Vyhledávání složitých vzorů: časté podgrafy

Christian Borgelt
(Jiří Kléma)

Katedra kybernetiky,
FEL, ČVUT v Praze

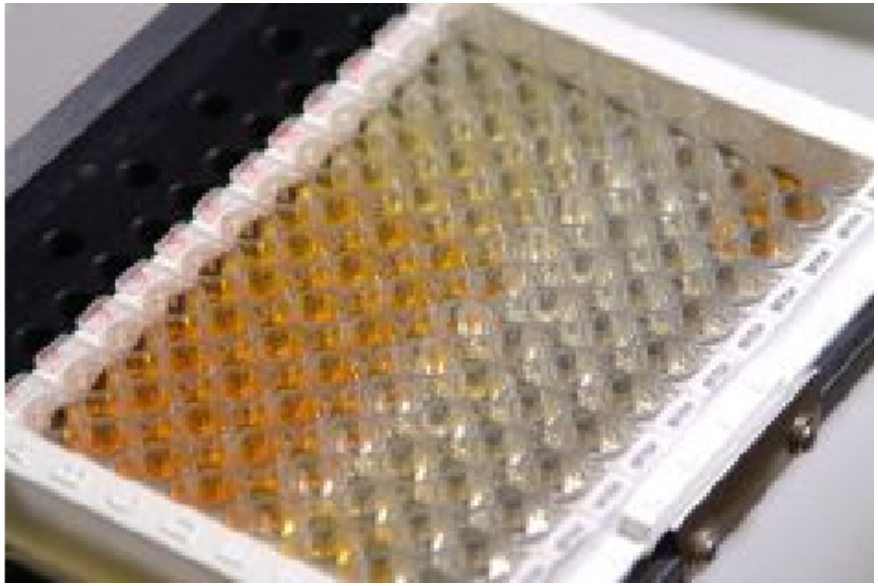
http://ida.felk.cvut.cz

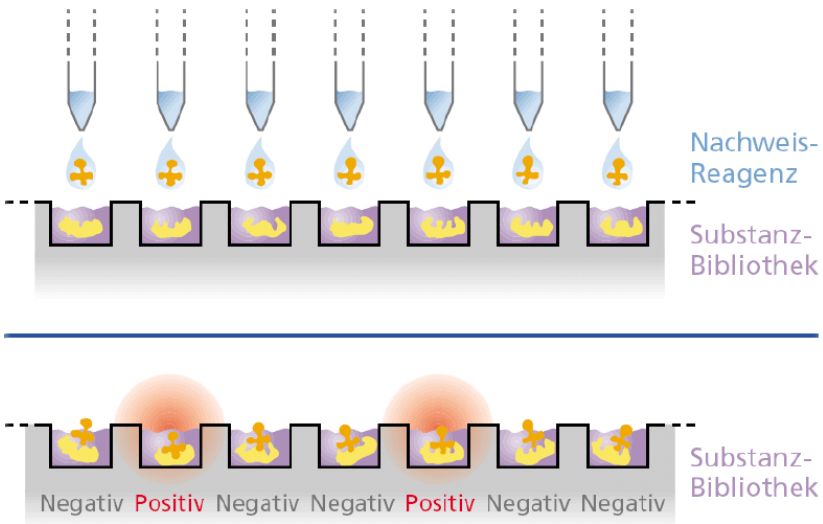
Struktura přednášky

- Motivace pro hledání častých podgrafů
  - praktické příklady, variabilita v zadání,
- nutné pojmy z teorie grafů
  - izomorfní podgraf,
  - častý, uzavřený a maximální podgraf,
- prohledávání prostoru podgrafů
  - kódová slova,
  - kanonická kódová slova,
  - v čem to urychluje prohledávání,
- shrnutí
  - co jsme se naučili,
  - co jsme nestihli probrat
    (přizpůsobení pro molekuly, stromy, jeden graf, repozitář fragmentů).

Časté podgrafy – ilustrace 1: molekulární fragmenty

- urychlení **vývoje léků**,
- př.: ochrana lidských CEM buněk před HIV infekcí (veřejná data),
  - velkokapacitní testování chemických sloučenin (37,171 substancí testováno)
    - 325 potvrzených aktivních (100% ochrana proti infekci),
    - 877 středně aktivních (50-99% ochrana proti infekci),
    - ostatní potvrzené neaktivní (<50% ochrana proti infekci),
  - problém: proč jsou některé sloučeniny aktivní a jiné ne?, kam dále směřovat test?



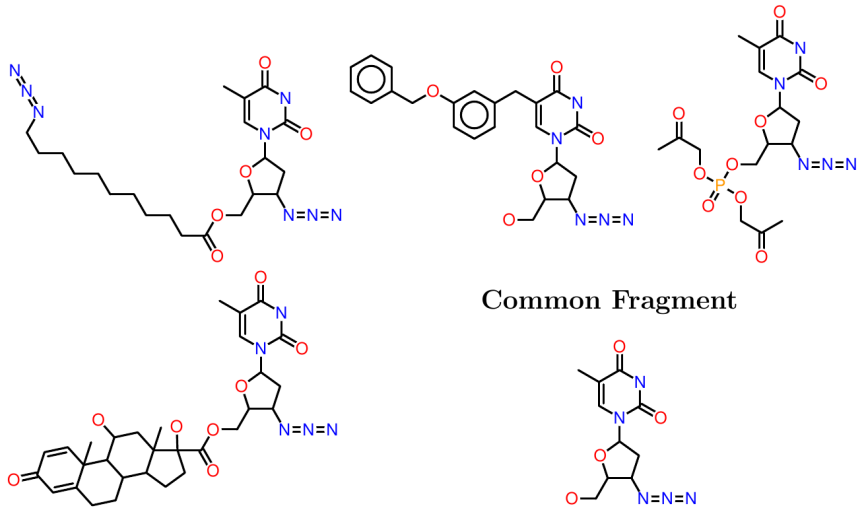


Časté podgrafy – ilustrace 1: molekulární fragmenty

- hledáme podstruktury společné pro aktivní substance – časté aktivní vzory = podgrafy,
  - jaké strukturální vlastnosti mají aktivní látky?
- hledáme diskriminativní vzory
  (tj. přidáváme podmínku, aby vzory byly řídké u inaktivních substancí),
  - kam směřovat další testy? jak by měl vypadat kandidát na příští lék?

