Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für Theoretische Informatik
Dr. B. Gärtner, Prof. Dr. J. Hromkovič

4. Dezember 2007

Informatik für Mathematiker und Physiker Serie 11 HS 07

URL: http://www.ti.inf.ethz.ch/ew/courses/Info1_07/

Skript-Aufgabe 90 (4 Punkte)

Write programs that produce turtle graphics drawings for the following Lindenmayer systems (Σ, P, s).

a) $\Sigma = \{F, +, -\}$, $s = F + F + F + F$ and P given by

$$F \mapsto FF + F + F + F + F + F - F.$$

b) $\Sigma = \{X, Y, +, -\}$, $s = Y$, and P given by

$$X \mapsto Y + X + Y$$
$$Y \mapsto X - Y - X.$$

For the drawing, use rotation angle $\alpha = 60$ degrees and interpret *both* X and Y as "move one step forward".

Modifizierte Skript-Aufgabe 93 (4 Punkte)

Define a type `Tribool` for three-valued logic; in three-valued logic, we have the truth values *true*, *false*, and *unknown*. You should model a Tribool as a `struct` containing one member of type `unsigned int`. The values 2, 0, and 1 can be used to represent the three possible truth values.

For the type `Tribool`, implement the logical operators

```
// POST: returns x AND y
Tribool operator&& (Tribool x, Tribool y);

// POST: returns x OR y
Tribool operator|| (Tribool x, Tribool y);
```

where AND ($\wedge$) and OR ($\vee$) are defined according to the following two tables.

$\wedge$	*false*	*unknown*	*true*
false	*false*	*false*	*false*
unknown	*false*	*unknown*	*unknown*
true	*false*	*unknown*	*true*

$\vee$	*false*	*unknown*	*true*
false	*false*	*unknown*	*true*
unknown	*unknown*	*unknown*	*true*
true	*true*	*true*	*true*

You should also implement a function that generates a Tribool from an integer.

```
// PRE: val in {0,1,2}
// POST: return value is a Tribool (0: false, 1: unknown, 2: true)
Tribool tribool(unsigned int val);
```

That way you can use the functionality in `tribool_truthtables.C` (see course homepage) to print the two truth tables.

Modifizierte Skript-Aufgabe 102 (4 Punkte)

We want to have a function that *normalizes* a rational number, i.e. transforms it into the unique representation in which numerator and denominator are relatively prime, and the denominator is positive. For example,

$$\frac{21}{-14}$$

is normalized to

$$\frac{-3}{2}.$$

The following function declaration seems like a good idea:

```
// POST: r is normalized
void normalize (rational& r);
```

Implement this function according to its postcondition.

Hint: You may want to use the function `gcd` from Section 3.2, modified for parameters of type `int` (how does this modification look like?). You may extend the programs shown in class (available from the course homepage) and use these programs to test the normalization.

Skript-Aufgabe 103 (4 Punkte)

Provide a definition of the following function.

```
// POST: return value indicates whether the linear equation
//       a * x + b = 0 has a real solution x ; if true is
//       returned, the value s satisfies a * s + b = 0
bool solve (double a, double b, double& s);
```

Test your function in a program for at least the pairs (a, b) from the set

$$\{(2,1),(0,2),(0,0),(3,-4)\}.$$

Challenge

You can solve the challenge exercise 106 from the lecture notes. It will be awarded a maxiumum of 8 points, and thus replaces two of the normal exercises.

Bemerkung

Dies ist die letzte Serie, für die die 50% Regelung des Testates gilt. Das bedeutet, dass Sie mit den letzten beiden Serien Bonuspunkte machen können, falls Sie knapp dran sind.

Abgabe: Bis 11. Dezember 2007, 15.15 Uhr.