Channel matching logic V2

Table of Contents

0. Changelog / Revisions

Date	Version	Who	Comment
2015-02-08	1.0	Michael Twarkowsky	Initial document for algorithm V2. First algorithm was without Country checking and doing poorly. Not documented.

1. Preface:

This channel matching logic assumes that the deploying SDP has access to the channel batches (REGION-*, ALL) of eyeQ or a similar database containing.
During quality control of the results some improvements have resulted which will be incorporated in the next iteration. Those improvements are:

- It may happen that a TVChannel contains the same triplet more than once. This leads to a false match of TripletNameRDMatch. Can be mitigated on various ways.
- It is arguable to skip channels marked as no listings available. Some of them have the correct Region/Country and contain a station logo. Maybe improved channel skipping algorithm should be applied according to the needs (EPG, Image, both).
- Current implementation of RegionDiff-check prefers placeholder over ISO3166-Alpha2. First hit gets responded as “RD matches”.
  Possible solution would be splitting the RD-check into two separate checks, RDd (RegionDiffdirect) and RDbg (RegionDiffbestGuess). First one tries to interpret the given RegionDiff solely as ISO3166-A2. Second tries to interpret the given RegionDiff as placeholder like “EU”. Modify analyzer results accordingly.
- Assuming there are two channels with same name “ABC”. One of them is assigned with RegionDiff=EU, one of them with RegionDiff=DEU. First one matches on the triplet, but the second one not. If RX-Country is set to DEU the matcher must return the second channel.
  This seems only feasible for complete name comparison and not to be done in any heuristics level.
- Add some weighting in the heuristic approach. Seen channel “DRF 1” matched on callsign “NDR FS”, HL6, with probability of 0.6. Possible solution: Play with average name length of all TVChannel’s names and some mathematical functions with rotational symmetry. Preferable one with asymptotic behavior.

2. Goal

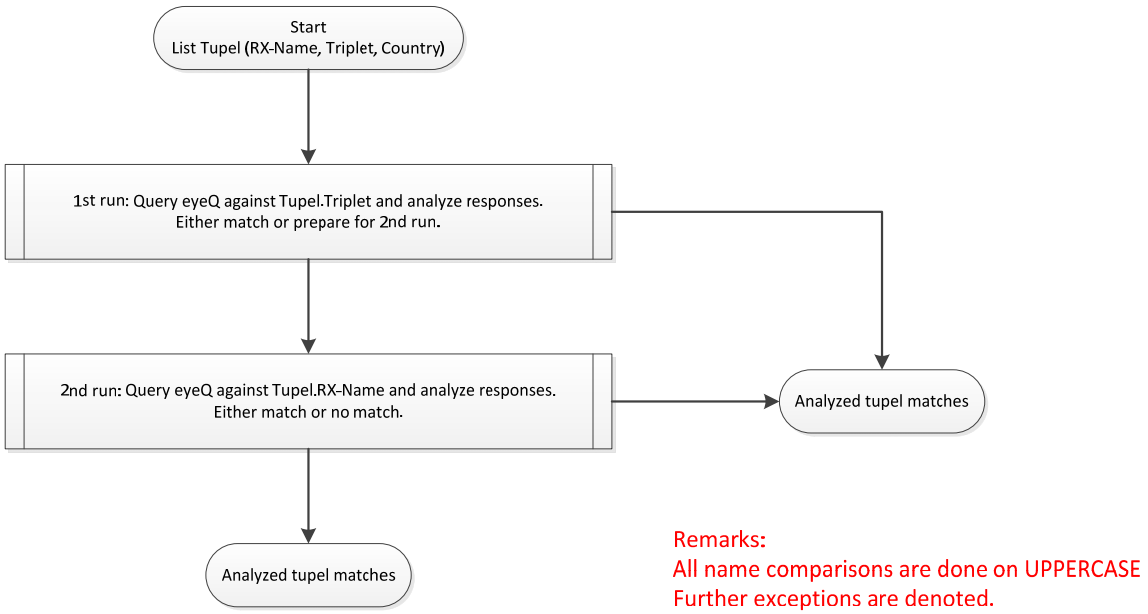
The algorithm described in this document is used to match a specific TV-channel received via DVB/ISDB with its equivalent in the eyeQ-database from Gracenote.

It is assumed that the implementation of the algorithm works on the “tvgridbatch” batch files obtainable via the eyeQ web-API.

3. How it works

3.1. Overview

A TV-set or STB sees the station name, the triplet of the digital video feed and the TV-set knows in which country it is operating.
For each given tuple {RX-Name, Triplet, Country} shall be found a match in the eyeQ database.
The algorithm uses a sequential two-way approach to find the matching item.
The first step is an eyeQ-lookup with the received Triplet and an analysis of the found TVChannels.
Any channel that doesn't get matched in the first step undergoes a second step. This second step is an eyeQ-lookup with the received Name and an analysis of the found TVChannels. As this is less reliable it should always be done if matching by triplet and name fails.



Picture 1 - Channel matcher, high level overview

Detailed flowcharts are shown in chapter 5.

