کند. اگر *b* شامل هیچ nestedOrs نباشد آنگاه eitherOr(b) باید [ ]~ باشد که مقدار آن false است. جمله‌ی CLIF متناظر (یعنی (or)) نیز طبق تعریف false است.

توضیح - دامنه‌ی کاربرد سورها در تمام زمینه‌های بولی باید با تودرتویی ترجمه‌ی آن‌ها به CGIF اصلی تعیین شود. هر برچسب تعریف‌کننده در یک زمینه از نوع If باید زمینه‌ی تودرتوی نوع Then را در دامنه‌ی کاربرد خود داشته باشد. هیچ کدام از دو زمینه‌ای که یک زمینه از نوع Either، Equivalence یا Equiv آن دو را به طور مستقیم دربرداشته باشد نباید در دامنه‌ی کاربرد زمینه‌ی دیگر باشد.

ب-۳-۵ مفهوم

تعریف: یک رشته‌ی *c* متشکل از چهار زیررشته که تمام آن‌ها قابل حذف بوده و عبارتند از: یک توضیح شروع *cm*، یک فیلد نوع، یک فیلد مرجع و یک توضیح انتهایی *ecm*.

فیلد مرجع *c* می‌تواند دربردارنده‌ی برچسب دنباله‌ی تعریف‌کننده با نشان‌گر دنباله‌ی *sqn* باشد. در این صورت فیلد نوع *c* باید تهی بوده، قبل از برچسب دنباله‌ی تعریف‌کننده می‌تواند «@every» قرار گیرد، و نباید هیچ مرجع یا گراف مفهومی در فیلد مرجع *c* وجود داشته باشد.

اگر هیچ *sqn* وجود نداشته باشد آنگاه فیلد نوع *c* یا باید دربردارنده‌ی یک عبارت نوع *tx* و یک علامت دو نقطه «:» یا دربردارنده‌ی یک مرجع اختیاری *ty* تحت عنوان یک *برچسب نوع* و یک علامت دو نقطه‌ی اختیاری باشد. همچنین فیلد ارجاعی *c* باید دربردارنده‌ی یک برچسب تعریف‌کننده اختیاری که نام CG آن *df* بوده (که قبل از آن می‌تواند «@every» قرار گیرد)، دنباله‌ای از صفر یا چند مرجع *rf* و یک گراف مفهومی *g* (که می‌تواند پوچ باشد) باشد. اگر تمام این گزینه‌ها حذف شوند مفهوم *c* باید رشته‌ی «[ ]» باشد.

ترجمه: یک سه‌تایی (*r,q,g*) متشکل از یک مرجع یا یک برچسب دنباله‌ی مقید *r*، یک سور *q* که باید «@every» یا رشته‌ی تهی باشد و یک گراف مفهومی g که باید حداقل دربردارنده‌ی یک مفهوم باشد.

```
concept = "[", [comment] ?cm?,
  ( (typeExpression ?tx?, ":"
| [["#" , "?"], CGname] ?ty?, [":"]),
  [["@every"] ?q?, "*", CGname ?df?], {reference} ?rf?, CG ?x?
| ["@every"] ?q?, "*", seqmark ?sqn?
 ), [endComment] ?ecm?, "]";
if (?sqn?~= ) ?r? = "?", ?sqn?; ?g1? = "[", ?cm?, "*", ?sqn?, ?ecm?];
elif (?df?~= ) ?r? = "?", ?df?; ?g1? = "[", ?cm?, "*", ?df?, ?ecm?];
if  (?rf?~= )  ?g2? = "[", ":", ?r?, ?rf?, "]"; end;
elif  (?rf?~= )  ?r? = first(?rf?);
?g2? = "[", ?cm?, ":", ?rf?, ?ecm?, "]";
else       ?df? = gensym(); ?r? = "?", ?df?;
```

```
?g1? = "[", ?cm?, "*", ?df?, ?ecm?, "]";
end;
if  (?tx?~= )  ?b? = first(typeExpression(?tx?));
?gx? = second(typeExpression(?tx?));
?g3? = substitute(?r?,?b?,?gx?);
elif(?ty?~= )?g3? = "(", ?ty?, ?r?, ")"; end;
if (?x?~= )  ?g4? = "[", CG(?x?), "]";
end;
?g? = ?g1?, ?g2?, ?g3?, ?g4?;
end;
```

چهار گزینه برای فیلد نوع مجاز است: یک عبارت نوع‌دار *tx*، یک برچسب هم‌مرجع مقید با پیشوند "#"، یک ثابت یا رشته‌ی تهی، یک علامت دونقطه پس از *tx* لازم و برای بقیه اختیاری است. قواعد بازنویسی ویژگی‌های مفهوم *c* را به چهار رشته انتقال می‌دهند. الحاق این رشته‌ها گراف مفهومی *g* را شکل می‌دهد بدین صورت که *g1* یک مفهوم وجودی است با یک برجسب تعریف‌کننده از *c* یا یک برچسب تولید شده با استفاده از *gensym*() است در صورتی که هیچ برچسب تعریف‌کننده یا مرجعی در *c* نباشد، *g2* یک مفهوم هم‌مرجع است در صورتی که مرجعی در *c* باشد، *g3* یا یک رابطه‌ی مفهومی با برچسب نوع *ty* است یا یک گراف مفهومی تولید شده از یک عبارت نوع *tx* و *g4* یک زمینه شامل هر گراف مفهومی غیرپوچ *x* CG است. توضیحات *cm* و *ecm* در اولین مفهوم غیرپوچ که باید *g1* یا *g2* باشد قرار می‌گیرند.

توضیح - به منظور تشریح ترجمه، جمله‌ی *A pet cat Yojo is on a mat* را می‌توان در CGIF گسترش‌یافته با دو گره‌ی مفهوم در دنباله‌ی کمان یک رابطه‌ی مفهومی به صورت زیر نمایش داد:

```
(On [@*x (Pet ?x) (Cat ?x): Yojo] [Mat])
```

برای تولید نمایش معادل عبارت فوق در CGIF اصلی مفاهیم از دنباله‌ی کمان حذف می‌شوند و به جای آن‌ها از مراجع برای پیوند دادن کمان‌ها به مفاهیم استفاده می‌شود. عبارت CGIF اصلی حاصل به صورت زیر است:

```
[: Yojo] (Pet Yojo) (Cat Yojo)

[*g00238] (Mat ?g00238) (On Yojo ?g00238)
```

نام Yojo مرجع مفهوم اول است و نام g00238 برای mat با استفاده از *gensym*() تولید شده است. بحث مربوط به عبارت نوع و ترجمه‌ی آن در بند ب-۳-۹ آمده است. *cg2cl*، CGIF اصلی را به نحو انتزاعی ترجمه می‌کند که با استفاده از عبارت CLIF زیر قابل بیان است:

```
(exists (g00238) (and (= Yojo Yojo) (Pet Yojo) (Cat Yojo)

(Mat ?g00238) (On Yojo ?g00238)))
```

یک مفهوم هم‌مرجع با تنها یک مرجع مانند [: Yojo] هیچ تأثیری بر ارزش درستی جمله نداشته و قابل حذف به‌وسیله یک کامپایلر بهینه‌ساز است مگر اینکه به عنوان یک نگهدارنده برای توضیحات مورد نیاز باشد.

ب-۳-۶ گراف مفهومي (CG)

تعریف: یک رشته‌ی *cg* متشکل از یک دنباله‌ی نامرتب از زیررشته‌هایی که مفاهیم، روابط مفهومی، عبارت‌های بولی و توضیحات را نمایش می‌دهند.

ترجمه: یک گراف مفهومی *g*.

```
CG(?cg?) -> ?g?;
CG = {concept | conceptualRelation | boolean | comment};
if (first(sortCG(?cg?)~= )
?g? = "~", "[", first(sortCG(?cg?)),
"~", "[", second(sortCG(?cg?), "]", "]";
else ?g? = second(sortCG(?cg?));
end; end;
```

sortCG(*cg*) باید زوج (*g1*,*g2*) باشد که در آن *g1* گراف مفهومی است که از تمام مفاهیم با سور عمومی در *cg* مشتق شده و *g2* گراف مفهومی است که از سایر مفاهیم، روابط مفهومی و توضیحات در *cg* مشتق شده است.

```
sortCG(?cg?) -> ?g1?,?g2?;
sortCG = ( (concept ?c? | conceptualRelation ?x?
| boolean ?x? | comment ?x?), sortCG ?rem?
| );
if (?c?= ) ?cg2? = CG(?x?));
elif (second(concept(?c?)) = "@every")
?cg1? = third(concept(?c?));
else   ?cg2? = third(concept(?c?));
end;
?g1? = ?cg1?, first(sortCG(?rem?)); ?g2? = ?cg2, second(sortCG(?rem?));
end;
```

توضیح: اگر هیچ مفهومی شامل سورهای عمومی در رشته‌ی ورودی وجود نداشته باشد آنگاه نتیجه باید یک رشته‌ی منفرد در CGIF اصلی باشد که از الحاق نتایج ترجمه‌ی مستقل گره‌ها بدست می‌آید. اما اگر رشته‌ی ورودی شامل مفاهیم عمومی باشد آنگاه رشته‌ی خروجی باید به صورت دو نفی تودرتو باشد. زمینه‌ی بیرونی باید شامل ترجمه‌های تمام مفاهیم عمومی و زمینه-ی درونی باید شامل ترجمه‌های سایر گره‌ها در ورودی باشد.

