Google Web Service

Rui José Gonçalves de Castro

March 12, 2003

1 Introduction

Web services, in the general meaning of the term, are services offered via the Web. In a typical Web services scenario, a business application sends a request to a service at a given URL using the SOAP protocol over HTTP. The service receives the request, processes it, and returns a response.

An example of a Web service, is the Google Web Service. this service provides all the funcionality that we can find at their web page (www.google.com).

This document contains the VDM-SL specification of the Google Web Service API's and the specification of the syntax of the queries that can me made throught it. A small application, is also, documented here. This application takes a VDM-SL type of a query constructs a query string and feeds it to the Web Service that returns the result of Google search engine for that query, form witch a corresponding VDM-SL type is constructed.

This document contains five appendixes. The first one (see appendix A) is a transcript of the document `Google Web APIs Reference`, that can be found at http://www.google.com/apis/reference.html. The second is the VDM-SL specification of the Google Web Service API, along with the contents of the file `GoogleSearch.wsdl`. The last tree contain the VDM-SL specifications of Google Query syntax, Language Restrict parameter syntax and the Restrict parameter syntax.

2 VDMTools and Google Web Service

2.1 Architecture

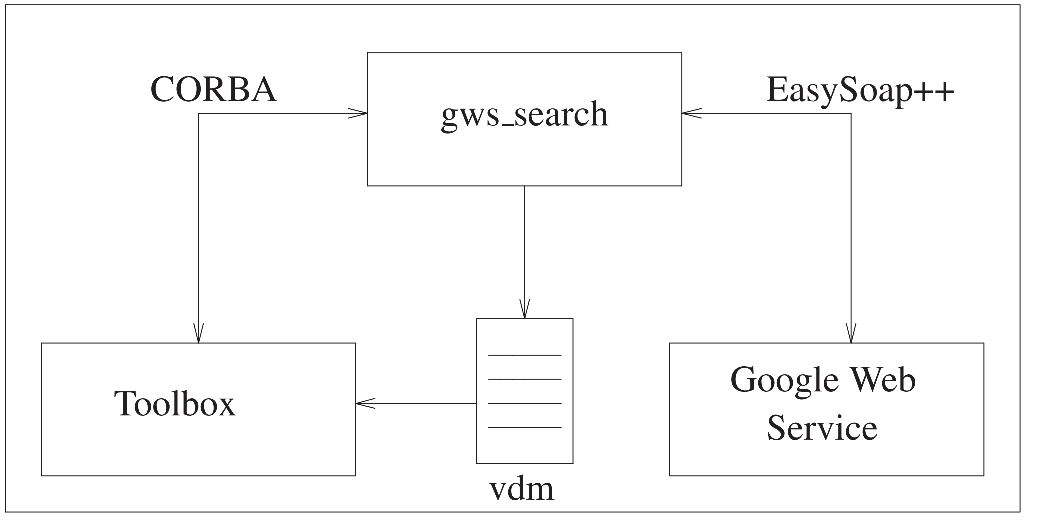
In this section is presented a description of the parts involved in creating the example aplication mentioned above.

2.1.1 VDMTools

IFAD VDMTools provides a facility for other applications to use its funcionality throught CORBA. This way we can local specification files into the Toolbox, do a syntax check and type check on then and even exceute any command that we could execute in the interpreter.

In this application, we load the Google Web Service API specification and the Google Search Query syntax, contained in this file, into the Toolbox. Then, we execute the function `queryToString` (p. 22) with one of the example query values found in section C.3 to obtain a query string that can be passed as an argument to the function `doGoogleSearh` (p. 19).

When the Web service returns a value, we construct a GoogleSearchResult value (p. 16), write it on a VDM-SL file and instruct the Toolbox to load that file.



2.1.2 Web Service Provider - EasySoap++

In order to access a web service we need a something that implements the SOAP protocol.
In this case the library EasySoap++ was used.
EasySoap++ is a lightweight SOAP implementation written in C++.
EasySoap++ provides us with the means to call the Google Web Service functions without even have to care with network operations or XML.

Ours application, passes the query string (obtained as described above) to a function, that uses EasySoap++ functionalities to call the remote Google's search, and extracts the information from the value returned by that function, in order to construct a VDM-SL value of the type GoogleSearchResult (p. 16) that is then loaded into the Toolbox.

2.1.3 Sample usage

Here is presented a sample usage of the application along with the results obtained.

```
$> gws_search "Dk99WfdQFHI6kLWG4FmXe/35bvCEh2pG"
/home/rui/aulas/5ano/1semestre/lmf/GoogleSearch/vdm/GoogleSearch.vdm
info_results.vdm info_query
```

This is a typical command line for the application.
The first parameter is the service key provided in the registration process. The second is the specification file for the Google Web Service API and the Google Query syntax (actually is this document). The third is the name of the file to write the results to. And, the last is the variable name of the value that contains the query (in this case the value is the one in page 31).

After calling the function `queryToString` with `info_query` as a parameter the result is `"info:www.google.com"`.

And the next box shows the contents of the file written by the application.

```
module Results
    imports
        from GoogleSearch all
    exports all
definitions
values
    doGoogleSearch-result : GoogleSearch`GoogleSearchResult = mk-GoogleSearch`GoogleSearchResult
        (
         false,
         [],
         1,
         true,
         [mk-GoogleSearch`ResultElement
             (
              "Enables users to search the Web, Usenet, and images. Features include PageRank, caching and translation...",
              "http:
//www.google.com/",
              "Happy Valentine's Day! Web, Images, Groups, Directory, News, ~ Advanced <
br > Search ~ Preferences ~ Language Tools. Advertise with Us < b >
... </b > ",
              "Google",
              "4k",
              true,
              [],
              mk-GoogleSearch`DirectoryCategory
                  (
                   "Top/Computers/Internet/Searching/Search_Engines/Google",
                   []),
              "Google")],
         "info:
www.google.com",
         1,
         1,
         [],
         [],
         0.023423)
end Results
```

A Google Web APIs Reference

Beta 2 - August, 2002

A.1 Overview

This document explains in detail the semantics of the function calls you can make using the Google Web APIs service. In this document, you will learn:

- How Google's query syntax works. (See appendix C)
- How to restrict your queries to portions of Google's index, such as a particular language or country.

- How to interpret the search results information sent back by the Google Web APIs service.

You may also find the following files from the Google Web APIs developer kit to be helpful:

- GoogleSearch.wsdl - WSDL description for Google Web APIs SOAP interface. (See appendix B)
- soap-samples/ - example SOAP messages and responses.
- javadoc/index.html - javadoc for the example Java libraries.

This is a beta document. If you have comments, find errors, or just have questions, please contact `api-support@google.com`.

A.1.1 Search Requests

Search requests submit a query string and a set of parameters to the Google Web APIs service and receive in return a set of search results (p. 16). Search results are derived from Google's index of over 2 billion Web pages.

The details of the interactions involved with search requests are covered in the Search Request Format (p. 4) and Search Results Format (p. 13) sections of this document.

A.1.2 Cache Requests

Cache requests submit a URL to the Google Web APIs service and receive in return the contents of the URL when Google's crawlers last visited the page (if available). Please note that Google is not affiliated with the authors of cached pages nor responsible for their content. The return type for cached pages is base64 (p. 17) encoded text.

A.1.3 Spelling Requests

Spelling requests submit a query to the Google Web APIs service and receive in return a suggested spell correction for the query (if available). Spell corrections mimic the same behavior as found on Google's Web site.

Spelling requests are subject to the same query string limitations as any other search request. (The input string is limited to 2048 bytes and 10 individual words.)

The return type for spelling requests is a text string.

A.2 Search Request Format

A.2.1 Search Parameters

This table lists all the valid name-value pairs that can be used in a search request and describes how these parameters will modify the search results.

Name	Description
key	Provided by Google, this is required for you to access the Google service. Google uses the key for authentication and logging.
q	(See Query Terms (p. 5) for details on query syntax and appendix C for the VDM-SL Specification)
start	Zero-based index of the first desired result.

maxResults	Number of results desired per query. The maximum value per query is 10. **Note:** If you do a query that doesn't have many matches, the actual number of results you get may be smaller than what you request.
filter	Activates or deactivates automatic results filtering, which hides very similar results and results that all come from the same Web host. Filtering tends to improve the end user experience on Google, but for your application you may prefer to turn it off. (See A.2.3 section for more details.)
restrict	Restricts the search to a subset of the Google Web index, such as a country like `"Ukraine"` or a topic like `"Linux."` (See A.2.4 section for more details.)
safeSearch	A Boolean value which enables filtering of adult content in the search results. See A.2.6 for more details.
lr	*Language Restrict* - Restricts the search to documents within one or more languages.
ie	*Input Encoding* - this parameter has been deprecated and is ignored. All requests to the APIs should be made with UTF-8 encoding. (See A.2.5 section for details.)
oe	*Output Encoding* - this parameter has been deprecated and is ignored. All requests to the APIs should be made with UTF-8 encoding. (See A.2.5 section for details.)

A.2.2 Query Terms - <q>

Default Search - By default, Google only returns pages that include all of the terms in the query string. There is no need to include "AND" between terms. Keep in mind that the order of the terms in the query will impact the search results.

Stop Words - Google ignores common words and characters such as *"where"* and *"how"* as well as certain single digits and single letters. Common words that are ignored are known as *stop words*. However, you can prevent Google from ignoring stop words by enclosing them in quotes, such as in the phrase "to be or not to be".

