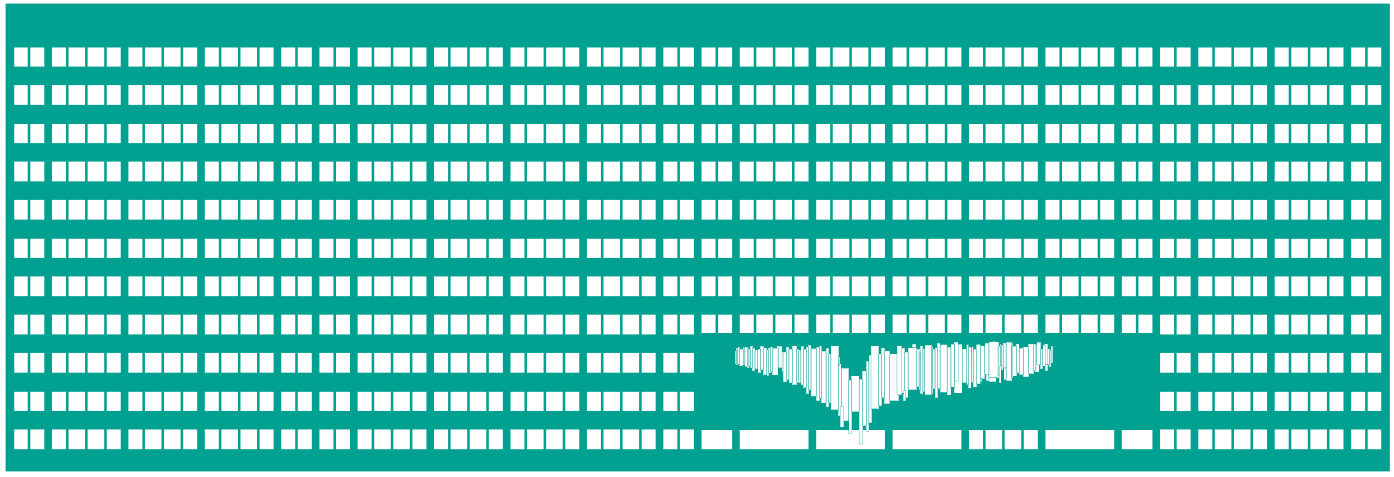
VŠB TECHNICKÁ UNIVERZITA OSTRAVA

VSB TECHNICAL UNIVERSITY OF OSTRAVA



www.vsb.cz

Algoritmy pro Bioinformatiku

Pokročilá témata

Michal Vašinek

VŠB – Technická univerzita Ostrava

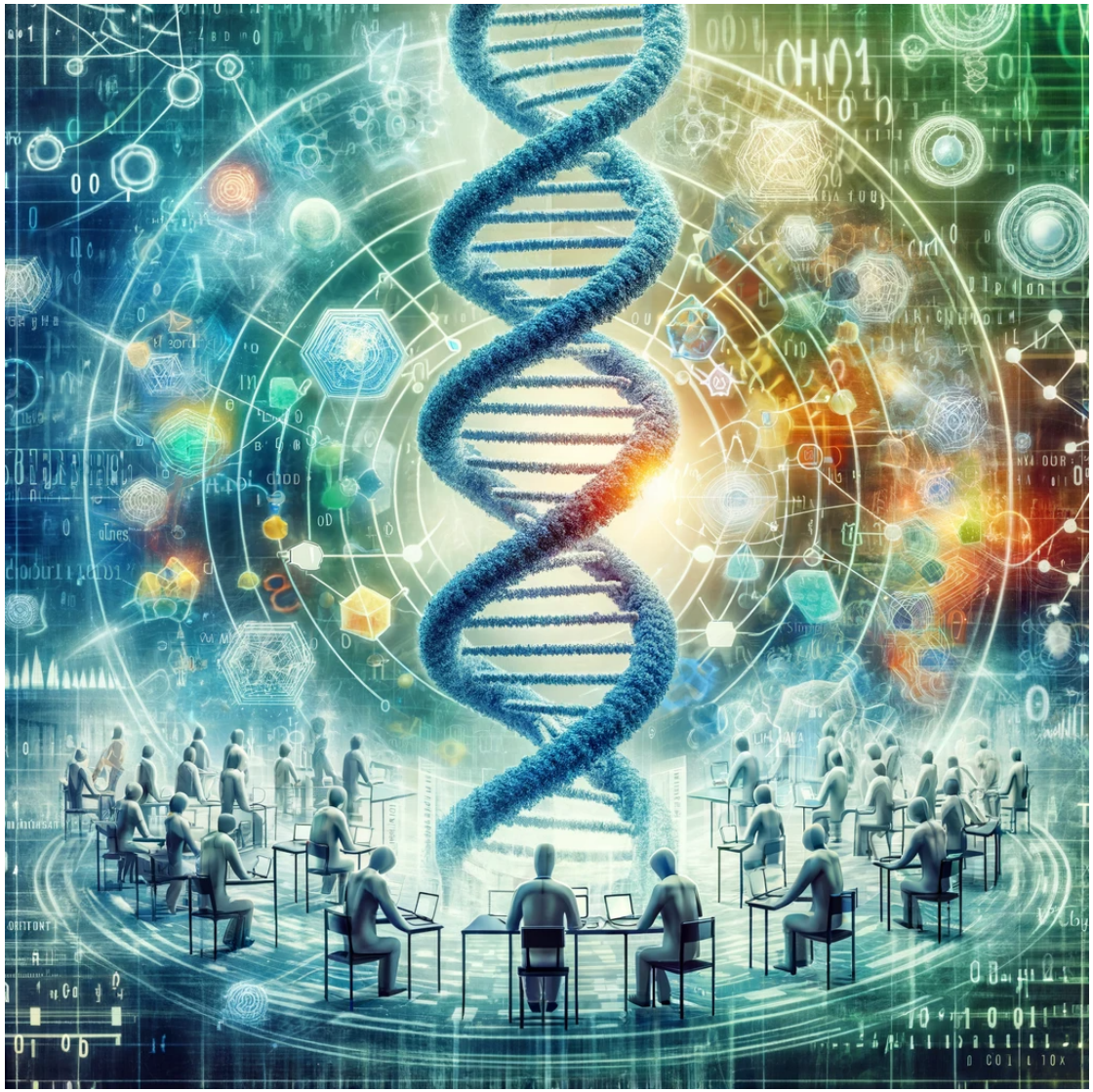
FEI/EA443

michal.vasinek@vsb.cz

5. prosinec 2023

VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY

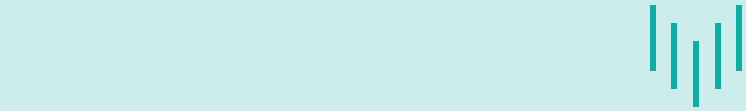
DNA Computing



Přehled přednášky

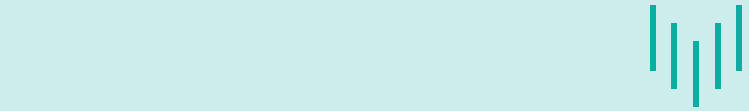
- DNA Computing.
- Adlemanovo řešení Hamiltonovy cesty.
- SAT problém
  - Liptonův algoritmus.
  - Smithův algoritmus.
  - Sakamotův algoritmus.

DNA computing



- Odvětví informatiky, které jako hardware využívá DNA, biochemii a molekulární biologii namísto křemíkových technologií.
- Počátek v roce 1994 - Adlemanův algoritmus.
- V roce 1997 Ogihara a Ray vyhodnocení boolovských obvodů.
- V roce 2002 první progamovatelný molekulární stroj složený z enzymů a DNA molekul.
- V roce 2004 první DNA počítač se schopností zpracovávat vstup a výstup - teoreticky schopen daignostikovat rakovinné projevy buněk.
- V dalších letech experimenty s ukládáním dat ve formě DNA.

Základní vlastnosti



- Jeden litr DNA teoreticky obsahuje až 2^{70} bitů.
- Výpočet jedné úlohy velmi pomalý.
- Potenciál pro masivně masivní paralelismus - obrovské množství molekul vzájemně interagují.
- Nejobtížnější je odečtení výsledků.
- Principiálně DNA počítač nemá schopnost řešit problémy jinak než standardní počítače. Třídy problémů jsou stejné, jako v případě Von Neumannových počítačů.