3.2. Name Matching

The name matching itself consists of four steps. Each step is less reliable than the previous one but resembles the seen variations of TV-station names in the field. Example "SR-Fernsehen" vs. "SR Fernsehen".

1st try to match the name is a lookup in the TVChannel fields: name, nameshort, callsign. The received name and the names in the TVChannel are compared always on uppercase. If a match is detected, it is being indicated and the algorithm ends.
All other tries to match the name are heuristic methods. They deploy various levels of heuristic analysis. Basically the higher the heuristics level is, the longer it takes.
It shall be noted that the heuristic levels 4-6 are the same as 1-3. The only difference is the handling of found TVChannels but non-matching country.

Heuristics level	Algorithm	Definition of ratio
HL1	Remove in received name unwanted characters, like blanks, dashes, slashes, etc. For each to be compared to TVChannel	Length of received name without unwanted characters vs. Length of unaltered

	remove in any field name, nameshort, callsign the same unwanted characters as above. If the modified received name is found in the modified TVChannel names, mark it as valid. Return all valid/found TVChannels.	received name.
HL2	Remove in received name unwanted characters as in HL1. For each to be compared TVChannel remove in any field name, nameshort, callsign the same unwanted characters. Then try to match parts of the modified received name with any modified name of the TVChannel. Always try to find the smaller string in the longer, that is: if the modified station name is longer or equal than the TVChannel-name, try to find parts of the TVChannel-name in the modified received name. If the modified station name is shorter than the TVChannel-name, try to find parts of the modified received name in the TVChannel-name. The parts of the modified received name always start from the beginning and with the longest possible substring (including the whole string). The minimum length of the pattern is two characters. Example: Received name “ABC 123-TV” Modified name “ABC123TV” First part to be searched “ABC123H” Second part to be searched “ABC123” … “AB” If the ratio is higher than the previously achieved highest ratio, every previously item is invalid and the newest one is solely valid. If the match ratio is the same as the so-far highest, it is a valid channel, too. This algorithm returns the TVChannels with the highest ratio.	Length of partial string of vs. length of found-in string. Example: Received name=”ABC 123 ENG” Database contains “ABC 123” Matching on “ABC123” and a ratio of 6/9 = 0.667 (ABC123 / ABC123ENG)
HL3	Remove in received name unwanted characters as in HL1. For each to be compared TVChannel remove in any field name, nameshort, callsign the same unwanted characters. Then try to match any parts of the modified received name with any modified name of the TVChannel. Always try to find the smaller string in the	Length of pattern vs. length of found-in string.

	longer, that is: if the modified station name is longer or equal than the TVChannel-name, try to find parts of the TVChannel-name in the modified received name. If the modified station name is shorter than the TVChannel-name, try to find parts of the modified received name in the TVChannel-name. The pattern is a windowed function with a given width that runs over the string. The width starts with the length-1 and is minimum 2 characters long. Example: from ABC123 will be derived the patterns “ABC12, BC123, ABC1, BC12, C123, ABC, BC1, C12, …”. If the ratio is higher than the previously achieved highest ratio, every previously item is invalid and the newest one is solely valid. If the match ratio is the same as the so-far highest, it is a valid channel, too. This algorithm returns the TVChannels with the highest ratio.	

3.3. Country Matching

Country matching is needed to differentiate between same named stations that send different broadcasts in different regions/countries. Good examples are “TV3” or “Euronews”.
TVChannels can have a Region Differentiator (RD) and should have at least one Country where it was reported being broadcast. The RD indicates the targeted country.

It is best to check first for the RD and if this fails (not matching or RD doesn’t exist) the Country.

Important: The RD may resemble an ISO3166-Alpha2 country or an GN-own overlay. This overlay can be a whole set of countries (EUrope, MiddleEast) or regions within a country (MElbourne, PEarth). Currently this algorithm first checks for the overlaid regions and if this fails, checks for ISO3166-Alpha2.
The algorithm works internally with ISO3166-Alpha3 as the Country-information is in ISO3166-Alpha3.

If the name matches but not neither RD nor country match the tuple-country, then it is handled very differently depending on the currently running step.
If looked up by triplet no RD/Country matches, then continue with Heuristics Level 1.
If looked up by name and no RD/Country matches, then indicate as result “NamesNoCountry”.
If looked up by name in any Heuristics Level (HL4-6), then check the ratio. If ratio is greater than a certain threshold, the result is regarded as “NamesNoCountryWeighted”. If the ratio is equal or lower than the threshold, the next Heuristics Level is started. As there is no higher Heuristics Level than HL6, a name-only match without country match is regarded as “NoMatch”

3.4. Possible matcher results

The matcher can have the following results.
NamesAll means all names from an eyeQ-TVChannel: Names, Nameshort, Callsign(s)
NameStripped means stripped from unwanted characters. This is HL1.
NamePartialSimple is HL2. See above for explanation.
RD Match stand for “Region Differentiator match”.

The algorithm follows this sequence. If a result is NameRDMatchSingle it is ensured that it does not match with the given triplet.

