

Salzgurken: A formal grammar for unambiguous grocery shopping

Eleftheria Chatziargyriou

EUR Store Coalitionist

Torshi Aisle (the one left to the big ice cream fridge)

Abstract—In this paper we are examining how to resolve ambiguities in grocery stories lists. We introduce a formal grammar for specifying an easy-to-understand grammar, easily parsable and tested in all currently available platforms (Gx1000000, GZ, GX and even the now-deprecated GBM). We also propose a range of restrictions with the aim of simplifying the information processed by the modern consumer.

I. INTRODUCTION

Ambiguity in grocery shopper instructions leads to countless incidents affecting consumers each year versed in the ways of programming. While sometimes said incidents undoubtedly offer some unparalleled entertainment value, this shouldn't determine the torment uncountable of our fellow citizens endure.

We aim at a rigorous definition of grocery shopping lists introducing a formal grammar, facilitating a pleasant shopping experience.

II. BACKGROUND

We begin by examining some of the most common incident categories. Some documented incidents are referenced in Table I and Figure 2.

TABLE I: Example of incident in tabular format

Failure Mode	Upvotes	Reference
Syntax ambiguity	2.3k	[5]
Non-terminating process	727	[4]
Syntax ambiguity	136	[6]
Syntax ambiguity	801	[3]
Syntax error	60	[2]

The observant reader will note that there are numerous more examples to be found, particularly in online platforms but the .bib isn't going to write itself. The curious reader with slight masochistic tendencies is invited to observe the phenomenon by themselves by duckduckgoing the following keywords we identified: "husband" "programmer" "grocery shopping".

A woman asks her husband, a **programmer**, to go shopping:

Dear, please, go to the nearby grocery store to buy some bread. Also, if they have eggs, buy 6.

O.K., hun.

Twenty minutes later the husband comes back bringing 6 loaves of bread.

His wife is flabbergasted:

Dear, why on earth did you buy 6 loaves of bread?

They had eggs.

Fig. 1: Example of incident but in pictorial format [1]

The main point to be highlighted is that victims seem to be predominantly male. The reason seems to be that the male brain is more logical and structured, making prone to syntactic ambiguities common in the female intellect (according to scientific study with at least 25.8k upvotes [7]).

We also observe that the failure modes are effectively eliminated with the introduction of a formal grammar. Namely the problem of syntactic ambiguity is resolved by using a well-designed grammar. Likewise, we tackle potentially non-terminating processes by prohibiting recursive and looping constructs.

III. REQUIREMENTS

Before proceeding to actually defining the grammar, we are going to elicit some requirements to maximise the stakeholder's satisfaction and efficiency.

The most important one is for the grammar to be clear, concise and easily understandable by both menfolk and womenfolk.

We also want to ensure that a polynomial-bound shopper can shop in a reasonable time. We are only considering classical shoppers as pesky quantum shoppers usually abide to much different lifestyles for them to be accurately examined here.

We can imagine semantically supporting clauses such as:

- "Buy either eggs or cherries"
- "Buy one butter for every egg you've bought"
- "If you have bought cherries, buy an avocado but only if there are vegan chicken nuggets"
- "Don't spend more than 10 money if BTC is trending downwards"

How will the optimal shopper proceed to maximize their shopping experience? Should they prioritize getting the most value for his money? Should he pick some items over others? He simply has no unambiguous way of deciding. Before they know it, they are going to start filling in abnormally large knapsacks with useless items bringing us one step closer to our eventual demise.

Therefore, we need to somehow restrict the semantic scope of our representation language in the following way:

- Any conditional statement on the procurement status of any single item shall not depend on the outcome of any other statement
- Arbitrary properties of the item can be referred to, assuming it is understandable by all parties. Said properties can only refer to the item that conditionally depends on them
- Make it easy for regular housewives to understand

The grammar has to obviously reflect the aforementioned well-defined, crystal-clear requirements.

IV. INNOVATION

Camera pans to a thirty year old man doing his weekly grocery shopping in a Lidl somewhere in Bavaria. He has a frustrated look on his face holding a sky blue package in one hand and a nearly-identical deep sky blue package in the other. The man is scratching his head and gives an intense look to the products A male narrator with stern yet oddly warm voice starts speaking:

[MN]: Has this ever happened to you?

Scene transition: The man is back to his rustic home and gets the groceries out of his eco-friendly, reusable bag. Tens of nearly-identical packages are laying on the table. The wife can be seen just entering the kitchen with a look of terror once she spots that the sky blue package is laying on their table

[MN:] Introducing: Salzgurken. The **modern** way of shopping

The husband is again doing his shopping on the Bavarian Lidl looking at two other identical packages: one lemon chiffon and one Chatreuse yellow

V. TADA

The grammar file can be found here¹ and it is extremely self-explanatory.

1) *Simple instructions if you can't self-explain:* Every entry in the grocery list has to start with *. This needs to be followed by a numerical value and an optional unit.

The numerical value can also specify a range:

- $N..$: at least N

- $..N$: at most N
- $N_1..N_2$: at least N_1 and at most N_2

The greedy shopper will always choose the greatest amount they can get that meet the given conditions.

Additionally, a condition can be specified. The shopper will proceed with the procurement only if said condition is met as is illustrated in the following double-explanatory example:

- * 6.. "EGGS" IF (WEIGHT < 199 kg)
- * 100..500 L "MILK" IF ((ATTRIBUTE \$quality IS "extra fresh") AND (PRICE = 100))

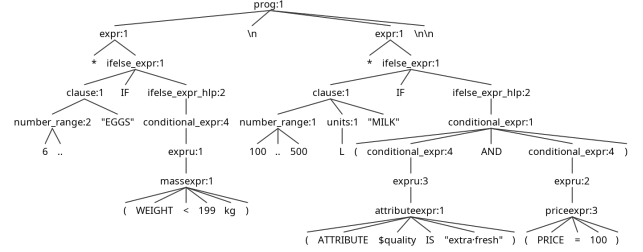


Fig. 2: Nice tree example

VI. FUTURE WORK

We will work on our time-scheduling skills so next SIG-BOVIK, we'll remember the deadline in time.

