map 'quest: Directions with a Lisp

Tony Lazenka

Advised by Drew McDermott

Department of Computer Science, Yale University

1 Introduction

This project is an attempt to rewrite driving directions (provided by MapQuest) differently depending on the context – local street or highway route – in which a given direction appears. Here is an example drawn from this paper’s test mule, a trip from New Haven to San Diego:

33: Stay STRAIGHT to go onto F ST. 0.5 mi
34: Turn RIGHT onto 9TH AVE. 0.1 mi
5: Take the I-287 W exit, EXIT 21, toward WHITE PLAINS/TAPPAN ZEE BR. 0.1 mi
6: Merge onto I-287 W/CROSS WESTCHESTER EXPY. 0.1 mi

Now, the first series (steps 33 to 34), in its local context of a city block, is perfectly concise: it is, so to speak, what the man on the street would tell you. The second, in its larger context of a rapidly diverging interstate, seems to beg for greater succinctness. To be sure, there are drivers that may prefer an account of each discrete segment of road or exit on their ride, but the extra step in this direction seems hardly useful when it is less than a mile long, and mirrors the preceding in every way except its leading verb and elaboration on exit name. Here is the sort of rewrite this project aims for:

5-6: Take the I-287 W/CROSS WESTCHESTER EXPY exit, EXIT 21, toward WHITE PLAINS/TAPPAN ZEE BR. 0.2 mi

An expanded goal of the project is to apply the principle of brevity to certain general types of directions, regardless of context. An example from the same trip of the “Becomes” type:

18: Merge onto I-44 W via EXIT 34 toward TULSA (Portions toll). 107 mi
19: I-44 W becomes TURNER TURNPIKE (Portions toll). 0.1 mi
20: TURNER TURNPIKE becomes JOHN KILPATRICK TURNPIKE (Portions toll). 25 mi

Ignoring the redundant distinction between “Turner Turnpike” and “John Kilpatrick Turnpike” here (which is really a mistake of the data, not its presentation), there is no reason the above steps can’t be more concisely written as:

18-20: Merge onto I-44 W / TURNER TURNPIKE / JOHN KILPATRICK TURNPIKE via EXIT 34 toward TULSA (Portions toll). 132.1 mi

To implement these and other rewrites, a program was developed with Allegro Lisp to first extract directions from MapQuest's output (since I have no access to their Geographic Information System, that is, their store of data that tells how roads are connected). These were then organized into blocks of local streets and highway routes. Next, rules were designed and applied to simplify cases like the above. The programming language chosen was Common Lisp, for its case at parsing (helpful in step 1) and code as data/data as code philosophy (helpful in step 3). The practical result is a Windows/WINE application that accepts straight-text MapQuest directions (UTF-8 or below) and transforms them into an html page in the new format.

2 Parsing

1Although the sense of “Becomes” (as part of a direction) is fairly close to that of “Stay Straight” (seen in steps 33-34 above), I would not apply this rewrite to the latter type of direction, as it would eliminate the useful orientation information that can indicate a road splitting off into two (“Stay Left” and “Stay Right” can also appear in MapQuest).
On review of a half-dozen or so MapQuest samples, the directions I got were for the most part a set of imperative sentences of a single verb modified by prepositions taking nouns as objects.\(^2\) With a vocabulary of a few other words to complete the sense of the sentences, these would mark how to approach a given segment of road. Here is the list of provisionally identified imperative verbs: “Start”, “Keep”, “Stay”, “Turn”, “Take”, “Merge”, and “Go”.\(^3\) The accompanying prepositions varied depending on the verb, but I found their object nouns to pretty uniformly fall under the following categories (associated prepositions are listed in parentheses): street to take ("onto", "on", "becomes"), means of approach ("via"), and general geographic destination ("toward").\(^4\)

For the purposes of this paper I will refer to the different verbs as direction “types”, the preposition objects as “variables”, and their combination as direction “templates”. All other words in the templates were treated as filler, but could always be inferred by the unique combination of template types and variables.