Special Characters - By default, all non-alphanumeric characters that are included in a search query are treated as word separators. The only exceptions are the following: double quote mark ("), plus sign (+), minus sign or hyphen (-), and ampersand (&). The ampersand character (&) is treated as another character in the query term in which it is included, while the remaining exception characters correspond to search features listed in the section below.

Special Query Terms - Google supports the use of several special query terms that allow the user or search administrator to access additional capabilities of the Google search engine.

- **Include Query Term**
  Ex: **`Star Wars Episode +I`**
  If a common word is essential to getting the results you want, you can include it by putting a "+" sign in front of it.
  (See page 22 for the VDM-SL specification.)

- **Exclude Query Term**
  Ex: **`bass -music`**
  You can exclude a word from your search by putting a minus sign ("-") immediately in front of the term you want to exclude from the search results.
  (See page 22 for the VDM-SL specification.)

- **Phrase Search**
  Ex: `"yellow pages"`
  Search for complete phrases by enclosing them in quotation marks or connecting them with hyphens. Words marked in this way will appear together in all results exactly as entered.
  Note: You may need to use a "+" to force inclusion of common words in a phrase.
  (See page 22 for the VDM-SL specification.)

- **Boolean OR Search**
  Ex: `vacation london OR paris`
  Google search supports the Boolean "OR" operator. To retrieve pages that include either word A or word B, use an uppercase OR between terms.
  (See page 22 for the VDM-SL specification.)

- **Site Restricted Search**
  Ex: `admission site:www.stanford.edu`
  If you know the specific web site you want to search but aren't sure where the information is located within that site, you can use Google to search only within a specific web site. Do this by entering your query followed by the string *"site:"* followed by the host name.
  Note: The exclusion operator ("-") can be applied to this query term to remove a web site from consideration in the search.
  Note: Only one *site:* term per query is supported.
  (See page 22 for the VDM-SL specification.)

- **Date Restricted Search**
  Ex: `Star Wars daterange:2452122-2452234`
  If you want to limit your results to documents that were published within a specific date range, then you can use the "*daterange:* " query term to accomplish this. The "*daterange:*" query term must be in the following format: **daterange:<start_date>-<end date>** where <**start_date**> = Julian date indicating the start of the date range <**end_date**> = Julian date indicating the end of the date range The Julian date is calculated by the number of days since January 1, 4713 BC. For example, the Julian date for August 1, 2001 is 2452122.
  (See page 22 for the VDM-SL specification.)

- **Title Search (term)**
  Ex: `intitle:Google search`
  If you prepend *"intitle:"* to a query term, Google search restricts the results to documents containing that word in the title. Note there can be no space between the *"intitle:"* and the following word.
  Note: Putting *"intitle:"* in front of every word in your query is equivalent to putting *"allintitle:"* at the front of your query.
  (See page 22 for the VDM-SL specification.)

- **Title Search (all)**
  Ex: `allintitle:  Google search`
  Starting a query with the term *"allintitle:"* restricts the results to those with all of the query words in the title.
  (See page 22 for the VDM-SL specification.)

- **URL Search (term)**
  Ex: `inurl:Google search`
  If you prepend *"inurl:"* to a query term, Google search restricts the results to documents containing that word in the result URL. Note there can be no space between the *"inurl:"* and the following word.

Note: *"inurl:"* works only on *words* , not URL components. In particular, it ignores punctuation and uses only the first word following the *"inurl:"* operator. To find multiple words in a result URL, use the *"inurl:"* operator for each word.
Note: Putting *"inurl:"* in front of every word in your query is equivalent to putting *"allinurl:"* at the front of your query.
(See page 22 for the VDM-SL specification.)

- **URL Search (all)**
  Ex: **`allinurl: Google search`**
  Starting a query with the term *"allinurl:"* restricts the results to those with all of the query words in the result URL. **Note:** *"allinurl:"* works only on *words,* not URL components. In particular, it ignores punctuation. Thus, *"allinurl: foo/bar"* restricts the results to pages with the words "foo" and "bar" in the URL, but does not require that they be separated by a slash within that URL, that they be adjacent, or that they be in that particular word order. There is currently no way to enforce these constraints.
  (See page 22 for the VDM-SL specification.)

- **Text Only Search (all)**
  Ex: **`allintext: Google search`**
  Starting a query with the term *"allintext:"* restricts the results to those with all of the query words in only the body text, ignoring link, URL, and title matches.
  (See page 22 for the VDM-SL specification.)

- **Links Only Search (all)**
  Ex: **`allinlinks: Google search`**
  Starting a query with the term *"allinlinks:"* restricts the results to those with all of the query words in the URL links on the page.
  (See page 22 for the VDM-SL specification.)

- **File Type Filtering**
  Ex: **`Google filetype:doc OR filetype:pdf`**
  The query prefix *"filetype:"* filters the results returned to include only documents with the extension specified immediately after. Note there can be no space between *"filetype:"* and the specified extension.
  Note: Multiple file types can be included in a filtered search by adding more *"filetype:"* terms to the search query.
  (See page 22 for the VDM-SL specification.)

- **File Type Exclusion**
  Ex: **`Google -filetype:doc -filetype:pdf`**
  The query prefix *"-filetype:"* filters the results to exclude documents with the extension specified immediately after. Note there can be no space between *"-filetype:"* and the specified extension.
  Note: Multiple file types can be excluded in a filtered search by adding more *"-filetype:"* terms to the search query.
  (See page 22 for the VDM-SL specification.)

- **Web Document Info**
  Ex: **`info:www.google.com`**
  The query prefix *"info:"* returns a single result for the specified URL if it exists in the index.
  Note: No other query terms can be specified when using this special query term.
  (See page 22 for the VDM-SL specification.)

- **Back Links**
  Ex: **`link:www.google.com`**

The query prefix *"link:"* lists web pages that have links to the specified web page. Note there can be no space between *"link:"* and the web page URL.
Note: No other query terms can be specified when using this special query term. (See page 22 for the VDM-SL specification.)

- **Related Links**
  Ex: **`related:www.google.com`**
  The query prefix *"related:"* lists web pages that are similar to the specified web page. Note there can be no space between *"link:"* and the web page URL.
  Note: No other query terms can be specified when using this special query term. (See page 22 for the VDM-SL specification.)

- **Cached Results Page**
  Ex: **`cache:www.google.com web`**
  The query prefix *"cache:"* returns the cached HTML version of the specified web document that the Google search crawled. Note there can be no space between *"cache:"* and the web page URL. If you include other words in the query, Google will highlight those words within the cached document. (See page 22 for the VDM-SL specification.)

You can see the formalization of the syntax described here in appendix C.

A.2.3 Automatic Filtering - <filter>

The <filter> parameter causes Google to filter out some of the results for a given search. This is done to enhance the user experience on Google.com, but for your application, you may prefer to turn filtering off in order to get the full set of search results.

When enabled, filtering takes the following actions:

- **Near-Duplicate Content Filter** - If multiple search results contain identical titles and snippets, then only one of the documents is returned.

- **Host Crowding** - If multiple results come from the same Web host, then only the first two are returned.

A.2.4 Restricts - <lr> <restrict>

Google provides the ability to search a predefined subset of Google's Web index. This is enabled by using the **`lr`** and **`restrict`** parameters.

<lr> - language restrict

To search for documents within a particular language, use the <lr> parameter, using one of the values in the table below.

Language	**<lr> value**	**Language**	**<lr> value**
Arabic	lang_ar	Icelandic	lang_is
Chinese S	lang_zh-CN	Italian	lang_it
Chinese T	lang_zh-TW	Japanese	lang_ja
Czech	lang_cs	Korean	lang_ko
Danish	lang_da	Latvian	lang_lv
Dutch	lang_nl	Lithuanian	lang_lt
English	lang_en	Norwegian	lang_no
Estonian	lang_et	Portuguese	lang_pt
Finnish	lang_fi	Polish	lang_pl
French	lang_fr	Romanian	lang_ro
German	lang_de	Russian	lang_ru

Greek	lang_el	Spanish	lang_es
Hebrew	lang_iw	Swedish	lang_sv
Hungarian	lang_hu	Turkish	lang_tr

<restrict> - Country and Topic Restricts

Google allows you to search for Web information within one or more countries, using an algorithm that considers the top level domain name of the server and the geographic location of the server IP address.