ب-۳-۷ رابطه‌ي مفهومي

تعریف: یک رشته‌ی *cr* که یک رابطه‌ی مفهومی معمولی یا یک کنش‌گر را نشان می‌دهد.

ترجمه: یک گراف مفهومی *g* که با استفاده از ordinaryRelation (*cr*) یا actor (*cr*) تعریف می‌شود.

```
conceptualRelation = ordinaryRelation | actor;
ordinaryRelation(?cr?) -> ?g?;
ordinaryRelation  =  "(", [comment] ?cm?, (["#", "?"], CGname) ?r?,
arcSequence ?s?, [endComment] ?ecm?, ")";
?g?  =  second(arcSequence(?s?)),
```

```
"(", ?cm?, ?r?, first(arcSequence(?s?)),
third(arcSequence(?s?)), ?ecm?, ")";
end;
```

اولین خط قاعده بازنویسی یک گراف مفهومی را از دنباله‌ی کمان s استخراج می‌کند. خط دوم توضیح شروع، برچسب نوع و دنباله‌ی کمان یک رابطه‌ی مفهومی را اضافه می‌کند. خط سوم نشان‌گر دنباله (در صورت وجود)، توضیح انتهایی و پرانتزهای بسته‌ی رابطه‌ی مفهومی را اضافه می‌کند.

توضیح - برای نمونه، رابطه‌ی مفهومی (On [Cat: Yojo] [Mat]) با استفاده از قواعد مربوط به روابط مفهومی، کمان‌ها، دنباله‌های کمان و مفاهیمی که برای تولید یک گراف مفهومی بیان شده در CGIF اصلی (مانند گراف مفهومی زیر) به کار می‌رود، به صورت زیر ترجمه می‌شود.

[: Yojo] (Cat Yojo) [*g00719] (Mat ?g00719) (On Yojo ?g00719)

ب-۳- ۸ متن

تعریف: یک زمینه‌ی *c* که هیچ زمینه‌ی دیگری آن را به‌طور مستقیم یا غیرمستقیم دربرنداشته باشد.

ترجمه: یک زمینه‌ی *cx*.

```
text(?c?) -> ?cx?;
text = "[", [comment] ?cm?, "Proposition", ":", [CGname] ?n?,
CG ?g?, [endComment] ?ecm?, "]";
?cx? = "[", ?cm?, "Proposition", ":", ?n?, CG(?g?), ?ecm?, "]";
end;
```

توضیح - CGIF برای پیمانه‌ها نحو صریحی فراهم نمی‌کند. به جای آن هر پیمانه‌ی CL باید بر اساس جدول الف-۲ ابتدا به یک متن در CGIF اصلی ترجمه شود. سپس نتیجه‌ی آن ترجمه باید با استفاده از تابع *cg2cl* تعریف شده در بند ب-۴ به یک متن در CGIF گسترش‌یافته ترجمه شود.

ب-۳-۹ عبارت نوع

تعریف: یک رشته‌ی *tx* شامل یک نام *n* و یک گراف مفهومی *g*.

ترجمه: یک زوج (*b*,*g*) متشکل از یک برچسب مقید *b* و یک گراف مفهومی *g*.

```
typeExpression(?tx?) -> ?b?,?g?;
typeExpression = "@", "*", CGname ?n?, CG ?g?;
?b? = "?", ?n?;
end;
```

اگر یک مفهوم *c* دارای یک عبارت نوع باشد آنگاه قواعد بازنویسی که concept(*c*) را مشخص می‌کنند از تابع substitute(?r?,?b?,?g?) به منظور جایگزینی مرجع *r* برای هر مورد از وقوع *b* در *g* استفاده می‌کنند.

توضیح - یک عبارت نوع متناظر است با یک عبارت لاندا[1] است که در آن نام n، پارامتر صوری را مشخص می‌کند و گراف مفهومی g بدنه‌ی عبارت است. اگر یک مفهوم c دربردارنده‌ی یک عبارت نوع باشد آنگاه قواعد ترجمه‌ای که c را پردازش می‌کنند باید مرجع مشتق شده از c را جایگزین هر مورد از وقوع برچسب مقید $n?$ در g کنند.

ب-٤ انطباق CGIF

نحو سه گویش منطق مشترک در این پیوست مشخص شده است که عبارتند از: یک نحو انتزاعی برای گراف‌های مفهومی، یک نحو محسوس برای CGIF اصلی و یک نحو محسوس برای CGIF گسترش‌یافته. هر سه‌ی این زبان‌ها گویش‌های منطبق بر منطق مشترک هستند، از این جهت که می‌توان هر جمله‌ی CL را به یک جمله در هر کدام از آن‌ها که از نظر معنایی هم‌ارز آن جمله باشد ترجمه کرد و در ضمن هر جمله متعلق به هر کدام از سه گویش را می‌توان به یک جمله در CL که از نظر معنایی معادل یکدیگر باشند ترجمه کرد. هم‌ارزی معنایی طبق تعریف برقرار است، بدین صورت که معنای هر جمله در CGIF گسترش‌یافته با ترجمه به یک جمله در CGIF اصلی، معنای هر جمله در CGIF اصلی با ترجمه به یک جمله در نحو انتزاعی CG و معنای هر جمله‌ی انتزاعی CG با ترجمه‌ی آن به نحو انتزاعی CL تعریف می‌شود.

به منظور نشان دادن انطباق کامل، این بند تابع *cl2cg* را تصریح می‌کند. این تابع باید هر جمله‌ی s در CL را به یک جمله‌ی *cl2cg*(s) در CGIF گسترش‌یافته ترجمه کند به‌طوری که ارزش درستی آن در تمام تفسیرهای CL یکسان باشد. برای اغلب عبارت‌های CL نگاشت به یک عبارت در CGIF گسترش‌یافته سرراست است. اما ترجمه‌ی اصطلاحات تابعی از CL به CGIF بیش از یک گام نیاز دارد. هر استفاده از یک تابع منطق مشترک را می‌توان به یک کنش‌گر ترجمه کرد که تابع را به همراه یک ارجاع به مفهومی که مرجع آن مقدار آن تابع است نمایش دهد. به منظور ترجمه‌ی دنباله‌ای از اصطلاحات CL به یک دنباله‌ی کمان در CGIF گسترش‌یافته، گره‌ی مفهوم باید دربردارنده‌ی گره‌ی کنش‌گر باشد.

برای نمونه، فرض کنید (F X1 X2) یک اصطلاح CLIF با عملگر F اعمال شده به آرگومان‌های X1 و X2 بوده و نام‌های X2 و X2 مقید به سور بوده در حالی که F مقید به سور نباشد. زمانی که این اصطلاح به‌وسیله cl2cg ترجمه شود تابع ()gensym باید به منظور تولید یک نام CG مانند g00592 استفاده شود. زمانی که «?» پیشوند این نام باشد این نام یک برچسب هم‌مرجع مقید خواهد بود که باید به عنوان کمان خروجی کنش‌گری که تابع F را نمایش می‌دهد استفاده شود. نتیجه‌ی ترجمه‌ی اصطلاح اصلی CLIF با استفاده از cl2cg باید (F ?X1 ?X2 | ?g00592) باشد. برچسب تعریف‌کننده g00592* باید مانند [g00592*] در یک مفهوم قرار گیرد. کنش‌گر نیز باید به عنوان یک گراف مفهومی تودرتو به صورت [(F X1 X2 | ?g00592) g00592*] در آن مفهوم قرار گیرد. این مفهوم باید زمانی که تابع *cl2cg* به اصطلاح تابعی اعمال می‌شود حاصل این تابع باشد و می‌تواند به عنوان یک کمان در یک دنباله‌ی کمان یک کنش‌گر یا رابطه‌ی مفهومی ظاهر شود.