Using a syntax borrowed from Peter Norvig (whose pattern matching tools I used in parsing), what follows is a subset of the templates. The variable names, marked by a “?” prefix and “=” suffix, are meant to be bound to all words in a direction delimited by the non-variable words immediately preceding and following them in the template, which were meant to be matched exactly. Despite this strictness, a relatively small number of templates were needed to parse the samples, due to the general sparseness of MapQuest vocabulary outside the template variables.\(^5\)

\begin{verbatim}
(Start
    start out going (?* ?dir=) on (?* ?street=) toward (?* ?destination=))

(Becomes
    (?* ?streetfrom=) becomes (?* ?street=))

(Turn
    turn (?* ?dir=) onto (?* ?street=))

(Turn
    turn (?* ?dir=))

(Take
    take (?* ?street=) via (?* ?exit=) toward (?* ?destination=))

(Take
    take (?* ?street=) via (?* ?exit=))

(Take
    take (?* ?exit=) on the (?* ?dir=) toward (?* ?destination=))

(Take
    take (?* ?exit=) toward (?* ?destination=))

(Take
    take (?* ?exit=) on the (?* ?dir=))

(Take
    take (?* ?exit=))

(Merge
    merge onto (?* ?street=) via (?* ?exit=) toward (?* ?destination=))

(Merge
    merge onto (?* ?street=) via (?* ?exit=))

(Merge
    merge onto (?* ?street=) toward (?* ?destination=))
\end{verbatim}

\(^2\)A necessary exception is the direction “Becomes”, which is not imperative, and takes no objects. It is however treated in the same way as the others as in Note 5.

\(^3\)I’m sure these aren’t exhaustive. I was told, for instance, there was a MapQuest U-turn direction, but could not find one in the samples.

\(^4\)This last category was somewhat problematic, since by using the preposition “toward” to mark a sense of general destination, I am ignoring its other common use by MapQuest to specify an upcoming street or route, as in the difference between “Take Exit 30 toward San Diego” and “Take Exit 30 toward CA-94”. Since the usage here would seem to mimic the actual language of a highway sign, I made the assumption that when a direction was “towards” a route, it meant the route was part of the immediately succeeding direction. This was important for establishing block structure later on.

\(^5\)Some words that were not prepositional objects, but either adverbs of place (such as the orientations “left” or “slight left”) or direct objects (such as the exits listed in the “Take” templates), still fit into these categories, so were assigned to the same variable names found in other templates whose sense they matched (as in the use of ?exit= in both the “Take” and “Merge” templates).

\(^6\)In the actual implementation, each of these also contains a “step” and “miles” variable to be matched against the step and mile count normally listed by each MapQuest direction.
Also defined was a set of special templates that I call “transitional”. These add extra orientation information when combined with any template in the previous set (though in practice these were only found alongside “Take” templates):

(Keep
  keep (?* ?dir=) to (?* ?firstorder=))
(Stay
  stay (?* ?dir=) to (?* ?firstorder=))
(Turn
  turn (?* ?dir=) to (?* ?firstorder=))

When parsing the following example, the ?firstorder= variable would be bound to the second “Take” template, whose variables would be bound in turn:

12: Turn SLIGHT LEFT to take the GOV'T CENTER ramp toward FANEUIL HALL/ FINANCIAL DIST/ AQUARIUM. 0.0 mi

Finally, here is an example of successful bindings for steps 5 and 6 in the trip to San Diego above:

(TAKEEXITTO (?STEP=. 5) (?EXIT= THE I - 287 W EXIT EXIT 21) (?DESTINATION= WHITE PLAINS/TAPPAN ZEE BR) (?MILES= . 0.1))
(MERGEONTO (?STEP=. 6) (?STREET= I - 287 W/CROSS WESTCHESTER EXPY) (?MILES= . 0.1))

3 Blocks of directions

To organize directions by their local and highway contexts per the introduction, a regular expression library by Dorai Sitaram was used to match bound variables against MapQuest’s nomenclature for interstates, US highways, and state routes. The nomenclature was determined by comparing the small icons on the left of a direction step in the MapQuest output that look like highway signs to the text in the directions these seemed to represent. Here is the regular expression:

(?:I|(?:[A-Z][A-Z])) \[ \] [NESW] - [0-9]+[A-Z]{0,1}

A route thus identified, along with the variable type of the binding in which it was found, was appended as an extra variable to each previously bound template, and the reference to the route removed from the original variable binding.

