1.8 Bewerten von Webseiten

Kennt man die Adressen von Webseiten, dann können wir diese über das Netz direkt erreichen. Was geschieht aber, wenn wir Seiten mit bestimmten Inhalten erst einmal suchen? Für diesen Zweck nutzen wir natürlich die *Suchmaschinen*, die uns zu bestimmten Stichworten Netzadressen vorschlagen, die sich in ihren Inhaltsverzeichnissen befinden. Diese Verzeichnisse selbst werden weitgehend erstellt, indem *Webcrawler* von Link zu Link automatisch möglichst viele erreichbare Webseiten besuchen und die dort aufgefundenen Stichworte in das Inhaltsverzeichnis der Suchmaschine aufnehmen. Auf diese Weise entstehen meist extrem umfangreiche Adressensammlungen zum gleichen Stichwort.

Da die Benutzer von Suchmaschinen mit großen ungeordneten Adressensammlungen nicht viel anfangen können, müssen die zu einem Stichwort gefundenen Seiten nach ihrer Wichtigkeit geordnet werden. Die Benutzer nutzen dann meist relativ wenige Adressen, die als Erstes erscheinen. Die weit „unten“ stehenden Links werden kaum beachtet. So müssen also zumindest die kommerziell arbeitenden Anbieter im Netz daran interessiert sein, möglichst weit „oben“ in den von Suchmaschinen erstellen Listen aufzutauchen, um von potentiellen Kunden überhaupt gefunden zu werden, und sie nutzen alle Tricks, um das zu erreichen.

Es ist bisher noch nichts über die Bedeutung der Informationen einer Seite für das Stichwort gesagt. Sein Auftauchen alleine bedeutet noch nicht viel. Enthält eine Seite z. B. den Text *„Hier steht nichts über Göttingen!“*, dann wird sie trotzdem in die Inhaltsverzeichnisse aufgenommen, die das Stichwort *„Göttingen“* betreffen. Wir brauchen also andere Bewertungskriterien.

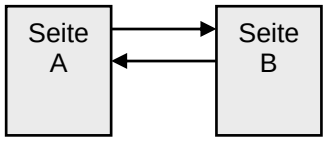
Im einfachsten Fall geben die Autoren einer Webseite in den Meta-Tags Stichworte zum Inhalt der Seite an:

```
<meta name = "keywords" content = "Delphi,Schule,Informatik">
```

Diese Möglichkeit wird aber oft missbraucht, indem oft verwendete Stichworte – die den Seiteninhalt gar nicht betreffen – angegeben werden, um potentielle „Opfer“ auf die Seite zu lenken. Nicht sehr hilfreich ist auch die Idee, zu zählen, wie oft das Stichwort auf der Seite vorkommt. Für diesen Fall enthalten Webseiten manchmal bestimmte Stichworte „unsichtbar“, z. B. indem das Stichwort sehr oft in weißer Schrift auf weißem Hintergrund geschrieben wird. Man kann Webseiten natürlich auch durch Menschen bewerten und in die Suchverzeichnisse eintragen lassen. Das ist aber eine sehr teure und relativ langsame Art, Verzeichnisse zu erstellen, und außerdem ist so eine Bewertung natürlich subjektiv. Auch ist es oft schwierig, Seiten mit speziellen Inhalten – z. B. aus der Archäologie – einzuschätzen. Im schlimmsten Fall ergibt sich der „Wert“ einer Seite nicht aus ihrem Inhalt, sondern aus dem Betrag, der für die Bewertung bezahlt wurde.