Triplet?	Name Match Type	RD Match?	Country Match	Match count	Match type
Yes	NamesAll	Yes	-	1	TripletNameRDMatchSingle
Yes	NamesAll	Yes	-	>1	TripletNameRDMatchMulti
Yes	NamesAll	No	yes	1	TripletNameCMatchSingle
Yes	NamesAll	No	yes	>1	TripletNameCMatchMulti
Yes	NameStripped	Yes	-	>0	TripletNameRDMatchHL1
Yes	NameStripped	No	yes	>0	TripletNameCMatchHL1
Yes	NamePartialSimple	Yes	-	>0	TripletNameRDMatchHL2
Yes	NamePartialSimple	No	yes	>0	TripletNameCMatchHL2
Yes	NamePartialComplex	Yes	-	>0	TripletNameRDMatchHL3
Yes	NamePartialComplex	No	yes	>0	TripletNameCMatchHL3
No	NamesAll	Yes	-	1	NameRDMatchSingle
No	NamesAll	Yes	-	>1	NameRDMatchMulti
No	NamesAll	No	yes	1	NamesCMatchSingle
No	NamesAll	No	yes	>1	NamesCMatchMulti
No	NamesAll	No	no	>0	NamesNoCountryAny
No	NameStripped	Yes	-	>0	NamesRDMatchHL4
No	NameStripped	No	yes	>0	NamesCMatchHL4
No	NameStripped	No	no	>0	NamesNoCountryHLWeighted
No	NamePartialSimple	Yes	-	>0	NamesRDMatchHL5
No	NamePartialSimple	No	yes	>0	NamesCMatchHL5
No	NamePartialSimple	No	no	>0	NamesNoCountryHLWeighted
No	NamePartialComplex	Yes	-	>0	NamesRDMatchHL6
No	NamePartialComplex	No	yes	>0	NamesCMatchHL6
No	NamePartialComplex	No	no	>0	NamesNoCountryHLWeighted
No	NamePartialComplex	No	no	>0	NoMatch
No	no name match	-	-	-	NoMatch

3.5. Suggested decider logic for TV-sets and STBs

This chapter is a suggestion on how to handle the channel matching results from the point of view of a TV-set, STB or any other device facing a user.

Match type	Suggested next step
TripletNameRDMatchSingle	Accept automatic match
TripletNameRDMatchMulti	If GN_IDs across responses differ, manual match
TripletNameCMatchSingle	Accept automatic match
TripletNameCMatchMulti	If GN_IDs across responses differ, manual match
TripletNameRDMatchHL1	Accept automatic match. Eventually use threshold for probability.
TripletNameCMatchHL1	Accept automatic match. Eventually use threshold for probability.
TripletNameRDMatchHL2	Accept automatic match. Eventually use threshold for probability.
TripletNameCMatchHL2	Accept automatic match. Eventually use threshold for probability.
TripletNameRDMatchHL3	Accept automatic match. Eventually use threshold for probability.
TripletNameCMatchHL3	Accept automatic match. Eventually use threshold for probability.
NameRDMatchSingle	Accept automatic match. Eventually use threshold for probability.
NameRDMatchMulti	If GN_IDs across responses differ, manual match
NamesCMatchSingle	Accept automatic match.
NamesCMatchMulti	If GN_IDs across responses differ, manual match
NamesNoCountryAny	Manual match (>1 elements) or manual confirm (=1 elements)
NamesRDMatchHL4	Accept automatic match. Eventually use threshold for probability.
NamesCMatchHL4	Accept automatic match. Eventually use threshold for probability.
NamesRDMatchHL5	Accept automatic match. Eventually use threshold for probability.
NamesCMatchHL5	Accept automatic match. Eventually use threshold for probability.
NamesRDMatchHL6	Accept automatic match. Eventually use threshold for probability.
NamesCMatchHL6	Accept automatic match. Eventually use threshold for probability.
NamesNoCountryHLWeighted	Manual confirm.
NoMatch	Display placeholder picture or indicate „no data“