Problém Hamiltonovy Cesty

Hamiltonova cesta

Mějme graf G. Hamiltonova cesta je taková cesta, která navštíví každý vrchol grafu G právě jednou.

Probém Hamiltonovy cesty

Mějme graf G. Existuje v grafu G Hamiltonova cesta?

- Problém je NP-úplný.

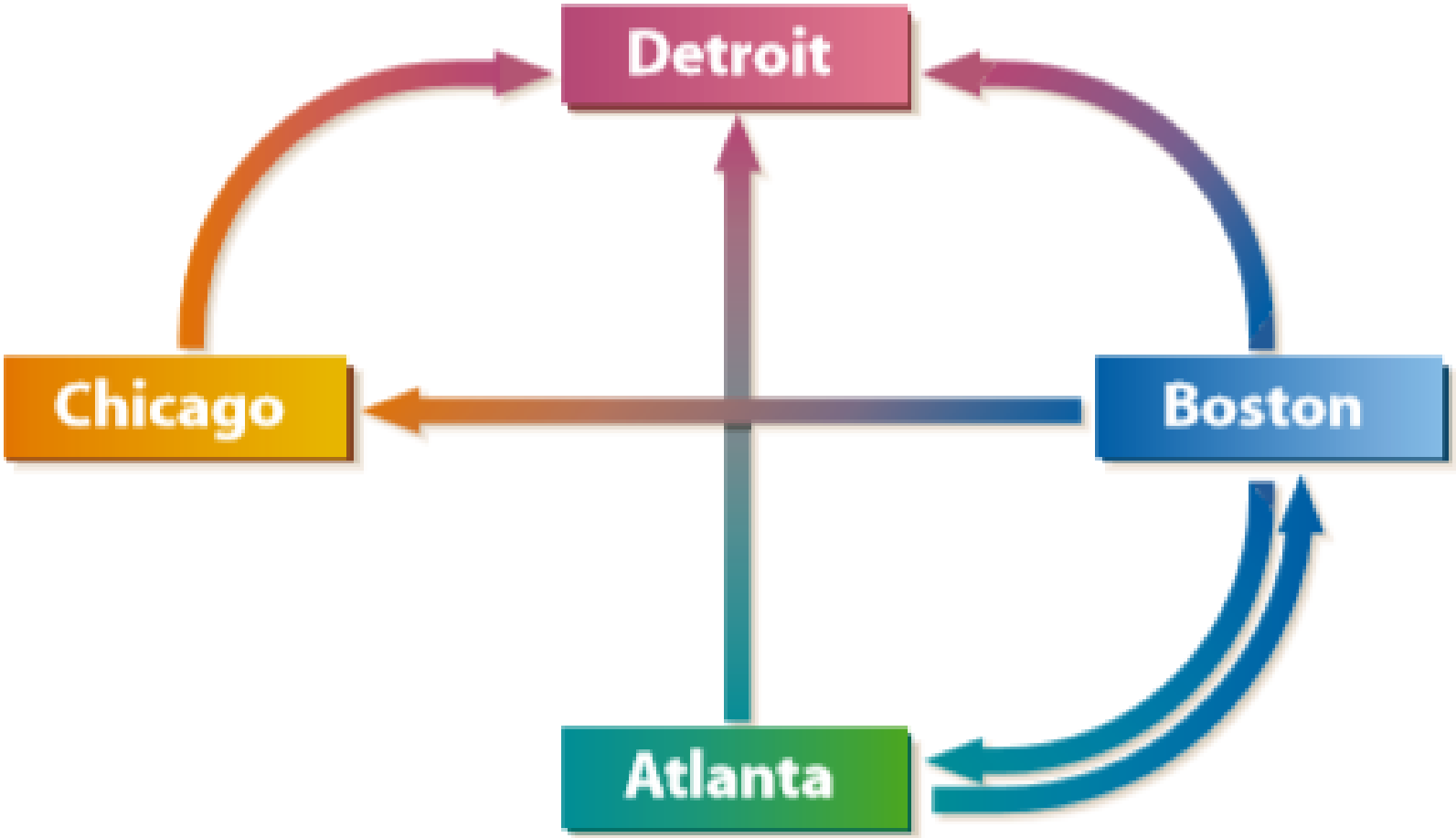
Adlemanův DNA stroj

- V roce 1994 Leonard Adleman[1] navrhl postup, kterým lze řešit problém Hamiltonovy cesty.[2]
- Použity 4 základní operace:
  - Watson-Crickovo párování komplementárních bazí.
  - Polymerázy - kopírování DNA.
  - Ligázy - spojování DNA.
  - Gelová eletroforéze

[1] Spoluautor šifrovacího algoritmu RSA.