Next, directions were assigned to blocks depending on whether their extracted routes formed an intersection with the routes of a previous step. The direction of a route (N, S, E or W) was ignored when forming the intersection. Also ignored was the type of preposition in which the route appeared, as it was assumed that a route mentioned in any preposition of a direction, by either an act of merging or exiting onto it, or going toward it, was part of the immediately succeeding direction (see Note 4).

As a (hypothetical) example, the following set of directions is blocked off by shared interstates 287, 80 and 76 (I give the results here in a form similar to the final print out):

Original:

7: Merge onto I-287 W. 29.5 miles

[^I did not take into account that route blocks were also often identifiable by the presence of a “Merge” direction.]
[^This nomenclature was not verified against whatever is defined by federal or state Departments of Transportation.]
[^State abbreviations were not matched explicitly to speed up processing time. Also, the extra test for a letter [A-Z] was for routes of the form “AZ-89A”.]
As a block:

I-287 W / I-287 S / I-80 W / I-76 W / I-76 (487.4 mi)
  I-287 W
    (29.5 mi) continue via EXIT 15 toward NEW JERSEY
  I-287 S / I-80 W
    (26.1 mi) continue via EXIT 41B toward DEL WATER GAP
  I-80 W / I-76 W
    (372.1 mi) continue toward AKRON
  I-76
    (59.5 mi) continue via EXIT 1 toward COLUMBUS
  I-71 S
    (90.4 mi) continue via EXIT 119B toward DAYTON...

Note that the I-71 S direction is in its own block, since it contains no reference to a route in the previous block.

Any direction with info I could not match to my regular expression I assumed local.

4 Transformation Rules

To rewrite directions as per the introduction, a set of rules was designed listing conditions to be met by the bound templates (or, more particularly, their bound variables), and what to do with the bound templates if the conditions were met. Manipulations could include, for instance, combining mile counts and exits (as in the first example in the introduction) or street names (as in the second example).

The rules were applied to two bound templates at a time that directly followed each other in the list, starting at the bottom. In this way, the second-to-last direction under consideration would have access to the transformed version of the last direction in the list, and so on. Also, only one rule in the set was applied at a time, so that if a series of two directions was matched to a rule, no others were considered.

To prepare bound templates for initial consideration, a “1” or “2” suffix was appended to their variable names depending on the relative position of the template in the list of directions. Here is the relative ordering of the bound templates for steps 19 and 20 of the second example of the introduction:

(BECOMES1 (?STEP1= . 19) (?STREETFROM1=) (?STREET1= TURNER TURNPIKE) (?MILES1= . 0.1) (?TYPE1= . BECOMES) (?ROUTES1= (INTERSTATE ((?INTERSTATE= . 44) (?DIR= . W)) (?STREETFROM=))))

(BECOMES2 (?STEP2= . 20) (?STREETFROM2= TURNER TURNPIKE) (?STREET2= JOHN KILPATRICK TURNPIKE) (?MILES2= . 24.5))

The rules themselves were represented as a four element list, with the name of the rule coming first. The second element listed verification functions and the ordered names of the variables of the two bound templates on which to call them. The last two elements, one for each bound template, included non-ordered variable names whose values were to be updated with the results of transformation functions, which were represented with the same conventions as the verification functions.\(^\text{10}\)

Here is a version of a rule for the second rewrite in the introduction (functions for combining the state and toll information of the two directions have been omitted):

(Becomes
  (((#'equal ?type2= ((becomes))))
   ((?street= (#'combine-arg-values ?street1= / ?street2=))))

\(^{10}\)Verification functions, apart from returning t or nil to indicate success or failure, could also return additional transformation functions to be appended to the list of transformation functions proper.
The second element lists the only condition for this rewrite: that the second bound template be of type “becomes”. The next element lists how we should update the first direction if verified: in this case, with function combine-arg-values to update its street/route variables with a reference to the street/routes in the second direction, and, with function add-mile-args, to update its mile count with that of the second direction. The next element in the list is ‘nil’, which indicates that we should delete the second direction from the resulting list.

