طراحی و پیاده‌سازی یک تحلیلگر صرفی برای زبان فارسی

وحید مواجی

استاد: دکتر بهرام وزیرنژاد

دانشکده زبان‌ها و زبان‌شناسی، دانشگاه صنعتی شریف

چکیده

در این پروژه با استفاده از قوانین موجود در دستور زبان، سیستمی برای تجزیه صرفی کلمات طراحی نموده‌ایم. سعی شده است تا سیستم از ویژگی‌هایی چون سبک بودن، سریع بودن، مبتنی بر قاعده بودن، کمترین میزان استثناها را داشتن، توجه به ویژگی‌های زبان فارسی و چالش های این زبان برخوردار باشد. در این پروژه سعی شده است با استفاده از زبان برنامه نویسی Java، الگوریتمی برای تحلیل صرفی و ریشه یابی واژگان زبان فارسی ارائه و پیاده‌سازی شود. مطالعه و بررسی الگوریتم‌های موجود مانند الگوریتم porter و بررسی دشواری‌های نوشتاری فارسی و ناهماهنگی‌های آن و مشکلات نوشتار رایانه‌ای زبان فارسی از چالش‌هایی بود که با آن مواجه بودیم. همچنین روشهای ریشه‌یابی شکل‌های صرف شده و مشتقات فعل‌های فارسی می‌توانند از موضوعات مورد بحث در آینده باشند. بدلیل محدودیت های موجود برای نگهداری، مدل‌سازی و ایجاد مستندات در زبان‌های برنامه نویسی دیگر مانند C, C++, Python, Ruby، سادگی و در عین حال قدرتمندی زبان Java و وجود توابع بسیار برای پردازش متن و همچنین وجود بسته‌های نرم افزاری موجود برای NLP مانند Stanford NLP، زبان Java برای پیاده سازی پروژه انتخاب گردید.

کلید واژگان: تجزیه‌گر صرفی، صرف واژگان فارسی، تحلیلگر صرفی، پردازش خودکار متن، زبان Java.

1. مقدمه

انواع کلمه یا اقسام کلمه (parts of speech) به طور سنتي براي اشاره به طبقات دستوري کلمات به کار مي‌رود و مهمترين انواع کلمه در دستورهايي که به پيروي از دستورنويسان روم و يونان باستان نوشته شده‌اند معمولاً عبارت است از اسم، ضمير، فعل، قيد، صفت، حرف اضافه، حرف ربط، صوت که اغلب مي‌توان حرف تعريف و ادات را نيز به آن افزود. اين تقسيم‌بندي از ويژگي‌هاي زبان‌هاي يوناني و لاتين است و به هيچ وجه جنبه عمومي و جهاني در همه زبانها ندارد و از طرف ديگر مفهوم "انواع کلمه" همانند مفهوم خود "کلمه" در دستورهاي سنتي داراي ابهام است. بنابراين براي رفع اين مشکل و محدود کردن اصطلاح "انواع کلمه" لازم است "واژه"(lexeme)ها را مدنظر قرار دهيم و صورت‌هاي تصريفي واحدهاي صرفي را با عنوان مثلاً "صورت‌هاي اسم" يا "صورت‌هاي فعل" مطرح کنيم، با اين کار تقسيم‌بندي کلمات، مبتني بر ملاک‌هاي صوري خواهد بود که از عموميت بيشتر برخوردار است.

واژه‌ها از نظر صرفي و نحوي رفتارهاي خاصي در نظام زبان دارند و واژه‌هايي که داراي رفتارهاي صرفي و نحوي و نيز مشخصه‌هاي معنايي مشابه باشند در گروه‌هاي جداگانه قرار مي‌گيرند. هرکدام از اين گروه با عنوان خاصي مانند اسم، صفت، فعل و غيره مشخص مي‌شود. مثلاً "اسم" به گروهي از واژه‌ها اطلاق مي‌شود که مي‌تواند با تکواژ جمع تصريف شود، و يا به عبارت ديگر جمع بسته شود و از نظر نحوي نقش فاعل، مفعول و غيره ايفا کند. اين مسئله دربارة بقية گروه‌هاي واژه‌ها نيز صادق است. از اين رو مي‌توان فرض کرد که واژه‌ها در واژگان ذهني (mental lexicon) اهل زبان نيز طبقه‌بندي شده‌اند و به همين دليل اهل زبان بدون آموزش و صرفاً با اتکا به توانش زباني خود مي‌توانند واژه‌هاي نوساخته را در طبقات خاص قرار دهند و متناسب با ويژگي صرفي و نحوي آنها را به کار برند. در ادامه به اختصار به توصيف ساختار تصريفي هرکدام از طبقات مورد اشاره در بالا مي‌پردازيم. لازم به ذکر است که برخي از اقسام کلمه مانند حروف ربط صورت تصريفي ندارند و به همين دليل به آنها نخواهيم پرداخت.

