A Python Library for Detection of Inconsistencies between Parcel Database and Plot Register

Mahesh THAPA, Nepal

Key words: Parcel Database, Cadastral Database, Plot Register, Parcel Subdivision Register, Inconsistencies, Errors, Digitization

SUMMARY

Accurate parcel databases are fundamental to effective land administration. This realization has prompted nations to develop parcel databases either through cadastral field surveys or the digitization of paper-based cadastral maps. Digitization of paper-based cadastral maps to develop parcel databases is a widely followed methodology as it is one of the most cost-effective approaches. However, during the digitization process large number of errors can be introduced into the parcel databases. Detection of these errors is necessary to make subsequent corrections. While detection of errors can be accomplished visually by overlaying digitized parcel data overlaid on scanned cadastral maps, this is a very tedious and time-consuming method. This paper discusses an alternate approach to identifying errors by detecting inconsistencies between plot register and parcel database and introduces a python library developed to identify different categories of inconsistencies between the Plot Register and Parcel Databases. These inconsistencies point out errors in cadastral databases. This python library is useful for quality checks during cadastral map digitization process as well as to identifying errors in parcel databases that are already in implementation.

A Python Library for Detection of Inconsistencies between Cadastral Database and Plot Register

Mahesh THAPA, Nepal

1. INTRODUCTION

Parcel databases are the foundation of effective land administration. Consequently, parcel databases are developed using various technologies and methods. One prevalent approach is the digitization of existing paper-based cadastral maps. The development process involves scanning cadastral maps in formats such as paper, cloth, etc., involves scanning of the maps and georeferencing of the scanned images. Features in the cadastral maps, such as parcels, are vectorized, and the parcel identification numbers are digitized and stored as attributes of the vectorized parcels.

While digitizing paper-based cadastral maps is a widely used method for developing parcel databases, the resulting databases can contain many errors. The illegibility of paper-based cadastral maps due to extensive use is the major source of errors in the parcel database. Additionally, the introduction of errors in the cadastral database is influenced by the quality of the digitization process and the effectiveness of quality control measures. Regardless of the causes, it is crucial to rectify the errors in cadastral databases, as these errors can lead to significant consequences.

One of the most challenging parts of rectifying errors in parcel databases is the detection of errors. This difficulty complicates quality control measures during the digitization process, resulting in undetected errors in cadastral databases. Additionally, it hinders initiatives aimed at detecting and rectifying errors in parcel databases.

A straightforward method for detecting errors involves the visual inspection of each parcel by overlaying vectorized parcel data onto scanned image of paper-based cadastral maps (Sapkota & Bhatta, 2071). While this method is simple, it a labor-intensive and time-consuming method. Alternative methodologies that are more efficient are necessary for identifying errors in cadastral database. (Roic, Krizanovic, & Pivac, 2021) and (Roic, Consistency of Data in Cadastral Systems, 2023) have discussed on identification of errors in cadastral databases based on consistency between the register and the cadastral database. An extensive discussion on the principles of identifying logical inconsistencies between the plot register and the cadastral database, including taxonomy for these, has been provided by (Thapa, 2024). This paper builds upon the principles discussed by the paper and extends it to develop a python library for detecting different categories of logical inconsistencies between plot register and parcel database.

2. THEORITICAL FOUNDATION

This review of the theoretical foundation for identifying inconsistencies between plot register and parcel database is primarily based on the concepts discussed in (Thapa, 2024). The functionalities provided by the python library are built upon this theoretical foundation.

2.1. Parcel Subdivision and Consolidation Process

The process of parcel subdivision and consolidation can be represented in a tree-like structure, referred to as “parcel tree”. This parcel tree exhibits characteristics of a tree-like structure; however, unlike a conventional tree structure, parcels can both subdivide and merge to form new parcels. Since the parcel tree structure and a conventional tree structure share similar characteristics, some of the terminology used to describe the parcel tree structure in this paper is derived from terms associated with conventional tree structures. A typical parcel tree structure is illustrated in Figure 1.