For now, only a limited number of transformation rules were created and applied, and mainly for the purpose of clearing up loose ends in the block structure. For instance, one transformation rule was explicitly written to join up bound templates of the type “Take (exit)” that referenced no route name to an immediately succeeding bound template, if it did contain a route. Another was written to clean up transitional templates by appending the orientation information of the bound transitional template to the direction that followed.

5 Printing

The main approach to printing was to orient directions around headings identifying the blocks extracted in section 3 as belonging to either route or local contexts. In the new print out, headings for the former showed a cumulative list of routes belonging to the block, and, for the latter, simply showed the letter ‘L’ in the text output or a picture of a streetlight in the html. Particular steps of directions in each block were themselves oriented towards the name of the routes or street that were their immediate destinations11, with the means of approach forming a connecting statement.

Here is an example of two local MapQuest directions thus oriented:

1: Start out going SOUTHWEST on COLLEGE ST toward CHAPEL ST. 0.3 miles
2: Turn LEFT onto S FRONTAGE RD. <0.1 miles

becomes (some notations omitted)

L (0.4 mi)
go SOUTHWEST toward CHAPEL ST
COLLEGE ST
(0.3 mi) turn LEFT
S FRONTAGE RD
(0.1 mi)…

The connectors here are the imperatives “go toward” and “turn” and their associated words, where the immediate destinations are “College St” and “S Frontage Rd”, which were bound to the ?street= variables of the original templates. The vocabulary of these connectors was in general drawn from the imperative verbs of the templates, with some more generalized substitutions for those I thought were close in sense. For example, both the “take” and “merge” imperatives were replaced with the word “continue”.

For an interstate:

13: Merge onto I-70 W via EXIT 8 toward INDIANAPOLIS. 168.5 miles
14: Merge onto I-70 W via EXIT 110B toward ST. LOUIS. 221.1 miles
15: Merge onto I-55 S via the exit on the LEFT toward ST LOUIS. 20.2 miles

becomes

continue via EXIT 8 toward INDIANPOLIS
I-70 W (389.6 mi)
(168.5 mi) continue via EXIT 110B toward ST LOUIS
(221.1 mi) continue via THE EXIT ON THE LEFT toward ST LOUIS
I-55 S…

11This distinction was meant to mirror the “zoom” feature of MapQuest, which allows you to isolate regions of a map and view them in more detail.
Note that, to avoid redundancy, although “I-70 W” appears two steps in a row in the original, it was not repeated in the reprint.

To further bring the block structure into relief, blocks were indented in the new print out depending on arbitrary levels of importance: blocks containing an interstate were indented once, US highways twice, state routes three times, and local blocks four times. Also, in the html icons matching the normal representation of route types on roadway signs (like a red and blue shield for an interstate) were included in the headers, to draw the eye in the same way. The actual number and direction of the route was printed in adjacent text, although of course it would be more ideal to draw it directly on the icon.

Further embellishments included appending the total mile count and toll information of belonging directions to the headings of a block, to better prepare drivers for a necessary turn-off for gas or an ATM. Similarly, crossings into new states were given prominence at the very left of the print out, alongside an icon in the html for a license plate (representing the invariable sighting of new license plates when going into a new state)\footnote{One can imagine printing other relevant state information in this spot, say voting preference in the last election.}, to more explicitly highlight a possible change in the exit numbering scheme of a highway when one crosses state boundaries (which is at least true for a trip from Connecticut to New York). Connectors of less than a mile length were also prefaced by an exclamation mark.\footnote{On the other hand, a simpler vocabulary, along with things like marking toll roads and state crossings pictorially, would seem beneficial when translating the directions into other languages. As of now MapQuest only provides a few non-English translations, all either Germanic or Romance.} A facility for adding notes in between steps in the input was also included, with the idea that one could add reminders to tank up, or to tag certain points of interest along a trip to share with others.