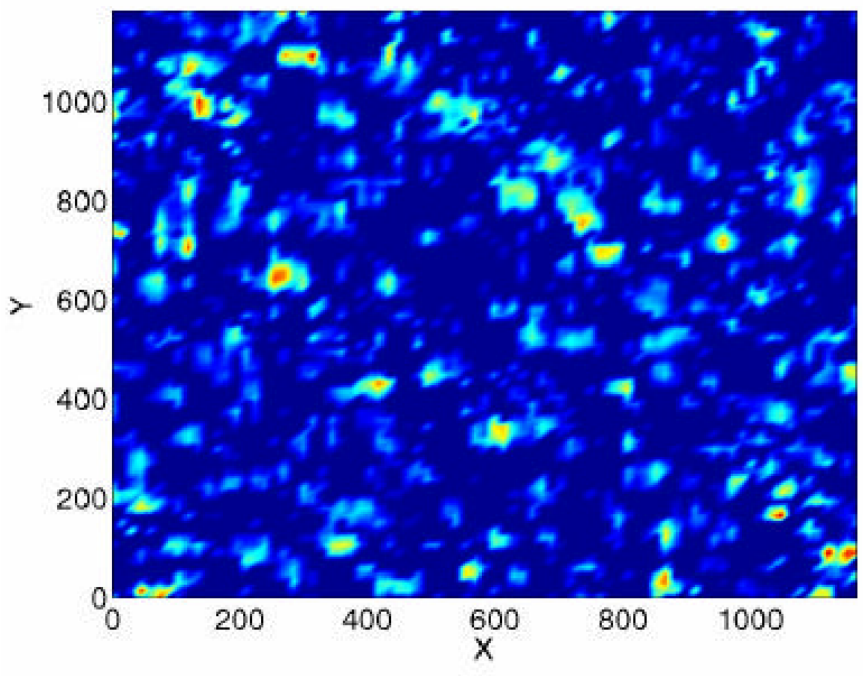
Excerpt from the NCI DTP HIV Antiviral Screen data set (SMILES format):

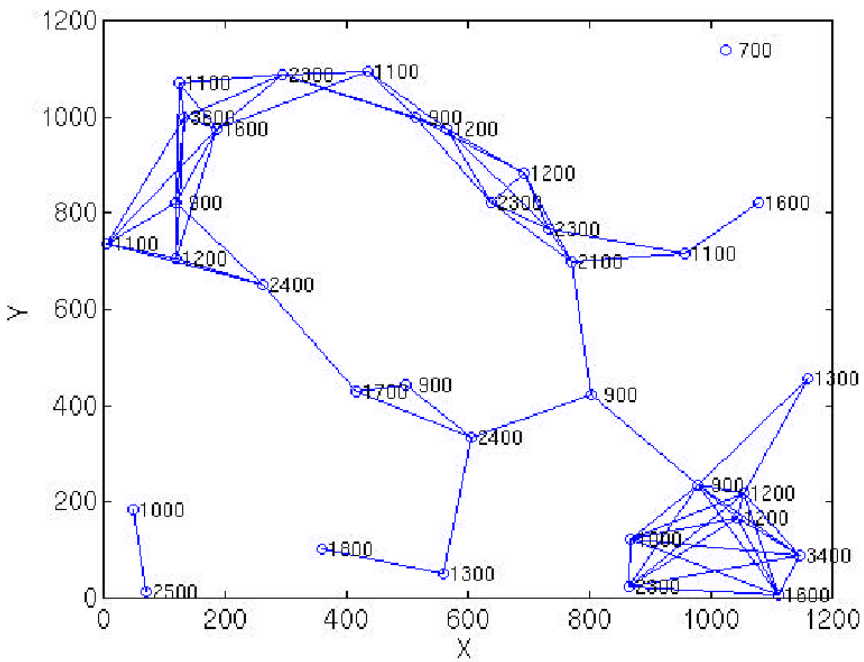
```
737,  0,CN(C)C1=[S+][Zn]2(S1)SC(=[S+]2)N(C)C
2018, 0,N#CC(=CC1=CC=CC=C1)C2=CC=CC=C2
19110,0,OC1=C2N=C(NC3=CC=CC=C3)SC2=NC=N1
20625,2,NC(=N)NC1=C(SSC2=C(NC(N)=N)C=CC=C2)C=CC=C1.OS(O)(=O)=O
22318,0,CCCCN(CCCC)C1=[S+][Cu]2(S1)SC(=[S+]2)N(CCCC)CCCC
24479,0,C[N+](C)(C)C1=CC2=C(NC3=CC=CC=C3S2)N=N1
50848,2,CC1=C2C=CC=CC2=N[C-](CSC3=CC=CC=C3)[N+]1=O
51342,0,OC1=C2C=NC(=NC2=C(O)N=N1)NC3=CC=C(Cl)C=C3
55721,0,NC1=NC(=C(N=O)C(=N1)O)NC2=CC(=C(Cl)C=C2)Cl
55917,0,O=C(N1CCCC[CH]1C2=CC=CN=C2)C3=CC=CC=C3
64054,2,CC1=C(SC[C-]2N=C3C=CC=CC3=C(C)[N+]2=O)C=CC=C1
64055,1,CC1=CC=CC(=C1)SC[C-]2N=C3C=CC=CC3=C(C)[N+]2=O
64057,2,CC1=C2C=CC=CC2=N[C-](CSC3=NC4=CC=CC=C4S3)[N+]1=O
66151,0,[O-][N+](=O)C1=CC2=C(C=NN=C2C=C1)N3CC3
...
```



Časté podsekvence – ilustrace 2: dynamika plynů a kapalin

- měření: velikost, rychlost a lokalizace vírů,
- grafová reprezentace: víry = vrcholy, hrany = blízkost,
- lze studovat časté vzory např. předcházející vzniku anomálií
  - meteorologie, aerodynamika, hydraulika.





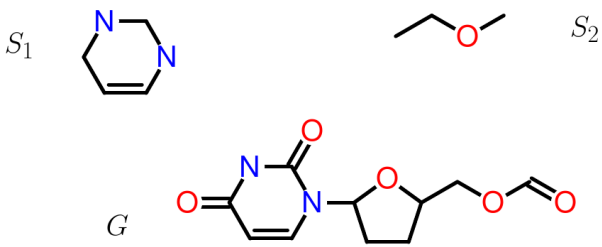
Grafy: základní pojmy

- Množina atributů, resp. značek $A = \{a_1, \ldots, a_m\}$,
  - př. atributů pro molekuly:
    chemický prvek, náboj, typ vazby (jednoduchá, dvojná, trojná, aromatická),
- ohodnocený graf je trojice $G = (V, E, \ell)$, kde
  - V je množina vrcholů/uzlů,
  - $E \subseteq V \times V - \{(v, v) \mid v \in V\}$ je množina hran,
  - $\ell : V \cup E \rightarrow A$ přiřazuje značky z množiny A uzlům i hranám,
  - G je neorientovaný, prostý a bez smyček,
  - více uzlů a hran může být označeno stejnou značkou,
- **podgraf** $S \subseteq G$
  - neformálně: vypustíme některé uzly a všechny jejich incidentní hrany
    (úplný, plný, indukovaný podgraf),
  - pokud vypustím i další hrany jde o podgraf bez přívlastku úplný,
  - vlastní podgraf $S \subset G$.