The automatic country sub-collections currently supported are listed below:

Country	**<restrict> value**	**Country**	**<restrict> value**
Afghanistan	countryAF	Albania	countryAL
Algeria	countryDZ	American Samoa	countryAS
Andorra	countryAD	Angola	countryAO
Anguilla	countryAI	Antarctica	countryAQ
Antigua and Barbuda	countryAG	Argentina	countryAR
Armenia	countryAM	Aruba	countryAW
Australia	countryAU	Austria	countryAT
Azerbaijan	countryAZ	Bahamas	countryBS
Bahrain	countryBH	Bangladesh	countryBD
Barbados	countryBB	Belarus	countryBY
Belgium	countryBE	Belize	countryBZ
Benin	countryBJ	Bermuda	countryBM
Bhutan	countryBT	Bolivia	countryBO
Bosnia and Herzegowina	countryBA	Botswana	countryBW
Bouvet Island	countryBV	Brazil	countryBR
British Indian Ocean Territory	countryIO	Brunei Darussalam	countryBN
Bulgaria	countryBG	Burkina Faso	countryBF
Burundi	countryBI	Cambodia	countryKH
Cameroon	countryCM	Canada	countryCA
Cape Verde	countryCV	Cayman Islands	countryKY
Central African Republic	countryCF	Chad	countryTD
Chile	countryCL	China	countryCN
Christmas Island	countryCX	Cocos (Keeling) Islands	countryCC
Colombia	countryCO	Comoros	countryKM
Congo	countryCG	Congo, The Democratic Republic of the	countryCD
Cook Islands	countryCK	Costa Rica	countryCR
Cote D'ivoire	countryCI	Croatia (local name: Hrvatska)	countryHR
Cuba	countryCU	Cyprus	countryCY
Czech Republic	countryCZ	Denmark	countryDK
Djibouti	countryDJ	Dominica	countryDM
Dominican Republic	countryDO	East Timor	countryTP
Ecuador	countryEC	Egypt	countryEG
El Salvador	countrySV	Equatorial Guinea	countryGQ

Eritrea	countryER	Estonia	countryEE
Ethiopia	countryET	European Union	countryEU
Falkland Islands (Malvinas)	countryFK	Faroe Islands	countryFO
Fiji	countryFJ	Finland	countryFI
France	countryFR	France, Metropolitan	countryFX
French Polynesia	countryPF	French Quiana	countryGF
French Southern Territories	countryTF	Gabon	countryGA
Gambia	countryGM	Georgia	countryGE
Germany	countryDE	Ghana	countryGH
Gibraltar	countryGI	Greece	countryGR
Greenland	countryGL	Grenada	countryGD
Guadeloupe	countryGP	Guam	countryGU
Guatemala	countryGT	Guinea	countryGN
Guinea-Bissau	countryGW	Guyana	countryGY
Haiti	countryHT	Heard and Mc Donald Islands	countryHM
Holy See (Vatican City State)	countryVA	Honduras	countryHN
Hong Kong	countryHK	Hungary	countryHU
Iceland	countryIS	India	countryIN
Indonesia	countryID	Iran (Islamic Republic of)	countryIR
Iraq	countryIQ	Ireland	countryIE
Israel	countryIL	Italy	countryIT
Jamaica	countryJM	Japan	countryJP
Jordan	countryJO	Kazakhstan	countryKZ
Kenya	countryKE	Kiribati	countryKI
Korea, Democratic People's Republic of	countryKP	Korea, Republic of	countryKR
Kuwait	countryKW	Kyrgyzstan	countryKG
Lao People's Democratic Republic	countryLA	Latvia	countryLV
Lebanon	countryLB	Lesotho	countryLS
Liberia	countryLR	Libyan Arab Jamahiriya	countryLY
Liechtenstein	countryLI	Lithuania	countryLT
Luxembourg	countryLU	Macau	countryMO
Macedonia, The Former Yugoslav Republic of	countryMK	Madagascar	countryMG
Malawi	countryMW	Malaysia	countryMY
Maldives	countryMV	Mali	countryML
Malta	countryMT	Marshall Islands	countryMH
Martinique	countryMQ	Mauritania	countryMR
Mauritius	countryMU	Mayotte	countryYT
Mexico	countryMX	Micronesia, Federated States of	countryFM
Moldova	countryMD	Monaco	countryMC
Mongolia	countryMN	Montserrat	countryMS

Morocco	countryMA	Mozambique	countryMZ
Myanmar	countryMM	Namibia	countryNA
Nauru	countryNR	Nepal	countryNP
Netherlands	countryNL	Netherlands Antilles	countryAN
New Caledonia	countryNC	New Zealand	countryNZ
Nicaragua	countryNI	Niger	countryNE
Nigeria	countryNG	Niue	countryNU
Norfolk Island	countryNF	Northern Mariana Islands	countryMP
Norway	countryNO	Oman	countryOM
Pakistan	countryPK	Palau	countryPW
Palestine	countryPS	Panama	countryPA
Papua New Guinea	countryPG	Paraguay	countryPY
Peru	countryPE	Philippines	countryPH
Pitcairn	countryPN	Poland	countryPL
Portugal	countryPT	Puerto Rico	countryPR
Qatar	countryQA	Reunion	countryRE
Romania	countryRO	Russian Federation	countryRU
Rwanda	countryRW	Saint Kitts and Nevis	countryKN
Saint Lucia	countryLC	Saint Vincent and the Grenadines	countryVC
Samoa	countryWS	San Marino	countrySM
Sao Tome and Principe	countryST	Saudi Arabia	countrySA
Senegal	countrySN	Seychelles	countrySC
Sierra Leone	countrySL	Singapore	countrySG
Slovakia (Slovak Republic)	countrySK	Slovenia	countrySI
Solomon Islands	countrySB	Somalia	countrySO
South Africa	countryZA	South Georgia and the South Sandwich Islands	countryGS
Spain	countryES	Sri Lanka	countryLK
St. Helena	countrySH	St. Pierre and Miquelon	countryPM
Sudan	countrySD	Suriname	countrySR
Svalbard and Jan Mayen Islands	countrySJ	Swaziland	countrySZ
Sweden	countrySE	Switzerland	countryCH
Syria	countrySY	Taiwan	countryTW
Tajikistan	countryTJ	Tanzania	countryTZ
Thailand	countryTH	Togo	countryTG
Tokelau	countryTK	Tonga	countryTO
Trinidad and Tobago	countryTT	Tunisia	countryTN
Turkey	countryTR	Turkmenistan	countryTM
Turks and Caicos Islands	countryTC	Tuvalu	countryTV
Uganda	countryUG	Ukraine	countryUA
United Arab Emirates	countryAE	United Kingdom	countryUK

United States	countryUS	United States Minor Outlying Islands	countryUM
Uruguay	countryUY	Uzbekistan	countryUZ
Vanuatu	countryVU	Venezuela	countryVE
Vietnam	countryVN	Virgin Islands (British)	countryVG
Virgin Islands (U.S.)	countryVI	Wallis and Futuna Islands	countryWF
Western Sahara	countryEH	Yemen	countryYE
Yugoslavia	countryYU	Zaire	countryZR
Zambia	countryZM		

Google also has four topic restricts:

Topic	<restrict> value
US. Government	unclesam
Linux	linux
Macintosh	mac
FreeBSD	bsd

Combining the <lr> and <restrict> parameters

Search requests which use the **`lr`** and **`restrict`** parameters support the Boolean operators identified in the items below (in order of precedence).

Note: If both **`lr`** and **`restrict`** parameters are used in a search request, the sub-collection strings will be combined together using "AND" logic.

- **Boolean NOT [-]**
  Ex: **`-lang_fr`**
  Removes all results which are defined as part of the sub-collection immediately following the "-" operator. The example **`restrict`** value would remove all results in French.

- **Boolean AND [.]**
  Ex: **`linux.countryFR`**
  Returns results which are in the intersection of the results returned by the sub-collection to either side of the "." operator. The example **`restrict`** value would return all results which are from both the "linux" subtopic and identified as being located in France.

- **Boolean OR [|]**
  Ex: **`lang_en|lang_fr`**
  Returns results which are in either of the results returned by the sub-collection to either side of the "|" operator. The example **`restrict`** value would return all results matching the query that are in either the French or English sub-collections.

- **Parentheses [( )]**
  Ex: **`(linux).(-(conutryUK|countryUS))`**
  All terms within the innermost set of parentheses in a sub-collection string will be evaluated before terms outside the parentheses are evaluated. Use parentheses to adjust the order of term evaluation. The example **`restrict`** value would return all results in the "linux" custom sub-collection that are not in either the United States or

United Kingdom sub-collections.
Note: Spaces are not valid characters in the `restrict` parameter.

A.2.5 Input and Output Encodings - <ie>, <oe>

In order to support searching documents in multiple languages and character encodings the Google Web APIs perform all requests and responses in the UTF-8 encoding. The parameters <ie> and <oe> are required in client requests but their values are ignored. Clients should encode all request data in UTF-8 and should expect results to be in UTF-8.

A.2.6 SafeSearch - <safeSearch>

Many Google users prefer not to have adult sites included in their search results. Google's SafeSearch feature screens for sites that contain this type of information and eliminates them from search results. While no filter is 100% accurate, Google's filter uses advanced proprietary technology that checks keywords and phrases, URLs, and Open Directory categories.

If you find sites that contain offensive content in your results, even with SafeSearch activated; please send an email with the site's URL to safesearch@google.com and Google will investigate it.

A.2.7 Limitations

There are some important limitations you should be aware of. Some of these are because Google's infrastructure is currently optimized for end users. However, in the future we hope to vastly increase the limits for Google Web APIs developers.

Component	**Limit**
Search request length	2048 bytes
Maximum number of words in the query	10
Maximum number of *site:* terms in the query	1 (per search request)
Maximum number of results per query	10
Maximum value of <start> + <maxResults>	1000

A.3 Search Results Format

A.3.1 Search Response

Each time you issue a search request to the Google service, a response is returned back to you. This section describes the meanings of the values returned to you.

<**documentFiltering**> A Boolean value indicating whether filtering was performed on the search results. This will be "true" only if (a) you requested filtering and (b) filtering actually occurred.

<**searchComments**> A text string intended for display to an end user. One of the most common messages found here is a note that "stop words" were removed from the search automatically. (This happens for very common words such as "and" and "as.")

<**estimatedTotalResultsCount**> The estimated total number of results that exist for the query. **Note:** The estimated number may be either higher or lower than the actual number of results that exist.

<**estimateIsExact**> A Boolean value indicating that the estimate is actually the exact value.

<**resultElements**> An array of <resultElement> items. This corresponds to the actual list of search results.

<**searchQuery**> This is the value of <q> for the search request.

<**startIndex**> Indicates the index (1-based) of the *first* search result in <resultElements>.