Eine andere Art, einerseits das Fachwissen von Webautoren für die Bewertung von Webseiten zu nutzen und andererseits den Bewertungsvorgang zu automatisieren, wird im so genannten *PageRank*-Verfahren verwirklicht, das die Suchmaschine *Google* benutzt. Im Unterschied zu den Meta-Tags, die die eigene Webseite bewerten, werden die Links einer Webseite auf andere Webseiten als eine auf Fachwissen basierende Abstimmung aufgefasst, mit der Autoren kundtun, dass andere Webseiten interessante Inhalte enthalten. Verweist also jemand auf eine Seite mit physikalischen Inhalten, dann wird der Autor mit hoher Wahrscheinlichkeit etwas von deren Inhalten verstehen. Da außerdem meist nicht bekannt sein kann, welche anderen Webseiten auf die eigene verweisen, können Webautoren dieses Verfahren nur schwer manipulieren.

Das PageRank-Verfahren bewertet nicht alle Links gleich. Es ermittelt für jede ihm bekannte Webseite einen Rang (eben den PageRank), der das „Gewicht“ dieser Seite beschreibt. Dieser Rang wird bei der „Abstimmung“ durch Links auf alle Verweise aufgeteilt, die von der Seite weg führen. Enthält eine Webseite also nur einen ausgehenden Link, dann erhält dieser das ganze Gewicht der Seite, enthält sie zwei, dann wird das Gewicht halbiert usw. (Enthält die Seite gar keinen ausgehenden Link, dann nimmt sie nicht an der Abstimmung teil. Bei der PageRank-Berechnung liefert sie den Wert 0.) Der Rang einer Webseite steigt also, wenn möglichst viele Seiten mit hohem Rang auf sie verweisen und wenn diese Seiten jeweils möglichst wenige Links enthalten.



Wählen wir als erstes Beispiel zwei Seiten, die wechselseitig auf einander verweisen. Zur Berechnung des PageRanks von Seite *A* – *PR(A)* – benötigen wir den PageRank *PR(B)* von Seite *B*, weil von *B* ein Link zur Seite *A* führt. In die Berechnung von *PR(B)* geht aber wiederum *PR(A)* ein. Wir benötigen also einen alten Wert von *PR(A)*, um den neuen zu bestimmen. Da sich diese Argumentation fortsetzen lässt, muss eine Methode entwickelt werden, um den Einfluss der alten Werte auf die Berechnung des neuen Rangs abklingen zu lassen, damit sich im Laufe der Berechnungen ein stabiles Ergebnis ergibt. Man erreicht dieses, indem man den Beitrag der eingehenden Links mit einem Faktor *d* multipliziert, der kleiner als 1 ist. Da dieser in jede Berechnung eingeht, werden die „sehr alten“ PageRanks mit d^n multipliziert, also einer Zahl die sich immer mehr der Null nähert. Meist wählt man den Wert *0,85* für *d*. Wenn wir die Zeitpunkte, zu denen der PageRank in der Vergangenheit berechnet wurde, mit t_1, t_2, t_3, … bezeichnen, wobei ein größerer Index einen früheren Zeitpunkt bedeuten soll, dann erhalten wir für unsere beiden Webseiten:

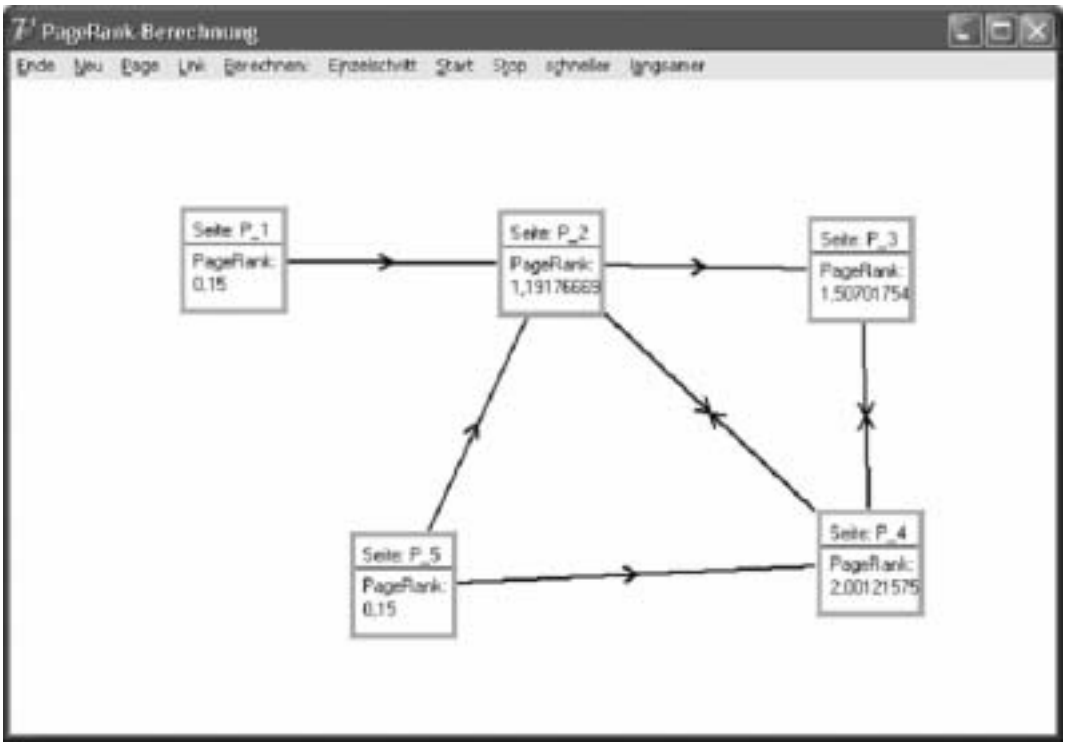
$$PR_{t_1}(A) = ... + 0{,}85 \cdot PR_{t_2}(B) = ... + 0{,}85 \cdot (... + 0{,}85 \cdot PR_{t_3}(A)) = ... + 0{,}85 \cdot ... + 0{,}85^2 \cdot PR_{t_3}(A) = ...$$

Hätte die Seite *B* mehr als einen ausgehenden Link, dann müssten wir in der Rechnung ihren Rang noch durch die Anzahl der Links – *C(B)* – teilen. Entsprechend müssen wir bei den anderen Seiten vorgehen, die Links auf die Seite *A* besitzen. Bezeichnen wir diese

n Webseiten mit T_1, T_2, …, T_n und ersetzen die drei Pünktchen in der oben angegebenen Beziehung durch *(1-d)*, dann erhalten wir die Originalformel, die anfangs für die PageRank-Berechnung von Google angegeben wurde:

$$PR(A) = (1-d) + d \cdot (\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + ... + \frac{PR(T_n)}{C(T_n)}),\ d = 0{,}85$$

Der Rang einer Webseite beträgt danach mindestens *0,15*. Aber welchen Einfluss haben die anderen Terme? Wir wollen die Frage durch ein Simulationsprogramm klären, in dem symbolische Webseiten erzeugt und verlinkt werden können. In einer so entstehenden „Website“ können dann die PageRanks berechnet werden.



Da in diesem Programm die Anzahl aller Größen vor dem Programmstart offen ist, handelt es sich um ein Musterbeispiel für die Anwendung von Listen. Wir können programmtechnischen Details der Seiten (*tPage*) und Links (*tLink*) weitgehend von unserem Simulationsprogramm für das Routing übernehmen (s. auch Anhang A3). Auch die Listentypen selbst unterscheiden sich nur wenig: wir benötigen eine Liste von Webseiten (*tPageListe*) und auf jeder von diesen jeweils Listen für eingehende und ausgehende Links (*tLinkListe*).

```
tLinkListe = class;

tPage = class(tPanel)
  constructor init(…);
  procedure MouseDown(…);
  …
  procedure berechnePageRank;
  outLinks,inLinks: tLinkListe;
  PageRank: double;
  end;
```

forward-Deklaration

ähnlich wie beim Routing

Link-Listen

der PageRank

Bei den Links handelt es sich um Objekte, die als *Quelle* und *Ziel* Webseiten haben, zwischen denen der Link verläuft. Die Koordinaten geben die Lage der augenblicklichen Linie sowie des Pfeils darauf an.

```
tLink = class
  constructor init(p1,p2: tPage);
  …
  quelle,ziel                    : tPage;
  x1,y1,x2,y2,x,y,xl,yl,xr,yr: integer;
  end;
```