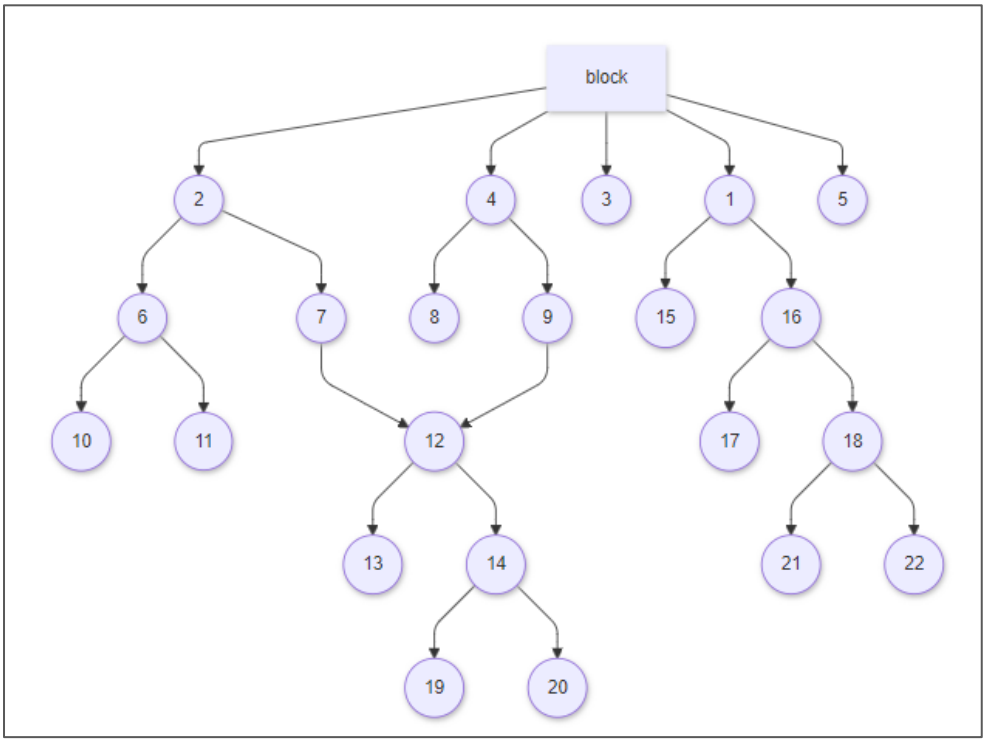

Figure 1 - A Typical Parcel Tree Structure

The root node of the parcel tree is usually a particular area where cadastral survey is conducted. In Figure 1, this is depicted as a rectangle labeled “block”. The child nodes of the root node “block” correspond to the parcels created during cadastral survey. In Figure 1, parcel nodes with Parcel IDs 1 to 5 represent the parcels created during cadastral survey. All other parcel nodes, with Parcel IDs from 6 to 22 are generated through subsequent subdivisions and consolidations.

During parcel subdivision, the Parcel ID of the parent parcel is annulled, and the newly created child parcels are assigned new Parcel IDs. Similarly, in parcel consolidation, the Parcel IDs of the parent parcels are annulled, and the newly created child parcel is assigned a new Parcel ID.

2.2. Record Keeping in Plot Register

In a plot register, parcel subdivision and consolidation records are maintained in a tabular format. Each time a parcel undergoes subdivision, new child parcels are formed. The *Parcel_Id* of the new child parcels are registered in the *child_parcel* column and the corresponding *Parcel_Id* of the

parent parcel is registered in the *parent_parcel* column. Same is done, when parcels are consolidated to form new parcels.

Although various categories of information are documented in a plot register, this research focuses on detecing inconsistencies based on the following columns: *child_parcel* and *parent_parcel.* The records of *child_parcel* and *parent_parcel* related to the parcel divisions and consolidations illustrated in the parcel tree of Figure 1 are presented in Table 1. The Plot Register has at least these columns of child_parcel and parent_parcel.

Table 1 - Tabular Record Keeping in Plot Register

child_parcel	parent_parcel	child_parcel	parent_parcel
1	block	12	7, 9
2	block	13	12
3	block	14	12
4	block	15	1
5	block	16	1
6	2	17	16
7	2	18	16
8	4	19	14
9	4	20	14
10	6	21	18
11	6	22	18

2.3. Leaf Parcels

When a parcel undergoes subdivision or multiple parcels are consolidated to form new parcels, the parent parcels are annulled. This means that all parcels listed in the *parent_parcel* column have already been annulled. Hence, among the parcels in *child_parcel* column, only those that are not present in the *parent_parcel* column remain. These parcels are referred as "leaf parcels". The term leaf is derived from a conventional tree structure which uses leaf node to describe nodes that do not have any children. Leaf parcels are simply the difference between the list of Child Parcels and Parent Parcels as expressed in Equation (1).

Leaf Parcels = Child Parcels – Parent Parcels (1)

For the plot register entries in Table 1, we can find the Leaf parcels using Equation (1) as follows:
Leaf Parcels = Child Parcels - Parent Parcels
= [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22] – [2,4,6,7,9,12,1,16,14,18]
= [3,5,10,11,13,15,17,19,20,21,22]

2.4. Principle of Consistency between Plot Register and Parcel Database and Categories of Inconsistencies

Based on the above discussions in Section 2.1, 2.2 and 2.3, the principle of consistency between plot register and parcel database can be expressed as follows:

"The plot register and parcel database are consistent if and only if parcel database contains exclusively leaf parcels - no more, no less."