<**endIndex**> Indicates the index (1-based) of the *last* search result in <resultElements>.

<**searchTips**> A text string intended for display to the end user. It provides instructive suggestions on how to use Google.

<**directoryCategories**> An array of <directoryCategory> items. This corresponds to the ODP directory matches for this search.

<**searchTime**> Text, floating-point number indicating the total server time to return the search results, measured in seconds.

A.3.2 Result Element

<**summary**> If the search result has a listing in the ODP directory, the ODP summary appears here as a text string.

<**URL**> The URL of the search result, returned as text, with an absolute URL path.

<**snippet**> A snippet which shows the query in context on the URL where it appears. This is formatted HTML and usually includes <B> tags within it. Note that the query term does not always appear in the snippet. **Note:** Query terms will be in highlighted in bold in the results, and line breaks will be included for proper text wrapping.

<**title**> The title of the search result, returned as HTML.

<**cachedSize**> Text (Integer + ”k”). Indicates that a cached version of the <URL> is available; size is indicated in kilobytes.

<**relatedInformationPresent**> Boolean indicating that the *”related:”* query term is supported for this URL.

<**hostName**> When filtering occurs, a maximum of two results from any given host is returned. When this occurs, the second resultElement that comes from that host contains the host name in this parameter.

<**directoryCategory**> See below.

<**directoryTitle**> If the URL for this resultElement is contained in the ODP directory, the title that appears in the directory appears here as a text string. Note that the directoryTitle may be different from the URL’s <title>.

A.3.3 Directory Category

<fullViewableName> Text, containing the ODP directory name for the current ODP category.

<specialEncoding> Specifies the encoding scheme of the directory information.

B VDM-SL Specification of the Google Web Service

This appendix contains the VDM-SL specification of the Google Web APIs, derived from the official description (`GoogleSearch.wsdl`), also embeded here.

```
<?xml version="1.0"?>

<!-- WSDL description of the Google Web APIs.
     The Google Web APIs are in beta release. All interfaces are subject to
     change as we refine and extend our APIs. Please see the terms of use
     for more information. -->

<!-- Revision 2002-08-16 -->

<definitions name="GoogleSearch"
             targetNamespace="urn:GoogleSearch"
             xmlns:typens="urn:GoogleSearch"
             xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
             xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
             xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
             xmlns="http://schemas.xmlsoap.org/wsdl/">

  <!-- Types for search - result elements, directory categories -->
```

```
module GoogleSearch
  exports all
definitions
```

B.1 Data types

```
  <types>
    <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
                targetNamespace="urn:GoogleSearch">

      <xsd:complexType name="GoogleSearchResult">
        <xsd:all>
          <xsd:element name="documentFiltering"           type="xsd:boolean"/>
          <xsd:element name="searchComments"              type="xsd:string"/>
          <xsd:element name="estimatedTotalResultsCount"  type="xsd:int"/>
          <xsd:element name="estimateIsExact"             type="xsd:boolean"/>
          <xsd:element name="resultElements"              type="typens:ResultElementArray"/>
```

```
    <xsd:element name="searchQuery"                type="xsd:string"/>
    <xsd:element name="startIndex"                 type="xsd:int"/>
    <xsd:element name="endIndex"                   type="xsd:int"/>
    <xsd:element name="searchTips"                 type="xsd:string"/>
    <xsd:element name="directoryCategories"        type="typens:DirectoryCategoryArray"/>
    <xsd:element name="searchTime"                 type="xsd:double"/>
  </xsd:all>
</xsd:complexType>
```

$$
\begin{aligned}
\mathit{GoogleSearchResult} :: \ & \mathit{documentFiltering} : \mathbb{B} \\
& \mathit{searchComments} : \mathit{string} \\
& \mathit{estimatedTotalResultCount} : \mathbb{Z} \\
& \mathit{estimateIsExact} : \mathbb{B} \\
& \mathit{resultElements} : \mathit{ResultElementArray} \\
& \mathit{searchQuery} : \mathit{string} \\
& \mathit{startIndex} : \mathbb{Z} \\
& \mathit{endIndex} : \mathbb{Z} \\
& \mathit{searchTips} : \mathit{string} \\
& \mathit{directoryCategories} : \mathit{DirectoryCategoryArray} \\
& \mathit{searchTime} : \mathbb{R}
\end{aligned}
$$

```
<xsd:complexType name="ResultElement">
  <xsd:all>
    <xsd:element name="summary" type="xsd:string"/>
    <xsd:element name="URL" type="xsd:string"/>
    <xsd:element name="snippet" type="xsd:string"/>
    <xsd:element name="title" type="xsd:string"/>
    <xsd:element name="cachedSize" type="xsd:string"/>
    <xsd:element name="relatedInformationPresent" type="xsd:boolean"/>
    <xsd:element name="hostName" type="xsd:string"/>
    <xsd:element name="directoryCategory" type="typens:DirectoryCategory"/>
    <xsd:element name="directoryTitle" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
```

$$
\begin{aligned}
\mathit{ResultElement} :: \ & \mathit{summary} : \mathit{string} \\
& \mathit{URL} : \mathit{string} \\
& \mathit{snippet} : \mathit{string} \\
& \mathit{title} : \mathit{string} \\
& \mathit{cachedSize} : \mathit{string} \\
& \mathit{relatedInformationPresent} : \mathbb{B} \\
& \mathit{hostName} : \mathit{string} \\
& \mathit{directoryCategory} : \mathit{DirectoryCategory} \\
& \mathit{directoryTitle} : \mathit{string}
\end{aligned}
$$

```
<xsd:complexType name="ResultElementArray">
  <xsd:complexContent>
    <xsd:restriction base="soapenc:Array">
```

```
        <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="typens:ResultElement[]"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
```

$ResultElementArray = ResultElement^*;$

```
  <xsd:complexType name="DirectoryCategoryArray">
    <xsd:complexContent>
      <xsd:restriction base="soapenc:Array">
        <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="typens:DirectoryCategory[]"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
```

$DirectoryCategoryArray = DirectoryCategory^*;$

```
  <xsd:complexType name="DirectoryCategory">
    <xsd:all>
      <xsd:element name="fullViewableName" type="xsd:string"/>
      <xsd:element name="specialEncoding" type="xsd:string"/>
    </xsd:all>
  </xsd:complexType>

 </xsd:schema>
</types>
```

$$\begin{aligned} DirectoryCategory :: & fullViewableName : string \\ & specialEncoding : string \end{aligned}$$

$string = \text{char}^*;$
$base64Binary = \text{char}^*$

B.2 Functions

```
<!-- Messages for Google Web APIs - cached page, search, spelling. -->

<message name="doGetCachedPage">
  <part name="key"            type="xsd:string"/>
  <part name="url"            type="xsd:string"/>
</message>

<message name="doGetCachedPageResponse">
  <part name="return"         type="xsd:base64Binary"/>
```

```
</message>

<message name="doSpellingSuggestion">
  <part name="key"            type="xsd:string"/>
  <part name="phrase"         type="xsd:string"/>
</message>

<message name="doSpellingSuggestionResponse">
  <part name="return"         type="xsd:string"/>
</message>

<!-- note, ie and oe are ignored by server; all traffic is UTF-8. -->

<message name="doGoogleSearch">
  <part name="key"            type="xsd:string"/>
  <part name="q"              type="xsd:string"/>
  <part name="start"          type="xsd:int"/>
  <part name="maxResults"     type="xsd:int"/>
  <part name="filter"         type="xsd:boolean"/>
  <part name="restrict"       type="xsd:string"/>
  <part name="safeSearch"     type="xsd:boolean"/>
  <part name="lr"             type="xsd:string"/>
  <part name="ie"             type="xsd:string"/>
  <part name="oe"             type="xsd:string"/>
</message>

<message name="doGoogleSearchResponse">
  <part name="return"         type="typens:GoogleSearchResult"/>
</message>

<!-- Port for Google Web APIs, "GoogleSearch" -->
```

Note that every function takes a `key` parameter. That is a requirement of Google's API, to ensure some level of security, it doesn't have nothing to do with the functionality provided.

B.2.1 Function $doGetCachedPage$

Specification:

$doGetCachedPage\ (key : string, url : string)\ result : base64Binary$
post
true;

Description:

This function takes a URL as input and delivers the contents of the URL when Google's crawlers last visited the page (if available).
(See section A.1.2 for more details.)

Calls:

Standard VDM-SL only

```
<portType name="GoogleSearchPort">

  <operation name="doGetCachedPage">
    <input message="typens:doGetCachedPage"/>
    <output message="typens:doGetCachedPageResponse"/>
  </operation>
```

B.2.2 Function *doSpellingSuggestion*

Specification:

$doSpellingSuggestion\ (key : string, phrase : string)\ result : string$
post
true;

Description:

This function takes a query string as input and returns a string with the suggested spell correction for that query (if available).
(See section A.1.3 for more details.)

Calls:

Standard VDM-SL only

```
  <operation name="doSpellingSuggestion">
    <input message="typens:doSpellingSuggestion"/>
    <output message="typens:doSpellingSuggestionResponse"/>
  </operation>
```

B.2.3 Function *doGoogleSearch*

Specification:

$doGoogleSearch\ (key : string, q : string, start : \mathbb{Z}, maxResults : \mathbb{Z}, filter : \mathbb{B}, restrict : string, safeSearch : \mathbb{B}, lr : string, ie : string, oe : string)\ result : GoogleSearchResult$
post
true

Description:

This function takes a query string and a set of parameters (such as the language restrict (lr), the topic and country restrict (restrict), etc) as input and returns a GoogleSearchResult (p. 16) that contains, among other things, a set of Result Elements (p. 16).
(See section A.1.1 for more details.)