4. Notes / Remarks

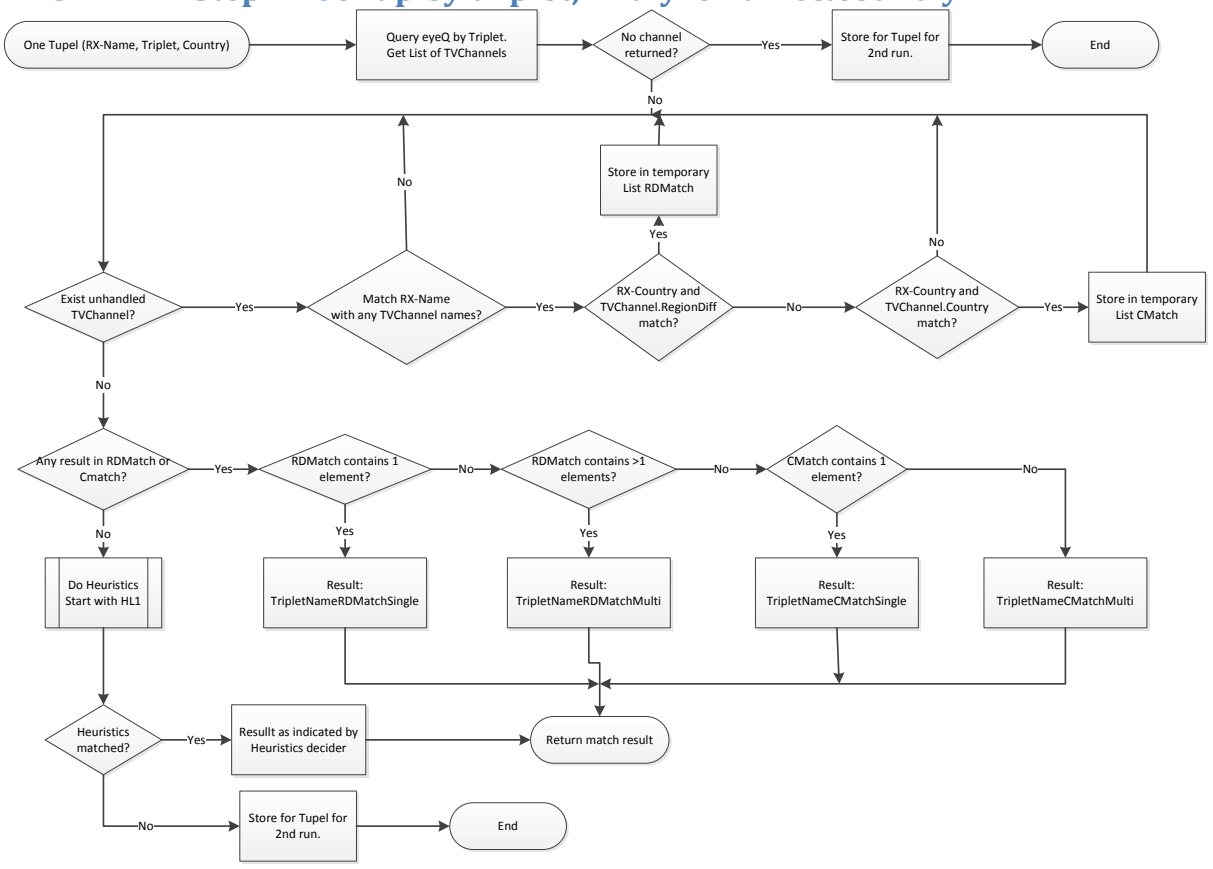
The large amount of different matching results is chosen to enable any afterwards set-up deciding logic to work best. This can be used to decide whether to ask the user to match the results manually or automatically accept the result.

The Heuristics always start only after every other fast-working and exact matching failed. Basically this is: for each found TVChannel check names. For each found TVChannel check names with heuristics HL1. For each found …

5. Algorithm flowcharts

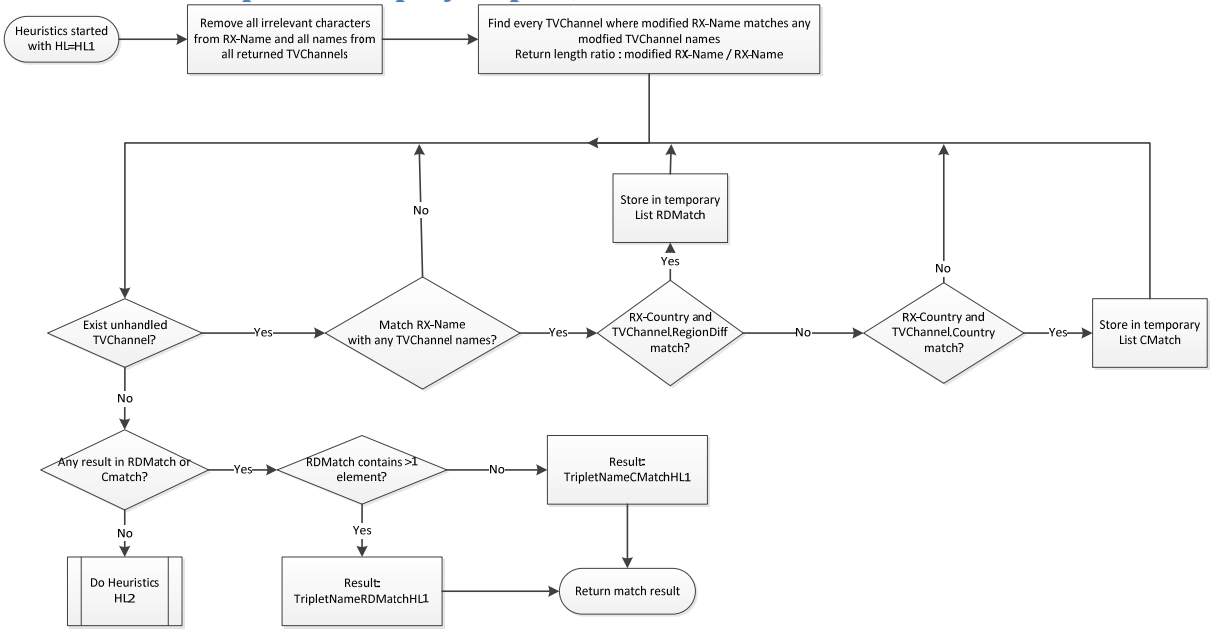
Not shown are decisions if a given Heuristics Level shall be done or not. It is up to the implementing engineer to do the trade-off to meet the boundary restrictions (performance limitations due to hardware, database, software, network connectivity).

