Ludwig-Maximilians-Universität
Centrum für Informations-
und Sprachverarbeitung

HS: Algorithmen und Strukturen zur Stringverarbeitung
(SoSe 2008)

Dozent: Dr. H Leiß
Referent: Peter Schrammel (peter.schrammel_AT_gmx.de) am 03.06.2008

Subword graphs

Subwortgraphen (SG) stellen eine kompakte Schreibweise für Suffixtries dar. Ein online Algorithmus erlaubt dabei ihren direkten Aufbau ohne zuerst einen Suffixtrie konstruieren zu müssen. Die kompakte Darstellung in Form eines gerichteten azyklischen Graphen (DAGs) wird gerichtetes azyklisches Wortgraph (DAWG) genannt.

Wie lassen sich Knoten, die identifiziert werden können erkennen?

Hilfsfunktion:

endpos(x): Die Menge aller Positionen im Text, an denen x endet.

Beispiel:

dababd	
Text	**endpos(x)**
d	1,6
da	2
dab	3
daba	4
dabab	5
dababd	6
a	2,4
b	3,5
ab	3,5
bd	6
aba	4
bab	5
abd	6
abab	5

dababd	
Text	**endpos(x)**
babd	6
ababd	6

Pfade, die die selbe *endpos* Menge haben führen zum selben Knoten.

Die *endpos* Mengen variieren jedoch, wenn man den DAWG online erstellen will.