2. ساختار تصریفی انواع کلمه در زبان فارسی

2.1. ساختار تصریفی اسم

$$[\text{اسم}] + [(\text{تکواژجمع})] + \begin{bmatrix} (\text{ياي نکره}) \\ (\text{ياي بند موصولي}) \\ (\text{واژه‌بست‌های شخصي / ضمایر متصل}) \\ (\text{کسرة اضافه}) \end{bmatrix} + [(\text{واژه‌بست‌های ربطي})]$$

2.2. ساختار تصریفی صفت

$$[\text{صفت}] + \begin{bmatrix}(\text{تکواژ صفت تفضیلی‌ساز})\\(\text{تکواژ صفت عالی‌ساز})\end{bmatrix} + [(\text{واژه‌بست‌های ربطی})]$$

2.3. ساختار تصریفی فعل

$$[\text{تکواژ نفی}] + \begin{bmatrix}(\text{تکواژ وجه امری و التزامی})\\(\text{تکواژ استمراری})\end{bmatrix} + [\text{فعل}] + [(\text{تکواژ نمود کامل})] + [(\text{شناسه‌ها})] + [(\text{واژه‌بست‌های شخصی})]$$

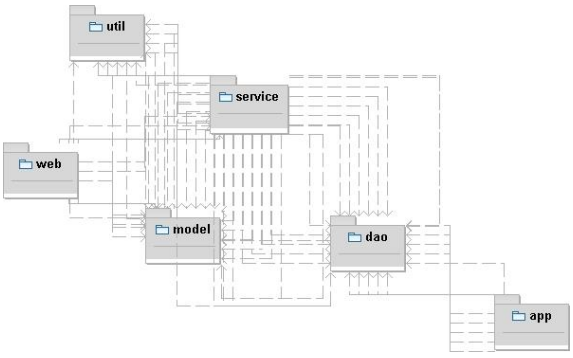
2.4. ساختار تصریفی قید

$$\begin{bmatrix}(\text{تکواژ قید تفضیلی‌ساز})\\(/-\mathbf{tar}/)\end{bmatrix} + [\text{قید}]$$