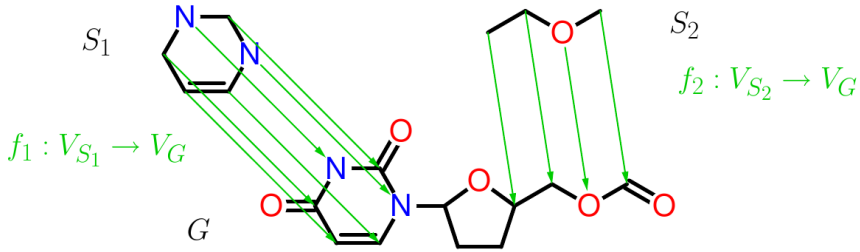
Grafy: základní pojmy

- souvislá komponenta grafu
  - souvislý podgraf, jakýkoli jeho nadgraf není souvislý,
- vrchol lze označit za
  - izolovaný – nemá žádnou incidentní hranu,
  - list – má právě jednu incidentní hranu,
- hranu lze označit za
  - most (bridge) – odebráním hrany zvýšíme počet souvislých komponent,
  - pravý (listový) most (proper (leaf) bridge) – je mostem a není (je) incidentní k listu,
- grafy $S = (V_S, E_S, \ell_S)$ a $G = (V_G, E_G, \ell_G)$ jsou **izomorfní** ($S \equiv G$), pokud
  - $\exists\ f\colon V_S \to V_G$ (bijekce) taková, že:
    - $\ell_S(v) = \ell_G(f(v))$
    - $(x, y) \in E_S \Leftrightarrow (f(x), f(y)) \in E_G \quad \wedge \quad \ell_S((u, v)) = \ell_G((f(u), f(v)))$,
- graf S je **izomorfním podgrafem** G (S se vyskytuje v G, $S \sqsubseteq G$), pokud
  - f omezíme na injektivní funkci $\forall v \in V_S$,
  - $S \sqsubseteq G \wedge G \sqsubseteq S \Leftrightarrow S \equiv G$.
- ověření zda S je izomorfním podgrafem G je NP-úplný problém!

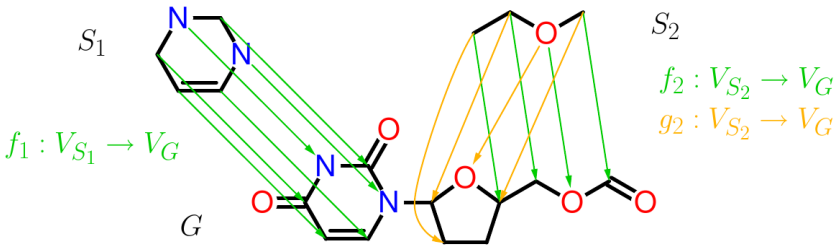
Příklady izomorfních podgrafů



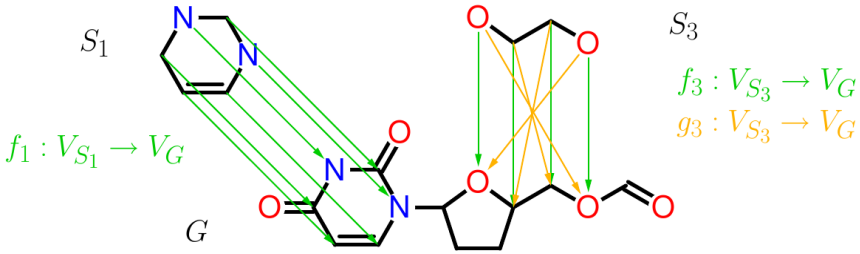
S_1 i S_2 jsou podgrafem G



nalezena izomorfní zobrazení f_1 a f_2
(zachovávající typ atomu i chemické vazby)



S_2 má více výskytů v G



pro S_3 existuje automorfismus
(S_3 lze neidenticky mapovat na sebe samé)
(S_3 má více výskytů na jediné lokaci v G)

Grafy: definice podpory

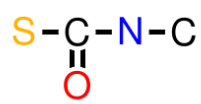
- G pokrývá S pokud
  - $S \sqsubseteq G$ (současně S je obsaženo (vyskytuje se) v G),
- S je řádně obsažen (řádně se vyskytuje) v G pokud
  - $S \sqsubset G \Leftrightarrow S \not\equiv G \wedge S \sqsubseteq G$,
- máme-li množinu grafů $\mathcal{G} = \{G_1, \ldots, G_n\}$, pokrytí S vzhledem ke $\mathcal{G}$ je
  - $K_{\mathcal{G}}(S) = \{k \in \{1, \ldots, n\} \mid S \sqsubseteq G_k\}$,
  - jde o množinu indexů všech grafů, které pokrývají S,
- (absolutní) podporou S vzhledem ke $\mathcal{G}$ je kladné celé číslo
  - $s_{\mathcal{G}}(S) = |K_{\mathcal{G}}(S)|$,
  - jde o počet grafů, které pokrývají S
    (neuvažujeme opakující se výskyty v jednom grafu),
- častý podgraf S vzhledem ke $\mathcal{G}$ je každý podgraf, pro který
  - $s_{\mathcal{G}}(S) \geq s_{min}$.

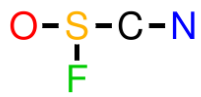
Problém vyhledání častých podgrafů v množině grafů

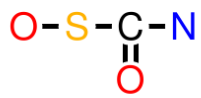
- vstup: grafy $\mathcal{G} = \{G_1, \ldots, G_n\}$ se značkami $A = \{a_1, \ldots, a_m\}$ a minimální podpora s_{min}
- výstup: množina častých podgrafů s podporou větší než minimální
  - $F_{\mathcal{G}}(s_{\min}) = \{S \mid s_{\mathcal{G}}(S) \geq s_{\min}\}$,
- obvyklé omezení
  - pouze souvislé podgrafy,
- hlavní problém
  - vyhnout se **redundanci** při prohledávání
    - kanonická reprezentace (pod)grafů,
    - částečné uspořádání prostoru (pod)grafů,
    - efektivní prořezávání prohledávacího prostoru,
    - úschovna zpracovaných fragmentů grafů.
- APRIORI vlastnost lze zobecnit i pro grafy
  - **Každý podgraf častého grafu je častý.** (antimonotónní)
  - **Podgraf, který není častý, nemůže mít častý nadgraf.** (monotónní)
  - $\forall S : \forall R \supseteq S : \quad s_{\mathcal{G}}(R) \leq s_{\mathcal{G}}(S)$.