Calls:

```
  <operation name="doGoogleSearch">
    <input message="typens:doGoogleSearch"/>
    <output message="typens:doGoogleSearchResponse"/>
  </operation>

</portType>


<!-- Binding for Google Web APIs - RPC, SOAP over HTTP -->

<binding name="GoogleSearchBinding" type="typens:GoogleSearchPort">
  <soap:binding style="rpc"
              transport="http://schemas.xmlsoap.org/soap/http"/>

  <operation name="doGetCachedPage">
    <soap:operation soapAction="urn:GoogleSearchAction"/>
    <input>
      <soap:body use="encoded"
                 namespace="urn:GoogleSearch"
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </input>
    <output>
      <soap:body use="encoded"
                 namespace="urn:GoogleSearch"
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </output>
  </operation>

  <operation name="doSpellingSuggestion">
    <soap:operation soapAction="urn:GoogleSearchAction"/>
    <input>
      <soap:body use="encoded"
                 namespace="urn:GoogleSearch"
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </input>
    <output>
      <soap:body use="encoded"
                 namespace="urn:GoogleSearch"
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </output>
  </operation>

  <operation name="doGoogleSearch">
    <soap:operation soapAction="urn:GoogleSearchAction"/>
    <input>
      <soap:body use="encoded"
                 namespace="urn:GoogleSearch"
                 encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </input>
```

```
      <output>
        <soap:body use="encoded"
                   namespace="urn:GoogleSearch"
                   encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
      </output>
    </operation>
  </binding>

  <!-- Endpoint for Google Web APIs -->
  <service name="GoogleSearchService">
    <port name="GoogleSearchPort" binding="typens:GoogleSearchBinding">
      <soap:address location="http://api.google.com/search/beta2"/>
    </port>
  </service>

</definitions>
```

end $GoogleSearch$

C VDM-SL Specification of the Google Query Syntax

This section contains the VDM-SL specification of the Google Query Syntax, derived from the desciption presented in section A.2.2.

module $GoogleSearchQuery$

exports all

definitions

C.1 Data types

$Query = NormalQuery \mid AllInQuery \mid InfoLinkRelated \mid Cache;$

$$
\begin{aligned}
&\mathit{NormalQuery} :: \mathit{site} : [\mathit{Site}]\\
&\qquad \mathit{daterange} : [\mathit{DateRange}]\\
&\qquad \mathit{filetype} : [\mathit{FileType}]\\
&\qquad \mathit{terms} : \mathit{Term}^*\\
&\mathit{Site} = \mathit{Word};\\
&\mathit{DateRange} :: \mathit{from\text{-}date} : \mathbb{N}\\
&\qquad \mathit{to\text{-}date} : \mathbb{N}\\
&\mathit{FileType} = \mathit{OnlyTypes} \mid \mathit{ExceptTypes};\\
&\mathit{OnlyTypes} :: \quad \mathit{Word}^+;\\
&\mathit{ExceptTypes} :: \quad \mathit{Word}^+;\\
&\mathit{Term} :: \quad \mathit{NormalTerm} \mid \mathit{ExcludeTerm} \mid \mathit{ExcludeSite};\\
&\mathit{ExcludeTerm} :: \quad \mathit{Word};\\
&\mathit{ExcludeSite} :: \quad \mathit{Word};\\
&\mathit{NormalTerm} :: \quad \mathit{IncludeTerm} \mid \mathit{PhraseTerm} \mid \mathit{ORTerm} \mid \mathit{InTitleTerm} \mid \mathit{InUrlTerm} \mid \mathit{Word};\\
&\mathit{IncludeTerm} :: \quad \mathit{Word};\\
&\mathit{PhraseTerm} :: \quad \mathit{string};\\
&\mathit{ORTerm} :: \mathit{left} : \mathit{NormalTerm}\\
&\qquad \mathit{right} : \mathit{NormalTerm}\\
&\mathit{InTitleTerm} :: \quad \mathit{Word};\\
&\mathit{InUrlTerm} :: \quad \mathit{Word};\\
&\mathit{Word} = \mathit{string}\\
&\mathsf{inv}\ w \triangleq \forall\, c \in (\mathsf{elems}\ w) \cdot c \neq \text{' '} \wedge c \neq \text{'}\backslash t\text{'} \wedge c \neq \text{'}\backslash n\text{'};
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{AllInQuery} :: \mathit{prefix} : \mathrm{ALLINTITLE} \mid \mathrm{ALLINURL} \mid \mathrm{ALLINTEXT} \mid \mathrm{ALLINLINKS}\\
&\qquad \mathit{query} : \mathit{string}
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{InfoLinkRelated} :: \mathit{prefix} : \mathrm{INFO} \mid \mathrm{LINK} \mid \mathrm{RELATED}\\
&\qquad \mathit{url} : \mathit{URL}
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{Cache} :: \mathit{url} : \mathit{URL}\\
&\qquad \mathit{words} : \mathit{Word}^*\\
&\mathit{string} = \mathsf{char}^*;\\
&\mathit{URL} = \mathit{Word}
\end{aligned}
$$

C.2 Functions

C.2.1 Function $\mathit{queryToString}$

Specification:

$$
\begin{aligned}
&\mathit{queryToString} : \mathit{Query} \to \mathit{string}\\
&\mathit{queryToString}\,(\mathit{gsq}) \triangleq\\
&\quad \mathsf{if\ is\text{-}}\mathit{NormalQuery}\,(\mathit{gsq})\\
&\quad \mathsf{then}\ \mathit{normalQueryToString}\,(\mathit{gsq})\\
&\quad \mathsf{else\ if\ is\text{-}}\mathit{AllInQuery}\,(\mathit{gsq})\\
&\qquad \mathsf{then}\ \mathit{allInQueryToString}\,(\mathit{gsq})\\
&\qquad \mathsf{else\ if\ is\text{-}}\mathit{InfoLinkRelated}\,(\mathit{gsq})\\
&\qquad\quad \mathsf{then}\ \mathit{infoLinkRelatedToString}\,(\mathit{gsq})\\
&\qquad\quad \mathsf{else}\ \mathit{cacheToString}\,(\mathit{gsq});
\end{aligned}
$$

Description:

This function takes a Query (p. 21) and returns a query string ready to be passed to the function doGoogleSearch (p. 19). This function requires all the others functions specified in this section to construct the various types of queries (consult p. 21).

Calls:

Standard VDM-SL only

C.2.2 Function $normalQueryToString$

Specification:

$$
\begin{array}{l}
normalQueryToString : NormalQuery \to string \\
normalQueryToString\,(\mathsf{mk}\text{-}NormalQuery\,(site, date\text{-}rng, filetype, terms)) \triangleq \\
\quad siteToString\,(site) \curvearrowright \texttt{" "} \curvearrowright \\
\quad dateRangeToString\,(date\text{-}rng) \curvearrowright \\
\quad \texttt{" "} \curvearrowright \\
\quad fileTypeToString\,(filetype) \curvearrowright \\
\quad \texttt{" "} \curvearrowright \\
\quad termsToString\,(terms);
\end{array}
$$

Description:

Auxiliary function of queryToString (p. 22).

Calls:

Standard VDM-SL only

C.2.3 Function $siteToString$

Specification:

$$
\begin{array}{l}
siteToString : [Site] \to string \\
siteToString\,(s) \triangleq \\
\quad \mathsf{if}\ s = \mathsf{nil} \\
\quad \mathsf{then}\ \texttt{""} \\
\quad \mathsf{else}\ \texttt{"}site:\texttt{"} \curvearrowright s;
\end{array}
$$

Description:

Auxiliary function of normalQueryToString (p. 23).

Calls:

Standard VDM-SL only

C.2.4 Function *dateRangeToString*

Specification:

$dateRangeToString : [DateRange] \to string$
$dateRangeToString\,(dr) \triangleq$
cases (dr) :
mk-$DateRange\,(fd, td) \to$ " $daterange$: " $\curvearrowright natToString\,(fd) \curvearrowright$
"-" $\curvearrowright natToString\,(td)$,
nil $\to$ ""
end;

Description:

Auxiliary function of normalQueryToString (p. 23).

Calls:

Standard VDM-SL only

C.2.5 Function *natToString*

Specification:

$natToString : \mathbb{N} \to string$
$natToString\,(n) \triangleq$
cases (n) :
$0 \to$ "0",
$1 \to$ "1",
$2 \to$ "2",
$3 \to$ "3",
$4 \to$ "4",
$5 \to$ "5",
$6 \to$ "6",
$7 \to$ "7",
$8 \to$ "8",
$9 \to$ "9",
others $\to natToString\,(n \text{ div } 10) \curvearrowright natToString\,(n \text{ mod } 10)$
end;

Description:

Auxiliary function of dateRangeToString (p. 23).

Calls:

Standard VDM-SL only

C.2.6 Function *fileTypeToString*

Specification:

$$
\begin{array}{l}
\mathit{fileTypeToString} : [\mathit{FileType}] \to \mathit{string} \\
\mathit{fileTypeToString}\,(\mathit{ft}) \ \triangleq \\
\quad \mathsf{cases}\ (\mathit{ft}) : \\
\quad\quad \mathsf{mk}\text{-}\mathit{OnlyTypes}\,(\mathit{ots}) \to \mathit{onlyTypesToString}\,(\mathit{ots}), \\
\quad\quad \mathsf{mk}\text{-}\mathit{ExceptTypes}\,(\mathit{ets}) \to \mathit{exceptTypesToString}\,(\mathit{ets}), \\
\quad\quad \mathsf{nil} \to \texttt{""} \\
\quad \mathsf{end};
\end{array}
$$

Description:

Auxiliary function of normalQueryToString (p. 23).

