Dr. Andreas Borchert
Norbert Heidenbluth

Allgemeine Informatik II für Mathematiker/Wirtschaftsmathematiker (SS 2005)

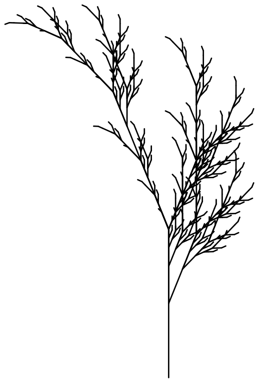
Abgabetermin: 21. April 2005

Aufgabe 1: Von der Uni in die Baumschule (9 Punkte)

In der Zeit, als Ihr Großvater noch hinter Ihrer Großmutter her war, lauschten die Menschen statt Sweety-Klingeltönen gerne deutschen Volksliedern. Um hier einen möglichen Trend nicht zu verschlafen, beschäftigen Sie sich gerade intensiv mit diesem Liedgut. Das Lied "Der Mai ist gekommen" (Lyra / Geibel) findet dabei Ihre besondere Aufmerksamkeit, denn im Text heißt es:

"Der Mai ist gekommen, die Bäume schlagen aus![...]"

Als kühle(r) Naturwissenschaftler(in) stellen Sie sich nämlich sogleich die Frage, *wie* die Bäume ausschlagen. Und von dieser Frage ist es dann nicht mehr weit zur Informatik, denn der Biologe Aristid Lindenmayer hat eine mathematische Theorie entwickelt (*Lindenmayer-Systeme*), mit der sich die Entwicklung von Pflanzen bzw. Bäumen simulieren lässt.



Konstruiert wird dieser Baum nach der im folgenden beschriebenen *rekursiven* Vorschrift. Rekursiv bedeutet hier, daß für die Konstruktion eines Astes wieder die Konstruktionsvorschrift verwendet wird, und für dessen Äste ebenfalls – solange, bis die momentane Rekursionsebene der gewünschten maximalen Rekursionstiefe entspricht.

Vorbedingungen

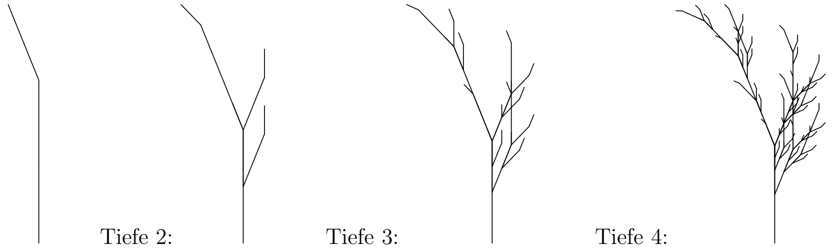
- Der Drehwinkel nach links oder rechts beträgt 22 Grad.
- Bei jedem Schritt zeichnen wir eine Linie ("*PenDown*").
- Auf jeder Rekursionsebene bedeutet das Zeichnen eines Astes, daß wir eine Rekursionsebene tiefer "steigen".
- Das Merken einer Position beinhaltet neben dieser (also neben der Position) auch das Merken der aktuellen Richtung!
- Gleiches gilt für das Zurückspringen an eine zuvor gemerkte Position!

Konstruktionsvorschrift

- Wir starten mit einem Schritt nach vorne.
- Wir drehen uns nach rechts und merken uns die aktuelle Position.
- Nur wenn die maximale Rekursionstiefe noch nicht erreicht ist, zeichnen wir Äste wie folgt:
  - Wir konstruieren an der aktuellen Position zunächst einen Ast nach dieser Konstruktionsvorschrift.
  - Wir springen zurück an die gemerkte Position, drehen uns nach links und zeichnen einen weiteren Ast nach dieser Konstruktionsvorschrift.
- Wir springen (ggf. erneut) an die zuvor gemerkte Position, drehen uns nach links und machen einen Schritt nach vorne.
- Wir merken uns diese Position, drehen uns nach links und machen einen Schritt nach vorne.
- Wenn die maximale Rekursionstiefe noch nicht erreicht ist, setzen wir hier einen neuen Ast (nach dieser Konstruktionsvorschrift) an.
- Wir springen zurück zur eben gemerkten Position.
- Wir drehen uns nach rechts.
- Falls die maximale Rekursionstiefe noch nicht erreicht ist, setzen wir hier einen neuen Ast (nach dieser Konstruktionsvorschrift) an.

