1 On the size of the formalization

The current formalization is the result of the original effort leading to the proof and a refinement step that reduced the size from 12000 to 7000 lines by combining the 16 possible associativity equations into a single proof.

On a second refinement, we should improve quotient reasoning. Paulson's method in *Defining functions on equivalence classes* is not directly applicable, since the type of the quotient depends on the type variable 'a representing the underlying field, which is fixed by the locale. The *types to sets* approach seems to address this issue. However, we think it would be enough to refactor the proof.

For instance, we have added a lemma of the form:

$$\forall p, p', l, l'.[(p, l)] \oplus [(p', l')] = [(p \oplus' p', l + l')]$$

This is cleaner than the lemmas *gluing_add* and *gluing_ext_add* (which take around 1000 lines). It could also factor out the duplication of code in some of the associativity lemmas (at the time we estimated this duplication in around 2000 lines). So overall, refactoring the proof with this lemma or a similar one would yield at least 2000-3000 lines of improvement.

Note that $\oplus'$ would be a modification of the original operation on points, such that if $\delta\, p_1\, p_2 = 0 = \delta'\, p_1\, p_2$ then it yields $p_1 \oplus p_2$, otherwise ($p_1 \oplus p_2$ undefined) it yields $p_1 \oplus \tau p_2$.

2 On the use of Groëbner basis

The original paper only used Groëbner basis in the proof of dichotomy. We actually produced a proof of the step sketched there without using Groëbner basis: only using basic arithmetic manipulations. This is not in the final version of the formalization, but it can be found in the history of the online repository.

Regarding the use of Groëbner basis in the current formalization. Indeed, they are used often. However, this can be seen as a convenient tool to avoid specifying all the steps and cases. A human reader should have enough with the high-level description of the paper.

Regarding our evaluation of the *algebra* method. If we are to be strict and consider it as a decision procedure, it did not always succeed in giving an answer yes/no. We reported the issue in the tool's mailing list. However, *algebra* did very well on most instances and only on a couple of occasions we needed some intuition on why it was failing.

3 On the state of the formalization

Some of the lemmas are kept to help reducing the size of the proof. Although they may not be used now, we prefer to keep them in scope so that they can be used if needed. They are marked with *TODO*'s to clean them up after the proof reaches its optimal size.

Finally, stating in the source code a clear correspondence with the lemmas in the paper, would be beneficial for the reviewers and future users. We keep this remark in mind, to clarify the script as soon as possible.