Časté podgrafy: příklad

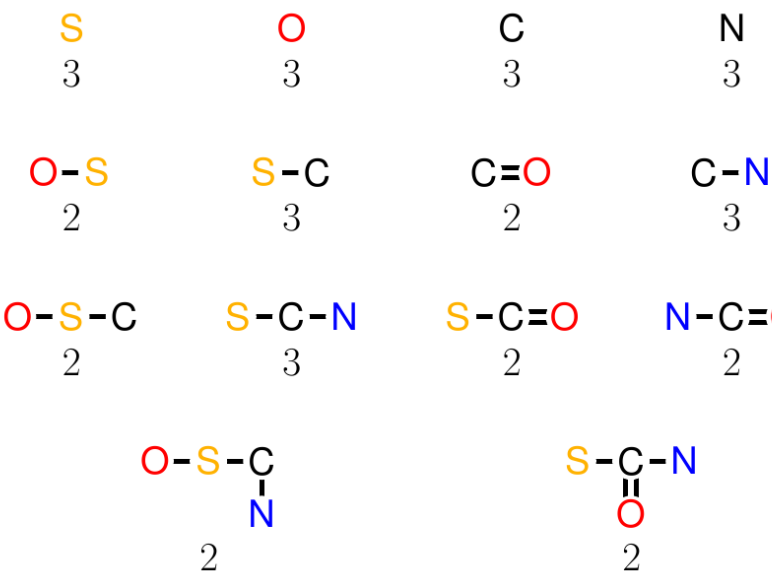
- $\mathcal{G}$ je tvořena trojicí molekul,
- existuje 15 podgrafů s podporou alespoň $s_{min} = 2$,
- prázdný podgraf je z definice řádně obsažen ve všech grafech.







$*$ $(s_{\min} = 2)$
3

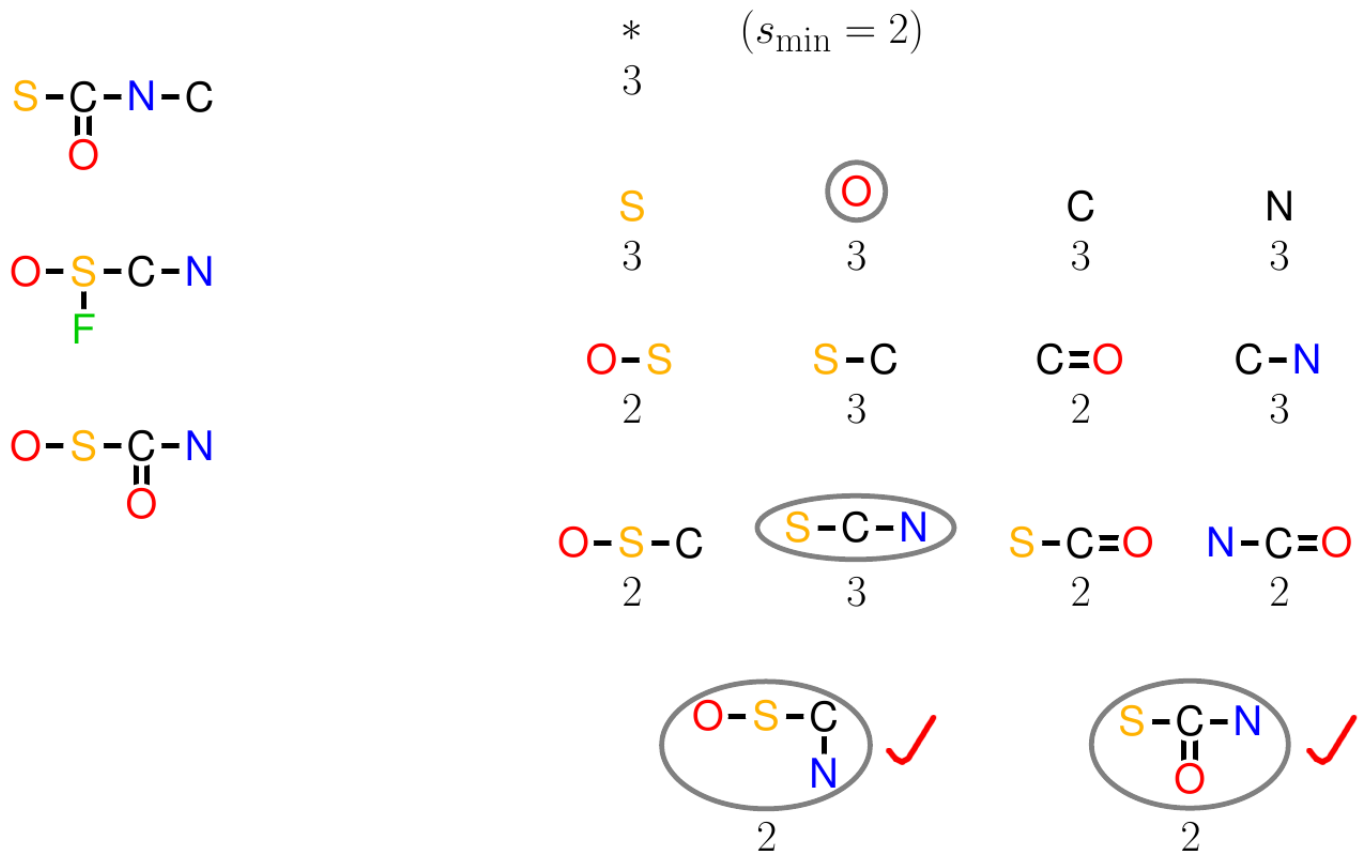


Typy častých podgrafů – uzavřenost a maximalita

- **maximální** podgraf (maximal subgraph)
  - je častý a žádný z jeho vlastních nadgrafů není častý, množina všech maximálních podgrafů: $M_{\mathcal{G}}(s_{\min}) = \{S \mid s_{\mathcal{G}}(S) \geq s_{\min} \wedge \forall R \supset S : s_{\mathcal{G}}(R) < s_{\min}\}$,
  - každý častý podgraf má maximální nadgraf,
  - žádný nadgraf maximálního grafu není častý,
  - z $M_{\mathcal{G}}(s_{\min})$ (a jejich podpory) nelze určit podporu všech častých podgrafů,
- **uzavřený** podgraf (closed subgraph)
  - je častý a žádný z jeho vlastních nadgrafů nemá identickou podporu,
  - množina všech uzavřených podgrafů: $C_{\mathcal{G}}(s_{\min}) = \{S \mid s_{\mathcal{G}}(S) \geq s_{\min} \wedge \forall R \supset S : s_{\mathcal{G}}(R) < s_{\mathcal{G}}(S)\}$,
  - každý častý podgraf má uzavřený nadgraf s identickou podporou,
  - z $C_{\mathcal{G}}(s_{\min})$ (a jejich podpory) lze určit podporu všech častých podgrafů,
- evidentní vztahy mezi pojmy
  - každý maximální i uzavřený podgraf je zároveň častý,
  - každý maximální podgraf je uzavřený.