Inconsistencies between Plot Register and Parcel Database can be categorized and expressed in terms of Parcel IDs as follows:

a) Parcel IDs not in Leaf Parcels List but existing in Parcel Database
   This category of inconsistency can be further divided into the following types –
   i. *Extra Parcels*
   These Parcel IDs are present in the parcel database but do not appear in the complete parcel tree. These inconsistencies may arise from incorrect Parcel IDs or, in some cases, from the erroneous inclusion of parcels from adjacent administrative units.
   ii. *Dead Parcels*
   Dead Parcels are parent parcels that should have been annulled from the parcel database but still exist because the database has not been updated in accordance with the Plot Register.

b) Parcel IDs in the Parcel database that have a greater count than those listed in the Leaf Parcels List – *Duplicate Parcels*
   These inconsistencies typically arise from incorrect Parcel IDs. However, they can also result from the duplication of digital data and the erroneous inclusion of parcels from adjacent administrative units.

c) Parcel IDs in Leaf Parcels List but not in Parcel database
   This category of inconsistency can be further categorized as:
   i. *Missing Parcels*
   If a parcel in the Leaf Parcels List is missing, and all of its ancestor parcels are also absent, these parcels are classified as Missing Parcels. Missing Parcels can occur due to incorrect Parcel IDs or when certain parcels or sections of administrative areas have not yet been digitized.
   ii. *Missing Child Parcels (of Dead Parcels)*
   If a parcel has undergone subdivision or parcels have undergone consolidation in the plot register but has not been accordingly updated in the Parcel database, the parent parcel(s) (Dead Parcel) will still appear in the cadastral database, while its child parcel(s) will be missing. These missing parcels are classified as *Missing Child Parcels of Dead Parcels*.

3. Python Library

A python library titled *Cadastral Consistency* has been developed which can be utilized to detect inconsistencies between Plot register and Parcel database. *Cadastral Consistency* library can be utilized to automate detection of inconsistencies. This library can be accessed at *https://github.com/maheshthapa/cadastral_consistency.git.*

3.1. Structure of the Library

The library has three modules - *plot_register.py*, *parcel_database.py* and *inconsistency_utils*. The module *plot_register.py* constitutes the *PlotRegister* class. The module *parcel_database.py* constitutes the *PlotRegister* class. The module *inconsistencyutils.py* consists of functions to detect the inconsistencies. The GeoPandas library is used to read Parcel database in shapefile format as geodataframe. The Panda library is used to read Plot Register in excel format as dataframe. The python library can be utilized to detect the following inconsistencies - a*) Extra Parcels b) Duplicate Parcels c) Missing parcels d) Dead Parcels*

4. SAMPLE IMPLEMENTATION

A sample parcel database in shapefile format containing the parcels as shown in *Table 2 – List of Parcels in Parcel Databas*e and Plot Register records as shown in *Table 1 – Tabular record Keeping in Plot Register* in excel format were input in the library to detect the inconsistencies.

Table 2- List of Parcels in Parcel Database

Parcels in Parcel Database
[1, 3, 4, 6, 7, 8, 9, 11, 12, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 22, 22, 23, 24, 25, 26, 27]

The inconsistencies detected are listed in *Table 3 – Detected Inconsistencies.*

Table 3 - Detected Inconsistencies

Sn	Inconsistency Categories	Parcels
1	Extra Parcels	['24', '25', '27', '26', '23']
2	Duplicate Parcels	['22']
3	Missing Parcels	['5']
4	Dead Parcels	['6']

5. CONCLUSION

An error free parcel database is essential in ensuring smooth land administration. A correct parcel database provides confidence in tenure security while errors in parcel database erodes confidence. The developed python library can be utilized to identify errors in parcel database through detection of inconsistencies between Plot Register and Parcel database.

REFERENCES

Roic, M. (2023). Consistency of Data in Cadastral Systems. *Fig Working Week 2023 Proceedings.* Orlando, Florida.

Roic, M., Krizanovic, J., & Pivac, D. (2021, January 14). An Approach to Resolve Inconsistencies of Data in the Cadastre. *Land, 10*(70). doi:ps://doi.org/10.3390/land10010070

Sapkota, R. K., & Bhatta, G. P. (2071). Technical Aspects of Digitization of Cadastral Maps. *Nepalese Jornal on Geoinformatics*, 42-50.

Thapa, M. (2024). Concpets and Principles for Identification of Logical Errors in Cadastral (Parcel) Database Developed by Digitization of Analogue Cadastral (Parcel) Maps. *Journal of Land Management and Geomatics Education*, 30-33.