Calls:

Standard VDM-SL only

C.2.7 Function *onlyTypesToString*

Specification:

$$
\begin{array}{l}
\mathit{onlyTypesToString} : \mathit{Word}^{+} \to \mathit{string} \\
\mathit{onlyTypesToString}\,(\mathit{ots}) \ \triangleq \\
\quad \mathsf{if}\ \mathsf{len}\,(\mathit{ots}) = 1 \\
\quad \mathsf{then}\ \texttt{"}\ \mathit{filetype} :\texttt{"} \frown \mathsf{hd}\,(\mathit{ots}) \\
\quad \mathsf{else}\ \texttt{"}\ \mathit{filetype} :\texttt{"} \frown \mathsf{hd}\,(\mathit{ots}) \frown \texttt{"}\ \mathit{OR}\ \texttt{"} \frown \mathit{onlyTypesToString}\,(\mathsf{tl}\ \mathit{ots});
\end{array}
$$

Description:

Auxiliary function of fileTypeToString (p. 24).

Calls:

Standard VDM-SL only

C.2.8 Function *exceptTypesToString*

Specification:

$$
\begin{array}{l}
\mathit{exceptTypesToString} : \mathit{Word}^{*} \to \mathit{string} \\
\mathit{exceptTypesToString}\,(\mathit{ets}) \ \triangleq \\
\quad \mathsf{if}\ \mathit{ets} = [\,] \\
\quad \mathsf{then}\ \texttt{""} \\
\quad \mathsf{else}\ \texttt{"}\ \text{-}\mathit{filetype} :\texttt{"} \frown \mathsf{hd}\,(\mathit{ets}) \frown \texttt{"}\ \texttt{"} \frown \mathit{exceptTypesToString}\,(\mathsf{tl}\ \mathit{ets});
\end{array}
$$

Description:

Auxiliary function of fileTypeToString (p. 24).

Calls:

Standard VDM-SL only

C.2.9 Function $termsToString$

Specification:

$termsToString : Term^* \to string$
$termsToString\,(ts) \triangleq$
  if $ts = []$
  then `""`
  else $termToString$ (hd ts) $\curvearrowright$ `" "` $\curvearrowright$ $termsToString$ (tl ts);

Description:

Auxiliary function of normalQueryToString (p. 23).

Calls:

Standard VDM-SL only

C.2.10 Function $termToString$

Specification:

$termToString : Term \to string$
$termToString$ (mk-$Term\,(t)$) $\triangleq$
  cases (t) :
    mk-$NormalTerm\,(nt) \to normalTermToString$ (mk-$NormalTerm\,(nt)$),
    mk-$ExcludeTerm\,(et) \to$ `"-"` $\curvearrowright et$,
    mk-$ExcludeSite\,(se) \to$ `"-`$site$ `:"` $\curvearrowright se$
  end;

Description:

Auxiliary function of termsToString (p. 25).

Calls:

Standard VDM-SL only

C.2.11 Function $normalTermToString$

Specification:

$normalTermToString : NormalTerm \to string$
$normalTermToString$ (mk-$NormalTerm\,(nt)$) $\triangleq$
  cases (nt) :
    mk-$IncludeTerm\,(it) \to$ `" + "` $\curvearrowright it$,
    mk-$PhraseTerm\,(pt) \to$ [’"’] $\curvearrowright pt \curvearrowright$ [’"’],
    mk-$ORTerm\,(l, r) \to normalTermToString\,(l)$ $\curvearrowright$ `" OR "` $\curvearrowright$
$normalTermToString\,(r)$,
    mk-$InTitleTerm\,(itt) \to$ `"`$intitle$ `:"` $\curvearrowright itt$,
    mk-$InUrlTerm\,(iut) \to$ `"`$inurl$ `:"` $\curvearrowright iut$,
    others $\to nt$
  end;

Description:

Auxiliary function of termToString (p. 26).

Calls:

Standard VDM-SL only

C.2.12 Function $allInQueryToString$

Specification:

$$
\begin{array}{l}
allInQueryToString : AllInQuery \to string \\
allInQueryToString\,(\mathsf{mk}\text{-}AllInQuery\,(prefix, q)) \triangleq \\
\quad \mathsf{cases}\ (prefix) : \\
\qquad \text{ALLINTITLE} \to \texttt{"}allintitle: \texttt{"} \frown q, \\
\qquad \text{ALLINURL} \to \texttt{"}allinurl: \texttt{"} \frown q, \\
\qquad \text{ALLINTEXT} \to \texttt{"}allintext: \texttt{"} \frown q, \\
\qquad \text{ALLINLINKS} \to \texttt{"}allinlinks: \texttt{"} \frown q \\
\quad \mathsf{end};
\end{array}
$$

Description:

Auxiliary function of queryToString (p. 22).

Calls:

Standard VDM-SL only

C.2.13 Function $infoLinkRelatedToString$

Specification:

$$
\begin{array}{l}
infoLinkRelatedToString : InfoLinkRelated \to string \\
infoLinkRelatedToString\,(\mathsf{mk}\text{-}InfoLinkRelated\,(prefix, url)) \triangleq \\
\quad \mathsf{cases}\ (prefix) : \\
\qquad \text{INFO} \to \texttt{"}info: \texttt{"} \frown url, \\
\qquad \text{LINK} \to \texttt{"}link: \texttt{"} \frown url, \\
\qquad \text{RELATED} \to \texttt{"}related: \texttt{"} \frown url \\
\quad \mathsf{end};
\end{array}
$$

Description:

Auxiliary function of queryToString (p. 22).

Calls:

Standard VDM-SL only

C.2.14 Function $cacheToString$

Specification:

$$
\begin{array}{l}
cacheToString : Cache \to string \\
cacheToString\,(\text{mk-}Cache\,(url, ws)) \triangleq \\
\quad \texttt{"}cache : \texttt{"} \frown url \frown \texttt{" "} \frown wordsToString\,(ws);
\end{array}
$$

Description:

Auxiliary function of queryToString (p. 22).

Calls:

Standard VDM-SL only

C.2.15 Function $wordsToString$

Specification:

$$
\begin{array}{l}
wordsToString : Word^{*} \to string \\
wordsToString\,(ws) \triangleq \\
\quad \text{if } ws = [\,] \\
\quad \text{then } \texttt{""} \\
\quad \text{else } (\text{hd } ws) \frown \texttt{" "} \frown wordsToString\,(\text{tl } ws)
\end{array}
$$

Description:

Auxiliary function of cacheToString (p. 27).

Calls:

Standard VDM-SL only

C.3 Example Values

In this section we have instances of the Query data type (see C.1) that correspond to the example queries presented in appendix A (page 5).

- **Include Query Term** (See page 5 for details.)

$$
\begin{array}{l}
\text{values} \\
\quad include\text{-}term\text{-}I : NormalTerm = \text{mk-}NormalTerm\,(\text{mk-}IncludeTerm\,(\texttt{"}I\texttt{"})); \\
\quad include\text{-}term\text{-}query : NormalQuery = \text{mk-}NormalQuery\,(\text{nil}, \\
\qquad \text{nil}, \\
\qquad \text{nil}, \\
\qquad [\text{mk-}Term\,(\text{mk-}NormalTerm\,(\texttt{"}Star\texttt{"})), \\
\qquad \text{mk-}Term\,(\text{mk-}NormalTerm\,(\texttt{"}Wars\texttt{"})), \\
\qquad \text{mk-}Term\,(\text{mk-}NormalTerm\,(\texttt{"}Episode\texttt{"})), \\
\qquad \text{mk-}Term\,(include\text{-}term\text{-}I)]);
\end{array}
$$

- **Exclude Query Term** (See page 5 for details.)

$$
\begin{aligned}
&\mathit{exclude\text{-}term\text{-}music} : \mathit{ExcludeTerm} = \mathsf{mk}\text{-}\mathit{ExcludeTerm}\,(\texttt{"}\mathit{music}\texttt{"});\\
&\mathit{exclude\text{-}term\text{-}query} : \mathit{NormalQuery} = \mathsf{mk}\text{-}\mathit{NormalQuery}\,(\mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad [\mathsf{mk}\text{-}\mathit{Term}\,(\mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{bass}\texttt{"})),\\
&\qquad\ \mathsf{mk}\text{-}\mathit{Term}\,(\mathit{exclude\text{-}term\text{-}music})]);
\end{aligned}
$$

- **Phrase Search** (See page 6 for details.)

$$
\begin{aligned}
&\mathit{phrase\text{-}term} : \mathit{NormalTerm} = \mathsf{mk}\text{-}\mathit{NormalTerm}\,(\mathsf{mk}\text{-}\mathit{PhraseTerm}\,(\texttt{"}\mathit{yellow\ pages}\texttt{"}));\\
&\mathit{phrase\text{-}search\text{-}query} : \mathit{NormalQuery} = \mathsf{mk}\text{-}\mathit{NormalQuery}\,(\mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad [\mathsf{mk}\text{-}\mathit{Term}\,(\mathit{phrase\text{-}term})]);
\end{aligned}
$$

- **Boolean OR Search** (See page 6 for details.)