1 - Lambda expression

از آنجا که مسند یک رابطه‌ی CL یا عملگر یک تابع CL می‌تواند یک اصطلاح تابعی باشد، باید از تبدیل یکسانی برای ترجمه‌ی مسند و عملگر به یک مفهوم استفاده کرد. برای نمونه، فرض کنید جمله‌ی ((F X1 X2) Y1 Y2)) یک جمله‌ی تجزیه ناپذیر CL باشد که مسند آن همان اصطلاح تابعی است که در مثال قبل ظاهر شد. بنابراین برچسب مقید «g00592?» که مقدار تابع را نمایش می‌دهد باید برچسب نوع رابطه‌ی مفهومی متناظر باشد. اگر هر دو آرگومان Y1 و Y2 در CL مقید به سور باشند رابطه‌ی مفهومی باید به صورت (#?g00592 ?Y1 ?Y2) باشد. به منظور تولید یک واحد نحوی منفرد به عنوان مقدار *cg2cl* این رابطه‌ی مفهومی باید درون مفهومی که اصطلاح تابعی را نمایش می‌دهد و بلافاصله قبل از «[» و به صورت [(#?g00592 ?Y1 ?Y2) (?g00592 | F X1 X2) g00592*] است، قرار گیرد. زمانی که تابع *cg2cl* به جمله‌ی تجزیه‌ناپذیر اصلی اعمال شود این مفهوم باید حاصل این تابع باشد. این مفهوم می‌تواند به صورت یک گره از یک گراف مفهومی که از ترجمه‌ی یک جمله‌ی CL بزرگتر که دربردارنده‌ی جمله‌ی تجزیه‌ناپذیر اصلی است بدست آید.

جدول ب-۱ عبارت CGIF گسترش‌یافته‌ای که *cg2cl*(**E**) را برای هر عبارت منطق مشترک **E** تعریف می‌کند مشخص می‌کند. برای اطمینان از اینکه محدودیت‌های CL برروی دامنه‌ی کاربرد سورها در ترجمه‌های صورت گرفته به‌وسیله *cg2cl* حفظ می‌شوند، از کروشه‌های زمینه (یعنی «[» و «]») برای دربرگرفتن ترجمه‌های عبارت‌های نوع E13 و E14 استفاده شده است. در بعضی موارد این کروشه‌ها ضروری نیستند و می‌توان آن‌ها را نادیده گرفت.

ستون اول جدول ب-۱ پیوندهایی را به سطرهای جدول معناشناسی منطق مشترک در بند ۶ نشان می‌دهد. ستون دوم از متازبان[1] و قراردادهای مورد استفاده در تعریف نحو انتزاعی CL استفاده می‌کند. ستون سوم آن متازبان را با نشانه‌گذاری مورد استفاده برای قواعد بازنویسی در بند ب-۱-۳-۲ ترکیب می‌کند. این ترکیب یک تابع *cg2cl* را تعریف می‌کند که هر جمله‌ی *s* از CGIF اصلی را به جمله‌ی *cg2cl*(*x*) از منطق مشترک که با آن هم‌ارز منطقی است ترجمه می‌کند.

جدول ب-۱ نگاشت نحو انتزاعی CL به نحو CGIF گسترش‌یافته

	اگر E یک عبارت CL به شکل زیر باشد	**آنگاه cl2cg(E) برابرخواهد بود با**
E1	یک عدد n	عدد n
E1	یک رشته‌ی نقل قول شده‌ی s	رشته‌ی نقل قول شده‌ی s
E1	یک نام قابل تفسیر n	اگر n شناسه‌ی CG نباشد باید بین علامت‌های نقل قول قرار گیرد، اگر n در سور یک جمله‌ی CL واقع شود باید با پیشوند "*" مشخص شود و اگر به یک سور مقید باشد باید با پیشوند "?"مشخص شود
E2	نشان‌گردنباله‌ی*S*	*S*

1 - Metalanguage

E3	یک دنباله‌ی اصطلاح <T1 … Tn> که با اصطلاح T1 شروع شود	یک دنباله‌ی کمان به صورت: *cl2cg*(T1) … *cl2cg*(Tn)
E4	یک دنباله‌ی اصطلاح T1 … Tn که با نشان‌گر دنباله‌ی T1 شروع شود	یک دنباله‌ی کمان به صورت: *cl2cg*(T2),…,*cl2cg*(Tn), *cl2cg*(T1)
E5	یک اصطلاح (O T1 … Tn)	یک مفهوم با نام تولید شده‌ی n که شامل یک کنش‌گر "[", "*", 'n', "(", *cl2cg*(O), *cl2cg*(T1, … Tn), ")", "|", "?", 'n', ")", "]" به صورت تودرتو باشد
	یک اصطلاح (cl:comment ‘string’ T)	یک کمان با یک توضیح و به صورت: "/*", 'string', "*/", cg2cl(T)
E6	یک تساوی (= T1 T2)	یک CG متشکل از یک، دو یا سه مفهوم. اگر T1 و T2 هر دو نام باشند یک مفهوم "[:",*cl2cg*(T1), *cl2cg*(T2), "]". اگر هر دو اصطلاح تابعی باشند سه مفهوم *cg2cl*(T1), "[", "?", 'n1', "?", 'n2', "]" *cg2cl*(T2) که در آن n1 نام تولید شده برای T1 و n2 نام تولید شده برای T2 است. اگر Ti (i = 1 یا i = 2) یک اصطلاح تابعی و اصطلاح دیگر Tj یک نام باشد آنگاه دو مفهوم "[", "?", *cl2cg*(Ti), 'ni', *cl2cg*(Tj), "]" که در آن ni نام تولید شده برای Ti است
E7	یک جمله‌ی تجزیه‌ناپذیر (P T1 … Tn)	یک CG متشکل از یک رابطه‌ی مفهومی یا یک رابطه. اگر P یک نام باشد آنگاه یک رابطه‌ی "(", *cl2cg*(P),*cl2cg*(T1 ...Tn), ")". اگر P یک اصطلاح تابعی باشد یک مفهوم *cl2cg*(P) به گونه‌ای که با درج رابطه‌ی مفهومی "(", 'n', *cl2cg*(T1 ... Tn), ")" (که در آن n نام تولید شده برای *cl2cg*(P) است) بلافاصله قبل از [تغییر یافته باشد.
E8	یک جمله‌ی بولی (not P)	یک نفی "~", "[", *cl2cg*(P), "]"
E9	یک جمله‌ی بولی (and P1 … Pn)	یک CG به صورت *cl2cg*(P1), ..., *cl2cg*(Pn)
E10	یک جمله‌ی بولی (or P1 … Pn)	یک CG به صورت زیر: [", "Either", "[", "Or", *cl2cg*(P1), "]", ..., "[", *cl2cg*(Pn), "]", "]"
E11	یک جمله‌ی بولی (if P Q)	یک CG به صورت زیر: "[", "If", *cl2cg*(P), "[", "Then", c*l2cg*(Q), "]", "]"
E12	یک جمله‌ی بولی (iff P Q)	یک CG به صورت زیر: "[", "Equiv", ":", "[", "Iff", *cl2cg*(P), "]", "[", “Iff”, *cl2cg*(Q), "]", "]"
	یک جمله (cl:comment ‘string’ P)	یک توضیح و یک CG به صورت زیر: "/*", 'string', "*/", *cl2cg*(P)
E13	یک جمله‌ی سوری (forall (N1 … Nn) B) که در	یک CG به صورت زیر:

	آن N1 تا Nn نام یا نشان‌گر دنباله هستند	"[", "[", "@every", "*", *cl2cg*(N1), "]", ..., "[", "@every", "*", *cl2cg*(Nn), "]", *cl2cg*(B), "]"
E14	یک جمله‌ی سوری (exists (N1 … Nn) B) که در آن N1 تا Nn نام یا نشان‌گر دنباله هستند	یک CG به صورت زیر: "[", "[", "*", *cl2cg*(N1), "]", ..., "[", "*", *cl2cg*(Nn), "]", *cl2cg*(B), "]"
	یک عبارت (cl:comment ‘string’ P)	یک توضیح به صورت "/*", 'string', "*/"
E17	یک عبارت (cl:imports N)	یک مفهوم به صورت زیر: "[", "cg_Imports", *cl2cg*(N), "]"
E18	یک پیمانه با نام N و لیست عدم شمول N1 … Nn و متن T	اگر M ترجمه به هسته‌ی CL به صورتی که در جدول الف-۲ مشخص شده باشد آنگاه یک متن به صورت زیر: "[", "Proposition", ":", *cl2cg*(M), "]"
E19	یک عبارت (cl:text T1 … Tn)	یک متن به صورت زیر: "[", "Proposition", *cl2cg*(T1 ... Tn), "]"
E20	(cl:text N T1 … Tn)	یک متن به صورت زیر: "[", "Proposition", ":", *cl2cg*(N), *cg2cl*(T1 ... Tn), "]"

به منظور تصریح ترجمه از CGIF گسترش‌یافته به CGIF اصلی، بند ب-۳ از ترکیبی از قواعد نحوی EBNF و قواعد بازنویسی مشخص شده در بند ب-۱-۳-۲ برای تعریف یک تابع ex2cor استفاده می‌کند. این تابع هر جمله‌ی s از CGIF گسترش‌یافته را به یک جمله‌ی CG(s) در CGIF اصلی ترجمه می‌کند که هم‌ارز منطقی s است.

پیوست پ
زبان علامت‌گذاري منطق مشترك گسترش‌پذير (XCL)
(الزامي)

پ-۱ مقدمه

XCL یک نشانه‌گذاری مبتنی بر XML برای منطق مشترک است. این زبان یک زبان مبادله به منظور ارتباط منطق مشترک در سرتاسر یک شبکه است. نگاشت نحو انتزاعی و معناشناسی CL به XML سرراست است.

