1. Zadání

Cílem tohoto programu bylo vytvořit generátor názvů organických sloučenin podle IUPAC nomenklatury. Ve výsledku program umí pojmenovat sloučeniny alifatické[1] a alicyklické[2] z prvků C, H, N, O, S, které načítá ve SMILES formátu nebo graficky z nákresu uživatele.

2. Algoritmus a program

2.1. Načtení molekuly

V případě textového vstupu postupným načítáním SMILES formátu třídou SMILESParser vytváříme jednotlivé atomy, přičemž první označíme za začátek molekuly. Atomy na sebe vážeme se zadanou vazností, kdy např. dvojnou vazbu mezi dvěma atomy reprezentujeme tak, že se v poli ligandů objeví navázaný atom dvakrát.

Pokud uživatel molekulu zakresluje, tak při vytvoření bodu se do seznamu přidá atom daného typu se stejným id jako bod. Při vytvoření hrany mezi dvěma body se vytvoří vazba dané násobnosti mezi atomy s odpovídajícími id. Pro pojmenování se za start molekuly zvolí uhlík a atomy musí tvořit spojitý graf.

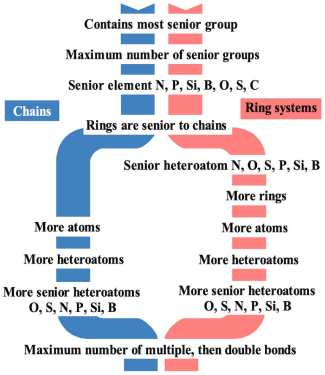
2.2. Tvorba molekuly

Základní jednotkou programu je instance třídy Atom, která slouží jako šablona pro konkrétní implementaci pro daný prvek. Prvek se vyznačuje svou vazností, která určuje velikost pole ligandů, a jednopísmenným symbolem. Atom si o sobě uchovává pro pojmenování relevantní informace, jako je množství násobných vazeb, navázané funkční skupiny a zda je či není součástí cyklu.

Navzájem navázané atomy tvoří instanci třídy Molekula, která je reprezentována jako neorientovaný multigraf. Celá molekula je přístupná ze startu, což je instance Atomu. Po načtení celou molekulu projdeme a poznamenáváme si informace o ní, jako je umístění funkčních skupin, násobných vazeb a začátků cyklů, tj. reference na atomy, kde se nacházejí.

2.3. Hledání řetězce

Po načtení a zpracování molekuly následuje nalezení hlavního řetězce podle vybraných IUPAC kritérii z následujících:



[1] Nearomatické, ne/větvené, ne/nasycené organické sloučeniny obsahují uhlík a další.

[2] Alifatické s uzavřeným uhlíkovým řetězcem.

Heteroatomy nejsou uvažovány, jelikož autor neumí jejich názvosloví a chybí mu vůle se ho doučit.

Nalezený řetězec je instancí třídy Chain a získáme ho filtrováním potenciálních řetězců pomocí statické třídy Filter a to následujícím způsobem:

1. Získáme množinu atomů, z nichž aspoň jeden musí být obsažen v řetězci, což je v následujícím pořadí buď množina atomů ...
    a. s funkční skupinou
    b. tvořící začátek cyklu
    c. které jsou listy

    Pokud je množina a prázdná, vezmeme b. Pokud i b je prázdné, tak c.
2. Pro každý atom z této množiny vytvoříme cesty do od nejvzdálenějších atomů a odstraňujeme z množiny atomy, které se v cestě vyskytnou. Takto pokračujeme, dokud množina není prázdná.

    Stojí za poznamenání, že v případě c je hlavní řetězec nejdelší cesta v grafu, kterou získáme pomocí dvou BFS, kdy nejprve ze startu molekuly nalezneme nejvzdálenější atomy (což budou listy), a nyní z těchto listů nalezneme nejvzdálenější atomy (opět listy), mezi nimiž nalezneme cestu.

    Ze snahy o optimalizaci vyplynulo, že v případě a a b, pokud zpracovávaný atom není list, chováme se k němu jako k listu, ale nalezené cesty ukládáme do hash mapy, kdy každou část zpracováváme zvlášť a poté spojíme nejlepší 2 části. To bylo problematické, pokud maximální počet zrovna filtrované struktury byl pouze v jedné takové částečné cestě, takže se musely ponechat i cesty s druhým takovým maximálním počtem, což vedlo k potřebě vytvořit vlastní funkce pro částečné cesty, což autor shledával velmi úmorným.
3. Získané množiny cest profiltrujeme podle daných kritérií, konkrétně podle:
    a. počtu funkčních skupin – v případě shody mají cykly přednost
    b. pro alifatické řetězce: počet atomů obsahují mezi ligandy začátek cyklu
    c. počtu uhlíků v řetězci
    d. počtu násobných vazeb

2.4. Tvorba názvu

Třídě Nomenclature se předá vybraný řetězec jako instance třídy Chain. Podle počtu atomů v řetězci se zvolí základ názvu.

Podle zvyklostí z autorovy střední, orientace řetězce je zvolena tak, aby součet lokantů byl co nejmenší. Do této chvíle měly oba atomy v násobné vazbě tento údaj o sobě poznačený, ale po zvolení orientace se tato informace uchovává pouze v prvním atomu ve vazbě.

Nyní k základu připojíme koncovku podle násobných vazeb, u dvojných a trojných i s lokanty.

Pokud má řetězec funkční skupinu vyšší než nitroskupinu, tak se to projeví na koncovce, jinak se případné další funkční skupiny objeví v předponě. V předponě se také objeví názvy vedlejších řetězců, které získáme rekurzivně tak, že z ligandů atomu značícího začátek vedlejšího řetězce odstraníme jeho souseda v hlavním řetězci a označíme ho za start nové molekuly a celý proces opakujeme. I přesto, že se v molekule

vyskytují cykly, tak se jedná o nezávislé podúlohy, neboť řetězce cyklů jsou zpracovávány samostatně. Z toho důvodu se dal cel proces urychlit použitím Parallel Stream.