$$
\begin{aligned}
&\mathit{or\text{-}term} : \mathit{NormalTerm} = \mathsf{mk}\text{-}\mathit{NormalTerm}\,(\mathsf{mk}\text{-}\mathit{ORTerm}\,(\mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{london}\texttt{"}),\\
&\qquad \mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{paris}\texttt{"})));\\
&\mathit{boolean\text{-}or\text{-}query} : \mathit{NormalQuery} = \mathsf{mk}\text{-}\mathit{NormalQuery}\,(\mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad [\mathsf{mk}\text{-}\mathit{Term}\,(\mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{vacation}\texttt{"})),\\
&\qquad\ \mathsf{mk}\text{-}\mathit{Term}\,(\mathit{or\text{-}term})]);
\end{aligned}
$$

- **Site Restricted Search** (See page 6 for details.)

$$
\begin{aligned}
&\mathit{site\text{-}restricted\text{-}query} : \mathit{NormalQuery} = \mathsf{mk}\text{-}\mathit{NormalQuery}\,(\texttt{"}\mathit{www.stanford.edu}\texttt{"},\\
&\qquad \mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad [\mathsf{mk}\text{-}\mathit{Term}\,(\mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{admission}\texttt{"}))]);
\end{aligned}
$$

- **Date Restricted Search** (See page 6 for details.)

$$
\begin{aligned}
&\mathit{date\text{-}restricted\text{-}query} : \mathit{NormalQuery} = \mathsf{mk}\text{-}\mathit{NormalQuery}\,(\mathsf{nil}\,,\\
&\qquad \mathsf{mk}\text{-}\mathit{DateRange}\,(2452122, 2452234),\\
&\qquad \mathsf{nil}\,,\\
&\qquad [\mathsf{mk}\text{-}\mathit{Term}\,(\mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{Star}\texttt{"})),\\
&\qquad\ \mathsf{mk}\text{-}\mathit{Term}\,(\mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{Wars}\texttt{"}))]);
\end{aligned}
$$

- **Title Search (term)** (See page 6 for details.)

$$
\begin{aligned}
&\mathit{intitle\text{-}google} : \mathit{NormalTerm} = \mathsf{mk}\text{-}\mathit{NormalTerm}\,(\mathsf{mk}\text{-}\mathit{InTitleTerm}\,(\texttt{"}\mathit{Google}\texttt{"}));\\
&\mathit{intitle\text{-}query} : \mathit{NormalQuery} = \mathsf{mk}\text{-}\mathit{NormalQuery}\,(\mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad [\mathsf{mk}\text{-}\mathit{Term}\,(\mathit{intitle\text{-}google}),\\
&\qquad \ \mathsf{mk}\text{-}\mathit{Term}\,(\mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{search}\texttt{"}))]);
\end{aligned}
$$

- **Title Search (all)** (See page 6 for details.)

$$\mathit{allintitle\text{-}query} : \mathit{AllInQuery} = \mathsf{mk}\text{-}\mathit{AllInQuery}\,(\textsc{allintitle}, \texttt{"}\mathit{Google\ search}\texttt{"});$$

- **URL Search (term)** (See page 6 for details.)

$$
\begin{aligned}
&\mathit{inurl\text{-}google} : \mathit{NormalTerm} = \mathsf{mk}\text{-}\mathit{NormalTerm}\,(\mathsf{mk}\text{-}\mathit{InUrlTerm}\,(\texttt{"}\mathit{Google}\texttt{"}));\\
&\mathit{inurl\text{-}query} : \mathit{NormalQuery} = \mathsf{mk}\text{-}\mathit{NormalQuery}\,(\mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad [\mathsf{mk}\text{-}\mathit{Term}\,(\mathit{inurl\text{-}google}),\\
&\qquad \ \mathsf{mk}\text{-}\mathit{Term}\,(\mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{search}\texttt{"}))]);
\end{aligned}
$$

- **URL Search (all)** (See page 7 for details.)

$$\mathit{allinurl\text{-}query} : \mathit{AllInQuery} = \mathsf{mk}\text{-}\mathit{AllInQuery}\,(\textsc{allinurl}, \texttt{"}\mathit{Google\ search}\texttt{"});$$

- **Text Only Search (all)** (See page 7 for details.)

$$\mathit{allintext\text{-}query} : \mathit{AllInQuery} = \mathsf{mk}\text{-}\mathit{AllInQuery}\,(\textsc{allintext}, \texttt{"}\mathit{Google\ search}\texttt{"});$$

- **Links Only Search (all)** (See page 7 for details.)

$$\mathit{allinlinks\text{-}query} : \mathit{AllInQuery} = \mathsf{mk}\text{-}\mathit{AllInQuery}\,(\textsc{allinlinks}, \texttt{"}\mathit{Google\ search}\texttt{"});$$

- **File Type Filtering** (See page 7 for details.)

$$
\begin{aligned}
&\mathit{only\text{-}doc\text{-}or\text{-}pdf} : \mathit{FileType} = \mathsf{mk}\text{-}\mathit{OnlyTypes}\,([\texttt{"}\mathit{doc}\texttt{"}, \texttt{"}\mathit{pdf}\texttt{"}]);\\
&\mathit{filetype\text{-}filtering\text{-}query} : \mathit{NormalQuery} = \mathsf{mk}\text{-}\mathit{NormalQuery}\,(\mathsf{nil}\,,\\
&\qquad \mathsf{nil}\,,\\
&\qquad \mathit{only\text{-}doc\text{-}or\text{-}pdf}\,,\\
&\qquad [\mathsf{mk}\text{-}\mathit{Term}\,(\mathsf{mk}\text{-}\mathit{NormalTerm}\,(\texttt{"}\mathit{Google}\texttt{"}))]);
\end{aligned}
$$

- **File Type Exclusion** (See page 7 for details.)

$$
\begin{array}{l}
\mathit{except\text{-}doc\text{-}pdf} : \mathit{FileType} = \mathsf{mk\text{-}}\mathit{ExceptTypes}\,([\texttt{"}\mathit{doc}\texttt{"}, \texttt{"}\mathit{pdf}\texttt{"}]);\\
\mathit{filetype\text{-}exclusion\text{-}query} : \mathit{NormalQuery} = \mathsf{mk\text{-}}\mathit{NormalQuery}\,(\mathsf{nil}\,,\\
\qquad \mathsf{nil}\,,\\
\qquad \mathit{except\text{-}doc\text{-}pdf}\,,\\
\qquad [\mathsf{mk\text{-}}\mathit{Term}\,(\mathsf{mk\text{-}}\mathit{NormalTerm}\,(\texttt{"}\mathit{Google}\texttt{"}))]);
\end{array}
$$

- **Web Document Info** (See page 7 for details.)

$$\mathit{info\text{-}query} : \mathit{InfoLinkRelated} = \mathsf{mk\text{-}}\mathit{InfoLinkRelated}\,(\textsc{info}, \texttt{"}\mathit{www.google.com}\texttt{"});$$

- **Back Links** (See page 7 for details.)

$$\mathit{link\text{-}query} : \mathit{InfoLinkRelated} = \mathsf{mk\text{-}}\mathit{InfoLinkRelated}\,(\textsc{link}, \texttt{"}\mathit{www.google.com}\texttt{"});$$

- **Related Links** (See page 8 for details.)

$$\mathit{related\text{-}query} : \mathit{InfoLinkRelated} = \mathsf{mk\text{-}}\mathit{InfoLinkRelated}\,(\textsc{related}, \texttt{"}\mathit{www.google.com}\texttt{"});$$

- **Cached Results Page** (See page 8 for details.)

$$\mathit{cache\text{-}query} : \mathit{Cache} = \mathsf{mk\text{-}}\mathit{Cache}\,(\texttt{"}\mathit{www.google.com}\texttt{"}, [\texttt{"}\mathit{web}\texttt{"}])$$

D VDM-SL Specification of the Language Restrict Syntax

This appendix contains the VDM-SL specification of the Language Restrict (<**lr**> parameter) syntax, derived from the desciption presented in section A.2.4.

D.1 Data types

$$
\begin{array}{l}
\mathit{LanguageRestrict} = \mathit{LRBooleanNOT} \mid \mathit{LRBooleanAND} \mid \mathit{LRBooleanOR} \mid \mathit{Language};\\
\mathit{LRBooleanNOT} :: \ \mathit{LanguageRestrict};\\
\mathit{LRBooleanAND} :: \mathit{left} : \mathit{LanguageRestrict}\\
\qquad\qquad \mathit{right} : \mathit{LanguageRestrict}\\
\mathit{LRBooleanOR} :: \mathit{left} : \mathit{LanguageRestrict}\\
\qquad\qquad \mathit{right} : \mathit{LanguageRestrict}\\
\mathit{Language} = \mathit{Word}\\
\mathsf{inv}\ w \triangleq \mathsf{len}\ w = 2
\end{array}
$$

D.2 Functions

D.2.1 Function $languageRestrictToString$

Specification:

$$
\begin{array}{l}
languageRestrictToString : LanguageRestrict \to string \\
languageRestrictToString\,(lr) \ \underline{\triangle} \\
\quad \mathsf{if}\ (\mathsf{is}\text{-}LRBooleanNOT\,(lr)) \\
\quad \mathsf{then}\ LRBooleanNOTToString\,(lr) \\
\quad \mathsf{else\ if}\ (\mathsf{is}\text{-}LRBooleanOR\,(lr)) \\
\qquad \mathsf{then}\ LRBooleanORToString\,(lr) \\
\qquad \mathsf{else\ if}\ (\mathsf{is}\text{-}LRBooleanAND\,(lr)) \\
\qquad\quad \mathsf{then}\ LRBooleanANDToString\,(lr) \\
\qquad\quad \mathsf{else}\ languageToString\,(lr);
\end{array}
$$

Description:

This function takes a LanguageRestrict (p. 31) and returns a string to be passed to the function doGoogleSearch (p. 31) as the ***lr*** *parameter. This function requires all the others functions specified in this section to construct the various forms of language restricts (consult p. 31).*

Calls:

Standard VDM-SL only

D.2.2 Function $LRBooleanNOTToString$

Specification:

$$
\begin{array}{l}
LRBooleanNOTToString : LRBooleanNOT \to string \\
LRBooleanNOTToString\,(\mathsf{mk}\text{-}LRBooleanNOT\,(lr)) \ \underline{\triangle} \\
\quad \texttt{"-("} \curvearrowright languageRestrictToString\,(lr) \curvearrowright \texttt{")"};
\end{array}
$$

Description:

Auxiliary function of languageRestrictToString (p. 32).