پ-۲ نحو XCL

از آنجا که نحو لغوی XCL همانند خود XML است، این نحو با استفاده از یک مشخصه‌ی نوع سند[1] که به صورت الکترونیکی در دسترس است، توصیف شده است.

```
<!-- ..... -->
<!-- XML Common Logic 1.0 DTD ..... -->
<!-- file: xcl1.dtd
-->
```

DTD نسخه‌ی 1.0 برای منطق مشترک مبتنی بر XML

```
<!--
```

این DTD دارای شناسه‌های صوری عمومی زیر است:

"ISO/IEC 24707:2006//DTD XML Common Logic (XCL) 1.0//EN"
"-//purl.org/xcl//DTD XML Common Logic (XCL) 1.0//EN"

این DTD قابل استناد با هر یک از اعلآن‌های زیر است:

```
<!DOCTYPE text PUBLIC
"ISO/IEC 24707:2006//DTD XCL Markup Language//EN">
"xcl1.dtd">
<!DOCTYPE text PUBLIC
"-//purl.org/xcl//DTD XML Common Logic (XCL) 1.0//EN"
"xcl1.dtd">
```

در این اعلان‌ها شناسه‌ی سامانه‌ای («xcl1.dtd») در صورت لزوم به منظور مشخص کردن مکان DTD قابل سفارشی‌سازی است.

1 - Document Type Definition (DTD)

در صورتی که هر گونه تفاوتی بین نثر استاندارد ملی و این XCL DTD وجود داشته باشد نثر[1] استاندارد ملاک است.

-->

<!-- **توضیحات در DTD**

توضیحات این DTD که از عبارت‌های «باید» یا «نباید» استفاده می‌کنند الزامات اصولی این استاندارد هستند. توضیحاتی که از افعال «توصیه می‌شود» یا «توصیه نمی‌شود» استفاده می‌کنند پیشنهادهای این استاندارد ملی هستند. توضیحاتی که از افعال «پیشنهاد می‌شود» و «شایسته نیست» استفاده می‌کنند نیز به ترتیب تعریف‌کننده پیشنهادها و مخالفت‌های این استاندارد ملی هستند.

-->

<!-- یک شناسه‌ی منبع یکسان، بند ۳-۲۷ و [8] ملاحظه شود

-->

```
<!ENTITY % URI.datatype "CDATA" >
```

<!-- **پشتیبانی از فضای نامی XML**.....-->

<!-- URI فضای نام XML[2] برای XCL نسخه‌ی 1.0 عبارت است از:

"http://purl.org/xcl/1.0/"

-->

```
<!ENTITY XCL1.xmlns "http://purl.org/xcl/1.0/" >
```

<!-- **۱ نحو عمومی**.....-->

<!-- **۱-۱ مدل‌های محتوا**.....-->

```
<!ENTITY % Quantified.class
"( quantified | forall | exists )"
>
<!ENTITY % Boolean.class
"( boolean | and | or | implies | iff | not )"
>
<!ENTITY % Atomic.class
"( atomic | relation | equal )"
>
<!ENTITY % Sentence.class
"( %Quantified.class; | %Boolean.class; | %Atomic.class; )"
>
<!ENTITY % Comment.class
"comment"
```

1 - Prose
2 - XML Namespace URI

>

<-- ۱-۲ صفات.....—!>

<!-- **۱-۲-۱ صفات عام**

صفات زیر برای تمام انواع عناصر XCL تعریف شده‌اند (گرچه مشمول متن توصیفی درون یادداشت‌ها نیستند).

xmlns (اختیاری): تمام عناصر XCL دارای صفت احتیاری و تعریف شده‌ی xmlns هستند که مقدار ثابت و پیش‌فرض آن با URI فضای نامی XML برای XCL نسخه‌ی 1.0 تطابق دارد. پردازشگرهای XML می‌توانند به این صفت در صورتی که به‌طور صریح در نمونه‌ی مدرک وجود نداشته باشد به صورت ضمنی اشاره داشته باشند.

id (اختیاری): تمام عناصر XCL دارای صفت اختیاری و اعلان شده‌ی id هستند که مقدار آن باید با نام XML[1] (تعریف شده در ساخت ۵ [XML]) تطابق داشته باشند. مقدار ID در صورت وجود ابزاری برای شناسایی یک عنصر ویژه در یک مدرک XCL به صورت منحصر به فرد است. توجه کنید که این در سطح نحو XML عمل می‌کند و هیچ اهمیت معنایی در XCL ندارد. هر مقدار id باید در یک مدرک XCL منحصر به فرد باشد.

-->

```
<!ENTITY % XCL.xmlns.attrib
"xmlns %URI.datatype; #FIXED '&XCL1.xmlns;'"
>
<!ENTITY % id.attrib
"id ID #IMPLIED"
>
<!ENTITY % Common.attrib
"%XCL.xmlns.attrib;
%id.attrib;"
>
```

<-- **۱-۲-۱ صفت CL dialect**.....--!>

<!--

نام:	dialect
URI:	http://purl.org/xcl/1.0/#dialect
اعلان:	http://purl.org/xcl/1.0/#dialect-xcl http://purl.org/xcl/1.0/#dialect-clif http://purl.org/xcl/1.0/#dialect-cgif
برچسب:	CL Dialect
توصیف:	یک شناسه برای گویش منطق مشترک محتوای عنصر. بند ۳-۸ و ۷-۱ این استاندارد ملی ملاحظه شود.

-->

1 - XML Name

ملاحظات: <!--

از صفت dialect به منظور نشان دادن گویش محتوای عنصر دارای این صفت استفاده می‌شود. dialect یک صفت پیوند دهنده است که مقدار آن (یک ارجاع URI) حاوی ارجاعی است به یکی از سه مجموعه‌ی ثابت شناسه‌های گویش منطق مشترک:

http://purl.org/xcl/1.0/#dialect-xcl
http://purl.org/xcl/1.0/#dialect-clif
http://purl.org/xcl/1.0/#dialect-cgif

برای سایر نمایش‌های محسوس نحو توصیه می‌شود یک URI مناسب نشان‌دهنده‌ی گویش استفاده شود. برای تمام عناصر XCL که صفت dialect برای آن‌ها تعریف شده غیبت آن نشان‌دهنده‌ی مقدار پیش‌فرض است: گویش XCL تعریف شده به‌وسیله این DTD.
توجه کنید وجود یک صفت dialect هر صفت dialect در عناصر پدر[1] را باطل می‌کند. به هر حال، چنین تصادم‌های گویشی بین پدر و فرزند شایسته نیست.
این صفت برای عناصر <text>، <module>، <import> و <phrase> اعلان شده است.

مثال:

```
    <text dialect="http://purl.org/xcl/1.0/#dialect-clif">
    (forall ex:romanceNovel ((x man)) (exists ((y woman))
    (and (loves x y) (not (loves y x))) ))
    </text>
-->
<!ENTITY XCL.dialect "http://purl.org/xcl/1.0/#dialect-xcl" >
<!ENTITY CLIF.dialect "http://purl.org/xcl/1.0/#dialect-clif" >
<!ENTITY CGIF.dialect "http://purl.org/xcl/1.0/#dialect-cgif" >
<!ENTITY % dialect.attrib
"dialect %URI.datatype; '&XCL.dialect;'"
>
```

<-- ۱-۳ توضیحات.....--!>

نام: comment <!--

URI: http://purl.org/xcl/1.0/#comment

برچسب: Comments

توصیف: یک توضیح درج می‌کند. هر عنصر XCL می‌تواند شامل عناصر <comment> باشد و این عناصر برای آن عنصر حکم توضیح را دارند. بند ۶-۱-۱-۳ این استاندارد ملاحظه شود.

1 - Parent

-->

ملاحظات: <!--

زمانی که پردازش XML خوش ترکیب قابل قبول باشد (بند مربوط به انطباق XCL ملاحظه شود) عناصر <comment> می‌توانند هر متنی را دربرداشته باشند، می‌توانند محتوای مرکب باشند و می‌توانند هر صفت تعریف شده به‌وسیله کاربر را داشته باشند. این صفات به‌وسیله پردازشگرهای منطقی نادیده گرفته می‌شوند، اما خود این صفات و مکان آن‌ها نسبت به سایر عناصر باید در کاربردهای XCL منطبق بر این استاندارد حفظ شود. توضیحاتی که درون توضیحات دیگر واقع شوند به عنوان توضیح برای توضیح در نظر گرفته می‌شوند. در بسیاری موارد مدل‌های محتوای XCL توضیحات را به عنوان آخرین فرزندان عنصر پدر دربرمی‌گیرند.

توجه کنید که نشانه‌گذاری XCL داخل یک توضیح بخشی از عنصر XCL دربردارنده‌ی آن محسوب نمی‌شود و باید به شیوه‌ی مناسبی محو شود.