5.1. 1st Step – Lookup by triplet, Analyze name&country



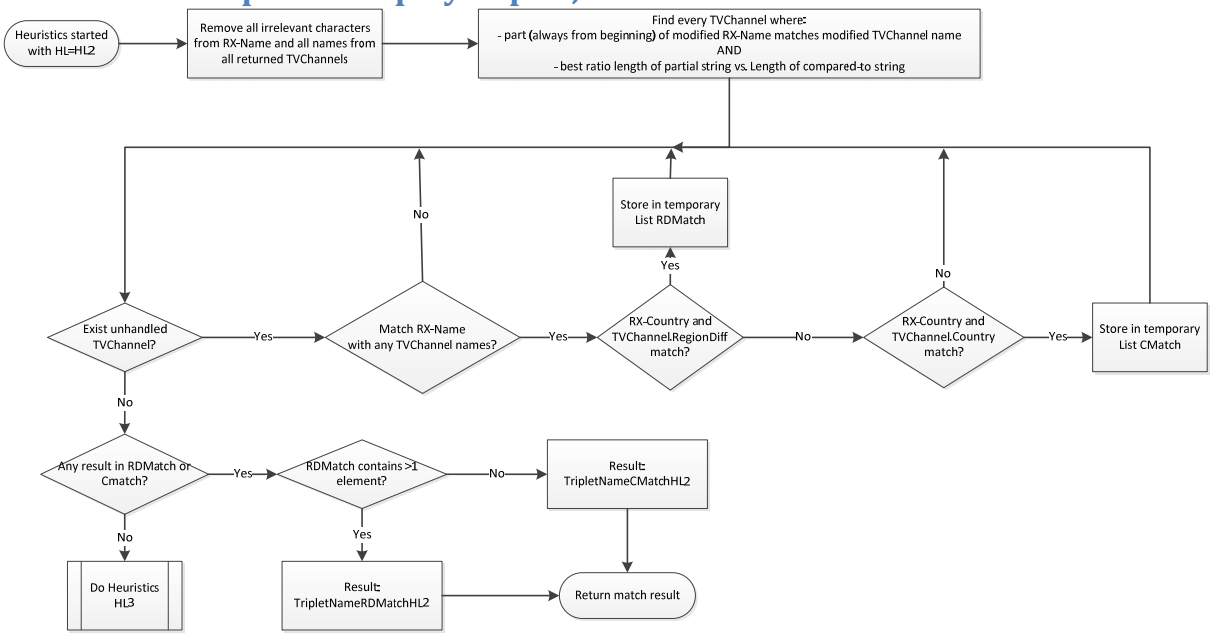
Note: On entering the two lists *RDMatch* and *CMatch* are regarded as empty.

5.2. 1st Step – Lookup by triplet, Heuristics Level 1



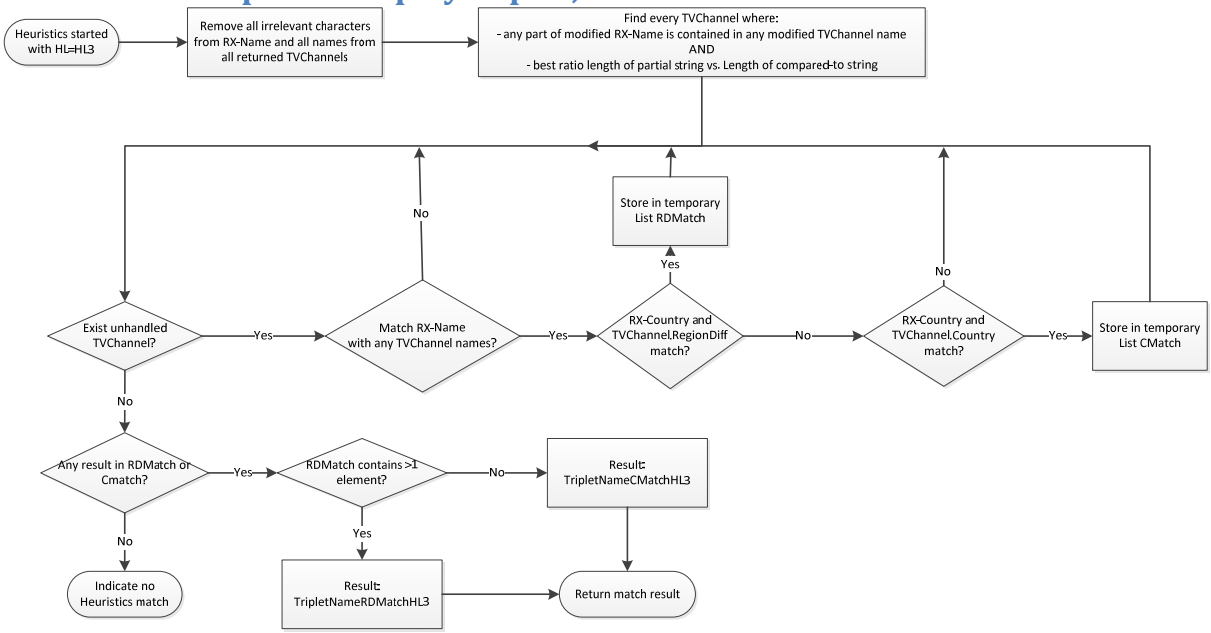
Note: On entering the two lists *RDMatch* and *CMatch* are regarded as empty.