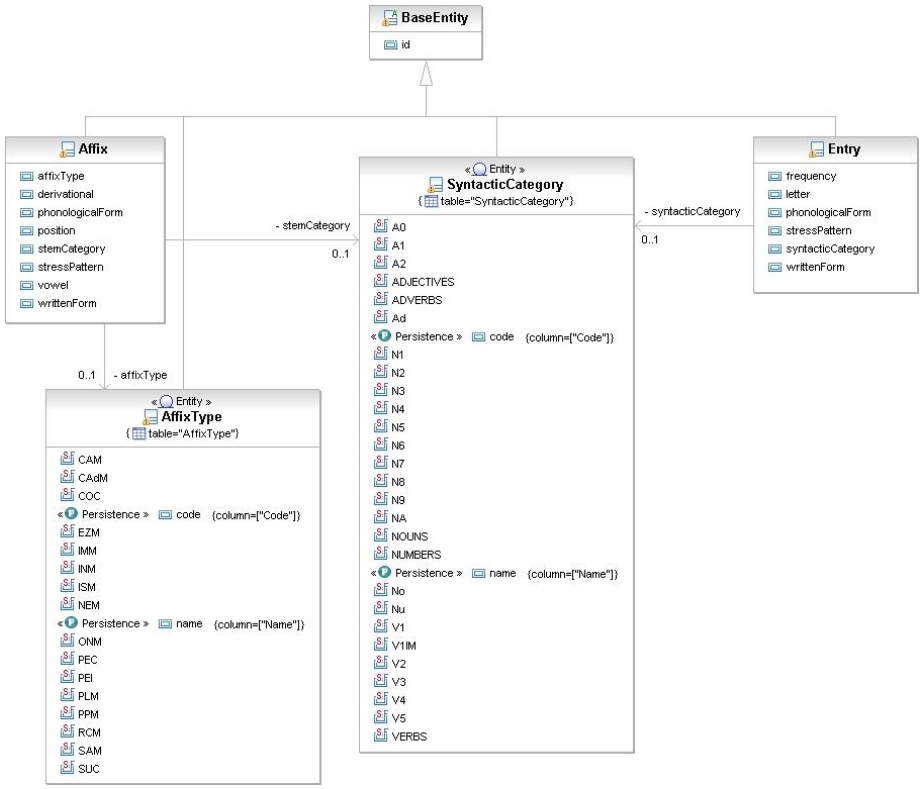
3. ساختار کلی برنامه

با استفاده از مبانی نظری و زیرساخت‌های تئوریکی که در بالا به اختصار به آن اشاره شد (و توضیح بیشتر در باره آن، خود یک مقاله جداگانه خواهد بود)، یک برنامه مبتنی بر وب و با معماری چندلایه برای پیاده‌سازی تحلیلگر صرفی نوشته شد. در لایه مدل برنامه از Hibernate 3.0 و Annotationهای معرفی شده در آن برای دسترسی به دادگان موجود در پایگاه داده استفاده می‌شود. پایگاه داده مبتنی بر MySql® 5.0 است که دادگان آن از پایگاه داده FLexicon پر شده است. در لایه وب از Struts 2.0 و Jsp و برای زیرساخت برنامه هم از Spring 2.5 استفاده شده است. به جای توضیح زبانی الگوریتم‌های استفاده شده، نمودارهای UML آنها از جمله Class Diagram ها و Sequence Diagram ها با استفاده از ابزار Eclipse Europa® و Omondo® با استفاده از روش مهندسی معکوس، از روی برنامه ساخته شده اند. بنابراین این نمودارها را در ادامه می‌آوریم و از توضیح بیشتر پرهیز می‌کنیم.

