Hinweise zum 1. Aufgabenblatt

31.10.2019

Sebastian Ruland

sebastian.ruland@es.tu-darmstadt.de

Fragen und Support:
se2@es.tu-darmstadt.de
oder ins **Moodle Forum**.

ES Real-Time Systems Lab

Prof. Dr. rer. nat. Andy Schürr

Dept. of Electrical Engineering and Information Technology

Dept. of Computer Science (adjunct Professor)

www.es.tu-darmstadt.de

Übungen

TECHNISCHE UNIVERSITÄT DARMSTADT

Übungen können dieses Semester wieder korrigiert werden.

- Es wird **kein** Bonussystem angeboten
  → Klausur zählt zu 100%
- Wir *kontrollieren* jede Abgabe, *markieren Fehler* und geben *Verbesserungsvorschläge*
- Teilnahme an den Übungen freiwillig
- Vorgehen dieses Semester:
  - Abgabe nach Besprechung der Musterlösung (bei Möglichkeit nur Teile abgeben, bei denen Korrektur erwünscht)
  - Abgabe erfolgt in Gruppen

Gruppeneinteilung

TECHNISCHE UNIVERSITÄT DARMSTADT

Bitte 5+ Personen pro Übungsgruppe!

→ Bei abweichenden Gruppengrößen werden wir evtl. Zuweisungen vornehmen

→ *Gruppen wechseln jetzt noch möglich*

→Ggf. als größere Gruppe anmelden und in Teilgruppen arbeiten

In eine der Gruppen in moodle eintragen!

Wichtig: Richtlinien für den Übungsablauf

TECHNISCHE UNIVERSITÄT DARMSTADT

- Jeden Donnerstag: Übungsblatt vom Web runterladen (vor der Übung)
  - Abgabe: meist Freitag darauffolgender Woche, 8:00 Uhr
    - manchmal auch später. Moodle zeigt die letzte Möglichkeit zur Abgabe an

  → spätere Abgaben können nicht berücksichtigt werden, da einmalig auf abgebene Korrekturen überprüft wird!

- Hochladen in moodle:
  - Name der Abgabedatei: **loesung<xx>_<gn>.pdf**
    - <xx> steht für die jeweilige Übungsblattnummer
    - <gn> steht für den jeweiligen Gruppennamen
    - Das **PDF** muss **alles** beinhalten, da wir **nur** das PDF ausdrucken und bewerten!
    - **Titelblatt:** Gruppenname und Namen aller Bearbeiter; Nummer Übungsblatt!
  - Bitte nur eine Abgabe pro Gruppe

Wichtig: Richtlinien für den Übungsablauf

TECHNISCHE UNIVERSITÄT DARMSTADT

- Aufgabenblatt zur aktuellen Übung wird spätestens am Mittwoch vor der Übung online gestellt

- In den Übungen besprechen wir:
  - Häufige Fehler & Fragen des letzten Übungsblattes
  - Hinweise & Fragen zum neuen Übungsblatt

Hinweise zum ersten Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Begriffe:

- Diff/Delta:

  Unterschied zwischen zwei Dateien (meist Revisionen)

- Patch:

  Technische Repräsentation eines Diffs

- Patchen:

  Durch anwenden eines Patches auf eine Version eine neue erzeugen

- Hunk:

  Abschnitt in einem Patch, der eine Änderung angibt

- Mögliche Änderungen:

  Zeile hinzufügen, Zeile löschen, Zeile ändern

Patch

TECHNISCHE UNIVERSITÄT DARMSTADT

Version 1

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMinimal;
04 int iMedian;
05 int iMinIndex = 0;
...
```

Version 2

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMin;
04 double iMittelwert;
05 int iMinIndex = 0;
...
```

Alte Datei (Version 1)

Neue Datei (Version 2)

```
*** 02, 04 *** Hunk 1 ****
int iArray[] = {0, 5, 3};
!      int iMinimal;
-      int iMedian;
--- 02, 04 ---
int iArray[] = {0, 5, 3};
!      int iMin;
+      double iMittelwert;
**********************************
```


Patch

TECHNISCHE UNIVERSITÄT DARMSTADT

Version 1

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMinimal;
04 int iMedian;
05 int iMinIndex = 0;
...
```

Version 2

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMin;
04 double iMittelwert;
05 int iMinIndex = 0;
...
```

Hunk: von letzter gleicher Zeile

bis letzte unterschiedliche

Alte Datei (Version 1)