VII. ACKNOWLEDGMENTS

This work wouldn't have been possible without Arch Linux. I would like to also thank my mother for teaching me the importance of generating exclusively human-made nonsense. Machines will not replace us.

REFERENCES

- [1] *A programmer goes shopping*. Oct. 21, 2010. URL: https://www.reddit.com/r/pics/comments/duf6p/a_programmer_goes_shopping/.
- [2] *A Python programmer and his wife are discussing grocery shopping*. Aug. 4, 2018. URL: https://www.reddit.com/r/ProgrammerHumor/comments/94ivqe/a_python_programmer_and_his_wife_are_discussing/.
- [3] *u/BlNG. A programmer goes shopping*. Jan. 30, 2017. URL: https://www.reddit.com/r/Jokes/comments/5qyxar/a_programmer_goes_shopping/.
- [4] *u/Kyraryc. A wife send her programmer husband grocery shopping*. Apr. 10, 2018. URL: https://www.reddit.com/r/Jokes/comments/8baxnd/wife_sends_her_programmer_husband_grocery_shopping/.
- [5] *u/pitzu. A wife send her programmer husband grocery shopping*. Apr. 10, 2018. URL: https://www.reddit.com/r/Jokes/comments/8baxnd/wife_sends_her_programmer_husband_grocery_shopping/.
- [6] *u/whitehat2k9. A man goes to the grocery store...* May 28, 2012. URL: https://www.reddit.com/r/ProgrammerHumor/comments/u85tr/a_programmer_goes_to_the_grocery_store_xpost_from/.

¹<https://gist.github.com/Sedictious/193bcb63d7d5960cb509713d78393e63>

- [7] *Women tend to prefer working with people, while men tend to prefer working with things, according to a new study based on an analysis of responses from people in 42 countries.* URL: https://www.reddit.com/r/science/comments/ziwfqe/women_tend_to_prefer_working_with_people_while/.

VIII. ANNEX

```

grammar salzgurken;
prog:  (expr NEWLINE)* ;

expr:
BULLET ifelse_expr |
BULLET clause
;

ifelse_expr:
clause IF ifelse_expr_hlp (ELSE ifelse_expr_hlp)?;
ifelse_expr_hlp:
clause IF conditional_expr (ELSE ifelse_expr_hlp)? |
conditional_expr;

conditional_expr:
LPAR conditional_expr AND conditional_expr RPAR |
LPAR conditional_expr OR conditional_expr RPAR |
LPAR NOT conditional_expr RPAR |
expru;

clause:
number_range (units)? STRING_LITERAL;

expru: massexpr | priceexpr | attributeexpr;

attributeexpr:
LPAR ATTRIBUTE ATTRIBUTE_VAR IS STRING_LITERAL RPAR |
LPAR ATTRIBUTE ATTRIBUTE_VAR IS ONEOF string_literal_list RPAR;

massexpr:
LPAR WEIGHT LT NUMBER MASSUNITS RPAR |
LPAR WEIGHT GT NUMBER MASSUNITS RPAR |
LPAR WEIGHT EQ NUMBER MASSUNITS RPAR;

priceexpr:
LPAR PRICE LT NUMBER RPAR |
LPAR PRICE GT NUMBER RPAR |
LPAR PRICE EQ NUMBER RPAR;

oneof: 'ONE OF' list;
list: number_list | string_literal_list;
number_list: '['NUMBER (NUMBER)* ']';
string_literal_list: '['STRING_LITERAL (COMMA STRING_LITERAL)* ']';
number_range:
NUMBER'..'NUMBER |
NUMBER'..' |
'..'NUMBER |
NUMBER;
units : MASSUNITS | VOLUMEUNITS;

MASSUNITS :
'dg' | 'cg' | 'mg' | 'µg' | 'ng' | 'pg' | 'fg' | 'ag' | 'zg' | 'yg' | 'rg' | 'qg' | 'dag' | 'hg' | 'kg' | 'Mg' | 'Gg' |
'Rg' | 'Qg';

VOLUMEUNITS:
'qL' | 'xL' | 'yL' | 'zL' | 'aL' | 'fL' | 'pL' | 'nL' | 'L' | 'mL' | 'cL' | 'dL' | 'L' | 'daL' | 'hL' |
'ZL' | 'YL' | 'RL' | 'QL';

IS: 'IS';
IF: 'IF';
ELSE: 'ELSE';
ONEOF: 'ONE OF';
WEIGHT: 'WEIGHT';
ATTRIBUTE: 'ATTRIBUTE';
FLAVOUR: 'FLAVOUR' | 'FLAVOR';
PRICE: 'PRICE';
LT: '<';
GT: '>';
EQ: '=';
AND: 'AND';
OR: 'OR';
NOT: 'NOT';
COMMA: ',';
LPAR: '(';
RPAR: ')';
ATTRIBUTE_VAR: '$'[A-Za-z]+;
STRING_LITERAL: '"'[^\r\n]*'"';
STRING: '[a-zA-Z]+';
NEWLINE: [\r\n]+;
BULLET: '*';
NUMBER: INTEGER | FLOAT;
fragment INTEGER: [0-9]+;
fragment FLOAT: [0-9]+'.'[0-9]+;
fragment SEPARATOR: '-' | '\\';
WS : ('\t' | ' ' | '\r' | '\n' | '\u000C')+ -> skip ;

```