در شرایطی که نشانه‌گذاری با توضیحات غنی مورد انتظار و در عین حال XCL معتبر مورد نیاز است توضیحات باید شامل یک پیوند با استفاده از صفت href به یک منبع مستندات خارجی باشند:

```
<comment href="http://www.acme.com/docs/sec7.html"/>
```

اگر هم محتوای عنصر و هم صفت href موجود باشند، صفت href اختیاری در نظر گرفته می‌شود بدین معنی که پیمایش پیوند به منظور تعیین محتوای توضیح ضروری نیست.

با اعلآن‌های مناسب فضای نامی XML می‌توان هستار پارامتر %Comment.class; را دوباره اعلان نمود تا دربردارنده‌ی محتوای XML جایگزین باشد، مانند XHTML یا DocBook.

```
<!ENTITY % Comment.class
"( xhtml:div | comment )"
>
```

-->

```
<!ENTITY % Comment.content
"( #PCDATA | %Comment.class; )*"
>
<!ELEMENT comment %Comment.content; >
<!ATTLIST comment
%Common.attrib;
href %URI.datatype; #IMPLIED
>
```

<-- ۲ عناصر بالاترین سطح.....--!>

<!-- **۲-۱ عنصر سند XML[1]**.....-->

نام: text <!--

URI: http://purl.org/xcl/1.0/#text

برچسب: XCL document element

توصیف: همانند جداکننده‌های یک سند XCL برای احاطه کردن هر قطعه از محتوای XCL استفاده می‌شود. متن درون این عنصر باید XCL معتبر باشد و لازم نیست یک پیمانه (هستان‌شناسی) باشد. برای جزئیات مربوط به قیود خوش ترکیبی[2] به بند انطباق XCL مراجعه شود (به بند ۶-۱-۱-۱ این استاندارد ملی مراجعه شود).

-->

ملاحظات: <!--

صفات:

xml:base (اختیاری): URI پایه‌ی مدرک را نشان می‌دهد.

dialect (اختیاری): توضیح ۱-۲-۲ این بند ملاحظه شود. در صورتی که به طور صریح مشخص نشده باشد این صفت به صورت پیش‌فرض به مقداری که نحو XCL نسخه‌ی 1.0 را نشان می‌دهد مقداردهی می‌شود.

href (اختیاری): به منظور تخصیص یک «نام واردسازی[3]» به یک متن استفاده می‌شود. این صفت ارجاعی به یک URI یا یک IRI است و اغلب با فضای نامی پیش‌فرض xmlns و/یا URL مدرک دربردارنده‌ی آن یکسان است، هر چند این تطبیق نام‌گذاری ضروری نیست. هیچ گونه ارتباط منطقی بین نام‌های مبتنی بر URL یا ساختار فضای نامی XML آن‌ها مفروض نیست. بنابراین استفاده از یک مرجع URI شامل یک ID قطعه[4] به منظور نام‌گذاری یک متن قابل قبول است.

فرزندان:

یک یا چند عنصر <module>، <phrase> و/یا <comment> به هر ترتیبی.

مثال:

```
<text dialect="http://purl.org/xcl/1.0/#dialect-xcl">
<phrase>
   ...
```

1 - XML Document Element
2 - Well-formedness constraints
3 - Importing name
4 - Fragment

```
        </phrase>
        </text>
-->

<!ENTITY % Text.content
"( module | phrase | %Comment.class; )*"
>
<!ELEMENT text %Text.content; >
<!ATTLIST text
%Common.attrib;
xml:base %URI.datatype; #IMPLIED
%dialect.attrib;
>
```

<-- ۲-۲ پیمانه‌های XCL .....--!>

نام: module <!--

URI: http://purl.org/xcl/1.0/#module

برچسب: XCL Module

توصیف: نشان می‌دهد که محتوای درون آن به عنوان یک پیمانه‌ی XCL در نظر گرفته می‌شود. بند ۶-۱-۱-۴ این استاندارد ملی ملاحظه شود.

-->

ملاحظات: <!--

یک پیمانه‌ی XCL بخشی از نشانه‌گذاری XCL است که دارای صفات ویژه‌ای بوده و قیود اضافی مشخصی را برقرار می‌کند. یک مدرک XCL می‌تواند تنها شامل یک پیمانه باشد. بنابراین این عنصر در صورت وجود تنها فرزند عنصر مدرک <text> خواهد بود و در این صورت استفاده از <module> برای مشخص کردن آن ضروری نیست. صفت href عنصر <module> ابزاری برای شناسایی یک پیمانه است. روش دیگر استفاده از صفت id به همراه URI پایه‌ی مدرک است. این استاندارد ملی هیچ کدام از این دو روش را بر دیگری برتر نمی‌داند چرا که این موضوع به کاربرد وابسته است. مزیت روش دوم مقید کردن URI پیمانه به URI پایه‌ی مدرک است (این موضوع می‌تواند به عنوان یک قابلیت یا یک نقص مطرح باشد) که البته متّکی به وجود صفت xml:base برای عنصر <text> است و در صورت عدم وجود آن URI پایه دلخواه درنظرگرفته می‌شود.

صفات:

href (اختیاری): به منظور تخصیص یک «نام واردسازی[1]» به یک پیمانه استفاده می‌شود. این صفت ارجاعی به یک URI است و اغلب با فضای نامی پیش‌فرض xmlns و/یا URL

1 - Importing name

مدرک دربردارنده‌ی آن یکسان است، هر چند این تطبیق نام‌گذاری ضروری نیست. هیچ گونه ارتباط منطقی بین نام‌های مبتنی بر URL یا ساختار فضای نامی XML آن‌ها مفروض نیست. بنابراین استفاده از یک مرجع URI شامل یک ID قطعه به منظور نام‌گذاری یک هستان‌شناسی قابل قبول است.

یک مدرک می‌تواند شامل چندین پیمانه باشد که نام آن‌ها ارجاعات URI شکل گرفته از URI خود مدرک است (به‌طور شهودی، با نام واردسازی هستان‌شناسی می‌توان همانند سایر نام‌ها در یک فضای نامی برخورد کرد بدون اینکه نیاز باشد این نام URI کامل یک فضای نامی باشد. بدین ترتیب، یک هستان‌شناسی می‌تواند با یک هستان‌شناسی دیگر به عنوان یک هستار در فضای نامی خود برخورد کند.)

dialect (اختیاری): توضیح بند ۱-۲-۳ از این بند ملاحظه شود.

مثال:

```
<module href="http://www.acme.com/vocab/#agent-vocab">
...
</module>
-->

<!ENTITY % Module.content
"( import | exclude | phrase | %Comment.class; )*"
>
<!ELEMENT module %Module.content; >
<!ATTLIST module
   %Common.attrib;
href %URI.datatype; #REQUIRED
%dialect.attrib;
>
```

<-- ۲-۳ عنصر Phrase.....--!>

<!-- نام: phrase

URI: http://purl.org/xcl/1.0/#phrase

برچسب: Phrase

توصیف: برای نشان دادن یک جمله در بالاترین سطح در XCL استفاده می‌شود. تنها عنصر فرزند این عنصر باید یک عنصر جمله باشد.

-->

ملاحظات: <!--

صفات:

dialect (اختیاری): توضیح بند ۱-۲-۲ از این بند ملاحظه شود.

مثال:

```
        <phrase>
        TBD
        </phrase>
-->

<!ENTITY % Phrase.content
"( %Sentence.class;, %Comment.class;* )"
>
<!ELEMENT phrase %Phrase.content; >
<!ATTLIST phrase
%Common.attrib;
%dialect.attrib;
>
```

<-- ۲-۴ عنصر Import.....--!>

نام: import
<!--

URI: http://purl.org/xcl/1.0/#import

برچسب: Import

توصیف: یک عنصر پیوند دهنده که به منظور وارد کردن یک پیمانه‌ی XCL به پیمانه‌ای دیگر استفاده می‌شود. مقدار صفت باید ارجاعی به یک شناسه‌ی یک پیمانه‌ی XCL باشد. بند ۶-۱-۱-۵ این استاندارد ملی ملاحظه شود.

-->

ملاحظات: <!—

صفات:

dialect (اختیاری): یک عنصر <import> می‌تواند چنین صفتی داشته باشد. این صفت گویشی که منبع باید در آن کدگذاری شود را نشان می‌دهد. برنامه‌های منطبق بر این استاندارد که منبع XCL را کدگذاری شده در گویشی غیر از گویش نشان داده شده بیابند مجازند با این وضعیت به صورت یک وضعیت خطا برخورد کنند یا منبع را به گویش نشان داده شده ترجمه کنند. هر خطایی که در حین ترجمه تولید شود باید به عنوان یک وضعیت خطای گویشی گزارش شود.

فرزندان:

صفر یا چند عنصر <comment>.

