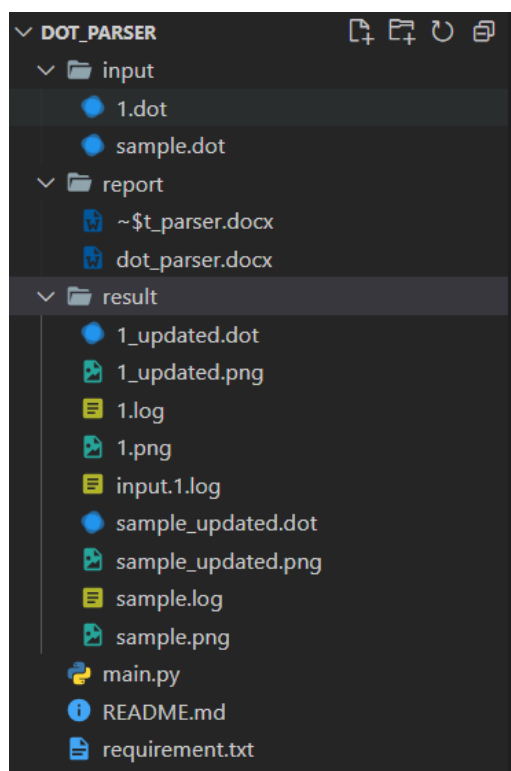


عنوان گزارش: تجزیه و ادغام نودها در فایل های .dat

۱-۱ مقدمه

این پروژه به منظور تجزیه فایل های .dat و شناسایی و ادغام نودهای متوالی طراحی شده است. هدف این گزارش ارائه توضیحات کامل در مورد ساختار کد، نحوه ورودی و خروجی دادن و نمونه های مرتبط می باشد.

۱-۲ ساختار کد



شکل (۱-۱) ساختار کد

با توجه به شکل (۱-۱) فایل main.py فایل اصلی اجرای برنامه است و پوشه های input و output به ترتیب مسیرهای فایل های ورودی برنامه و خروجی برنامه هستند.

۱-۳ ورودی‌های برنامه

این پروژه به دو صورت `command_line` و `user_input` ورودی می‌پذیرد.

```
Home@MShadow MINGW64 /d/work/codes/dot_parser (main)
● $ python main.py input/1.dot
<IPython.core.display.Image object>
<IPython.core.display.Image object>
```

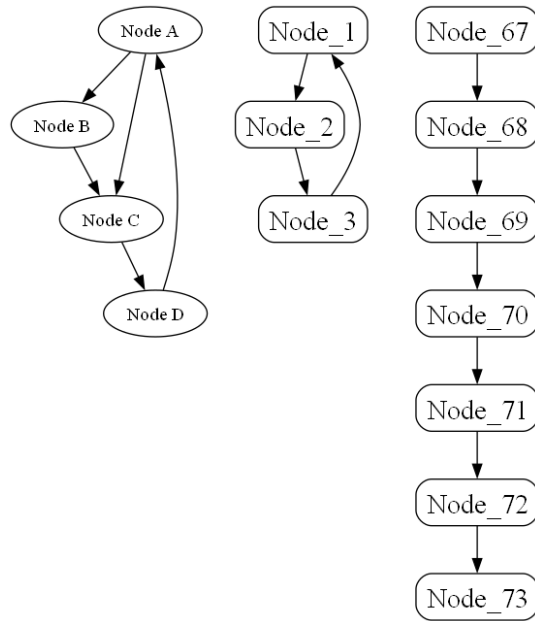
شکل (۱-۲) ورودی `command_line`

```
Home@MShadow MINGW64 /d/work/codes/dot_parser (main)
$ C:/Users/Home/AppData/Local/Programs/Python/Python312/python.exe d:/work/codes/dot_parser/main.py
● Enter the path to the DOT file: input/sample.dot
<IPython.core.display.Image object>
<IPython.core.display.Image object>
Home@MShadow MINGW64 /d/work/codes/dot_parser (main)
```