„verlinkte“ Seiten

Koordinaten

Linklisten enthalten Links und die üblichen Listenoperationen. Zusätzlich sollen sie den Beitrag zum PageRank der durch sie verbundenen Seiten ermitteln können, also

$$\frac{PR(T_1)}{C(T_1)}+\frac{PR(T_2)}{C(T_2)}+\ldots+\frac{PR(T_n)}{C(T_n)}$$

```
tLinkListe = class
   inhalt: tLink;
   naechster: tLinkListe;
   …
   function RankSumme: double;
   end;
```

den Beitrag zum PageRank berechnen

Die Pageliste arbeitet ähnlich und kann die in ihr gespeicherten Seiten bitten, ihre PageRanks zu bestimmen.

```
tPageListe = class
  inhalt: tPage;
  …
  procedure berechnePageRanks;
  end;
```

alle PageRanks berechnen

Wie berechnet man nun den PageRank?

Zunächst einmal benötigen wir die Anzahl der Links in der Linkliste der ausgehenden Links.

```
function tLinkListe.anzahl: integer;
var summe: integer;
begin
if inhalt = nil then summe := 0 else summe := 1;
if naechster = nil then result := summe
else result := summe + naechster.anzahl
end;
```

Danach kann der Beitrag der Linkliste zum PageRank leicht rekursiv berechnet werden.

```
function tLinkListe.RankSumme: double;
var summe: double;
    i    : integer;
begin
summe := 0;
if inhalt <> nil then
  begin
  i := inhalt.quelle.outLinks.anzahl;
  if i > 0 then summe := inhalt.quelle.PageRank/i
  end;
if naechster = nil then result := summe
else result := summe + naechster.RankSumme
end;
```

Der Beitrag der **eingehenden** Links wird berechnet, also der Quellen!

Jede Seite kann damit leicht ihren PageRank bestimmen:

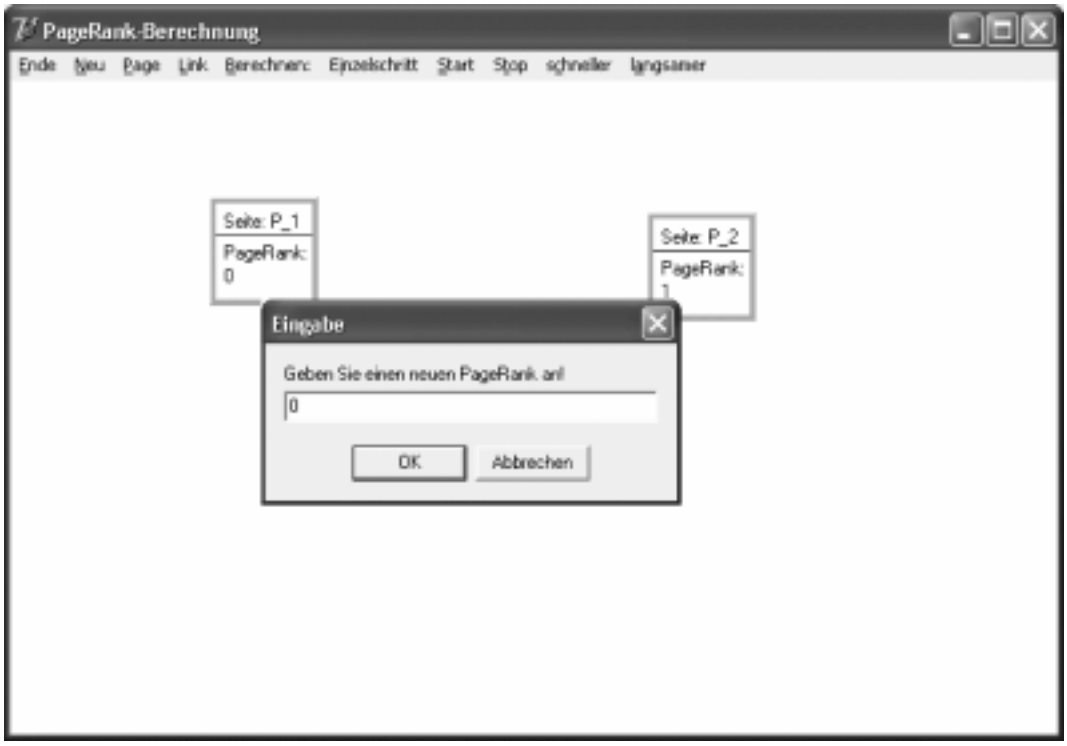
```
procedure tPage.berechnePageRank;
begin
PageRank := 0.15 + 0.85*inLinks.RankSumme;
end;
```