6 Conclusion

An unfortunate consequence of this project’s approach is that it presents a loss, or at least misplacement, of some of the original MapQuest information. Route names have been removed from their normal place in the directions that presumably points to what actually appears on a highway exit sign (also see Note 4), and similarly because of the transformation rules, the mile count and name of a terminating street of a direction like “Becomes” has been lost in aggregation with that of the step immediately prior. Additionally, by making the vocabulary of connecting statements even more limited than that of the MapQuest templates, I am ignoring those who might prefer a more verbose language to capture the distinction between, say, taking an exit and simply merging onto a route.\footnote{The strictness of the original template structure has its failings, too, if we were to try to adapt it to interpret other web directions, especially Google’s, which has a wider range of classes that are used in more varied combinations with each other. [I handled MapQuest’s extra “continue to follow” clause, included at the end of some directions, in a very ad-hoc way in the code.] While testing the program I did have some success reconfiguring the templates for use with Yahoo! directions, though these may or may not work in the final product.}

At this other extreme, for future work, the new block structure would seem to facilitate the construction of an even more verbose representation of the directions, especially if new semantics were inferred from different combinations of MapQuest templates.\footnote{It may have been possible to simulate more global transformations by writing a set of rules to update directions with references to much later directions in a continuous chain upwards, but this seemed like too poor of a hack to be worth it.} For instance, any local segment part of a “Merge” template of length less than one mile leading to a route could be renamed an “on-ramp” (or “off-ramp” in the opposite direction, or simply “ramp” if between routes). However, the set of transformation rules as it stands is ill-equipped for this, as it is only meant to process two steps of directions at a time, and only one rule in the set can be applied to a single series of two directions.\footnote{The block structure may also suggest an alternative method to actually store and retrieve MapQuest directions, if oft-used blocks, say the I-95 corridor from New England to Florida, rather than their constituent steps, were cached for ready access, with necessary turn-offs inserted for the final destination.}

Where the project succeeds, I think, is in providing an alternative representation of directions for those of us who are either more or less attentive to just the local aspects of a drive along the mighty US roads system.
Appendix

Here is the MapQuest print out of a trip from New Haven to San Diego:

A. New Haven, CT

1. Start out going SOUTHWEST on COLLEGE ST toward CHAPEL ST. 0.3 mi
2. Turn LEFT onto S FRONTAGE RD. 0.0 mi
3. Merge onto CT-34 E via the ramp on the LEFT. 0.7 mi
4. Merge onto I-95 S toward N.Y. CITY (Crossing into NEW YORK). 48.2 mi
5. Take the I-287 W exit, EXIT 21, toward WHITE PLAINS/TAPPAN ZEE BR. 0.1 mi
6. Merge onto I-287 W/CROSS WESTCHESTER EXPY. 0.1 mi
7. Merge onto I-287 W (Portions toll). 29.5 mi
8. Merge onto I-287 S via EXIT 15 toward NEW JERSEY (Crossing into NEW JERSEY). 26.2 mi
9. Take I-76 W toward AKRON. 59.5 mi
10. Merge onto I-71 W via EXIT 3 toward COLUMBUS. 90.4 mi
11. Merge onto I-70 W via EXIT 19 toward DAYTON. 17.0 mi
12. Merge onto I-70 W via EXIT 8 toward INDIANAPOLIS (Crossing into INDIANA). 168.6 mi
13. Merge onto I-70 W via EXIT 110 toward ST LOUIS (Crossing into ILLINOIS). 221.1 mi
14. Merge onto I-55 S via the exit on the LEFT toward ST LOUIS (Crossing into MISSOURI). 20.3 mi
15. Merge onto I-55 S via EXIT 40C toward I-44 W. 1.1 mi
16. Merge onto I-44 W via EXIT 207/290C (Portions toll) (Passing through OKLAHOMA). 378.1 mi
17. Merge onto I-44 W via EXIT 34 toward TULSA (Portions toll). 106.7 mi
18. I-44 W becomes TURNER TURNPIKE (Portions toll). 0.1 mi
19. TURNER TURNPIKE becomes JOHN KILPATRICK TURNPIKE (Portions toll). 24.6 mi
20. Take the exit toward I-40 WEST/YUKON/AMARILLO. 0.7 mi
21. Merge onto I-40 W via the exit on the LEFT toward AMARILLO (Passing through TEXAS and NEW MEXICO, then crossing into ARIZONA). 852.2 mi
22. Take EXIT 195 toward AZ-89A N/FLAGSTAFF/GRAND CANYON. 0.2 mi
23. Merge onto I-17 S via the exit on the LEFT toward SEDONA/PHOENIX. 124.0 mi
24. Take the AZ-101-LOOP W exit, EXIT 214C. 1.3 mi
25. Merge onto AZ-101 LOOP W. 20.9 mi
26. Merge onto I-10 W via EXIT 1A toward LOS ANGELES. 20.6 mi
27. Merge onto AZ-85 S via EXIT 112 toward I-8/YUMA/SAN DIEGO. 37.0 mi
28. Stay STRAIGHT to go onto W PIMA ST/I-8 BL. 0.4 mi
29. Merge onto I-8 W toward SAN DIEGO (Crossing into CALIFORNIA). 272.8 mi
30. Merge onto CA-125 S toward CA-94. 2.7 mi
31. Keep LEFT to take CA-94 W. 8.8 mi
32. Stay STRAIGHT to go onto F ST. 0.5 mi
33. Turn RIGHT onto 9TH AVE. 0.1 mi
34. End at San Diego, CA