شکل (۱-۳) ورودی `user_input`

۱-۴ خروجی‌های برنامه

برای هر فایل ورودی با پسوند `.dot` برنامه در مرحله اولین فایل را تجزیه کرده و در صورت صحت ساختار فایل `.dot`، گراف متناظر با نام فایل و پسوند `.png` ذخیره می‌گردد. نمونه گراف حاصل از تجزیه یک فایل `.dot` در شکل (۱-۴) قابل مشاهده است. سپس در روند اجرای برنامه یک فایل متنی با نام فایل و پسوند `.log` با توجه به فرمت مشخص شده در شکل (۱-۵) برای لاگ مسیرهای اجرایی برنامه فراهم شده است.



شکل (۴-۱) نمونه تصویر ذخیره شده حاصل از تجزیه یک فایل dot

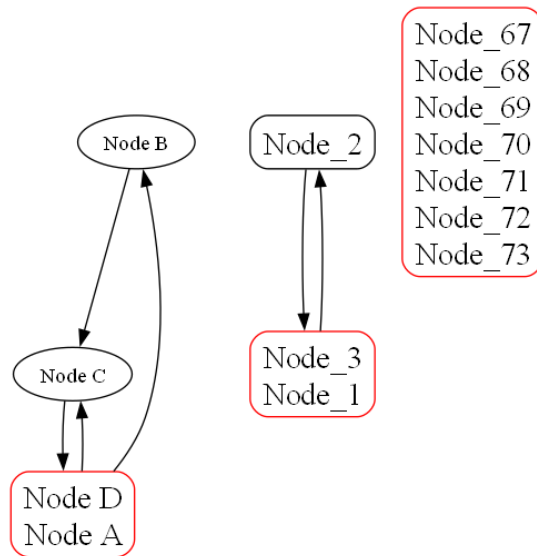
```

result > 1.log
You, 4 minutes ago | 1 author (You)
1 2024-07-27 23:50:11,043 - DEBUG - File path: D:\work\codes\dot_parser\input\1.dot
2 2024-07-27 23:50:11,605 - INFO - Original graph saved to: D:\work\codes\dot_parser\result\1.png
3 2024-07-27 23:50:11,615 - INFO - processing "sub_graph_1" subgraph ...
4 2024-07-27 23:50:11,615 - DEBUG - The initial number of nodes: 3
5 2024-07-27 23:50:11,615 - DEBUG - The number of deleted nodes: 2
6 2024-07-27 23:50:11,615 - DEBUG - The number of inserted nodes: 1
7 2024-07-27 23:50:11,615 - INFO - processing "sub_graph_2" subgraph ...
8 2024-07-27 23:50:11,615 - DEBUG - The initial number of nodes: 7
9 2024-07-27 23:50:11,615 - DEBUG - The number of deleted nodes: 12
10 2024-07-27 23:50:11,615 - DEBUG - The number of inserted nodes: 6
11 2024-07-27 23:50:11,615 - INFO - processing root graph ...
12 2024-07-27 23:50:11,615 - DEBUG - The initial number of nodes: 4
13 2024-07-27 23:50:11,615 - DEBUG - The number of deleted nodes: 2
14 2024-07-27 23:50:11,615 - DEBUG - The number of inserted nodes: 1
15 2024-07-27 23:50:11,742 - INFO - Updated graph saved to: D:\work\codes\dot_parser\result\1_updated.png
16 2024-07-27 23:50:11,742 - INFO - Updated DOT file saved to: D:\work\codes\dot_parser\result\1_updated.dot
17

```

شکل (۵-۱) لاگ اجرای برنامه

کد برنامه پس از شناسایی راس‌های متوالی گراف و ادغام آن‌ها فایل جدید dot. حاصل را تولید و خروجی جدید فایل به روزرسانی شده را در شکل فلان ذخیره می‌کند.



شکل (۶-۱) خروجی نهایی

۵-۱ آدرس گیت سورس کد برنامه

سورس کد پروژه در github موجود است. برای دسترسی به کد و مستندات کامل به آدرس زیر مراجعه کنید:

https://github.com/meysamkhazaei/dot_parser

پایان گزارش