[2] `http://152.2.128.56/~montek/teaching/Comp790-Fall11/Home/Home_files/Adleman-ScAm94.pdf`

Příklad



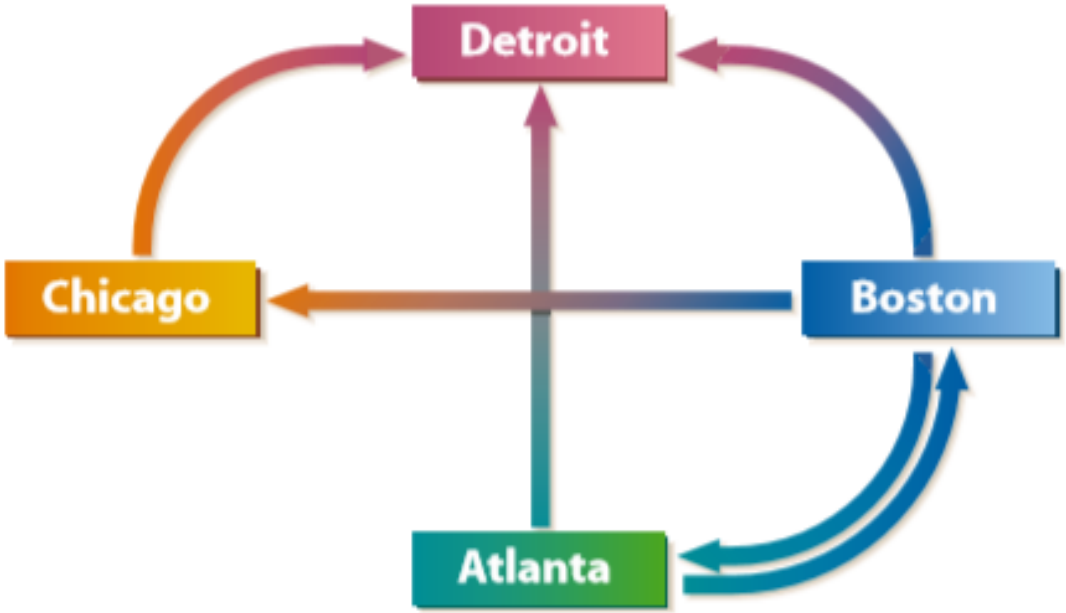
Příklad

CITY	DNA NAME	COMPLEMENT
ATLANTA	ACTTGCAG	TGAACGTC
BOSTON	TCGGACTG	AGCCTGAC
CHICAGO	GGCTATGT	CCGATACA
DETROIT	CCGAGCAA	GGCTCGTT

FLIGHT	DNA FLIGHT NUMBER
ATLANTA - BOSTON	GCAGTCGG
ATLANTA - DETROIT	GCAGCCGA
BOSTON - CHICAGO	ACTGGGCT
BOSTON - DETROIT	ACTGCCGA
BOSTON - ATLANTA	ACTGACTT
CHICAGO - DETROIT	ATGTCCGA

Postup

- Problém formulován následovně: Mějme dvě letiště: počáteční a koncové, zjistěme, zda lze cestovat mezi počátečním a koncovým letištěm, takovým způsobem, že procestujeme přes všechna ostatní letiště.
- Graf je orientovaný.
- Nechť počáteční letiště je Atlanta a koncové Detroit.



Obecný algoritmus

Mějme graf s n vrcholy:

1. Vygenerujte množinu náhodných cest skrze graf.
2. Pro každou cestu:
   - Ověřte, že cesta začíná a končí v počátečním a koncovém vrcholu. Pokud ne, odeberte cestu z množiny cest.
   - Ověřte, zda cesta prochází přesně n vrcholy. Pokud ne, odeberte cestu z množiny cest.
   - Pro každý vrchol ověřte, zda cesta vrcholem prochází. Pokud ne, odeberte cestu z množiny cest.
3. Pokud je výsledná množina neprázdná, pak existuje Hamiltonova cesta, v opačném případě neexistuje.