B. San Diego, CA
Here is the output from my program:

- (0.4 mi) go SOUTHWEST toward CHAPEL ST COLLEGE ST
- (0.3 mi) turn LEFT S FRONTAGE RD
- (0.1 mi) continue via THE RAMP ON THE LEFT

- CT-34 E
- (0.7 mi) continue toward NY CITY

- I-95 S
- NY
  - (48.2 mi) continue via THE EXIT EXIT 21 toward WHITE PLAINS / TAPPAN ZEE BR

- I-287 W / I-287 S
- I-287 W
  - (0.2 mi) continue

  - (29.5 mi) continue via EXIT 15 toward NEW JERSEY
  - I-287 S

- NJ
  - (26.1 mi) continue via EXIT 41B toward DEL WATER GAP

- I-80 W

- PA OH
  - (372.1 mi) continue toward AKRON

- I-76 W
  - (59.5 mi) continue via EXIT 1 toward COLUMBUS

- I-71 S
  - (90.4 mi) continue via EXIT 119B toward DAYTON

- I-270 W
  - (17.0 mi) continue via EXIT 8 toward INDIANAPOLIS

- I-70 W
  - (389.6 mi)

- IN
  - (168.5 mi) continue via EXIT 110B toward ST LOUIS

- IL
  - (221.1 mi) continue via THE EXIT ON THE LEFT toward ST LOUIS

- I-55 S / I-44 W
  - (530.7 mi)

  - I-55 S

8
MO
(20.2 mi) continue via EXIT 40C
(1.1 mi) continue via EXIT 207 / 290C
I-44 W

OK
(378.1 mi) continue via EXIT 34 onto TURNER TURNPIKE / JOHN KILPATRICK TURNPIKE toward TULSA
(131.3 mi) continue via THE EXIT toward YUKON / AMARILLO
I-40 W (852.9 mi)
(0.6 mi) continue via THE EXIT ON THE LEFT toward AMARILLO

TX NM AZ
(852.3 mi) continue via EXIT 195 toward FLAGSTAFF / GRAND CANYON
AZ-89A N
(0.1 mi) continue via THE EXIT ON THE LEFT toward SEDONA / PHOENIX

I-17 S
(124.0 mi) continue via THE LOOP W EXIT EXIT 214C
AZ-101 (22.0 mi)
(1.2 mi) continue onto LOOP W
(20.8 mi) continue via EXIT 1A toward LOS ANGELES

I-10 W
(20.5 mi) continue via EXIT 112 toward YUMA / SAN DIEGO
AZ-85 S / I-8 / I-8 W (310.0 mi)
AZ-85 S / I-8 (36.9 mi) go STRAIGHT onto W PIMA ST / BL
I-8
(0.3 mi) continue toward SAN DIEGO
I-8 W

CA
(272.8 mi) continue
CA-125 S / CA-94 / CA-94 W (11.3 mi)
CA-125 S / CA-94 (2.6 mi) continue on the LEFT
CA-94 W
(8.7 mi) go STRAIGHT onto F ST
(0.5 mi)
F ST
(0.4 mi) turn RIGHT
9TH AVE