Software Engineering

WS 2022/23, Assignment 01

Prof. Dr. Sven Apel
Annabelle Bergum
Sebastian Böhm
Christian Hechtl

Handout: 11.11.2022
Handin: 23.11.2022 23:59 CET

Organizational Section:

- The assignment must be accomplished by yourself. You are not allowed to collaborate with anyone. Plagiarism leads to failing the assignment.
- The deadline for the submission is fixed. A late submission leads to a desk reject of the assignment.
- The submission must consist of a PDF file fulfilling the following criteria:
  - You used the LaTeXtemplate provided in the materials section on the course's CMS page.
  - Your name and matriculation number are included as specified by the template.
  - Your answers to the tasks including feature diagrams are exported from FeatureIDE.
  - No screenshots or photos of handwritten solutions will be accepted.
- Any violation of these submission format rules leads to a desk reject of the assignment.
- Questions regarding the assignment can be asked in the forum or during tutorial sessions. Please do not share any parts that are specific to your solution, as we will have to count that as attempted plagiarism.
- If you encounter any technical issues, inform us immediately.
- FeatureIDE[1] is an extension to eclipse–java. It can be downloaded via the Eclipse marketplace.

Task 1 [5 Points]

Create a feature diagram for all the pizzas presented in the menu in Table 1 using FeatureIDE.
The features correspond to the ingredients listed in the table. Include all crosstree constraints in addition to the feature diagram. The feature diagram must model *exactly* the pizzas listed in the table. It must not leave out any of the listed pizzas and it must not include any additional pizzas. The ingredients should be modeled as features.

Task 2 [3 Points]

Create a feature diagram (including cross-tree constraints) for the command-line backup tool duplicity using FeatureIDE. Your model has to include all features and constraints that are included in the *simplified man–page* of the tool given in Figure 3.

[1] `http://www.featureide.com/`

Task 3 [5.5 Points]

a) Convert the abstract syntax in Figure 1 of the command-line tool GREP into a propositional formula, following the definition of the lecture. [4 Points]

b) Propose each a *generalization, specialization, and refactoring* of the propositional formula from a).
Note: You do not have to write the entire formula again, it suffices to only write down the old and new versions of the clause(s) you changed. [1.5 Points]

$$
\begin{aligned}
fn \; = \bigstar \langle\!\langle \wedge \langle\!\langle \, & Grep, \\
& \bigcirc \langle\!\langle \, Grep, \; \triangle \langle\!\langle \, ProgramInfo, Help, VersionInfo \, \rangle\!\rangle \, \rangle\!\rangle, \\
& \bigcirc \langle\!\langle \, Grep, \; \triangle \langle\!\langle \, MatcherSelection, extRegex, string, basicRegex, perlRegex \, \rangle\!\rangle \, \rangle\!\rangle, \\
& \bigcirc \langle\!\langle \, Grep, \; \blacktriangle \langle\!\langle \, MatchingControl, ignoreCase, invertMatch, wordRegex, lineRegex \, \rangle\!\rangle \, \rangle\!\rangle, \\
& \bigcirc \langle\!\langle \, Grep, \; \blacktriangle \langle\!\langle \, OutputControl, count, filesWithoutMatch, filesWithMatch, onlyMatching, quiet, noMsg \, \rangle\!\rangle \, \rangle\!\rangle, \\
& \bigcirc \langle\!\langle \, Grep, \; \wedge \langle\!\langle \, FileDirSelection, \bigcirc \langle\!\langle \, FileDirSelection, recursive \, \rangle\!\rangle \, \rangle\!\rangle \, \rangle\!\rangle, \\
& \rangle\!\rangle \, \rangle\!\rangle
\end{aligned}
$$

Figure 1: The abstract syntax of grep

.

Task 4 [6.5 Points]

a) Derive the abstract syntax for the feature diagram in Figure 2 that models a shoe generator. [4.5 Points]

b) Give four *valid* configurations for the shoe generator in *functional* notation [1 Points]

c) Give four *invalid* configurations for the shoe generator in *formula* notation [1 Points]

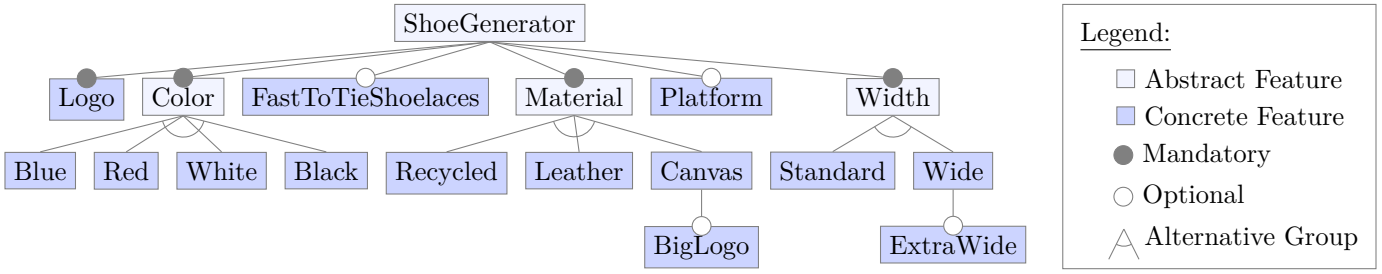

Figure 2: The feature diagram of the shoe generator.

