Final Report of AndroidTeX

Haotian Tang, Benjie Miao, Hanbo Wen, Feixiang Xu

May 21, 2017

Contents

1 Brief Introduction

1.1 Introduction

AndroidTeX is an android app which allows users to edit TeX files on android devices. We provide a more user-friendly approach than traditional WinEdt for users to view and edit TeX files, which includes hand-writing cognition and easy table generation and sturcture modification with GUI.

1.2 Key Function

A whole LaTeX file is definitely a complicated one. So we decided to choose the most frequently-used and the most important functions to contain in our system. In our opinions, the most amazing function of a LaTeX essay is its Math Formula, Table and Picture insertion, but all of them require complicated codings, which is not user-friendly. So we decided to make it easier. For math formula, we want our user to insert and modify them in the most nature way to input a formula – to write it. Fortunately, we find an API Myscript to implement this; As to Table, we try to enable users to edit it like Excels; For picture, we decide to provide the visualized interface. And the primary target of the app is: in the whole LaTeX project, the users are able to not contact with any coding. The app can deal with the coding process by itself.

2 UI and Struct Tree

In this section, we are going to talk about the basic UI components and the struct tree, which is a very important data structure used in our app.

2.1 Struct Tree

To store the structure of an essay, we use a Tree Structure to represent the structure. The different level of nodes represents the different level of sections. Due to the consideration of saving the memory uses, we choose to use the Left Son Right Sibling representation of a tree.

```
1  class treeNode {
2      int index;
3      int parent;
4      int firstChild = -1;
5      //-1 means no Child
6      int nextSibling = -1;
7      //-1 means it is the last child of his parent(i.e. no next sibling)
8      boolean isDeleted = false;
9      //is Deleted means the node is deleted( but is able to be recovered)
10     int height = 0;
11     //The height of the node
```

Thus, we can use a small amount of information to represent the whole structure tree. However, it is relatively difficult to edit, add or delete nodes or to modify the structure of the tree. Therefore, we design a series of Methods to construct the primitive operations: When it comes to show the whole structure

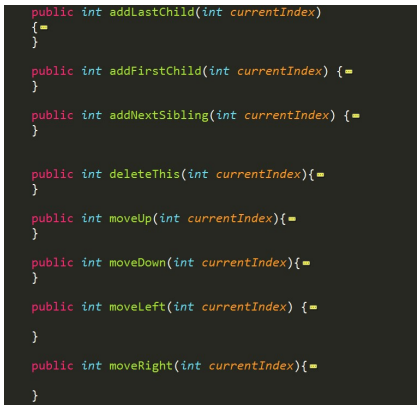

Figure 1: Overall data structure

in a ListView like below: A DFSTraversal is first done. And the sequence of

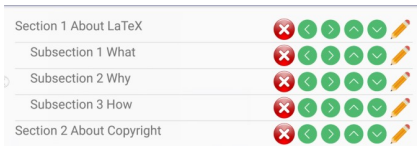

Figure 2: Structure showing

each node indicates the order to be shown. And the height of node decides the indents before the item.

2.2 User Interface

We put the most important things on the user interface. The main interface is the structure of the Essay.

If one want to see and edit one section, he simply clicks the certain item and the content of the section will appears.

If one term is long-clicked, then the system enters the modifying mode. By another long-click, the system return to the main interface.

In the modifying mode, one can simply click the cross or the arrow to delete or move the section. The last icon is used to edit the name of section. The buttons on the bottom is used to create a new section and the undo/redo function.

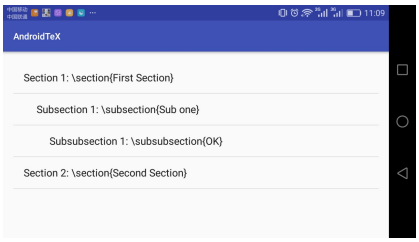

Figure 3: Document structure overview

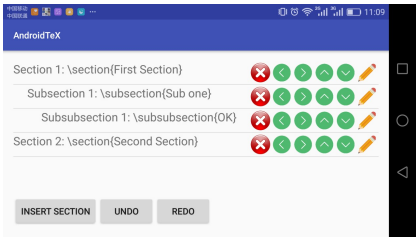

Figure 4: After long click

3 Syntax Analysis and Inner Structure

3.1 Syntax Analysis

Latex documents are well-written in structure.
Section level structure:
Adjacent sections can have only 3 relationships:
1. Father node and son node;
2. Siblings;
3. Different level but not father and son.