Převod do DNA

- Každému městu přiřadit náhodnou DNA sekvenci.
- Uvažujme, že sekvence se skládá ze dvou částí.
- Kód pro let mezi městy se skládá z druhé části startovacího města a první části koncového města.

CITY	DNA NAME	COMPLEMENT
ATLANTA	ACTTGCAG	TGAACGTC
BOSTON	TCGGACTG	AGCCTGAC
CHICAGO	GGCTATGT	CCGATACA
DETROIT	CCGAGCAA	GGCTCGTT

FLIGHT	DNA FLIGHT NUMBER
ATLANTA - BOSTON	GCAGTCGG
ATLANTA - DETROIT	GCAGCCGA
BOSTON - CHICAGO	ACTGGGCT
BOSTON - DETROIT	ACTGCCGA
BOSTON - ATLANTA	ACTGACTT
CHICAGO - DETROIT	ATGTCCGA

Postup

- Syntetizovány sekvence pro komplementární názvy měst a sekvence kódů pro jednotlivá spojení.
- Přibližně 10^{14} sekvencí v testovacím mediu.
- Do média přidána ligáza.
- Reakce trvající cca 1s, ve výsledné směsi je obsaženo řešení.

Postup - spojování molekul

- Uvažujeme sekvenci pro spojení Atlanta - Boston:

```
GCAGTCGG
```

- Komplementární název pro Boston:

```
AGCCTGAC
```

- Sekvence jsou komplementární, proto dojde ke spojení:

```
GCAGTCGG
    AGCCTGAC
```

- Ligáza zároveň spojuje sekvence do delších sekvencí.

Postup - eliminace nesprávných řešení

- Obrovské množství molekul většina z nich s nesprávným řešením.
- Použití dvou primerových sekvencí pro PCR pro popis počátečního a koncového města.
- Několik cyklů PCR => dojde k exponenciálnímu nárůstu sekvencí s korektním počátkem a koncem.
- Pomocí gelové elektroforézy výběr molekul o správné délce, ostatní zahozeny.
- Na závěr použita speciální sonda s komplementárními sekvencemi k přestupním městům. Sekvence s daným městem se komplementárně připojí k sondě. Ostatní sekvence jsou odstraněny. Proces se opakuje pro další města. Na závěr se ve směsi buď vysyktuje DNA sekvence s řešením a nebo se nevyskytuje DNA vůbec.

SAT Problém

Definice

SAT (Splnitelnost booleovských formulí) je rozhodovací problém, kde se snažíme určit, zda existuje přiřazení pravdivostních hodnot proměnným v booleovské formuli tak, aby formule byla pravdivá (splnitelná).

- Literálem je proměnná nebo její negace: x_i nebo $\neg x_i$.
- Klauzule C_i je disjunkcí literálů: $(x_1 \vee x_2)$.
- Boolovská formule je konjunkcí klauzulí: $C_1 \wedge C_2 \wedge \cdots \wedge C_n$.

Příklad SAT Formule

Formule

Uvažujme formuli s klauzulemi:

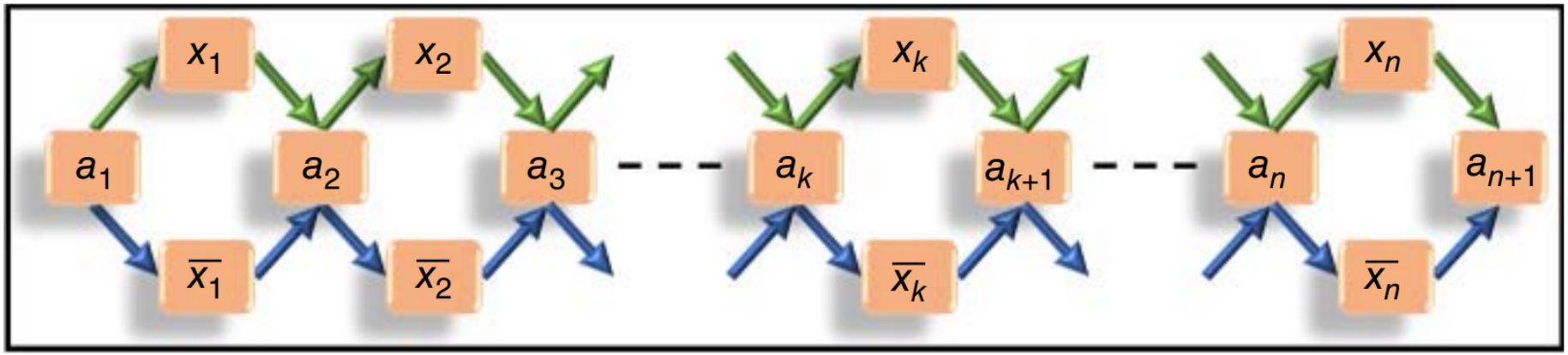
1. $x_1 \vee \neg x_2$
2. $x_2 \vee x_3$
3. $\neg x_1 \vee \neg x_3$