Wichtig: Die Stengel (also alles bis auf das Ende eines Astes) sollen bei jedem Schritt ihre Größe verdoppeln!

Nachstehend sind die enstehenden Bäume für die maximalen Rekursionstiefen 1 bis 4 zu sehen (der Baum auf der ersten Seite hat die maximale Rekursionstiefe 5).



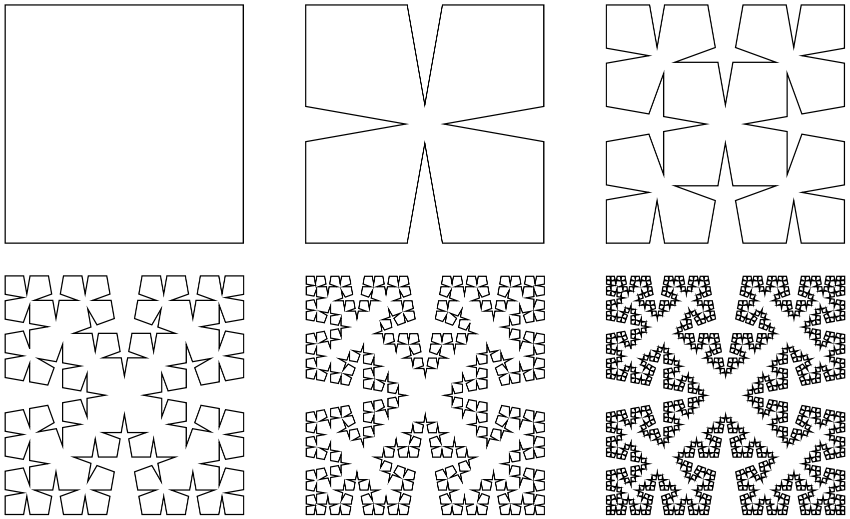
Tiefe 1: Tiefe 2: Tiefe 3: Tiefe 4:

Aufgabe: Schreiben Sie ein Oberon-Programm, das unter Verwendung der vorstehenden Konstruktionsvorschrift einen Baum zeichnet. Die zu verwendende maximale Rekursionstiefe muß dabei als Argument angegeben werden, der zu verwendende Winkel kann als Argument angegeben werden (sonst wird ein Default-Winkel verwendet). Desweiteren sollte mit der Option "-o" die Ausgabe in eine *plot(5)*-Datei möglich sein; die Voreinstellung ist die Ausgabe auf dem Bildschirm.

```
Usage: Trees [-a angle] [-o outfile] level
```

Aufgabe 2: Abstrakte Kunst (6 Punkte)

Die nachstehenden Abbildungen resultieren ebenfalls aus einer rekursiven Konstruktionsvorschrift und zeigen die Rekursionstiefen 1 bis 6:



Aufgaben:

a) Versuchen Sie, die Konstruktionsvorschrift aus den gegebenen Abbildungen zu erkennen.

b) Implementieren Sie Ihr Ergebnis aus (a) in einem Oberon-Programm. Die Argumentverarbeitung sollte dabei analog zu der aus Aufgabe 1 sein!

Generelle Hinweise

- Der generelle Rahmen für beide Programme kann aus den Beispielen (Vorlesung bzw. Übung) übernommen werden!
- **Wichtig:** Die makefiles für Ihre Programme müssen auf unseren Suns für diese Aufgabe(n) wie folgt erzeugt werden:

  ```
  mmo -c makefile +l plot +l x11
  ```

- Die Übersetzung unter Linux funktioniert wie üblich, allerdings müssen Sie mit der neuesten Version arbeiten (0.8.1).
- Eventuell lohnt auch ein Blick in die Dokumentation:
  - *Manualseite zu Plot5Streams*
  - *Manualseite zu Plotters*
  - *Manualseite zu TurtleGraphics*
  - *Manualseite zu XPlotters*
- Sie müssen bei der Einwahl via SSH das "X11-Forwarding" aktivieren, um "aus der Ferne" grafisch arbeiten zu können. Dies erreichen Sie mit der Option "-X" wie folgt:

  ```
  ssh -X IhrLogin@theseus.mathematik.uni-ulm.de
  ```

  (natürlich gilt dies für alle anderen Rechnernamen ebenso!)

Wir wünschen Ihnen ein erfolgreiches Sommersemester 2005!