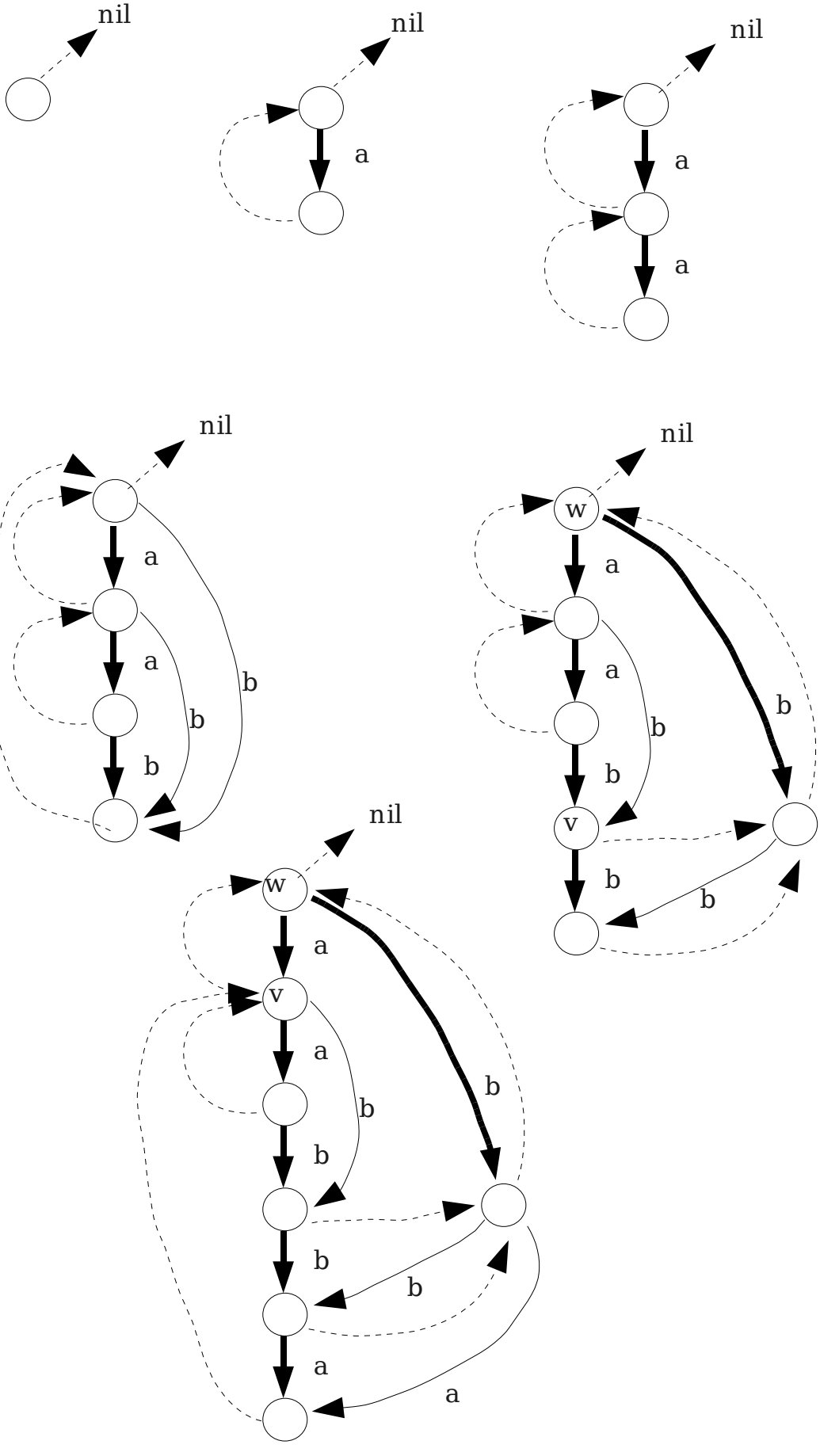
Onlinekonstruktion von DAWGs

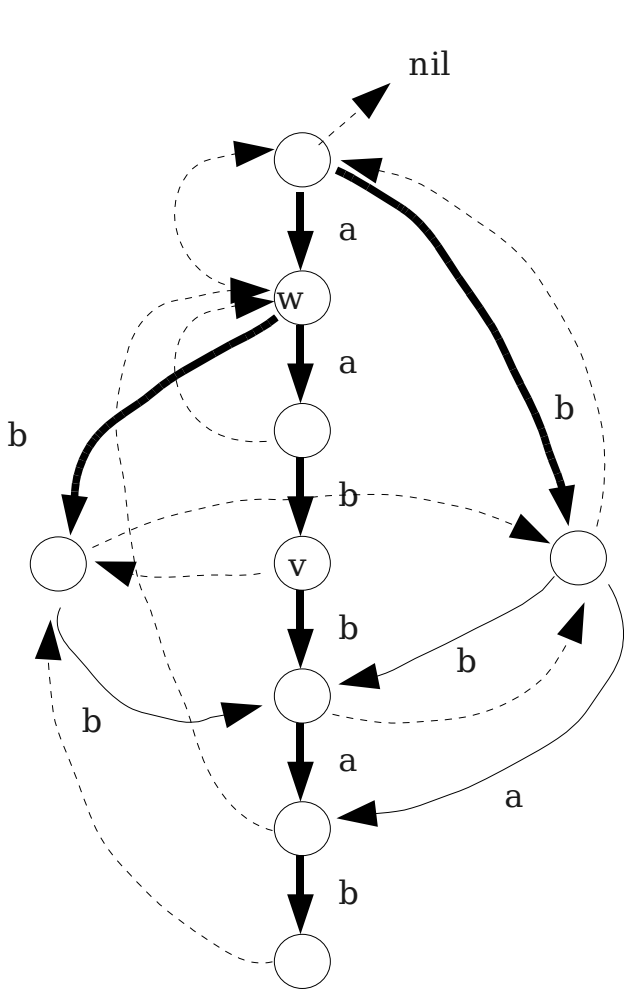
Die Konstruktion eines DAWGs in linearer Zeit ist möglich und kann als Simulation der Konstruktion eines on-line Tries mit gleichzeitiger Identifizierung von isomorphen Subbäumen aufgefasst werden.
Der Algorithmus verwendet als Hilfsmittel typisierte Kanten und die von der Trie Online-Konstruktion bekannten Suffixzeiger.
Es gibt feste (solid) und nicht feste (non-solid) Kanten, wobei die festen Kanten sich nicht mehr verändern, die nicht festen sich jedoch noch verändern können.
Der Algorithmus geht zunächst davon aus, dass sich alle Blätter des Trie identifizieren lassen. Erst wenn sich diese Annahme als falsch erweist, wird der Knoten geklont.