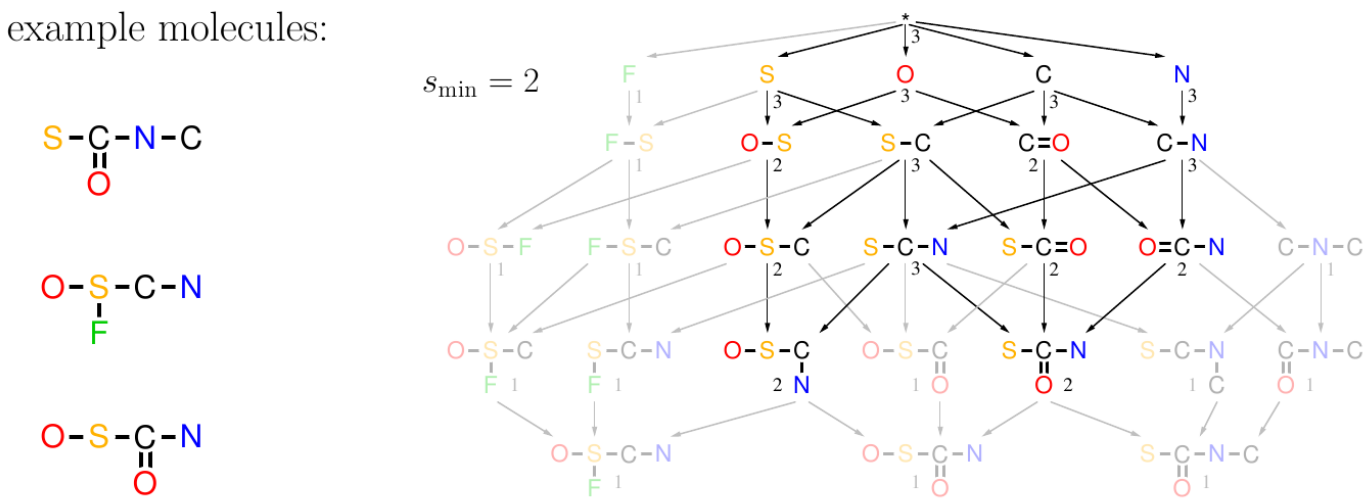
Uzavřené a maximální podgrafy: příklad

- $\mathcal{G}$ je tvořena trojicí molekul, $s_{min} = 2$,
- existují 4 uzavřené podgrafy, 2 z nich jsou maximální.



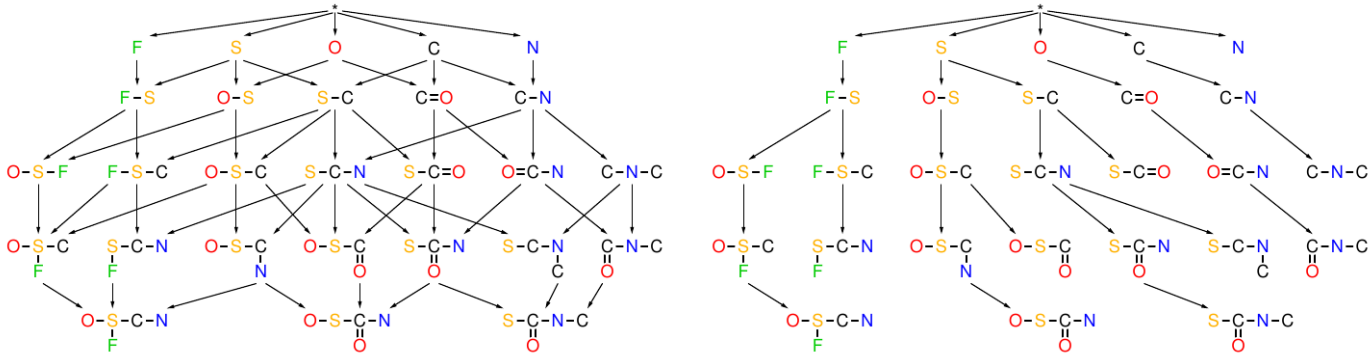
Prohledávání částečně uspořádané množiny podgrafů

- Izomorfismus podgrafů definuje částečné uspořádání podgrafů
  - lze vytvořit Hasseův diagram, prázdný podgraf je infimem, přirozené supremum neexistuje,
  - diagram lze úplně prohledat směrem od prázdného podgrafu,
  - pro velký větvící faktor upřednostníme obvykle prohledávání do hloubky.
- hlavní problém
  - jak zajistit, že podgraf nebude generován více způsoby?
  - diagram musíme převést na strom – každý podgraf má jediného rodiče.



Prohledávání částečně uspořádané množiny podgrafů

- Algoritmus prohledávání stromu podgrafů
  - (podgrafy s jediným rodičem),
- Počáteční cyklus
  - prozkoumej všechny možné atributy uzlu,
  - rekurzivně rozvíjej všechny časté atributy jako podgrafy,
- Rekurzivní zanořování pro každý častý podgraf S
  - generuj všechna rozšíření R podgrafu S přidáním hrany nebo uzlu a hrany
    - přidáváme hranu $(u, v) \notin E_S$, $u \in V_S \vee v \in V_S$,
    - pokud $u \notin V_S \vee v \notin V_S$, přidáme i chybějící uzel,
  - S musí být unikátním rodičem R,
  - je-li R časté, dále jej rozšiřuj, jinak STOP.



Přiřazení jediného rodiče

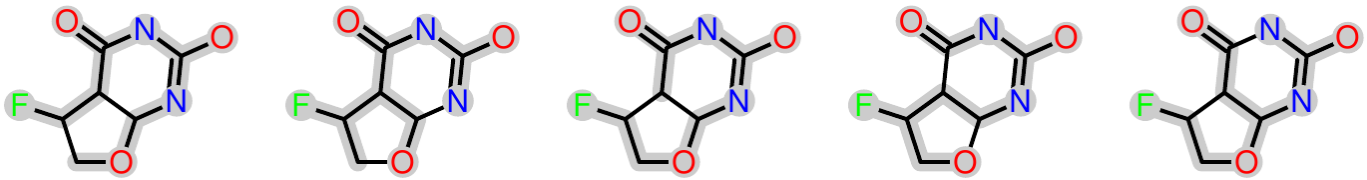
- Jak lze množinu rodičů podgrafu S vymezit
  - podgrafy, které obsahují právě o jednu hranu méně než S,
  - jde o maximální vlastní podgrafy,
- **Kanonický (unikátní) rodič** $p_c(S)$ podgrafu S
  - zavedení předchází definice uspořádání hran podgrafu S,
  - nechť e^* je poslední hranou podgrafu S, která není řádným mostem,
  - pak $p_c(S)$ je podgrafem S bez hrany e^*,
  - je-li e^* listovým mostem odebereme také izolovaný uzel, který vznikne,
  - pro situace s jedinou hranou v S potřebujeme zavést i pořadí uzlů.
- Pořadí hran bude vycházet z **kanonické reprezentace grafu**
  - graf bude popsán kódovým slovem,
  - to bude jednoznačně identifikovat graf (nerozliší pouze automorfismus),
  - pokud existuje více slov pro jeden graf, lexikograficky vybereme jediné slovo.

Kanonická reprezentace grafů

- Základní myšlenka
  - znaky kódového slova popíší hrany grafu,
  - značky ale nejsou jednoznačné, je třeba zavést jednoznačné číslování vrcholů,
- obvyklý požadavek na kanonickou formu
  - prefixovost – každý prefix kanonického kódového slova je také kanonickým kódovým slovem,
  - odebráním poslední hrany e^* vznikne kanonické slovo a půjde o kanonického rodiče,
- dopad prefixovosti na algoritmus prohledávání stromu podgrafů
  - z rodiče vygenerujeme následníka rozšířením jeho kanonického kódového slova,
  - stačí ověřit, že jde o kanonické kódové slovo,
  - důsledek: rychle generujeme neredundantní následníky rodičů,
- nejčastější přístupy ke kanonické formě
  - kostry grafu,
  - matice sousednosti.