Und die Liste der Webseiten kann nacheinander alle Seiten bitten, genau dieses zu tun:

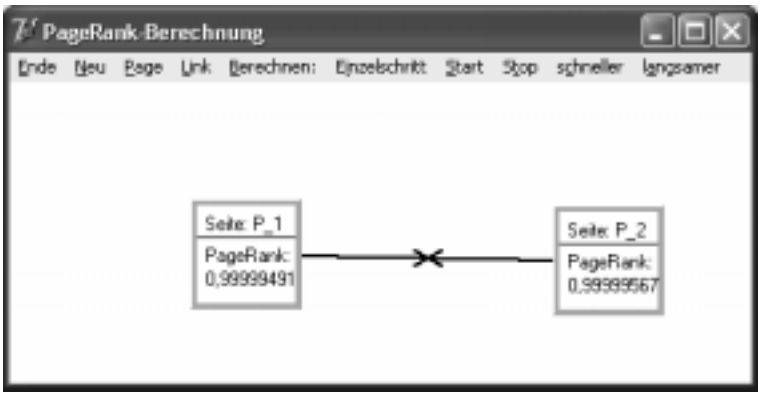
```
procedure tPageListe.berechnePageRanks;
begin
if inhalt <> nil then inhalt.berechnePageRank;
if naechster <> nil then naechster.berechnePageRanks;
end;
```

Wir wollen jetzt unser Simulationsprogramm benutzen. Zuerst einmal testen wir, wie sich unterschiedliche Anfangs-PageRanks auf die Berechnung auswirken.

Klicken wir eine Seite mit der rechten Maustaste an, dann können wir deren PageRank eingeben:

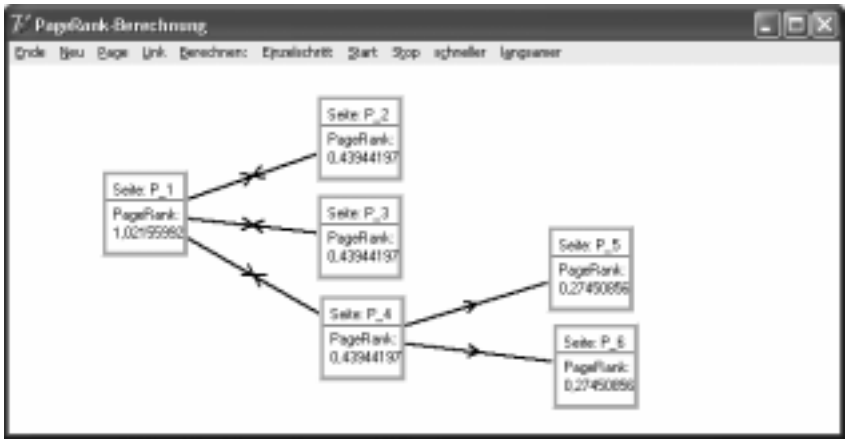


Wir erzeugen jetzt zwei Webseiten, weisen ihnen unterschiedliche Anfangs-PageRanks zu und berechnen fortlaufend die PageRanks – entweder in Einzelschritten oder automatisch.



