4 Assignment 4 (11 Points)

Hinweis:	Abgabe in $\{2, 3, 4\}$-er Gruppen.
Abgabe:	**20.05.2017, 23.59**
Email:	Betreff "[Compsec] Ex4"
	(bitte nur .pdf oder .txt, kein .doc, .jpeg, etc)
	Source code: bitte inkl. `signify` Signatur

Exercise 9 (Password Generation using Context Free Grammars (**3 Points**))**.** Consider the following probabilistic context-free grammar.

Production rule	Probability
$S \to D_1 L_3 S_2 D_1$	0.75
$S \to L_3 D_1 S_1$	0.25
$D_1 \to 4$	0.60
$D_1 \to 5$	0.20
$D_1 \to 6$	0.20
$S_1 \to !$	0.65
$S_1 \to \%$	0.30
$S_1 \to \#$	0.05
$S_2 \to \$\$$	0.70
$S_2 \to **$	0.30

and the following priority queue:

Base Struct	Pre-Terminal	Probability	Pivot Value
$D_1 L_3 S_2 D_1$	$4L_3\$\$4$	0.188	0
$L_3 D_1 S_1$	$L_3 4!$	0.097	0

Calculate the next five pre-terminal structures that the enumerator (as discussed in class) does output and the resulting priority queue.

Solution: Output:

1. $4L_3\$\$4$
2. $L_3 4!$
3. $4L_3 * *4$
4. $5L_3\$\$4$
5. $4L_3\$\$5$

Priority Queues:

Base Struct	Pre-Terminal	Probability	Pivot Value
$L_3D_1S_1$	$L_34!$	0.097	0
$D_1L_3S_2D_1$	$4L_3**4$	0.081	2
$D_1L_3S_2D_1$	$5L_3\$\$4$	0.063	0
$D_1L_3S_2D_1$	$4L_3\$\$5$	0.063	3

Base Struct	Pre-Terminal	Probability	Pivot Value
$D_1L_3S_2D_1$	$4L_3**4$	0.081	2
$D_1L_3S_2D_1$	$5L_3\$\$4$	0.063	0
$D_1L_3S_2D_1$	$4L_3\$\$5$	0.063	3
$L_3D_1S_1$	$L_34\%$	0.045	2
$L_3D_1S_1$	$L_35!$	0.0325	1

Base Struct	Pre-Terminal	Probability	Pivot Value
$D_1L_3S_2D_1$	$5L_3\$\$4$	0.063	0
$D_1L_3S_2D_1$	$4L_3\$\$5$	0.063	3
$L_3D_1S_1$	$L_34\%$	0.045	2
$L_3D_1S_1$	$L_35!$	0.0325	1
$D_1L_3S_2D_1$	$4L_3**5$	0.027	3

Base Struct	Pre-Terminal	Probability	Pivot Value
$D_1L_3S_2D_1$	$4L_3\$\$5$	0.063	3
$D_1L_3S_2D_1$	$6L_3\$\$4$	0.063	0
$L_3D_1S_1$	$L_34\%$	0.045	2
$L_3D_1S_1$	$L_35!$	0.0325	1
$D_1L_3S_2D_1$	$5L_3**4$	0.027	2
$D_1L_3S_2D_1$	$4L_3**5$	0.027	3
$D_1L_3S_2D_1$	$5L_3\$\$5$	0.021	3

Depending on how the first two lines (same probability) are sorted you get

Base Struct	Pre-Terminal	Probability	Pivot Value
$D_1L_3S_2D_1$	$4L_3\$\$6$	0.063	3
$D_1L_3S_2D_1$	$6L_3\$\$4$	0.063	0
$L_3D_1S_1$	$L_34\%$	0.045	2
$L_3D_1S_1$	$L_35!$	0.0325	1
$D_1L_3S_2D_1$	$5L_3**4$	0.027	2
$D_1L_3S_2D_1$	$4L_3**5$	0.027	3
$D_1L_3S_2D_1$	$5L_3\$\$5$	0.021	3

or

Base Struct	Pre-Terminal	Probability	Pivot Value
$D_1 L_3 S_2 D_1$	$4L_3\$\$5$	0.063	3
$L_3 D_1 S_1$	$L_3 4\%$	0.045	2
$L_3 D_1 S_1$	$L_3 5!$	0.0325	1
$D_1 L_3 S_2 D_1$	$5L_3 * *4$	0.027	2
$D_1 L_3 S_2 D_1$	$4L_3 * *5$	0.027	3
$D_1 L_3 S_2 D_1$	$6L_3 * *4$	0.027	2
$D_1 L_3 S_2 D_1$	$5L_3\$\$5$	0.021	3
$D_1 L_3 S_2 D_1$	$6L_3\$\$5$	0.021	3

Exercise 10 (Case study μ-`shout` (II): a multi-client echo server (**4 Points**))**.** Extend your implementation from Exercise 3 to support multiple simultaneous connections and password-based authentication. Each message from one client should be sent to every other client connected and password-information should be stored in the file `/etc/ushoutd.passwd` (note that you do not have to provide sophisticated user management such as user registration).

Additionally, fix all previously discovered bugs and vulnerabilities (if any) and try not to introduce new ones.

Exercise 11 (Bad Password Practice: Users (**1+3 Points (Bonus +2)**))**.** You somehow managed to acquire an excerpt of a password database (see the `a4_master.passwd` file in KVV). Upon close inspection, you suspect it might be a `master.passwd` file extracted from an OpenBSD 6.1 system.

1. Explain how passwords are stored (encoded) in this file. How are the lines in `master.passwd` calculated? **Solution:** See `passwd(5)` for details on the contents of the file. From this manpage we find `crypt(3)`, where we find that by default the `bcrypt` hash function is used (a construction using the Blowfish cipher). The `bcrypt` hash function (more precisely: key derivation function) takes as input a password (*key*) and a *salt* value as well as a configuration variable *cost*. The *key*, salt and *cost* parameters are used to initialize a blowfish *state*. This state is then used to encrypt the string "OrpheanBeholderScryDoubt" 64 times. The original paper by Provos and Mazières [1] covers the details.

References

[1] Niels Provos and David Mazières. *A Future-Adaptable Password Schemee*. Proceedings of the annual conference on USENIX Annual Technical Conference, 1999