Kanonická reprezentace grafů vycházející z koster

- Kódové slovo grafu je tvořeno při vytváření **kostry grafu**
  - vrcholy jsou číslovány v pořadí daném pořadím průchodu kostrou,
  - hrany jsou popsány čísly výchozího a koncového vrcholu, svojí značkou a značkami uzlů,
  - kódové slovo popisuje hrany, v pořadí jejich průchodu kostrou (hrany mimo kostru musí být zařazeny také v jiném režimu),
- jak lze kostru tvořit
  - prohledávání: do hloubky $\times$ do šířky,
  - každý ze způsobů vyžaduje vlastní způsob konstrukce kódového slova,
- jeden graf může být vyjádřen velkým množstvím kódových slov
  - graf má více koster (počáteční uzel, volby při větvení),
  - jak rychle najít to lexikograficky nejmenší = kanonické?
  - prefixová vlastnost je zajištěna přidáváním hran na konec,
  - stačí ověřit, že rozšíření je také kanonické.



Kanonická reprezentace grafů vycházející z koster

- Zavedeme uspořádání značek
  - kvůli efektivitě je vhodné přihlížet k frekvenci,
  - časté značky vrcholů je vhodné řadit dozadu,
- Regulární výrazy pro kódová slova
  - do hloubky: $a\,(i_d\,\underline{i_s}\,b\,a)^m$,
    (<u>výjimka: indexy v klesajícím pořadí</u>)
  - do šířky: $a\,(i_s\,b\,a\,i_d)^m$ (nebo $a\,(i_s\,i_d\,b\,a)^m$),
  - vysvětlení symbolů:

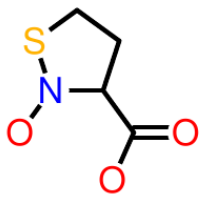
    n počet vrcholů grafu,
    m počet hran grafu,
    i_s index výchozího vrcholu hrany, $i_s \in \{0, \ldots, n-1\}$,
    i_d index cílového vrcholu hrany, $i_d \in \{0, \ldots, n-1\}$,
    a značka vrcholu,
    b značka hrany.

Kanonické kostry: příklad

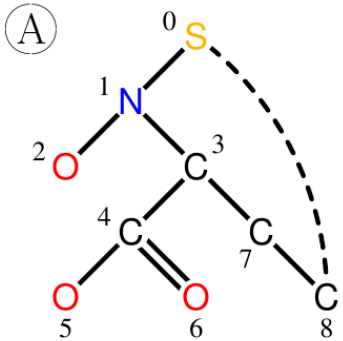
- Zavedeme uspořádání značek
  - vrcholy: $S \prec N \prec O \prec C$, hrany: $- \prec =$,
- Kódová slova

A: `S 10-N 21-O 31-C 43-C 54-O 64=O 73-C 87-C 80-C`

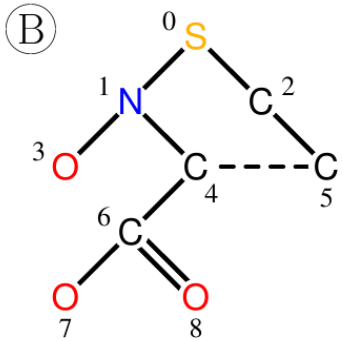
B: `S 0-N1 0-C2 1-O3 1-C4 2-C5 4-C5 4-C6 6-O7 6=O8`



example molecule



depth-first



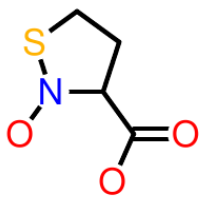
breadth-first

Rekurzivní ověření kanonicity slova

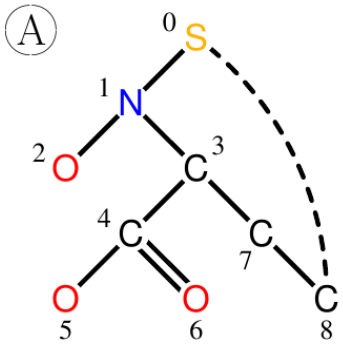
- vyjdi ze všech vrcholů se značkou s nižším a stejným pořadím než je současný kořen kostry,
- rekurzivně přidávej hrany, srovnej slovo s potenciálně kanonickým
  - je-li nova hrana lex. větší než aktuální v ověřovaném slově, backtrack
  - je-li nova hrana lex. menší než aktuální v ověřovaném slově, ověřované slovo není kanonické,
  - je-li nova hrana lex. identická s aktuální v ověřovaném slově, pokračuj v rozvoji.

A: `S 10-N 21-O 31-C 43-C 54-O 64=O 73-C 87-C 80-C`

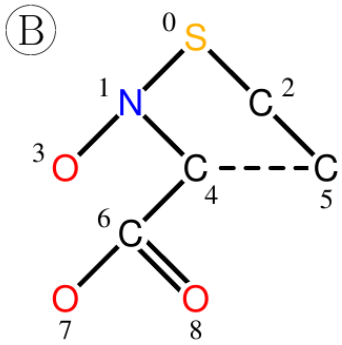
B: `S 0-N1 0-C2 1-O3 1-C4 2-C5 4-C5 4-C6 6-O7 6=O8`



example molecule



depth-first



breadth-first

Rychlé odmítnutí rozšíření kanonického slova

- Postup při rekurzivním prohledávání stromu podgrafů
  - generuj všechna rozšíření kanonického kódového slova častého rodiče,
  - rozšíření vznikají doplněním hrany, která rozšiřuje daný podgraf,
  - prefixová reprezentace: popis hrany přidáme na konec kódového slova,
  - ověříme kanonicitu slova, pokud je splněna pokračujeme rekurzivně, jinak větev ukončíme,
- jak ověřit, zda jde o kanonické slovo?
  - obecně nutno nalézt lex. menší slovo se stejným kořenovým vrcholem,
  - lze uplatňovat jednoduchá pravidla, která rozšíření odmítají na základě lokálního kritéria
    - k některým vrcholům nelze hranu připojit,
    - některé cykly nelze uzavřít,
    - jde o podmínky nutné, nikoli postačující.