مثال:

```
        <import href="http://www.acme.com/vocab/agent-vocab.xcl"/>
-->

<!ENTITY % Import.content
"( %Comment.class;* )"
>
<!ELEMENT import %Import.content; >
<!ATTLIST import
%Common.attrib;
href %URI.datatype; #REQUIRED
%dialect.attrib;
>
```

<- ۲-۵ عنصر Exclude.....-->

<!--

نام: exclude

URI: http://purl.org/xcl/1.0/#exclude

برچسب: Module Exclusions

توصیف: شامل دنباله‌ای از یک یا چند عنصر <term> به عنوان پوشش‌هایی[1] برای مجموعه‌ای از نام‌های CL است که به عنوان مجموعه‌ی عدم شمول پیمانه درنظرگرفته می‌شوند. این مجموعه هر نامی در متن که به‌طور صریح از جهان محلی پیمانه بیرون نگه داشته شده باشد را شامل می‌شود. بند ۶-۱-۱-۴ این استاندارد ملی ملاحظه شود.

-->

ملاحظات: <!--

هدف از استفاده از این عنصر این است که نشان داده شود این اصطلاحات چیزی را در هستان‌شناسی کنونی نشان نمی‌دهند، یعنی آن‌ها عملگرهای غیرگفتمانی هستند که مشمول حوزه‌ی محلی عبارت‌های حاوی سور نمی‌شوند. این موضوع به یک هستان‌شناسی اجازه می‌دهد کنترل دقیقی بر اندازه‌ی حوزه‌ی گفتمان خود داشته باشد. با نام‌هایی که در چنین سرایندی نباشند به عنوان مشخص کننده‌ی اصطلاحات در هستان‌شناسی برخورد می‌شود.

1 - Wrappers

اصطلاحات غیرگفتمانی تنها در رابطه یا تابع تجزیه‌ناپذیرها یا اصطلاحات در این پیمانه قابل استفاده هستند و نمی‌توان در این پیمانه از آن‌ها به عنوان آرگومان استفاده کرد یا آن‌ها را به سور مقید نمود. توجه کنید که یک اصطلاح مجاز است در یک هستان‌شناسی غیرگفتمانی و در هستان‌شناسی دیگر معنی‌دار باشد. تخصیص وضعیت غیرگفتمانی به یک اصطلاح به‌صورت محلی در این پیمانه صورت می‌گیرد.
این عنصر هیچ صفت یا عنصر فرزندی ندارد.

مثال:

```
        <exclude>
        <term name=”cow”>
        <term name=”pig”>
        <term name=”mouse”>
        </exclude>
-->
<!ENTITY % Exclude.content
"( term )*"
>
<!ELEMENT exclude %Exclude.content; >
<!ATTLIST exclude
%Common.attrib;
>
```

<-- ۳ عناصر جمله.....-->

<-- ۳-۰-۱ صفت syntaxType.....-->

<!-- نام: syntaxType

URI: http://purl.org/xcl/1.0/#syntaxType

برچسب: Syntax Type

توصیف: شکل منطقی یک جمله‌ی عمومی[1] را نشان می‌دهد. بندهای ۳-۱-۳ و ۳-۲-۶ از این بند ملاحظه شود.

-->

ملاحظات: <!--

جمله‌های سوری و بولی را می‌توان به صورت «عمومی» و با یک صفت اختیاری syntaxType بیان کرد. مقدار این صفت یک ارجاع URI است که شکل منطقی مناسب را نشان می‌دهد. پیوند syntaxType به منظور گسترش‌پذیری فراهم شده است تا بدین ترتیب امکان گنجاندن شکل‌ها و الگوهای نحوی منطقی جایگزین مانند سورهای عددی یا عملگرهای بولی جایگزین

1 - Generic

مانند یک ترکیب عطفی دودویی یا ضربه شِفِر[1] وجود داشته باشد. هم‌اکنون چنین گسترش-هایی تعریف نشده‌اند.

پیوند syntaxType به همراه سور XCL معمولی و عناصر جمله‌ی بولی که می‌توان به آن‌ها به عنوان کوته‌نوشت‌های برای عناصر جمله‌ی عمومی با انواع نحوی تعریف شده در XCL نگاه کرد استفاده نمی‌شود. برای نمونه، عنصر زیر را درنظربگیرید:

```
<forall>
...
</forall>
```

عنصر فوق شکل مختصر شده‌ی عنصر زیر است:

```
<quantified syntaxType="http://purl.org/xcl/1.0/#forall">
...
</quantified>
```

توجه کنید صفت dialect برای عناصر جمله اعلان نمی‌شود.

-->

```
<!ENTITY % syntaxType.attrib
"syntaxType %URI.datatype; #IMPLIED"
>
```

<!-- **۳-۰-۲ صفت logicalFormOf**.....-->

<!--

نام: logicalFormOf

URI: http://purl.org/xcl/1.0/#logicalFormOf

برچسب: Logical-Form-Of

توصیف: یک صفت اختیاری برای تمام عناصر جمله‌ی XCL که مقدار آن یک ارجاع URI است و یک رده‌ی نحوی در یک مشخصه‌ی زبان خارجی را نشان می‌دهد.

-->

<!--

ملاحظات:

هدف از این صفت این است که نشان داده شود جمله‌ی مورد نظر نتیجه‌ی یک ترجمه با حفظ محتوا به جمله‌ی XCL با شکل نحوی نشان داده شده است. توصیه می‌شود صفت logicalFormOf رده‌های مناسب زیرعبارت‌های عبارت خارجی را نشان دهد.

برای یک قطعه XCL داده شده تنها یک ترجمه‌ی «خارجی» می‌توان مشخص نمود. برای نشان دادن اینکه یک جمله‌ی منفرد XCL شکل منطقی دو نحو متمایز خارجی است لازم

1 - Scheffer stroke

است این جمله دو بار نوشته شود و برای هر مشخصه‌ی خارجی مقدار مناسبی برای صفت logicalFormOf مشخص شود (این تکرار را می‌توان با استفاده از استانداردهای XML خارج از XCL نشان داد). در حال حاضر هیچ نگاشت خارجی این چنینی تعریف نشده است.

مثال:

یک قاعده هُرن[1] می‌تواند به صورت یک دلالت ضمنی با سور عمومی بین یک ترکیب عطفی و یک تجزیه‌ناپذیر در XCL ظاهر شود. در این صورت مقدار صفت logicalFormOf به عنصر پدر آن یعنی <forall> اعمال می‌شود. مقدار این صفت ممکن است http://acme.com/rulespec/#hornrule باشد. مقدار این صفت برای ترکیب عطفی و تجزیه‌ناپذیر داخل این عنصر نیز ممکن است به صورت http://acme.com/rulespec/#body و http://acme.com/rulespec/#head باشد.

```
<forall logicalFormOf="http://acme.com/rulespec/#hornrule">
...
</forall>
-->

<!ENTITY % logicalFormOf.attrib
"logicalFormOf %URI.datatype; #IMPLIED"
>
```

<-- ۳-۱ عناصر جمله‌ی سوری.....—!>

```
<!ENTITY % Quantified.content
"( guard?, var+, %Sentence.class;, %Comment.class;* )"
>
```

<-- ۳-۱-۱ عنصر forall.....--!>

<!--

نام: forall

URI: http://purl.org/xcl/1.0/#forall

برچسب: Universal Quantifier

توصیف: یک جمله‌ی سوری را دربرمی‌گیرد و سور عمومی را نشان می‌دهد. بند ۶-۱-۷-۱ از این استاندارد ملی ملاحظه شود.

-->

```
<!ELEMENT forall %Quantified.content; >
<!ATTLIST forall
```

1 - Horn rule

```
%Common.attrib;
%logicalFormOf.attrib;
syntaxType %URI.datatype; #FIXED 'http://purl.org/xcl/1.0/#forall'
>
```

<!-- ۳-۱-۲ عنصر exists.....--!>

<!-- نام: exists

URI: http://purl.org/xcl/1.0/#exists

برچسب: Existential Quantifier

توصیف: یک جمله‌ی سوری را دربرمی‌گیرد و سور وجودی را نشان می‌دهد. بند ۶-۱-۷-۱ این استاندارد ملی ملاحظه شود.

-->

```
<!ELEMENT exists %Quantified.content; >
<!ATTLIST exists
%Common.attrib;
%logicalFormOf.attrib;
syntaxType %URI.datatype; #FIXED 'http://purl.org/xcl/1.0/#exists'
>
```

<!-- ۳-۱-۳ جمله‌ی سوری - شکل عمومی.....--!>

<!-- نام: quantified

URI: http://purl.org/xcl/1.0/#quantified

برچسب: Quantified Sentence – Generic Form

توصیف: بند ۶-۱-۱-۷ این استاندارد ملی و بند ۳-۰-۱ از این بند ملاحظه شود.