- Formule je splnitelná např. pro $x_1 = 1$, $x_2 = 1$ a $x_3 = 0$,

Liptonův DNA Model

Přehled

Liptonův DNA model používá principy molekulární biologie k řešení SAT problému tím, že reprezentuje proměnné a klauzule pomocí DNA sekvencí a využívá biochemické procesy k nalezení splnitelných kombinací.



Reprezentace DNA

DNA Sekvence

Proměnné x_1, x_2, a x_3 a jejich negace jsou reprezentovány unikátními DNA sekvencemi. Například:

- x_1: ‘ATCG‘
- $\neg x_1$: ‘GCAA‘
- x_2: ‘TGCAG‘
- $\neg x_2$: ‘CCATG‘
- x_3: ‘GTACT‘
- $\neg x_3$: ‘AAGCT‘

Linkovací Sekvence

- $(x_1)a_2(x_2)$: ‘GCAC‘
- $(x_2)a_3(x_3)$: ‘GTCCA‘

Příprava dat/směsi

1. Vložíme všechny jednovláknové DNA sekvence reprezentující literály $x_1, \overline{x_1}, \ldots x_n, \overline{x_n}$ společně s linkujícími sekvencemi pro a_1 až a_n do testovací nádoby.
2. Linkující sekvence a_i jsou navrženy tak, aby se komplementárně párovaly k x_{i-1} a x_i. Tím dojde k vytvoření sekvence: $a_i x_i a_{i+1} x_{i+1} \ldots$.
3. Použití biochemických technik (např. elektroforéza) k oddělení a identifikaci DNA molekul, které reprezentují splnitelné kombinace, tj. mají požadovanou délku.