5.3. 1st Step – Lookup by triplet, Heuristics Level 2



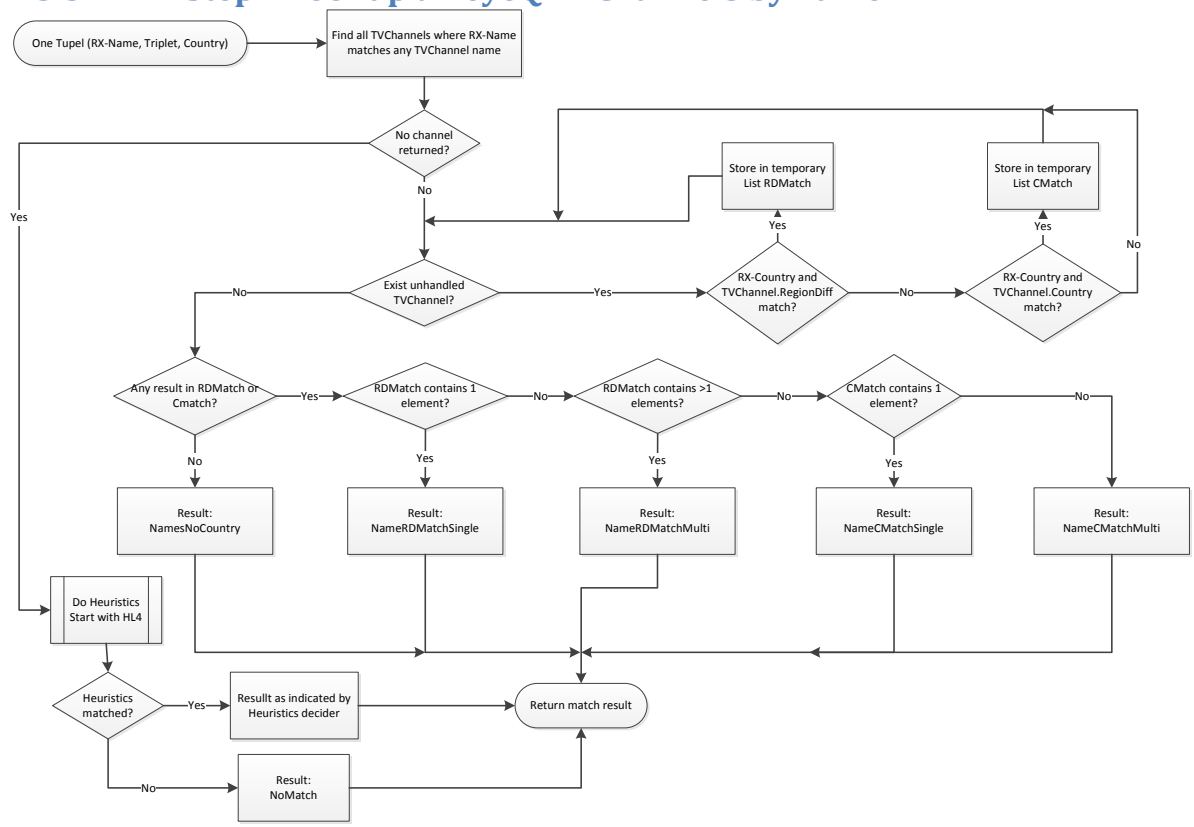
Note: On entering the two lists *RDMatch* and *CMatch* are regarded as empty.