Table 1: This table lists all possible pizzas, no other combination is possible.

All pizzas with tomato sauce and cheese on widespread dough		
Pizza Magaritha		5.00 €
Pizza Vegetaria	with paprika, corn and pineapple	8.75 €
Pizza Sea	with salmon and tuna	7.50 €
Pizza Meatlover	with ham and salami	7.50 €
Pizza Salmon	with salmon	6.25 €
Pizza Tonno	with tuna	6.25 €
Pizza Prosciutto	with ham	6.25 €
Pizza Salami	with salami	6.25 €
Pizza Paprika	with paprika	6.25 €
Pizza Mexican	with paprika and corn	7.50 €
Pizza Hawaii (veg.)	with paprika and pineapple	7.50 €
Pizza Redsea	with salmon, tuna and paprika	8.75 €
Pizza Speziale	with ham, salami and paprika	8.75 €
Pizza Redsalmon	with salmon and paprika	7.50 €
Pizza Tunica	with tuna and paprika	7.50 €
Pizza Normale	with ham and paprika	7.50 €
Pizza Hawaii	with ham, paprika and pineapple	8.75 €
Pizza Kids	with salami and paprika	7.50 €
Pizza Frutti di Mare	with salmon, tuna, paprika and corn	10.00 €
Pizza Tutti	with ham, salami, paprika and corn	10.00 €
Pizza Nordsea	with salmon, paprika and corn	7.50 €
Pizza Ocean	with tuna, paprika and corn	7.50 €
Pizza Farm	with ham, paprika and corn	7.50 €
Pizza a la chef	with ham, paprika, corn and pineapple	10.00 €
Pizza Mister	with salami, paprika and corn	7.50 €
Additional:		
Garlic		0.25€
Chilli		0.25€

duplicity Manpage (simplified)

Name

`duplicity` - Encrypted backup using rsync algorithm

Synopsis

```
duplicity cleanup [options] [--force] target_url
duplicity full [options] source_directory target_url
duplicity incremental [options] source_directory target_url
duplicity restore [options] source_url target_directory
duplicity verify [options] source_url target_directory
```

Description

`Duplicity` incrementally backs up files and directory by encrypting tar-format volumes with GnuPG and uploading them to a remote (or local) file server. Currently local, ftp, sftp/scp, rsync, WebDAV, WebDAVs, Google Docs, HSi and Amazon S3 backends are available. Because duplicity uses librsync, the incremental archives are space efficient and only record the parts of files that have changed since the last backup. Currently duplicity supports deleted files, full Unix permissions, directories, symbolic links, fifos, etc., but not hard links.

Actions

cleanup Delete the extraneous duplicity files on the given backend. Non-duplicity files, or files in complete data sets will not be deleted. This should only be necessary after a duplicity session fails or is aborted prematurely. Note that –force will be needed to delete the files rather than just list them.

full Indicate full backup. If this is set, perform full backup even if signatures are available.

incremental If this is requested an incremental backup will be performed. Duplicity will abort if old signatures cannot be found.

restore Restore backup

verify Enter verify mode instead of restore. If the –file-to-restore option is given, restrict verify to that file or directory. duplicity will exit with a non-zero error level if any files are different. On verbosity level 4 or higher, log a message for each file that has changed.

Options

–**dry-run** Calculate what would be done, but do not perform any backend actions

–**encrypt-key key** When backing up, encrypt to the given public key, instead of using symmetric (traditional) encryption. Can be specified multiple times.

–**encrypt-secret-keyring filename** This option can only be used with –encrypt-key, and changes the path to the secret keyring for the encrypt key to filename This keyring is not used when creating a backup. If not specified, the default secret keyring is used which is usually located at .gnupg/secring.gpg

–**extra-clean** When cleaning up, be more aggressive about saving space. For example, this may delete signature files for old backup chains. See the cleanup argument for more information.

–**file-to-restore path** This option may be given in restore mode, causing only path to be restored instead of the entire contents of the backup archive. path should be given relative to the root of the directory backed up.

–**full-if-older-than time** Perform a full backup if an incremental backup is requested, but the latest full backup in the collection is older than the given time. See the TIME FORMATS section for more information.

–**force** Proceed even if data loss might result. Duplicity will let the user know when this option is required.

Figure 3: The simplified Manpage of duplicity.