-->

```
<!ELEMENT quantified %Quantified.content; >
<!ATTLIST quantified
%Common.attrib;
sort NMTOKEN #IMPLIED
href %URI.datatype; #REQUIRED
%logicalFormOf.attrib;
%syntaxType.attrib;
>
```

<!-- ۳-۱-۴ عنصر Guard.....--!>

<!-- نام: guard

URI: http://purl.org/xcl/1.0/#guard

برچسب: Quantifier Guard

توصیف: این عنصر اختیاری دربردارنده‌ی یک عنصر منفرد <term> است که در

صورت وجود سور حکم محافظ آن را دارد. بند الف-۲-۳-۸ این استاندارد ملی و بند ۳-۱ از این بند ملاحظه شود.

-->

ملاحظات: <!--

عنصر <guard> تنها درون عناصر سور استفاده می‌شود و در صورت وجود باید اولین فرزند باشد.

این عنصر هیچ صفتی ندارد.

فرزندان:

یک عنصر ضروری <term> که پس از آن صفر یا چند عنصر <comment> آمده باشد.

مثال:

```
<forall>
<guard>
   <term name="cow"/>
</guard>
TBD
</forall>
-->

<!ENTITY % Guard.content
"( term, %Comment.class;* )"
>

<!ELEMENT guard %Guard.content; >
<!ATTLIST guard
%Common.attrib;
>
```

<-- ۳-۱-۵ عنصر Variable.....--!>

نام: var <!--

URI: http://purl.org/xcl/1.0/#var

برچسب: Variable

توصیف: یک نام مقید به یک سور را دربرمی‌گیرد. این نام باید یک نام گفتمانی باشد.

-->

ملاحظات: <!--

صفات:

sort (اختیاری): مقدار این صفت باید یک نام نشان‌دهنده‌ی نوع متغیر باشد. این نام می‌تواند یک نام گفتمانی باشد.

فرزندان:

صفر یا چند عنصر <comment>.

مثال:

```
        <var name=”person” />
-->

<!ENTITY % Var.content
"( %Comment.class;* )"
>

<!ELEMENT var %Var.content; >
<!ATTLIST var
%Common.attrib;
name NMTOKEN #REQUIRED
sort NMTOKEN #IMPLIED
>
```

<!-- ۳-۲ عناصر جمله‌ی بولی.....-->

ملاحظات: <!--

```
<and> </and>
<or> </or>
<not> </not>
<implies> </implies>
<iff> </iff>
```

تمام عناصر فوق جمله‌های بولی را دربرمی‌گیرند و تابع درستی[1] را نشان می‌دهند. شکل عمومی جایگزین به صورت زیر است:

```
<boolean syntaxType="http://purl.org/xcl/1.0/#and">
...
</boolean>
```

1 - Truth function

از بین این عناصر تنها عنصر <implies> حسّاس به ترتیب است. طبق قرارداد، فرزند اول مقدم و فرزند دوم نتیجه است. در XCL استفاده از عنصر <role> برای جمله‌های بولی ضروری نیست.

صفات:

syntaxType (اختیاری): بند ۳-۰-۱ از این بند ملاحظه شود. توجه کنید برای عناصر جمله‌های غیرعمومی (مانند <and>) صفت syntaxType با یک مقدار ثابت و پیش‌فرض مطابق با شناسه‌ی URI مربوط به نوع عنصر اعلان می‌شود.

logicalFormOf (اختیاری): بند ۳-۰-۲ از این بند ملاحظه شود.

فرزندان:

هر عنصر تعریف‌کننده یک جمله‌ی بولی می‌تواند جمله‌هایی از هر نوع را به عنوان عنصر فرزند داشته باشد. عناصر <and> و <or> می‌توانند هر تعداد (حتی صفر) فرزند داشته باشند. عنصر <not> به‌طور دقیق یک فرزند و عناصر <implies> و <iff> به‌طور دقیق دو فرزند دارند.

صفر یا چند عنصر <comment> نیز ممکن است پس از این فرزندان قرار گیرند.

-->

```
<!ENTITY % Boolean.content
"( %Sentence.class; | %Comment.class; )*"
>
```

<!-- ۳-۲-۱ عنصر and .....-->

نام: and <!--

URI: http://purl.org/xcl/1.0/#and

برچسب: Conjunction

توصیف: بند ۶-۱-۱-۸ از این استاندارد ملی ملاحظه شود.

-->

```
<!ELEMENT and %Boolean.content; >
<!ATTLIST and
%Common.attrib;
%logicalFormOf.attrib;
syntaxType %URI.datatype; #FIXED 'http://purl.org/xcl/1.0/#and'
>
```

<!-- ۳-۲-۲ عنصر or .....-->

نام: or <!--

URI: http://purl.org/xcl/1.0/#or

برچسب: Disjunction

توصیف: بند ۶-۱-۱-۸ از این استاندارد ملی ملاحظه شود.

-->

<!ELEMENT or %Boolean.content; >
<!ATTLIST or
%Common.attrib;
%logicalFormOf.attrib;
syntaxType %URI.datatype; #FIXED 'http://purl.org/xcl/1.0/#or'
>

<!-- ۳-۲-۳ عنصر not.....-->

<!-- نام: not

URI: http://purl.org/xcl/1.0/#not

برچسب: Negation

توصیف: بند ۶-۱-۱-۸ از این استاندارد ملی ملاحظه شود.

-->

<!ENTITY % Not.content
"(%Sentence.class;, %Comment.class;*)"
>
<!ELEMENT not %Not.content; >
<!ATTLIST not
%Common.attrib;
%logicalFormOf.attrib;
syntaxType %URI.datatype; #FIXED 'http://purl.org/xcl/1.0/#not'
>

<!-- ۳-۲-۴ عنصر implies.....-->

<!-- نام: implies

URI: http://purl.org/xcl/1.0/#implies

برچسب: Implication

توصیف: بند ۶-۱-۱-۸ از این استاندارد ملی ملاحظه شود.

-->

<!ENTITY % ImpliesIff.content
"((%Sentence.class;, %Sentence.class;), %Comment.class;*)"
>
<!ELEMENT implies %ImpliesIff.content; >
<!ATTLIST implies

```
%Common.attrib;
%logicalFormOf.attrib;
syntaxType %URI.datatype; #FIXED 'http://purl.org/xcl/1.0/#implies'
>
```

<!-- ۳-۲-۵ عنصر iff.....-->

```
<!--
```
نام: iff

URI: http://purl.org/xcl/1.0/#iff

برچسب: Biconditional

توصیف: بند ۶-۱-۱-۸ از این استاندارد ملی ملاحظه شود.

```
-->

<!ELEMENT iff %ImpliesIff.content; >
<!ATTLIST iff
%Common.attrib;
%logicalFormOf.attrib;
syntaxType %URI.datatype; #FIXED 'http://purl.org/xcl/1.0/#iff'
>
```

<!-- ۳-۲-۶ جمله‌ی بولی - شکل عمومی.....-->

```
<!--
```
نام: boolean

URI: http://purl.org/xcl/1.0/#boolean

برچسب: Boolean Sentence – Generic Form

توصیف: بند ۶-۱-۱-۸ از این استاندارد ملی و بند ۳-۲ از این بند ملاحظه شود.

```
-->

<!ELEMENT boolean %Boolean.content; >
<!ATTLIST boolean
%Common.attrib;
sort NMTOKEN #IMPLIED
href %URI.datatype; #REQUIRED
%syntaxType.attrib;
>
```

<!-- ۳-۳ عناصر جمله‌ی تجزیه‌ناپذیر.....-->

ملاحظات:

```
<!--
```
جمله‌ی تجزیه‌ناپذیر یک رابطه‌ی درست بین چند آرگومان را نشان می‌دهد. عنصر فرزند اول همواره رابطه را نشان می‌دهد. این رابطه می‌تواند یک نام غیرگفتمانی باشد. در XCL آرگومان‌های یک رابطه را می‌توان به دو شکل مشخص کرد که عبارتند از: یک لیست مرتب یا

یک مجموعه. در شکل دوم از یک صفت syntaxType برای عنصر <atomic> که مقدار صفت href آن http://purl.org/xcl/1.0/#roleset است استفاده می‌شود.

صفات:

syntaxType (اختیاری): بند ۳-۰-۱ از این بند ملاحظه شود.

logicalFormOf (اختیاری): بند ۳-۰-۲ از این بند ملاحظه شود.

فرزندان:

یک عنصر ضروری <relation> و پس از آن یا صفر یا چند عنصر <term> یا صفر یا چند عنصر <role> می‌آید. پس از این عناصر صفر یا چند عنصر <comment> قرار می‌گیرد.

مثال:

```
<atomic>
TBD
</atomic>
-->
```

<!-- ۳-۳-۱ عنصر Atomic.....-->

نام: atomic <!--

URI: http://purl.org/xcl/1.0/#atomic

برچسب: Atomic Sentence

توصیف: بند ۶-۱-۱-۹ از این استاندارد ملی ملاحظه شود.