5.4. 1st Step – Lookup by triplet, Heuristics Level 3



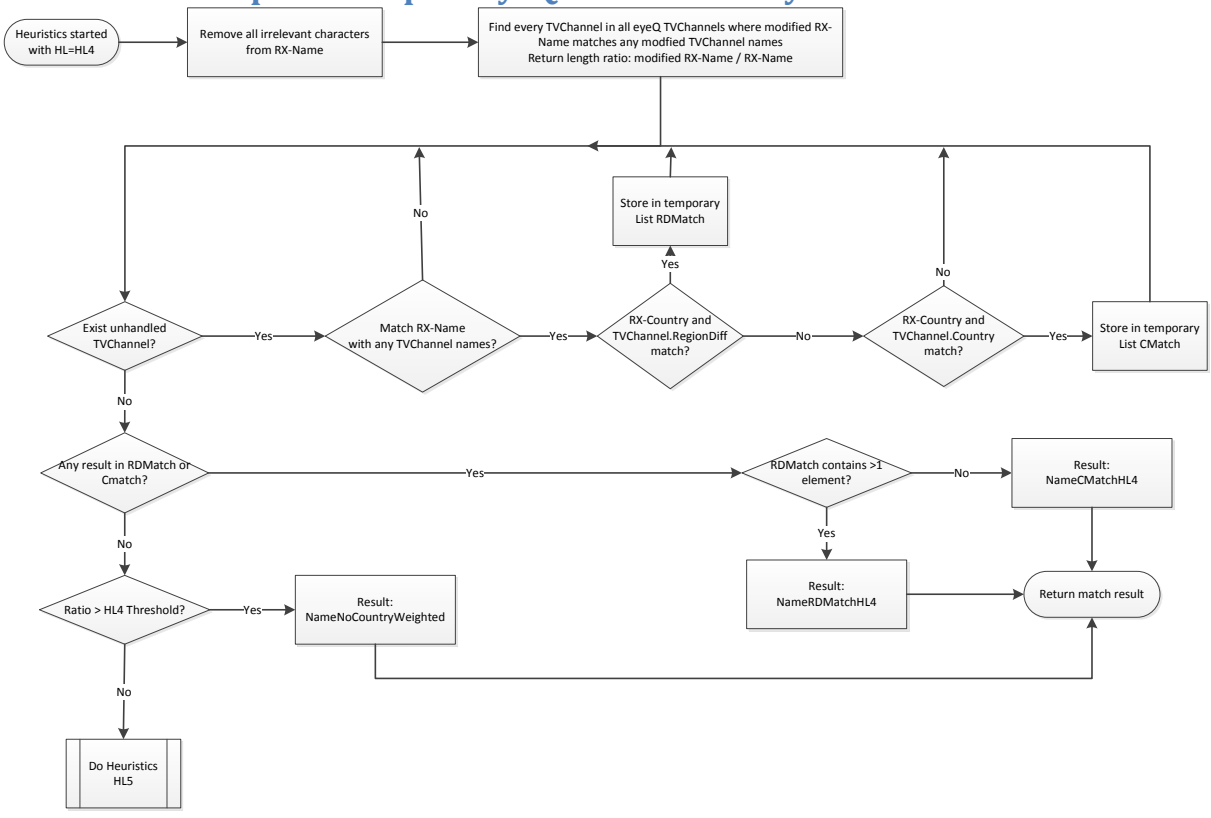
Note: On entering the two lists *RDMatch* and *CMatch* are regarded as empty.

5.5. 2nd Step – Lookup all eyeQ TVChannels by name



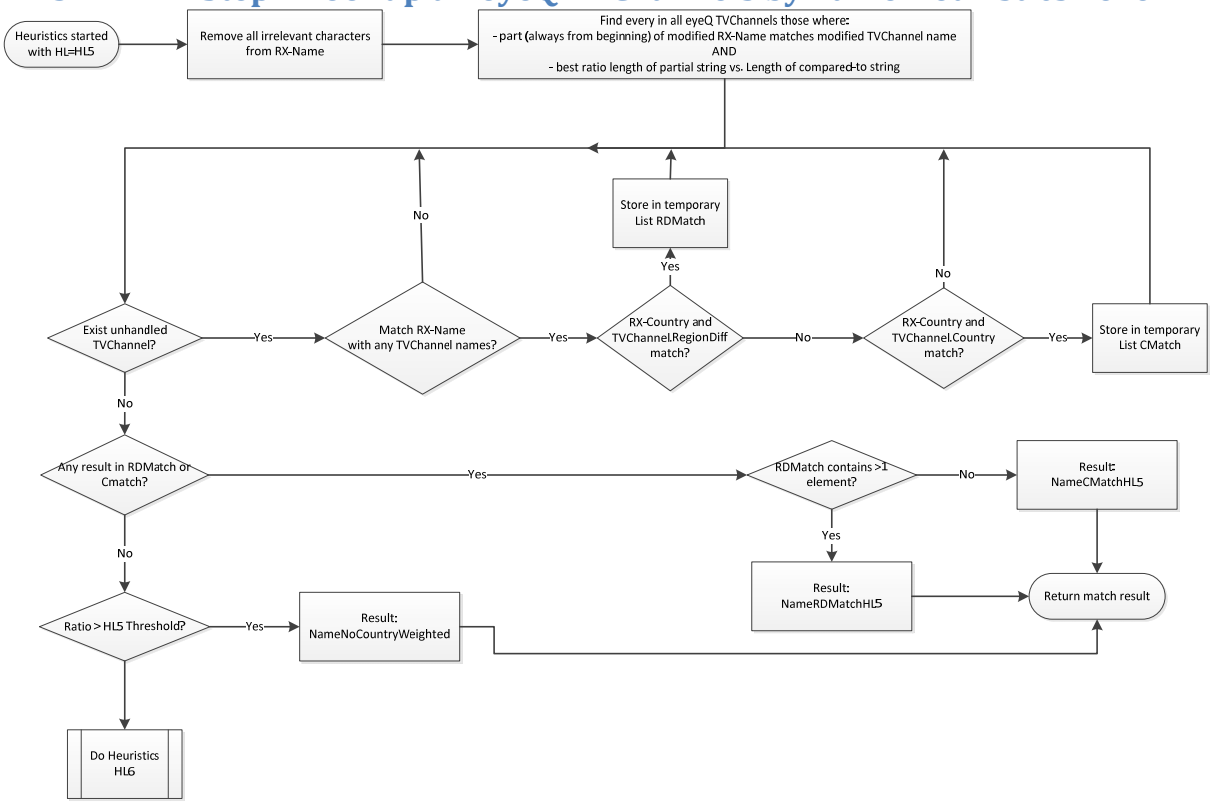
Note: On entering the two lists *RDMatch* and *CMatch* are regarded as empty.