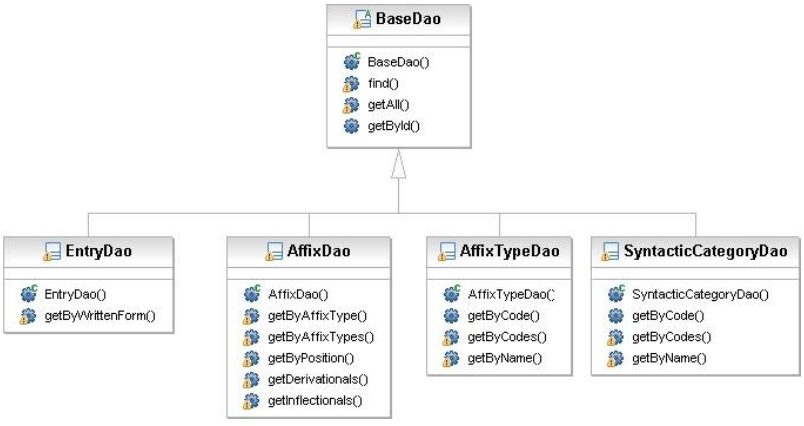
3.1. ساختار بسته‌های برنامه



3.2. ساختار کلاس‌های لایه مدل

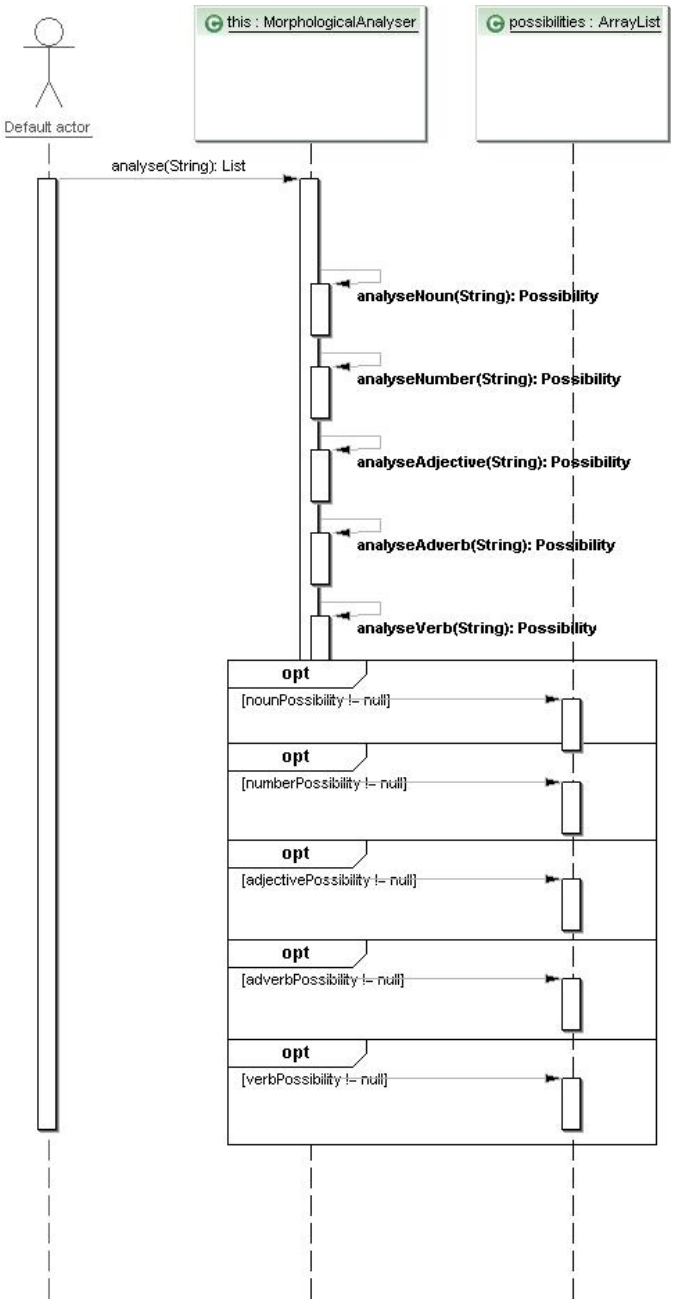


3.3. ساختار کلاس‌های لایه DAO

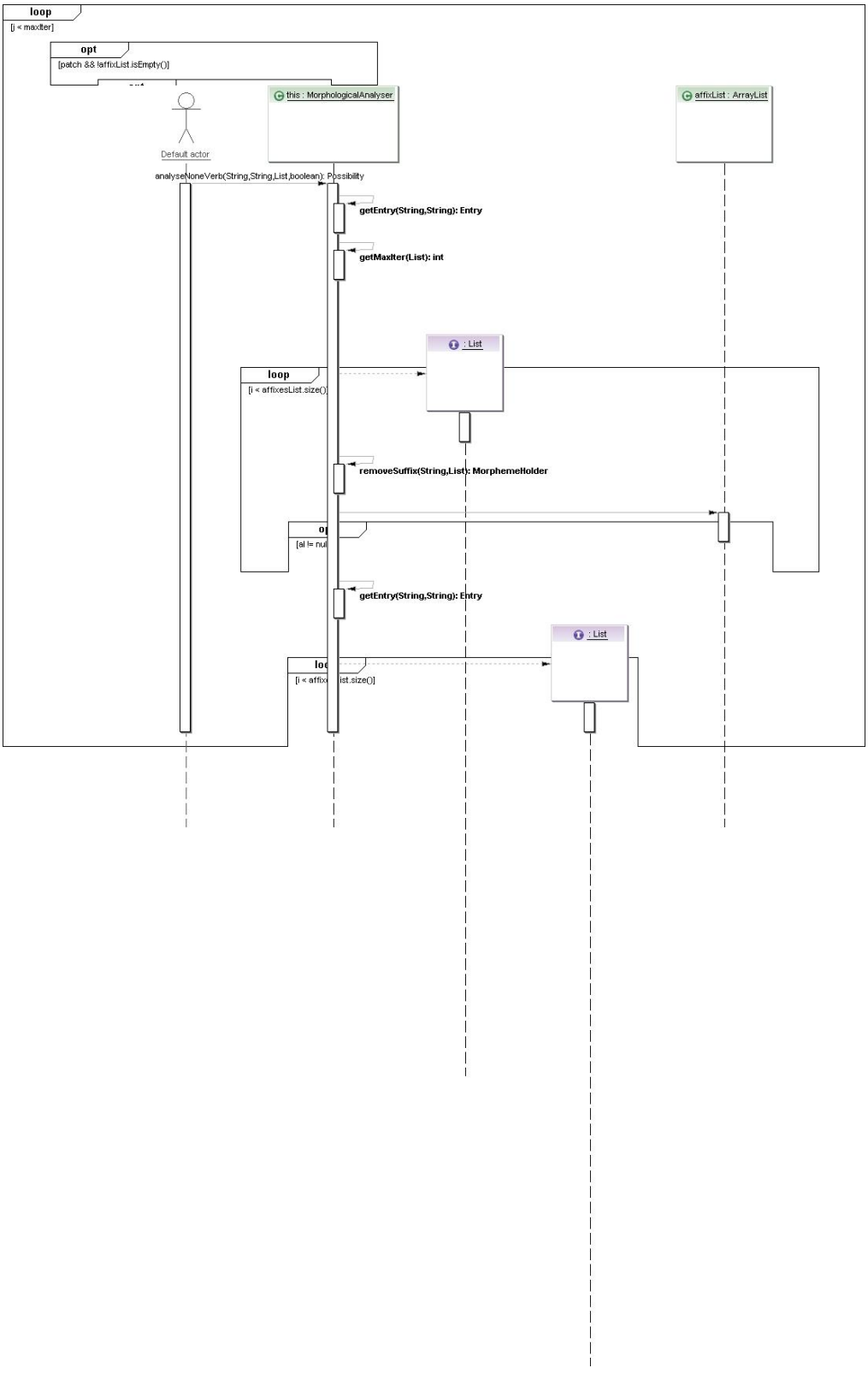


3.4. ساختار کلاس‌های لایه سرویس

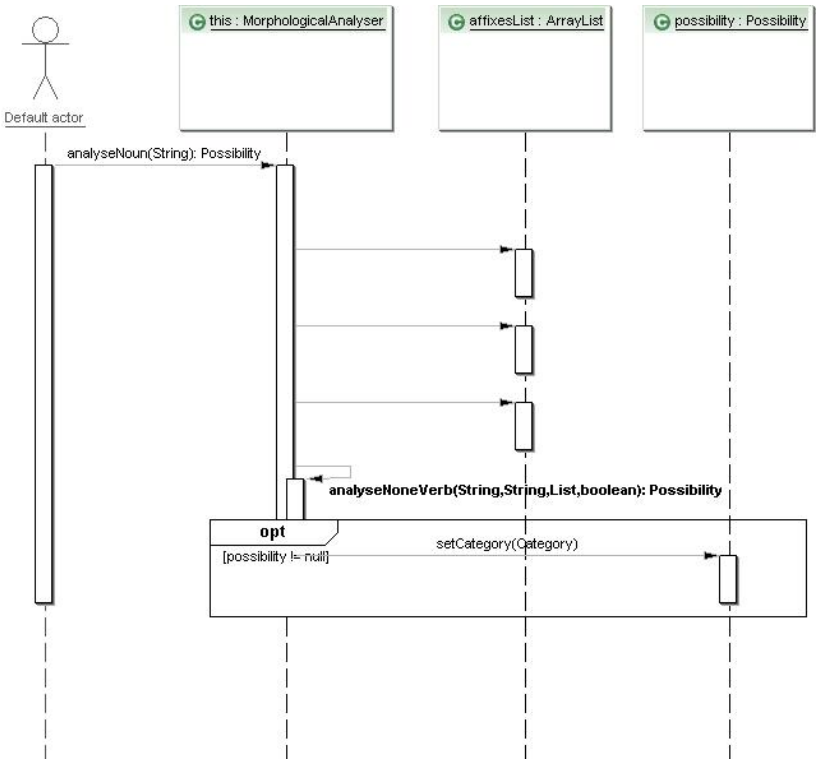
3.5. نمودار الگوریتم آنالیز کلی



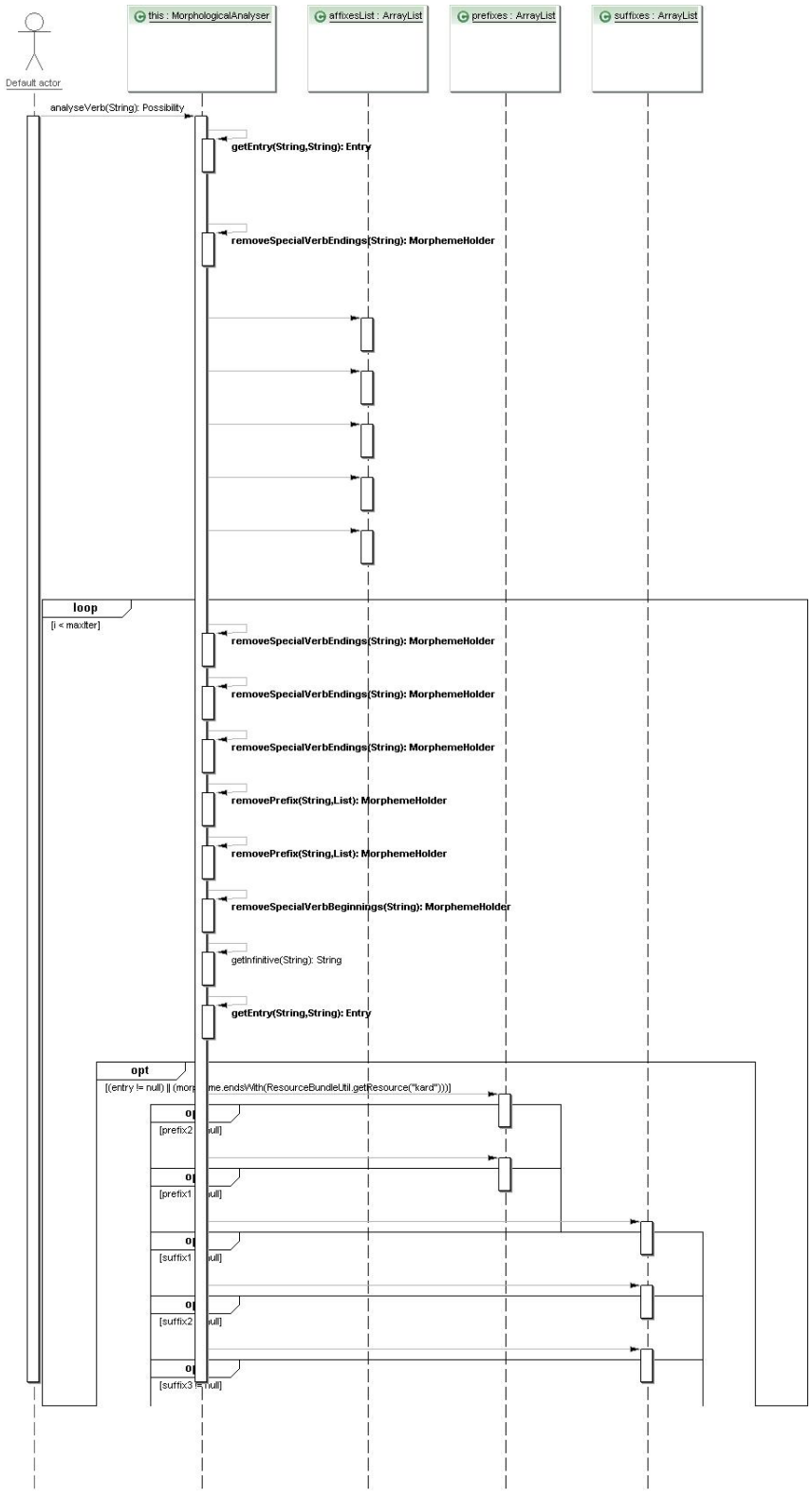
3.6. نمودار الگوریتم آنالیز انواع کلمه غیر از فعل



3.7. نمودار الگوریتم آنالیز اسم



3.8. نمودار الگوریتم آنالیز فعل


Default actor
this : MorphologicalAnalyser
affixesList : ArrayList
prefixes : ArrayList
suffixes : ArrayList
analyseVerb(String): Possibility
getEntry(String,String): Entry
removeSpecialVerbEndings(String): MorphemeHolder
loop
[i < maxIter]
removeSpecialVerbEndings(String): MorphemeHolder
removeSpecialVerbEndings(String): MorphemeHolder
removeSpecialVerbEndings(String): MorphemeHolder
removePrefix(String,List): MorphemeHolder
removePrefix(String,List): MorphemeHolder
removeSpecialVerbBeginnings(String): MorphemeHolder
getInfinitive(String): String
getEntry(String,String): Entry
opt
[(entry != null) || (mor me.endsWith(ResourceBundleUtil.getResource("kard")))]
[prefix2
[prefix1
[suffix1
[suffix2
[suffix3 != null]