Neue Datei (Version 2)

```
*** 02, 04 *** Hunk 1 ****
int iArray[] = {0, 5, 3};
!      int iMinimal;
-      int iMedian;
--- 02, 04 ---
int iArray[] = {0, 5, 3};
!      int iMin;
+      double iMittelwert;
****************************
```


Patch

TECHNISCHE UNIVERSITÄT DARMSTADT

Version 1

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMinimal;
04 int iMedian;
05 int iMinIndex = 0;
...
```

Version 2

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMin;
04 double iMittelwert;
05 int iMinIndex = 0;
...
```

Hunk: von letzter gleicher Zeile

bis letzte unterschiedliche

Alte Datei (Version 1)

Neue Datei (Version 2)

```
*** 02, 04 *** Hunk 1 ****
int iArray[] = {0, 5, 3};
!      int iMinimal;
-      int iMedian;
--- 02, 04 ---
int iArray[] = {0, 5, 3};
!      int iMin;
+      double iMittelwert;
****************************
```


Patch

TECHNISCHE UNIVERSITÄT DARMSTADT

Version 1

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMinimal;
04 int iMedian;
05 int iMinIndex = 0;
...
```

Version 2

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMin;
04 double iMittelwert;
05 int iMinIndex = 0;
...
```

Hunk: von letzter gleicher Zeile bis letzte unterschiedliche

Alte Datei (Version 1)

Neue Datei (Version 2)

```
*** 02, 04 *** Hunk 1 ****
int iArray[] = {0, 5, 3};
!      int iMinimal;
-      int iMedian;
--- 02, 04 ---
int iArray[] = {0, 5, 3};
!      int iMin;
+      double iMittelwert;
**********************************
```

Patch

Version 1

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMinimal;
04 int iMedian;
05 int iMinIndex = 0;
...
```

Version 2

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMin;
04 double iMittelwert;
05 int iMinIndex = 0;
...
```

Hunk: von letzter gleicher Zeile bis letzte unterschiedliche Kontextzeile

Alte Datei (Version 1)

Neue Datei (Version 2)

```
*** 02, 04 *** Hunk 1 ****
int iArray[] = {0, 5, 3};
!      int iMinimal;
-      int iMedian;
--- 02, 04 ---
int iArray[] = {0, 5, 3};
!      int iMin;
+      double iMittelwert;
**********************************
```

Patch

Version 1

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMinimal;
04 int iMedian;
05 int iMinIndex = 0;
...
```

Version 2

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMin;
04 double iMittelwert;
05 int iMinIndex = 0;
...
```

Hunk: von letzter gleicher Zeile bis letzte unterschiedliche Kontextzeile

Zeile geändert

Alte Datei (Version 1)

Neue Datei (Version 2)

```
*** 02, 04 *** Hunk 1 ****
int iArray[] = {0, 5, 3};
!      int iMinimal;
-      int iMedian;
--- 02, 04 ---
int iArray[] = {0, 5, 3};
!      int iMin;
+      double iMittelwert;
****************************
```

Patch

Version 1

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMinimal;
04 int iMedian;
05 int iMinIndex = 0;
...
```

Version 2

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMin;
04 double iMittelwert;
05 int iMinIndex = 0;
...
```

Hunk: von letzter gleicher Zeile bis letzte unterschiedliche

Kontextzeile

Zeile geändert

Zeile gelöscht

Alte Datei (Version 1)

Neue Datei (Version 2)

```
*** 02, 04 *** Hunk 1 ****
int iArray[] = {0, 5, 3};
!     int iMinimal;
-     int iMedian;
--- 02, 04 ---
int iArray[] = {0, 5, 3};
!     int iMin;
+     double iMittelwert;
****************************
```

Patch

Version 1

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMinimal;
04 int iMedian;
05 int iMinIndex = 0;
...
```

Version 2

```
01 int counter = 0;
02 int iArray[] = {0, 5, 3};
03 int iMin;
04 double iMittelwert;
05 int iMinIndex = 0;
...
```

Alte Datei (Version 1)

Neue Datei (Version 2)