5.6. 2nd Step – Lookup all eyeQ TVChannels by name Heuristics Level 1



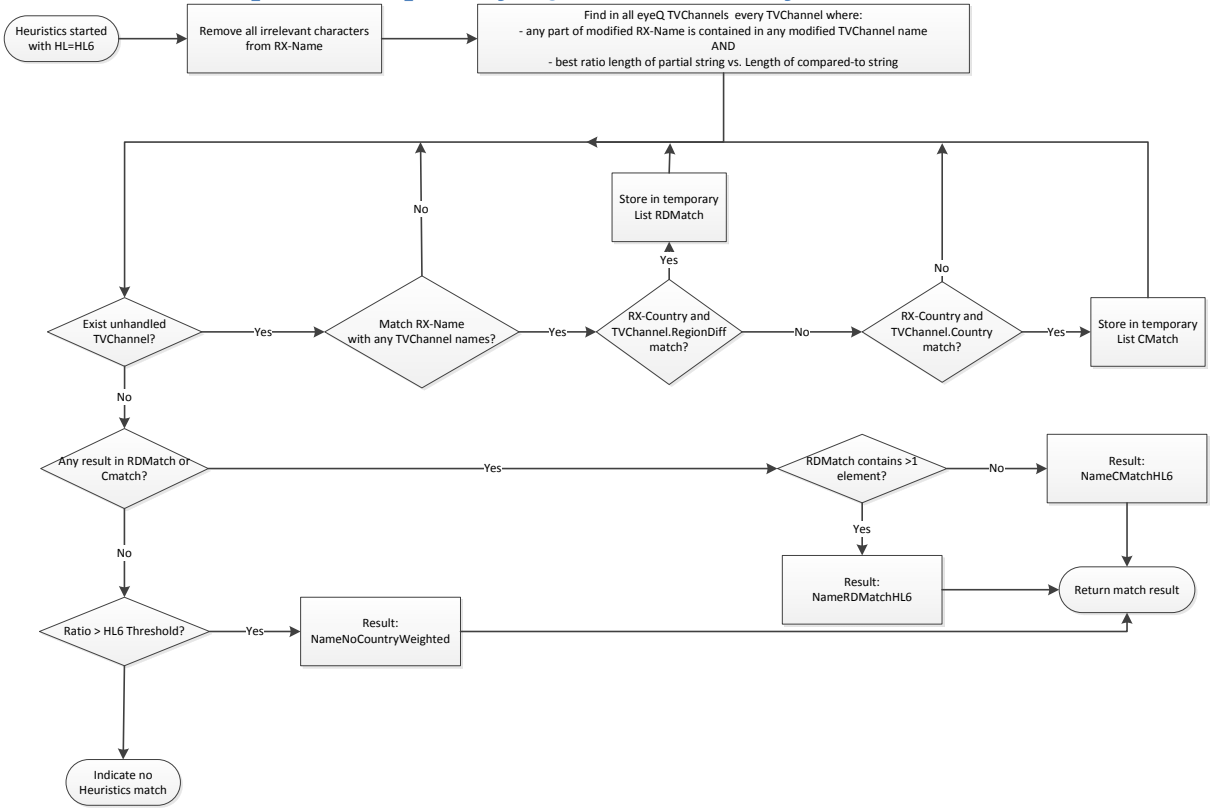
Note: On entering the two lists *RDMatch* and *CMatch* are regarded as empty.

5.7. 2nd Step – Lookup all eyeQ TVChannels by name Heuristics Level 2



Note: On entering the two lists *RDMatch* and *CMatch* are regarded as empty.

5.8. 2nd Step – Lookup all eyeQ TVChannels by name Heuristics Level 3



Note: On entering the two lists *RDMatch* and *CMatch* are regarded as empty.

6. Benchmarks

An example implementation was engineered. This program handles a list of received tuples and evaluates them. All heuristic levels have been activated. Runtime includes all database accesses for channel matching.
Note: The example implementation uses dumb multithreading. The tuples are initially split in equal chunks and then started on all existing cores. It can happen that one analyzer thread finishes far earlier than other ones. This will be fixed in a future implementation.

6.1. Intel Core i7 (model 860)

Used with eyeQ Batch Stamp: 1000144546
Number of tuples: 2.132.386
Used cores: 8
Runtime matching: 1280 seconds

6.2. Intel Core i5 (model 4200H @ 3.30GHz)

Used with eyeQ Batch Stamp: 1000144546
Number of tuples: 2.132.386
Used cores: 4
Runtime matching: 1610 seconds

6.3. Intel Core i7 (model 4600U @ 2.70GHz)

Used with eyeQ Batch Stamp: 1000144546
Number of tuples: 2.132.386
Used cores: 4
Runtime matching: 2117 seconds