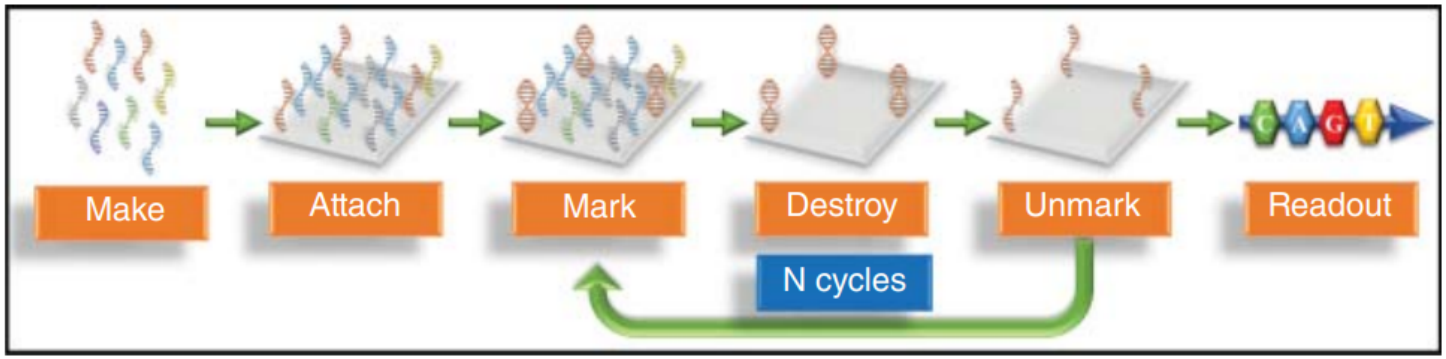
Extrakce řešení

1. Nyní aplikujeme extrakční operátor $E(t_j, i, v)$, kde i je index literálu a $v \in \{0, 1\}$. Operátorem E vybíráme ohodnocení literálu, tak aby i-tá klauzule byla splnitelná. Extrakcí rozdělíme do dvou nádob t_{j+1} a t_{j+2} sekvence na ty, které mají hodnotu i-té proměnné v a na sekvence v t_{j+2} aplikujeme další operátor E odpovídající splnitelnosti podle druhé proměnné.
2. Extrahované sekvence spojíme do nádoby t_{j+3} a opakujeme pro další klauzuli.
3. Pokud po extrakci všech klauzulí najdeme ve výsledné nádobě nějaké sekvence, pak se jedná o řešení.

Smithův Model pro DNA Computing

Úvod

Smithův model DNA computing pro řešení SAT problému využívá oligomery uchycené na povrchu mikročipu. Tento přístup umožňuje efektivní manipulaci a analýzu DNA sekvencí.



Obrázek: Vizualizace postupu výpočtu ve Smithově algoritmu. Zdroj: DNA and RNA based Computing Systems.

Reprezentace SAT Problému

Kódování Proměnných a Klauzulí

V tomto modelu jsou proměnné a klauzule SAT problému reprezentovány jako specifické sekvence oligomerů DNA. Tyto sekvence jsou uchyceny na povrchu mikročipu a jsou použity pro reprezentaci logických vztahů v SAT formuli.

- Vytvoříme všechny kombinace proměnných a jejich negací ve zkoumané formuli.

$$T_{15}GCTTxxxyyyyzzzTTCC$$

Zde x, y a z zastupují kódové sekvence pro jednotlivé proměnné či jejich negace.

Mark, Destroy, Unmark

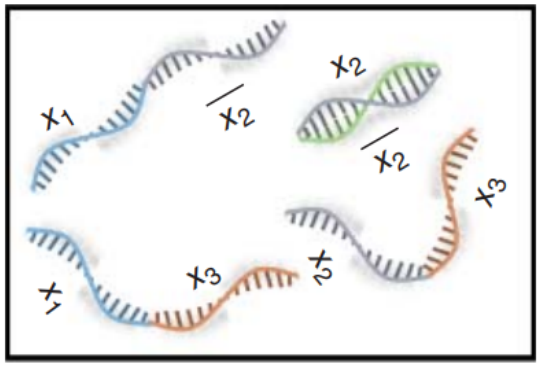
Základní Princip

1. **Mark:** Na sekvence(oligomery) reprezentující možná řešení se váží komplementární sekvence reprezentující splnitelnost klauzule(jsou komplementární k proměnným x, y, ... a utvoří tzv. duplex.
2. **Destroy:** Sekvence, které neodpovídají žádné klauzuli, jsou selektivně zničeny. Používá se např. z E.Coli exonukleáza I, ta přestřihne všechna řešení která jsou jednovláknová.
3. **Unmark:** Zbylé sekvence, které reprezentují platná řešení, jsou odznačeny a zachovány pro opakování s další klauzulí.

Sakamotův algoritmus

Základní princip

Sakamotův model využívá komplementarity sekvencí. Uvažujme, že literál x je reprezentován sekvencí $ACGT$, pak $\neg x$ bude reprezentován komplementární sekvencí $TGCA$. Pokud se v řešení objeví x i $\neg x$, dojde ke komplementárnímu navázání za vzniku smyčky, pak jde o spor a nejedná se o řešení SAT problému.



Obrázek: Tvorba smyček u komplementárních sekvencí. Zdroj: DNA and RNA based Computing Systems.

Algoritmus

1. **Příprava:** Pomocí ligázy spojíme různé kombinace literálů: $x, y, z, \ldots$. Délka sekvencí je rovna počtu klauzulí krát počet nukleotidů použitých pro každý literál.
2. **Vytvoření smyček:** Použitím restrikčních enzymů přestřihneme všechny smyčky.
3. **Eliminace:** Gelovou elektroporézí eliminujeme řešení s nesprávnou délkou.
4. **Přečtení výsledku:** Sekvenováním jednotlivých sekvencí obdržíme řešení problému.

Děkuji za pozornost

Michal Vašinek

VŠB – Technická univerzita Ostrava

FEI/EA443

michal.vasinek@vsb.cz

5. prosinec 2023

VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY