Gliederung Proseminar Datenkompression

Clemens Damke - Thema 5

1. Problemstellung

1.1. Was ist ein Suffix Array?

Lediglich eine kurze Beschreibung des Problems.

1.2. Wozu benötigt man Suffix Arrays?

Ebenfalls kurz gehalten, aber anschaulich, da SAs sonst vermutlich recht abstrakt wirken. Anwendungen z. B.:

- Schnelle exakte Suchanfragen (Häufigkeit, Position) mittels binärer Suche auf SA.
- Kompressionsalgorithmen: z. B. LZ77
- ... (evtl. weitere Einsatzgebiete)

2. Bisherige Lösungen

Kurzer Überblick über andere Lösungen ist vermutlich hilfreich; zum Einen, da dies hilft zu verstehen, wo allgemein Probleme bei der Konstuktion von SAs liegen und zum Anderen, da dies erklärt, warum ein weiterer SACA überhaupt relevant ist.

2.1. Naiver $O(n^2 \log n)$ Ansatz

1. Alle Suffixe erzeugen.
2. Suffixe (z. B. mit Quicksort) in $O(n^2 \log n)$ sortieren.

2.2. $O(n)$ Algorithmen

1. **Rekursiv (z. B. Skew):** $O(n)$

2. **Induzierte Sortierung:** *O(n)* aber worst case oft $O(n^2 \log n)$

3. GSACA Algorithmus

Nicht rekursive SA Konstruktion in *O(n)*. Vorgestellt anhand eines Beispiels.

Das Beispiel `graindraining` aus dem Paper kann hier definitiv verwendet werden, da es alle interessanten Grenzfälle produziert. Vielleicht gibt es aber auch ein kürzeres Beispiel, welches im Rahmen eines Vortrags leichter im Kopf nachzuvollziehen ist. (Muss noch gefunden werden...)

3.1. Erläuterung der Syntax

Der GSACA Algorithmus basiert auf verschiedenen Klassen von Substrings. Diese müssen zu Anfang kurz eingeführt werden.

3.2. Erläuterung des Grundprinzips

Der Algorithmus arbeitet in zwei Phasen. Hier wird kurz erklärt, was diese Phasen prinzipiell tun und wie sie zusammenspielen. Dies wird anhand des gewählten Beispiels veranschaulicht.

3.3. Erklärung der ersten Phase

Sofern das Beispiel geschickt gewählt ist, sollte es hier möglich sein zu Beginn kurz das Grundprinzip der ersten Phase zu erklären und dies dann einfach iterativ anzuwenden, sodass der Reihe nach alle interessanten Spezialfälle auftreten, die dann näher erläutert werden können.

3.4. Erklärung der zweiten Phase

Hier sollte es möglich sein, analog zur ersten Phase vorzugehen, da die zweite Phase ebenfalls iterativ abläuft.

3.5. Kurzer Rückblick auf die beiden Phasen

Nach dem Erklären der beiden Phasen, ist es wahrscheinlich hilfreich noch einmal kurz auf beide zurückzublicken, da sonst u. U. nur die Details der zweiten Phase in Erinnerung bleiben.

4. Performance

Performance Ergebnisse aus Paper vorstellen. Erklären, warum das Ergebnis aktuell schlechter ist, als das anderer Verfahren. Von hier aus kann gut übergeleitet werden in den Ausblick.

5. Ausblick

Erklären, wie die Implementation sich u. U. verbessern ließe (Ideen hierzu im Paper).