Dear Editors,

We would like to thank the reviewers for the insightful reviews that they provided us with. In the revised version that we are submitting, we have compared the proposed grammar to the one implemented by Apache POI and we have included an extended discussion in a new section (6.1). Also, in this version we included references to recent works that further strengthened the motivation towards a defined grammar for Excel formulas and we improved the introduction, related work and conclusions sections.

Below we respond to the reviewers’ comments and we explain how we revised the paper to address them.

Looking forward to your feedback on the revised version of the paper.

Efthimia Aivaloglou

**Reviewing: 2**

**Comment 2.1** The “Conclusion” section is really tiny, and it does not present and ignore the main results described in section 5. Thus, I suggest to include the main results also in the conclusion (results highlighted in the boxes), exploring more the results that the abstract (the abstract is okay).  

**Answer 2.1** We agree, the Conclusion section needed an update. We added a description of the main results of the datasets analysis.

**Comment 2.2** In order to update the related work section, I strongly suggest to cite the paper  
Wensheng Dou, Shing-Chi Cheung, Chushu Gao, Chang Xu, Liang Xu, and Jun Wei. 2016. Detecting table clones and smells in spreadsheets. In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016). ACM, New York, NY, USA, 787-798.

**Answer 2.2** We updated the introduction section and the related work sections with references to this paper, along with other relevant papers presented within 2016.

**Comment 2.3** The paper claims that the previous research work ([7, 9, 10, 11, 12, 13]) would support the motivation to propose a new grammar for researching. However, it is a biased argument (confirmation bias), because the references 7, 9, 10, 11, and 13 are from the paper’s co-author, decreasing the motivation reliability. In addition, the paper use a single external reference ([12]) as a motivation, and despite the Badame and Dig ([12] paper’s authors) discussed, they have issues and found limitations when adopted Apache POI Java Library in their research, but they \*did\* not discuss exactly what kind of issues or limitations they had had using the tool. Consequently, all referenced work construct a weak foundation for motivating the paper.

**Answer 2.3** We strengthened the motivation for the grammar by including 3 recent works by other researchers that could have utilized a tested and published formula grammar instead of an undefined grammar or string comparison operations.

**Comment 2.4** Analysing the paper, I would highlight one major concern: ignoring/hiding previous grammar proposals and parser implementations.  
The paper abstract starts claiming “there does not exist a \*reliable\* grammar that is \*concise\* enough to facilitate formula parsing and analysis and to support research on spreadsheet codebases.” However, there is at least one grammar that the paper ignore completely: Apache POI ( [https://poi.apache.org/spreadsheet/formula.html](https://webmail.tudelft.nl/owa/redir.aspx?C=Whba2o_0RjgTkQ9jNYwgFS04yGoZhGxe80iQvUfyp3HbxPRVMmXUCA..&URL=https%3a%2f%2fpoi.apache.org%2fspreadsheet%2fformula.html" \t "_blank) ).  
In order to sustain the main claim of the paper, I strongly suggest to include an evaluation of Apache POI grammar and the POI tool, because XLParser is also a paper contribution. The Apache POI propose a grammar implementation on [https://apache.googlesource.com/poi/+/refs/heads/trunk/src/java/org/apache/poi/ss/formula/ptg/Ptg.java](https://webmail.tudelft.nl/owa/redir.aspx?C=cv0UPYTt2Lwh-mkQV-qzPHzgTneqte_SlbF0XufwbLDbxPRVMmXUCA..&URL=https%3a%2f%2fapache.googlesource.com%2fpoi%2f%2b%2frefs%2fheads%2ftrunk%2fsrc%2fjava%2forg%2fapache%2fpoi%2fss%2fformula%2fptg%2fPtg.java) .   
Apache POI is the current state of practice on open source Excel formula parsing (best of my knowledge). Consequently, the paper could be strongly motivated just showing that the proposed grammar (and XLParser) is better than POI, performing the same experimental protocol using POI, or at least making a table putting side-by-side POI and your proposal features. After that, you could also discuss XLParser advantages and weaknesses comparing on POI. Actually, reading the paper, it is just a “believing” that  XLParser could obtain a better success performance that Apache POI. What exactly are POI advantages and weakness comparing with the presented grammar?  
In the paper,  there is a paragraph  
“The above analyses are our main motivation towards defining a formula grammar. Having such a grammar will enable parsing spreadsheet formulas into processable parse trees which can in turn be used to analyze cell references, extract metrics, find code smells and explore the structure of spreadsheets. Essentially, a reliable and consistent grammar and its parser implementation, available to the spreadsheet research community, can support research on spreadsheet formula codebases and can enhance the understanding and usability of research results.”  
Thus, what is a piece of evidence supports the claim that the proposed grammar is more “reliable and consistent” or “concise” than Apache POI? Maybe XLParser is more reliable and consistent than Apache POI, but I could not find any evidence in the paper, because there is any evaluation of POI as a previous work.  
In the “spreadsheet formula grammar requirements”, I found a list:  
1. Compatibility with the official language  
2. Produce parse trees suited for further manipulation and analysis with minimal post-processing required  
3. Recognize the spreadsheet formula elements required for supporting spreadsheets research  
Consequently, before start a new proposal, probably the authors evaluated the POI features and identified that it does not follow the requirements all that requirements.  The concern is addressed just reporting that analysis. However, if POI is better (more “reliable and consistent” or “concise”) and the proposal grammar does not bring a new contribution, it means that the paper has a major issue.  
Thus, does Apache POI follow that requirements, or not? Why not? What kind of limitations are found in POI grammar that the proposal is able to address? What kind of features POI is better, equal or worst than XLParser?  
I suggest to clarify why POI grammar and parser is not able to support research needs as functions calls (of build-in and user-defined functions), function arguments, data (of different types) or references (to internal and external cells and ranges of different types). As illustrated on the Figure 1, maybe show a similar comparison using POI grammar, or performing the same experimental protocol using Apache POI, comparing parsing success and non-functional requirements as the time to processing the corpus or memory footprint, as examples.