Calls:

Standard VDM-SL only

D.2.3 Function $LRBooleanORToString$

Specification:

$$
\begin{array}{l}
LRBooleanORToString : LRBooleanOR \to string \\
LRBooleanORToString\,(\mathsf{mk}\text{-}LRBooleanOR\,(l, r)) \ \underline{\triangle} \\
\quad \texttt{"("} \curvearrowright languageRestrictToString\,(l) \curvearrowright \texttt{")} \quad | \quad \texttt{("} \curvearrowright \\
languageRestrictToString\,(r) \curvearrowright \texttt{")"};
\end{array}
$$

Description:

Auxiliary function of languageRestrictToString (p. 32).

Calls:

Standard VDM-SL only

D.2.4 Function $LRBooleanANDToString$

Specification:

$LRBooleanANDToString : LRBooleanAND \to string$
$LRBooleanANDToString\,(\mathsf{mk\text{-}}LRBooleanAND\,(l, r)) \triangleq$
$\texttt{"("} \frown languageRestrictToString\,(l) \frown \texttt{").("} \frown languageRestrictToString\,(r) \frown \texttt{")"};$

Description:

Auxiliary function of languageRestrictToString (p. 32).

Calls:

Standard VDM-SL only

D.2.5 Function $languageToString$

Specification:

$languageToString : Language \to string$
$languageToString\,(l) \triangleq$
$\texttt{"}lang\_\texttt{"} \frown l$

Description:

Auxiliary function of languageRestrictToString (p. 32).

Calls:

Standard VDM-SL only

D.3 Example Values

In this section we can see example values if the type LanguageRestrict (p. 31). (Consult page 12 for a description of the semantic of this values.)

The following example appers in page 12.

values
$lr\text{-}restrict\text{-}not : LanguageRestrict = \mathsf{mk\text{-}}LRBooleanNOT\,(\texttt{"}fr\texttt{"});$

The following example appers in page 12.

$lr\text{-}restrict\text{-}or : LanguageRestrict = \mathsf{mk\text{-}}LRBooleanOR\,(\texttt{"}en\texttt{"}, \texttt{"}fr\texttt{"});$

$lr\text{-}restrict\text{-}and : LanguageRestrict = \mathsf{mk\text{-}}LRBooleanAND\,(\texttt{"}en\texttt{"}, \texttt{"}fr\texttt{"})$

E VDM-SL Specification of the Restrict Syntax

This appendix contains the VDM-SL specification of the Country and Topic Restricts (the <**restrict**> parameter) syntax, derived from the desciption presented in section A.2.4.

E.1 Data types

$Restrict = BooleanNOT \mid BooleanAND \mid BooleanOR \mid Topic \mid Country;$
$Topic :: \ \text{UNCLESAM} \mid \text{LINUX} \mid \text{MAC} \mid \text{BSD};$
$Country :: \ Word$
$\mathsf{inv}\ \mathsf{mk}\text{-}Country\,(w) \triangleq \mathsf{len}\,(w) = 2;$
$BooleanNOT :: \ Restrict;$
$BooleanAND :: left : Restrict$
$\qquad right : Restrict$
$BooleanOR :: left : Restrict$
$\qquad right : Restrict$

E.2 Functions

E.2.1 Function $restrictToString$

Specification:

$restrictToString : Restrict \rightarrow string$
$restrictToString\,(r) \triangleq$
$\quad \mathsf{if}\ (\mathsf{is}\text{-}BooleanNOT\,(r))$
$\quad \mathsf{then}\ booleanNOTToString\,(r)$
$\quad \mathsf{else\ if}\ (\mathsf{is}\text{-}BooleanOR\,(r))$
$\qquad \mathsf{then}\ booleanORToString\,(r)$
$\qquad \mathsf{else\ if}\ (\mathsf{is}\text{-}BooleanAND\,(r))$
$\qquad\quad \mathsf{then}\ booleanANDToString\,(r)$
$\qquad\quad \mathsf{else\ if}\ (\mathsf{is}\text{-}Topic\,(r))$
$\qquad\qquad \mathsf{then}\ topicToString\,(r)$
$\qquad\qquad \mathsf{else}\ countryToString\,(r);$

Description:

*This function takes a Restrict (p. 34) and returns a string to be passed to the function doGoogleSearch (p. 19) as the **restrict** parameter. This function requires all the others functions specified in this section to construct the various forms of restricts (consult p. 34).*

Calls:

Standard VDM-SL only

E.2.2 Function $booleanNOTToString$

Specification:

$$
\begin{array}{l}
booleanNOTToString : BooleanNOT \to string \\
booleanNOTToString\,(\mathsf{mk}\text{-}BooleanNOT\,(r)) \triangleq \\
\quad \texttt{"-("} \curvearrowright restrictToString\,(r) \curvearrowright \texttt{")"};
\end{array}
$$

Description:

Auxiliary function of restrictToString (p. 34).

Calls:

Standard VDM-SL only

E.2.3 Function *booleanORToString*

Specification:

$$
\begin{array}{l}
booleanORToString : BooleanOR \to string \\
booleanORToString\,(\mathsf{mk}\text{-}BooleanOR\,(l, r)) \triangleq \\
\quad \texttt{"("} \curvearrowright restrictToString\,(l) \curvearrowright \texttt{")|("} \curvearrowright restrictToString\,(r) \curvearrowright \texttt{")"};
\end{array}
$$

Description:

Auxiliary function of restrictToString (p. 34).

Calls:

Standard VDM-SL only

E.2.4 Function *booleanANDToString*

Specification:

$$
\begin{array}{l}
booleanANDToString : BooleanAND \to string \\
booleanANDToString\,(\mathsf{mk}\text{-}BooleanAND\,(l, r)) \triangleq \\
\quad \texttt{"("} \curvearrowright restrictToString\,(l) \curvearrowright \texttt{").("} \curvearrowright restrictToString\,(r) \curvearrowright \texttt{")"};
\end{array}
$$

Description:

Auxiliary function of restrictToString (p. 34).

Calls:

Standard VDM-SL only

E.2.5 Function *topicToString*

Specification:

$$
\begin{array}{l}
topicToString : Topic \to string \\
topicToString\,(\textsf{mk-}Topic\,(t)) \triangleq \\
\quad \textsf{cases}\ (t) : \\
\quad\quad \text{UNCLESAM} \to \texttt{"}unclesam\texttt{"}, \\
\quad\quad \text{LINUX} \to \texttt{"}linux\texttt{"}, \\
\quad\quad \text{MAC} \to \texttt{"}mac\texttt{"}, \\
\quad\quad \text{BSD} \to \texttt{"}bsd\texttt{"} \\
\quad \textsf{end};
\end{array}
$$

Description:

Auxiliary function of restrictToString (p. 34).

Calls:

Standard VDM-SL only

E.2.6 Function *countryToString*

Specification:

$$
\begin{array}{l}
countryToString : Country \to string \\
countryToString\,(\textsf{mk-}Country\,(c)) \triangleq \\
\quad \texttt{"}country\texttt{"} \curvearrowright c
\end{array}
$$

Description:

Auxiliary function of restrictToString (p. 34).

Calls:

Standard VDM-SL only

E.3 Example Values

In this section we can see example values if the type Restrict (p. 34). (Consult page 12 for a description of the semantic of this values.)

$$
\begin{array}{l}
\textsf{values} \\
\quad restrict\text{-}or : Restrict = \textsf{mk-}BooleanOR\,(\textsf{mk-}Topic\,(\text{LINUX}), \\
\quad\quad\quad\quad\quad\quad\quad\quad \textsf{mk-}Topic\,(\text{BSD}));
\end{array}
$$

The following example appers in page 12.

$$
\begin{array}{l}
restrict\text{-}and : Restrict = \textsf{mk-}BooleanAND\,(\textsf{mk-}Topic\,(\text{LINUX}), \\
\quad\quad\quad\quad\quad\quad\quad\quad \textsf{mk-}Country\,(\texttt{"}FR\texttt{"}));
\end{array}
$$

The following example appers in page 12.

$$restrict\text{-}uk\text{-}or\text{-}us : Restrict = \mathsf{mk}\text{-}BooleanOR\,(\mathsf{mk}\text{-}Country\,(\texttt{"}UK\texttt{"}),\ \mathsf{mk}\text{-}Country\,(\texttt{"}US\texttt{"}));$$

$$restrict\text{-}complex : Restrict = \mathsf{mk}\text{-}BooleanAND\,(\mathsf{mk}\text{-}Topic\,(\textsc{linux}),\ \mathsf{mk}\text{-}BooleanNOT\,(restrict\text{-}uk\text{-}or\text{-}us))$$

end *GoogleSearchQuery*

Contents