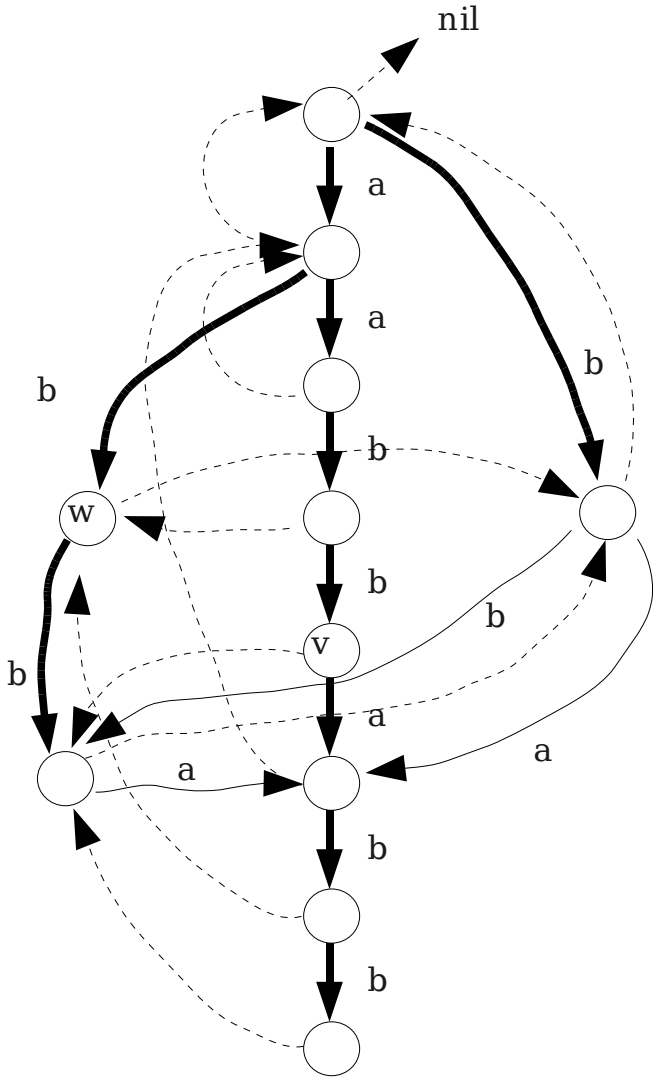
Der Algorithmus

erzeuge einen Graphen G=DAWG(ε) mit einem Knoten *root*;
sink := *root*; *suf*[*root*] := nil;
for *i*:= 1 to *n* **do begin**
    a:= *text*[*i*]; erzeuge einen neuen Knoten *newsink*;
    ziehe eine feste *a*-Kante (*sink*,*newsink*); *w*:= *suf*[*sink*];
    while (*w* ≠ nil) **and** (*son*(*w*,*a*) = nil) **do begin**
        ziehe ein nicht feste *a*-Kante (*w*,*newsink*); *w*:= *suf*[*w*];
    end;
    if *w* = nil **then** *suf*[*newsink*] := *root*
    else
        v:= *son*(*w*,*a*);
        if (*w*,*v*) eine feste Kante **then** *suf*[*newsink*]:= *v*
        else
            erzeuge *newnode* mit den selben,jedoch nicht festen,
            Ausgangskannten wie *v*;
            wandle (*w*,*v*) in die feste Kante (*w*,*newnode*);
            suf[*newsink*]:=*newnode*; *suf*[*newnode*]=*suf*[*w*];
            suf[*v*]:=*newnode*; *w*:=*suf*[v];
            while *w* ≠ nil **and** (*w*,*v*) ist eine nicht feste *a*-Kante **do begin**
                diese Kante nach *newnode* verbiegen; *w*:= *suf*[*w*];
            end;
        end;
    end;
    sink:= *newsink*;
end

Beispiel: aabbabb




nil
a
w
a
b
b
b
v
b
b
b
a
a
b


nil
a
a
b
b
b
w
b
b
v
b
b
a
a
a
b
b