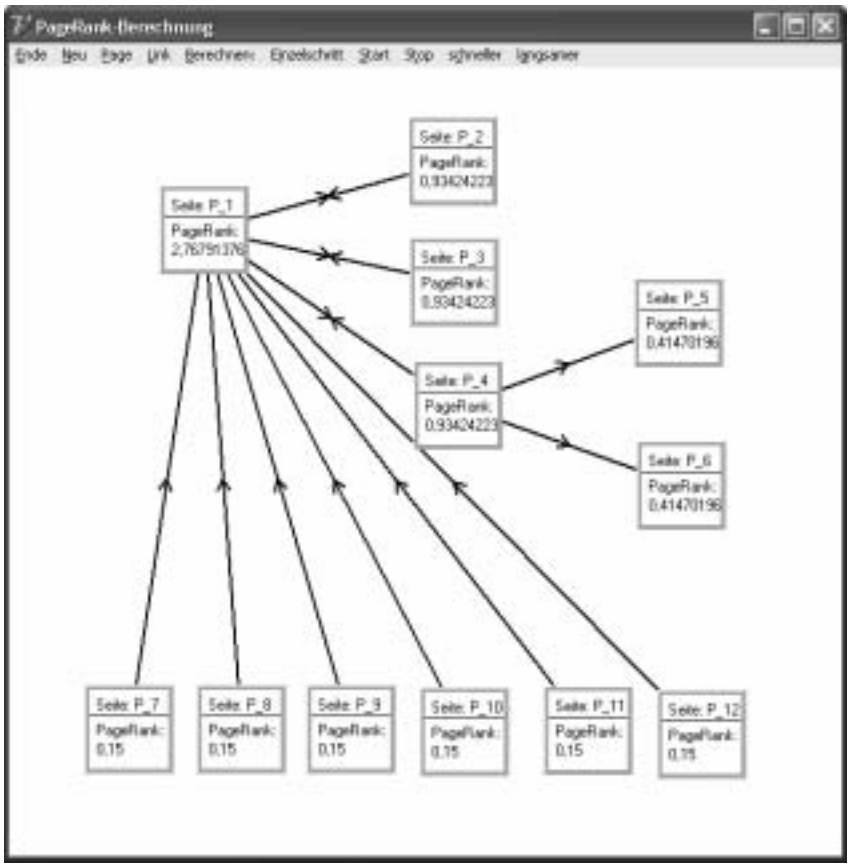
Wir erkennen schnell, dass sich im Laufe der Berechnungen ein Endwert unabhängig vom Anfangswert einstellt – hier 1. Das ist natürlich keine Überraschung, denn genau dieses haben wir mit der Einführung des „Dämpfungsfaktors“ *0,85* ja beabsichtigt.

Als nächstes Beispiel wählen wir den Aufbau einer typischen Homepage mit einer Baumstruktur, die von einer Indexseite ausgeht und in Unterverzeichnisse verzweigt.