**Answer 2.4** Indeed, we had omitted discussing the Apache POI formula parser. The reason was that its grammar is not published or defined anywhere, only the parser implementation is available. Still, we agree that it should be discussed. Based on your suggestions, on this version of the paper:

* We compared the parse trees produced by Apache POI to the ones from the proposed grammar
* We tested the Apache POI formula parser against the latest improvements that we had to include to our grammar in order to cover all cases found in the four datasets. We found 6 cases where Apache POI produces parse errors or incorrect parse trees, and we opened issues for those.
* We examined the drawbacks of using the Apache POI parser for research, and especially that the parser is marked ‘for internal use only’ and the parse trees are not exposed.

To discuss those issues, we added a separate section 6.1, comparing our grammar to the 2 alternatives (the official Excel formula grammar and the Apache POI parser). Moreover, we updated the introduction section to enrich the motivation towards our grammar based on this comparison.

**Reviewing: 3**

**Comment 3.1** Motivation for making the spreadsheet grammar is not clear. They authors noted, they needed a grammar (Section 3) for their previous and ongoing research (in the intro they do have some use cases). If that is required for a research purpose, this paper should have been a side work or part of that research. What is the use case for making such grammar? I find Section 3 is unprofessional, in particular the first sentence. I suggest the authors make it clearer about the use cases of making the grammar. As a software engineering researchers, how much do I care about this work?

**Answer 3.1** In this version we enriched the motivation towards the proposed grammar, citing more recent works that involve processing spreadsheet formulas but either do not detail which grammar they use, or they are based on error-prone string comparison operations to bypass formula parsing. We also improved Section 3, both in terms of the motivation for the grammar and mentioning the existing alternatives.

**Comment 3.2** In building the grammar the authors followed the trial and error approach without showing any further meaningful ways to generalize the grammar. With such an approach, the current grammar might be incomplete for any new dataset. This was also evident from their experiment of new dataset. I am not convinced that such an approach is actually meaningful. The authors should have attempted to find a generalized approach for building the grammar. What did we learn from this paper in grammar construction? what is the novelty actually?  At least how good the proposed parser is? can you guarantee whether this is a complete and correct one? What is the take home?

**Answer 3.2** Our intention with this paper it to contribute to the spreadsheet research community, not the language engineering one. The paper does not introduce novelties in the process of grammar construction. The problem that we are addressing with the parser is that there are numerous works in the spreadsheet research field that analyze spreadsheet formulas using either simple grammars which have not been evaluated and often contain obvious errors, or using implied, undefined grammars. We believe that a reliable grammar and its parser implementation, available to the spreadsheet research community, can support spreadsheet research and can enhance the understanding and usability of research results. Issues related to generalizing the grammar and to the correctness of the parser are further discussed in Section 6.

**Comment 3.3** The paper does not have any threats to validity section. Are they sure that there have been no threats to the validity of the approach? I only see something about compatibility with the official Excel grammar. There seem to be other threats in the data collection, analysis and the nature of data as well. It would be great if the authors point out of such Internal Validity, Construct Validity, External Validity,  Reliability Validity and so on.

**Answer 3.3** We discuss the threats to validity in Section 6 (Discussion and Limitations). The issues include dialects of formulas of various Excel versions, internationalization issues, and issues related to the rejection of invalid formulas and the correctness of the parse trees.

**Comment 3.4** The Related work section is incomplete. It only talks about studies with Spreadsheet/Excel datasets. I thought the paper is largely on the category of building grammars. Unfortunately, I don't see any single citations in building grammars in general. This seems to me a big weakness since as a reader I am not sure how people usually write such papers and valuate the proposed approaches. This is even more important for this paper, since they follow a trail and error approach. Is this a common approach for building grammar? Can they justify?

**Answer 3.4** The first paragraph of the related work section concerns reverse-engineering languages based on existing code, but we chose to keep this discussion limited for the reasons explained in Answer 3.2. In this version we have enriched the related work discussion, but most additions concern works relating to spreadsheet formula analysis.