-->

```
<!ENTITY % Atomic.content
"( relation, ( term* | role* ), %Comment.class;* )"
>
<!ELEMENT atomic %Atomic.content; >
<!ATTLIST atomic
%Common.attrib;
href %URI.datatype; #IMPLIED
%logicalFormOf.attrib;
%syntaxType.attrib;
>
```

<!-- ۳-۳-۲ عنصر Relation.....-->

نام: relation <!--

URI: http://purl.org/xcl/1.0/#relation

برچسب: Relation

توصیف: یک عنصر <term> را دربرمی‌گیرد. نام می‌تواند غیرگفتمانی باشد.

-->

<!-- **ملاحظات:**

این عنصر هیچ صفتی ندارد.

فرزندان:

یک عنصر <term> که پس از آن صفر یا چند عنصر <comment> قرار گیرد.

مثال:

```
<relation>
TBD
</relation>
-->

<!ENTITY % Relation.content
"( term, %Comment.class;* )"
>
<!ELEMENT relation %Relation.content; >
<!ATTLIST relation
%Common.attrib;
>
```

<!-- **۳-۳-۳ عنصر Equal**.....-->

<!-- نام: equal

URI: http://purl.org/xcl/1.0/#equal

برچسب: Equality

توصیف: بند الف-۲-۳-۴ از این استاندارد ملی ملاحظه شود.

-->

<!-- **ملاحظات:**

این عنصر هیچ صفتی ندارد.

فرزندان:

باید به‌طور دقیق دو عنصر <term> وجود داشته باشد.

مثال:

```
<equal>
<term name="Robert Dobbs"/>
<term name="Bob Dobbs"/>
</equal>
```

-->

```
<!ENTITY % Equal.content
"( term, term )"
>
<!ELEMENT equal %Equal.content; >
<!ATTLIST equal
%Common.attrib;
>
```

<!-- **۳-۳-۴ عنصر Role**.....-->

<!--

نام: role

URI: http://purl.org/xcl/1.0/#role

برچسب: Role

توصیف: بند الف-۲-۳-۵ از این استاندارد ملی ملاحظه شود.

-->

ملاحظات: <!--

صفات:

name (الزامی): مقدار این صفت نامی است که نقش محتوای تجزیه‌ناپذیر را نشان می‌دهد. نام‌های نقش‌ها روابط دودویی در CL را نشان می‌دهند و می‌توانند غیرگفتمانی باشند.

فرزندان:

یک عنصر ضروری <term> نشان دهنده‌ی آرگومان «filler» که پس از آن صفر یا چند عنصر <comment> قرار می‌گیرد.

مثال:

یک تجزیه‌ناپذیر که از نحو Role استفاده می‌کند:

```
<atomic href="http://purl.org/xcl/1.0/#roleset">
<relation>married</relation>
<role name="wife">
   <term name="Jill"/>
</role>
<role name="husband">
   <term name="Jack"/>
</role>
</atomic>
```

-->

```
<!ENTITY % Role.content
"( term, %Comment.class;* )"
>
<!ELEMENT role %Role.content; >
<!ATTLIST role
%Common.attrib;
name NMTOKEN #REQUIRED
>
```

<-- ۳-۴ عناصر اصطلاح و تابع.....--!>

<-- ۳-۴-۱ عنصر Term.....--!>

نام: term <!--

URI: http://purl.org/xcl/1.0/#term

برچسب: Term

توصیف: بند ۶-۱-۱-۱۰ از این استاندارد ملی ملاحظه شود.

-->

ملاحظات: <!--

عنصر <term> می‌تواند یکی از دو شکل زیر را داشته باشد (البته با نادیده گرفتن عناصر comment که از نظر نحوی مهم نیستند):

۱- یک عنصر تهی دربردارنده‌ی یک نام که با استفاده از صفت name آن مشخص می‌شود و به عنوان فرزند ضروری عناصر <guard>، <var>، <sort>، <relation>، <function> یا <role> مجاز است.

۲- یک عنصر دارای محتوا که اصطلاحی را نشان می‌دهد که از اعمال یک تابع بر تعدادی آرگومان بدست می‌آید. اولین عنصر فرزند تابع را مشخص می‌کند و می‌تواند یک عنصر اصطلاح باشد. نام این عنصر می‌تواند غیرگفتمانی باشد.

صفات:

اگر هیچ عنصر <function> یا <term> وجود نداشته باشد صفت name الزامی است (گرچه محدودیتی برای این موضوع در DTD مشخص نشده است).

فرزندان:

یک دنباله‌ی اختیاری از یک عنصر <function> که پس از آن یک یا چند عنصر <term> و سپس صفر یا چند عنصر <comment> قرار گیرد.

مثال‌ها:

```
<term name="cow"/>
```

```
            <term>
               <function>
                  <term name="pig"/>
                  <term name="cow"/>
                  </function>
               <term name="barnyard"/>
               </term>
-->

<!ENTITY % Term.content
"( ( function, term+ )?, %Comment.class;* )"
>
<!ELEMENT term %Term.content; >
<!ATTLIST term
%Common.attrib;
name CDATA #IMPLIED
>
```

<!-- **۳-۴-۲ عنصر Function**.....-->

نام:	function
URI:	http://purl.org/xcl/1.0/#function
برچسب:	Function
توصیف:	یک عنصر <term> را دربرمی‌گیرد. نام می‌تواند غیرگفتمانی باشد. بند ۶-۱-۱-۱۱ این استاندارد ملی ملاحظه شود.

<!--

ملاحظات:

این عنصر هیچ صفتی ندارد.

فرزندان:

یک عنصر <term> که پس از آن صفر یا چند عنصر <comment> قرار گیرد.

مثال:

```
            <function>
            TBD
            </function>
-->

<!ENTITY % Function.content
"( term, %Comment.class;* )"
```

```
>
<!ELEMENT function %Function.content; >
<!ATTLIST function
%Common.attrib;
>
```

<-- اعلان پشتیبانی از معماری.....--!>

به منظور استفاده از مشخصه‌ی نوع مدرک XCL نسخه‌ی ۱/۰ به عنوان یک معماری مبنا برای سایر کاربردهای SGML/XML، توصیه می‌شود از اعلان پشتیبانی از معماری زیر استفاده شود:

```
<?IS24707
arch name="html"
public-id="ISO/IEC 24707:2006//DTD XML Common Logic (XCL) 1.0//EN"
dtd-system-id="http://purl.org/xcl/1.0/xcl1.dtd"
renamer-att="XCLnames"
doc-elem-form="text"
?>
-->
```

پ-۳ معناشناسي XCL

منطق مشترک را می‌توان به‌صورت مستقیم به XCL به گونه‌ای ترجمه کرد که نحو منطق مشترک را به طور مستقیم در نشانه‌گذاری XML نمایش دهد.. از آنجا که XML هیچ معناشناسی ذاتی ندارد هدف از بند ج-۲ این است که هر ساختار در نحو به‌طور دقیق همان معنایی را که در بند ۶-۲ این استاندارد مشخص شده منتقل کند. هیچ معنای دیگری به‌طور صریح یا به صورت ضمنی بیان نشده است.

پ-٤ انطباق XCL

XCL به دلایل زیر یک گویش به‌طور دقیق منطبق بر منطق مشترک است. نخست، نحو محسوس XCL در بند ج-۲ به‌طور دقیق منطبق بر نحو انتزاعی منطق مشترک بیان شده در بند ۶-۲ است. علاوه بر این از آنجا که معناشناسی XCL به‌طور مستقیم برگرفته از معناشناسی منطق مشترک که در بند ۶-۲ بیان شده است، این گویش هم از نظر نحوی و هم از نظر معناشناسی منطبق بر بدنه‌ی این استاندارد ملی است.

کتاب‌نامه

[1] SOWA, J.F. *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley, 1984

[2] DÜRST, M. and SUIGNARD, M. *Internationalized Resource Identifiers (IRIs)*, The Internet Society, 2005. URL: http://www.ietf.org/rfc/rfc3987.txt

[3] GENESERETH, M.R. and FIKES, R.E. *Knowledge Interchange Format Version 3.0 Reference Manual*, Computer Science Department, Stanford University, Technical Report KSL-92-86, June 1992

[4] MEALLING, M. and DENENBERG, R. *RFC 3305 - Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations*, 2002

[5] RUMBAUGH, J., JACOBSEN, I. and BOOCH, G. *Unified Modeling Language Reference Manual*. Reading, MA: Addison-Wesley, 1998

[6] CARROL, J.J., BIZER, C., HAYES, P. and STICKLER, P. *Named graphs, provenance and trust*, in *14th Intl. Conf. on World Wide Web*. Chiba, Japan: ACM Press, 2005

[7] FIELDING, R.T. Architectural Styles and the Design of Network-Based Software Architectures, in *Information and Computer Science*, Ph.D. dissertation, Irvine, CA, USA: University of California, Irvine, 2000

[8] BERNERS-LEE, T., FIELDING, R. IRVINE, J.C. and MASINTER, L. *Uniform Resource Identifiers*, IETF, 1998

[9] BERNERS-LEE, T., HENDLER, J. and LASSILA, O. The Semantic Web, in *Scientific American*, vol. 284, 2001