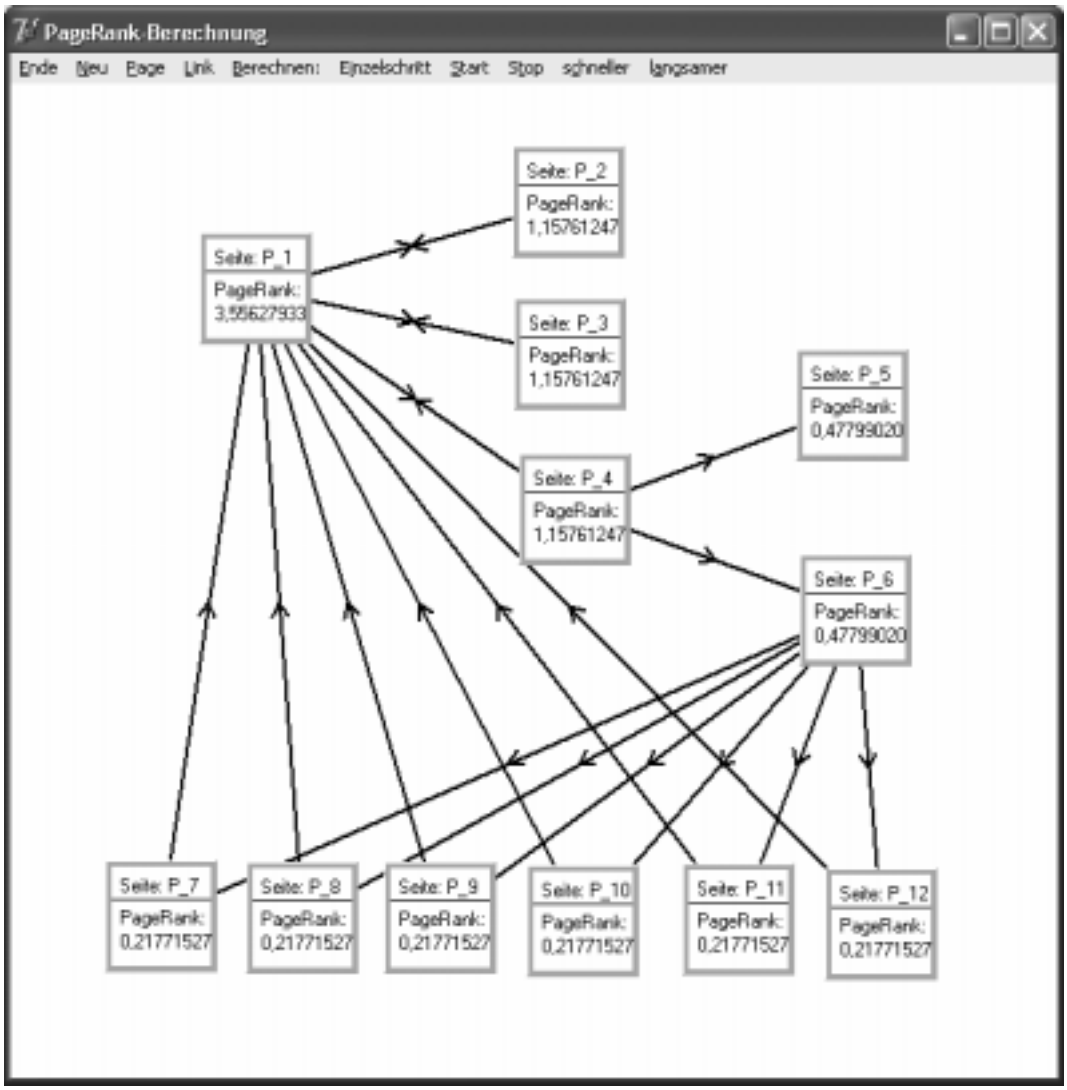


Wir nehmen jetzt an, dass es zusätzliche externe Seiten gibt, die auf unsere Homepage verweisen.

Der PageRank der Homepage steigt erheblich, Auch das Gewicht der internen Seiten nimmt zu.



Zuletzt wollen wir annehmen, dass die externen Seiten wiederum in einer Linkliste der Homepage referenziert werden.



Der Rang der Homepage steigt weiter. Man sieht, wie in einem Netz von Seiten, die wechselseitig aufeinander verweisen, um ihre „Hochachtung“ vor einander auszudrücken, die Bedeutung der Seiten wächst.