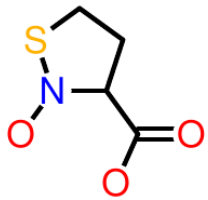
Rychlé odmítnutí rozšíření kanonického slova

- Do hloubky: **rozšíření nejpravější cesty**
  - rozšiřitelné vrcholy
    - musí ležet na větvi nejvíce vpravo
      (k jiným vrcholům nelze hranu v dané větvi prohledávacího stromu připojit),
    - pokud vrchol není listem, nelze přeskakovat hrany
      (značka hrany nesmí předcházet značku stávající nejpravější hrany,
      při shodě značek musí totéž platit pro cílové vrcholy hran),
  - hrany uzavírající cykly
    - musí začínat v rozšiřitelném vrcholu,
    - musí končit v rozšiřitelném listu
      (jediný vrchol v podgrafu),
    - index počátečního vrcholu musí předcházet indexy všech počátečních vrcholů hran vedoucích do rozšiřitelného listu,

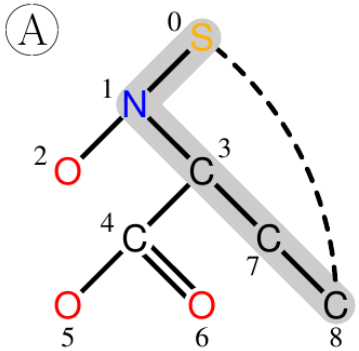
Rychlé odmítnutí rozšíření kanonického slova

- Do šířky: **rozšíření podle maximálního počátku**
  - rozšiřitelné vrcholy
    - pouze s indexem stejným nebo vyšším než je maximální index počátečního vrcholu v podgrafu,
    - pokud má hrana stejný počáteční index jako je ten maximální, je třeba ověřit i precedenci hran (viz do hloubky)
  - hrany uzavírající cykly
    - musí začínat v rozšiřitelném vrcholu,
    - musí vest vpřed, od vrcholu s nižším indexem do vrcholu s vyšším.

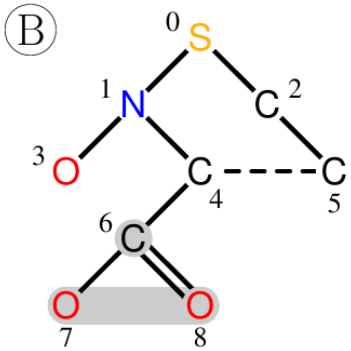
Rozšíření kanonického slova: příklad



example molecule



depth-first



breadth-first

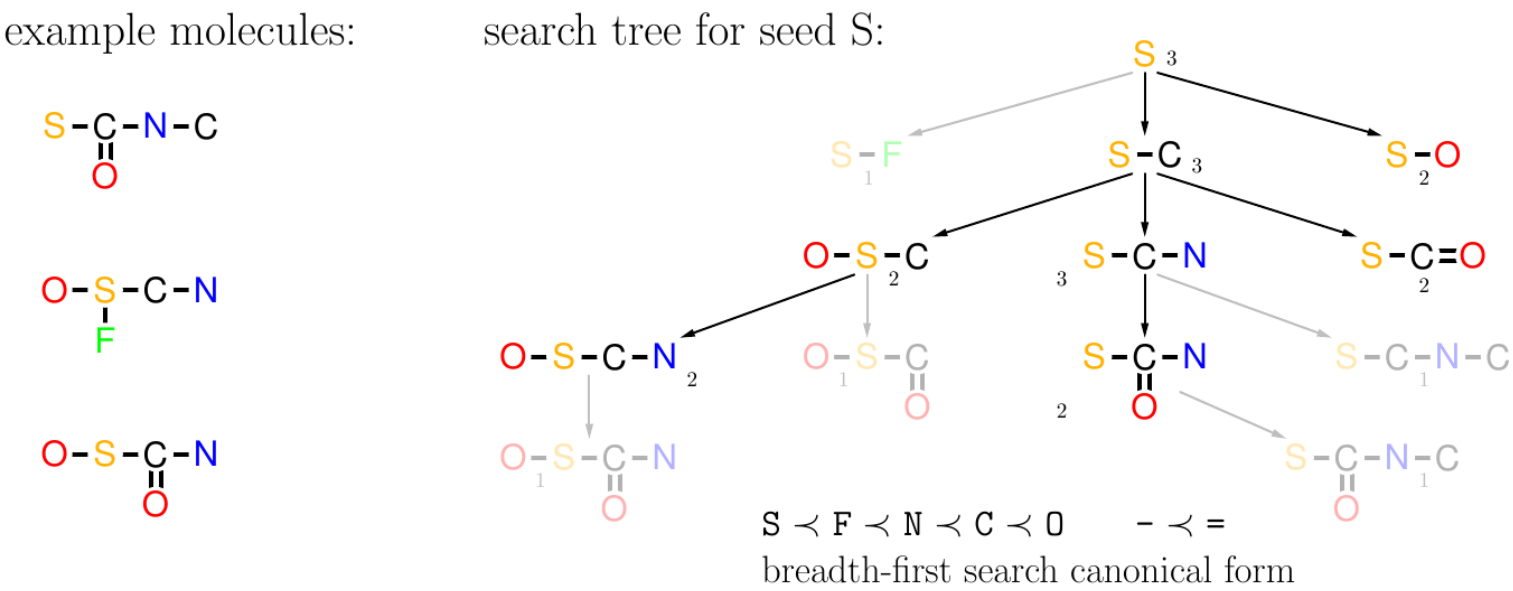
- Rozšiřitelnost
  - vrcholy: ad A: 0, 1, 3, 7, 8, ad B: 6, 7, 8,
  - uzavírací hrany: ad A: žádná, ad B: hrana od 7 k 8,
- Rozšíření: přidej jednovazební uhlík k nejlevějšímu kyslíku

A: `S 10-N 21-O 31-C 43-C 54-O 64=O 73-C 87-C 80-C 92-C`
`S 10-N 21-O 32-C` ···

B: `S 0-N1 0-C2 1-O3 1-C4 2-C5 4-C5 4-C6 6-O7 6=O8 3-C9`
`S 0-N1 0-C2 1-O3 1-C4 2-C5 3-C6` ···

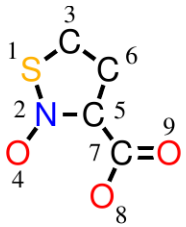
Prohledávaní prostoru častých podgrafů s kanonickými slovy: příklad

- Začneme s jádrovým vrcholem,
- přidáme hranu (a možná i uzel) v každém kroku,
- rychle odmítáme nekanonická rozšíření, ověříme kanonicitu zbylých,
- ověříme podporu kandidátů (mimo prostor kódových slov),
- prořežeme podgrafy s nedostatečnou podporou.

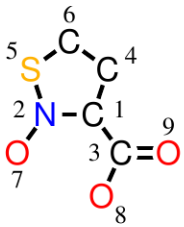


Kanonická reprezentace grafů vycházející z matice sousednosti

- **Matice sousednosti**
  - častá reprezentace grafu, více tvarů závislých na číslování vrcholů,
  - graf G s n vrcholy je popsán maticí $n \times n$ maticí $\mathbf{A} = (a_{ij})$,
  - $a_{ij} = 1$ pokud mezi vrcholy s čísly i a j vede hrana, jinak $a_{ij} = 0$.
- **Rozšířená matice sousednosti**
  - pro ohodnocený graf G (se značkami u vrcholů a hran),
  - obsahuje navíc sloupec se značkami vrcholů,
  - a_{ij} obsahuje značku hrany, pokud nějaká hrana mezi vrcholy s čísly i a j vede, jinak $a_{ij} = \times$.