When analyzing the structure, thanks to the left son right sibling representation, we can perform following operations when building the struct tree.
1. Add first child to the first node.
2. Add next sibling to the first node.
3. Find LCA(Least Common Ancestor) of the first node and the second node, and add the second node as a next sibling of LCAs first son.(need a graph to clarify)

We use an Article class to store the information of the whole passage. Article class basically contains all the inner structures and some interfaces which can interact with inner structures and the StructTree, which basically only contains the position information of the article components. The Article class is instantiated in a global class(Application in Android programming) ga.art. We know that using a global class is not a good idea in programming, but we compromised finally because the project requires so much information interchange and it seems that Intent alone is not powerful enough under some tricky circum-

stances.

3.2 Inner Structure

We have a class ArticleComponent to store the inner information of each latex document. In our case, due to the limit of time, we can only take into consideration three simple elements of a typical experiment report which is often used in our course: table, mathematical formula and picture. The ArticleComponent class majorly deals with the storage and processing of such 3 basic components. We use position vector to represent the position-relationships between different substructures. We discovered later that it might be not high-efficient to do so. The structure of the class is: And the methods are shown as follows:

```
private intline_number;
private intid;
private intstart_line= -1;
private intend_line=-1;
private String title="";
private ArrayList<String>raw = new ArrayList<>();
private ArrayList<String>content_array= new ArrayList<>();
private ArrayList<Table>table_array= new ArrayList<>();
private ArrayList<Graphic>picture_array= new ArrayList<>();
private ArrayList<Formula>formula_array= new ArrayList<>();
private ArrayList<Integer>table_pos= new ArrayList<>();
private ArrayList<Integer>picture_pos= new ArrayList<>();
private ArrayList<Integer>formula_pos= new ArrayList<>();
private ArrayList<String>independent_raw= new ArrayList<>();
```

Figure 5: Inner Structure

```
    publicintgetId()
    ArticleComponent(ArrayList<String>r,String title, intstart_line
intid)//Constructor, where r is the whole document stored in a string
array, title is the title of the section, start_line is where this
section starts in the whole document, and id is the number of this
component in the struct tree.
    voidclean();//Something like a destructor, cleans every array in
the class.
    voidset(ArrayList<String> r);//read from string r and reset the
content of the current component. Used when refreshing the content
after editing one section.
    voidaddAllComponents(intst);//Used in constructor. Do the
parsing job.
    intadd_table(intln, ArrayList<String> raw);//Used in
addAllComponents(intst), the ln parameter means where the table starts
in the whole raw input, and the arraylist stores the whole passage.
    intadd_graphic(intln, ArrayList<String> raw);//Similar to the
previous function.
    voidpush_graphic(Graphic x);//An interface to add a graph into the
ArrayListpicture_array. It also refreshes picture_pos.
    StringgetRaw();//Get all the raw literals in this section.(not
containing tables, mathematical formulas and pictures.
    And some other interfaces.
```

Figure 6: Inner Structure

3.3 Basic Components

There are three basic components in our app as is mentioned previously. Respectively they are the Table Component, the Picture Component and the Math Component.

3.3.1 Table Component

It is basically a two-dimensional String array with interfaces to parse from and to Latex code.
Equipped with a simple GUI.
Only valid for simple m*n tables. Merging is not supported currently. Far less powerful than TablesGenerator, but works locally.

3.3.2 Picture Component

Allows user to insert a picture from android device and set the name, label and caption of the picture. Then the picture will be automatically implanted into the document and the user needn't write a single line of code to insert a picture.

3.3.3 Math Component

Hand-writing coginition:
Using Myscript api.
What is supported:
Real time hand-writing cognition and converting into latex strings.
Restore the content that is recognized last time. What to be supported:(We have been working on this but due to the time limit we havent succeeded):
Use this Api to modify existing mathematical formulas in the document.

3.4 Rendering

With the help of java api jLatexmath, which is widely used in latex rendering. Basically the API can only render single-line tex code, after doing some modifications to the API, it can render the whole section now.(Due to the limitation of computational ability of an Android phone, we are not trying to render the whole passage at a time on a device.)
Notice that what is rendered by the API is never what is exhibited on your screen but what is stored in our ArticleComponent structure, which is very important in our designing spirit. After each modification, the ArticleComponent in the global class ga.art is refreshed and it ensures that all the changes the user makes will be updated real time.

4 To Improve

- A better GUI
- Modify existing mathematical formulas in the document.
- Not only simple m*n tables, support merging.
- Able to generate the result .pdf file on the mobile.
- Allow Bar-code transmission from computer to android devices and from android devices to computer.