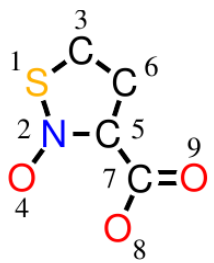
		1	2	3	4	5	6	7	8	9
1	S	×	–	–	×	×	×	×	×	×
2	N	–	×	×	–	–	×	×	×	×
3	C	–	×	×	×	×	–	×	×	×
4	O	×	–	×	×	×	×	×	×	×
5	C	×	–	×	×	×	–	–	×	×
6	C	×	×	–	×	–	×	×	×	×
7	C	×	×	×	×	–	×	×	–	=
8	O	×	×	×	×	×	–	–	×	×
9	O	×	×	×	×	×	×	=	×	×



		1	2	3	4	5	6	7	8	9
1	C	×	–	–	–	×	×	×	×	×
2	N	–	×	×	×	–	×	–	×	×
3	C	–	×	×	×	×	×	×	–	=
4	C	–	×	×	×	×	–	×	×	×
5	S	×	–	×	×	×	–	×	×	×
6	C	×	×	×	–	–	×	×	×	×
7	O	×	–	×	×	×	×	×	×	×
8	O	×	×	–	×	×	×	×	×	×
9	O	×	×	=	×	×	×	×	×	×

Od matice sousednosti ke kódovému slovu

- jednoduchým zápisem elementů matice po řádcích,
- pro neorientované grafy je matice symetrická – stačí zapsat horní trojúhelník,
- zahuštěná reprezentace kódového slova
  - zapisujeme pouze existující hrany,
  - je nutné přiřadit i identifikátory sloupců,
  - vhodné pro řídké matice.



Regular expression (non-terminals):
$(a\ (\ i_c\ b\ )^*\ )^n$

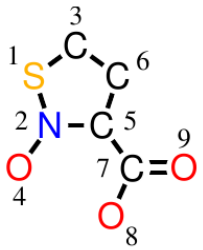
		1	2	3	4	5	6	7	8	9
1	S	×	–	–	×	×	×	×	×	×
2	N	–	×	×	–	–	×	×	×	×
3	C	–	×	×	×	×	–	×	×	×
4	O	×	–	×	×	×	×	×	×	×
5	C	×	–	×	×	×	–	–	×	×
6	C	×	×	–	×	–	×	×	×	×
7	C	×	×	×	×	–	×	×	–	=
8	O	×	×	×	×	×	–	–	×	×
9	O	×	×	×	×	×	×	=	×	×

code word:

```
S 2 - 3 -
N 4 - 5 -
C 6 -
O
C 6 - 7 -
C
C 8 - 9 =
O
O
```

Kanonické rozšířené matice sousednosti

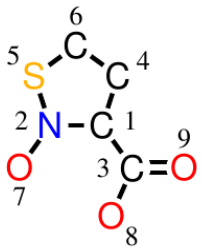
- klíčovým problémem je nalezení kanonického kódového slova
  - opět vychází z lexikografického uspořádání značek
    vrcholy: $S \prec N \prec O \prec C$, hrany: $- \prec =$,
  - kanonické slovo je lexikograficky nejmenší,
  - u matic sousednosti je ještě větší počet kandidátských slov než u koster,
  - konstrukce slova z horní matice po řádcích má výhodu prefixovosti.
- ukázka kanonického a nekanonického kódového slova



S 2 - 3 - N 4 - 5 - C 6 - O C 6 - 7 - C C 8 - 9 = O O

$<$

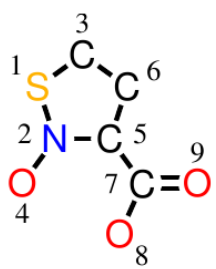
C 2 - 3 - 4 - N 5 - 7 - C 8 - 9 = C 6 - S 6 - C O O O



- triviální pozorování
  - jeden z vrcholů s minimální značkou musí mít index 1,
  - různě značkované hrany definují pořadí dalších vrcholů jednoznačně,
  - nejsnažší konstrukce pro unikátní značky, jinak nutný backtracking.

Kanonické rozšířené matice sousednosti

- jak rozlišit mezi vrcholy a hranami se stejnou značkou?
  - zavedeme **podpis vrcholu**
  - pro každý vrchol zaznamenáme celou sousední strukturu grafu,
  - tu rozšiřujeme tak dlouho, dokud nerozlišíme podpisy libovolné dvojice vrcholů,
  - iterativně rozdělujeme **třídy ekvivalence** vrcholů.



vertex	signature
1	S
2	N
4	O
8	O
9	O
3	C
6	C
5	C
7	C

vertex	signature
1	S
2	N
4	O -
8	O -
9	O =
3	C --
6	C --
5	C ---
7	C --=

vertex	signature
1	S
2	N
4	O - N
8	O - C --=
9	O =
3	C -- S C --
6	C -- C -- C ---
5	C ---
7	C --=

Další témata

- Repozitář fragmentů
  - dominantní způsob omezení redundance jsou kanonická slova,
  - alternativou je uchovávat už zpracované grafy, ty už se znovu neprocházejí,
  - klíčová témata: paměť, rychlý přístup (hash),
- přizpůsobení pro molekuly
  - často se vyskytující fragmenty molekul zpracovávat najednou,
  - cykly, uhlíkové řetězce, třídy ekvivalentních atomů,
- jeden graf
  - jiná definice podpory (obtížnější),
- stromy
  - uspořádané $\times$ neuspořádané,
  - v zásadě snazší než obecné grafy.

Časté podgrafy – shrnutí

- Problém podobný vyhledávání častých množin položek
  - APRIORI vlastnost,
  - komplikovanější je ale zamezení redundance při prohledávání,
    - větší větvící faktor,
    - množiny položek nemají strukturu,
  - netriviální kanonická reprezentace grafů
    - zajistí, že četnost každého podgrafu ověřujeme nejvýše jednou (další nutnou podmínkou je četnost rodiče),
    - volba reprezentace souvisí s volbou prohledávacího algoritmu,
    - prefixovost reprezentace umožní rychlé zamítnutí nekanonických kandidátů,
  - dvě ukázky kanonické reprezentace
    - založená na kostrách,
    - založená na matici sousednosti,
- příště: aplikační a přehledová přednáška – texty a web.

Doporučené doplňky – zdroje přednášky

:: Četba

- Borgelt: **Frequent Pattern Mining.**
  - přednáška je výběrem z grafového bloku z tohoto kurzu,
  - http://www.borgelt.net/teach/fpm/slides.html.
- Nijssen, Kok: **The Gaston Tool for Frequent Subgraph Mining.**
  - často používaný nástroj Gaston, aplikace na molekulární databáze,
  - http://www.liacs.nl/~snijssen/gaston/index.html,
- Yan, Han: **gSpan: Graph-Based Substructure Pattern Mining.**
  - často používaný nástroj gSpan,
  - http://www.cs.ucsb.edu/~xyan/software/gSpan.htm.