```
*** 02, 04 *** Hunk 1 ****
int iArray[] = {0, 5, 3};
!     int iMinimal;
-     int iMedian;
--- 02, 04 ---
int iArray[] = {0, 5, 3};
!     int iMin;
+     double iMittelwert;
**********************************
```

Hunk: von letzter gleicher Zeile bis letzte unterschiedliche

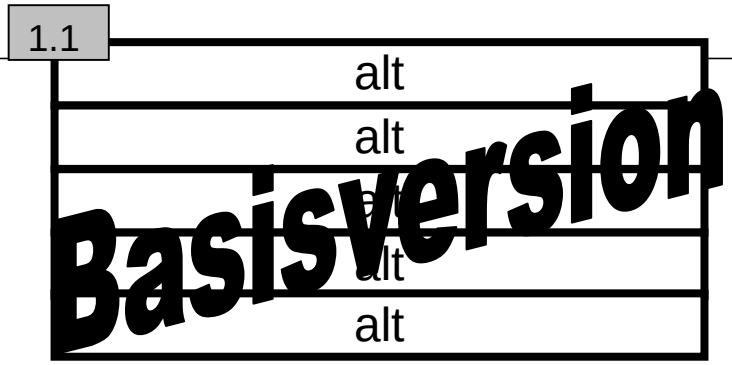
Kontextzeile

Zeile geändert

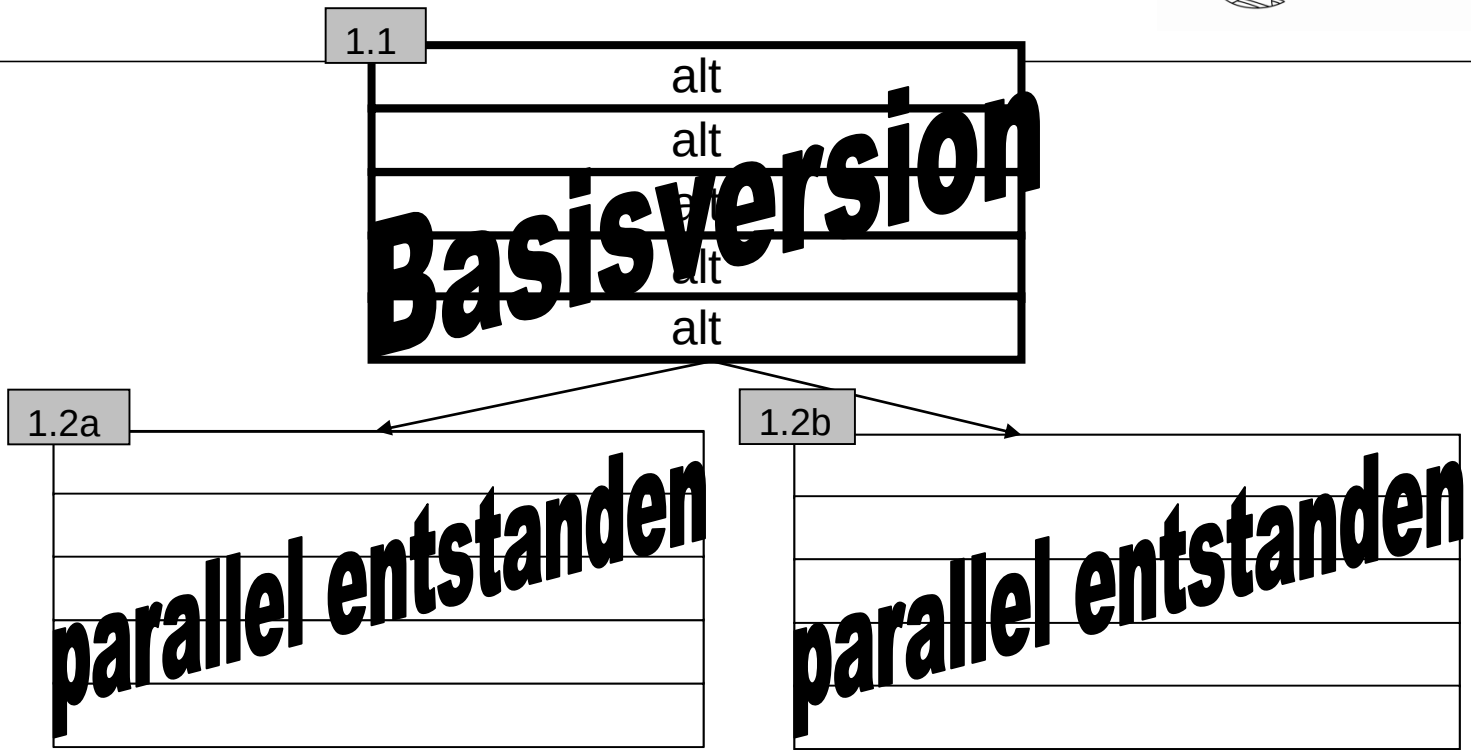
Zeile gelöscht

Zeile hinzugefügt

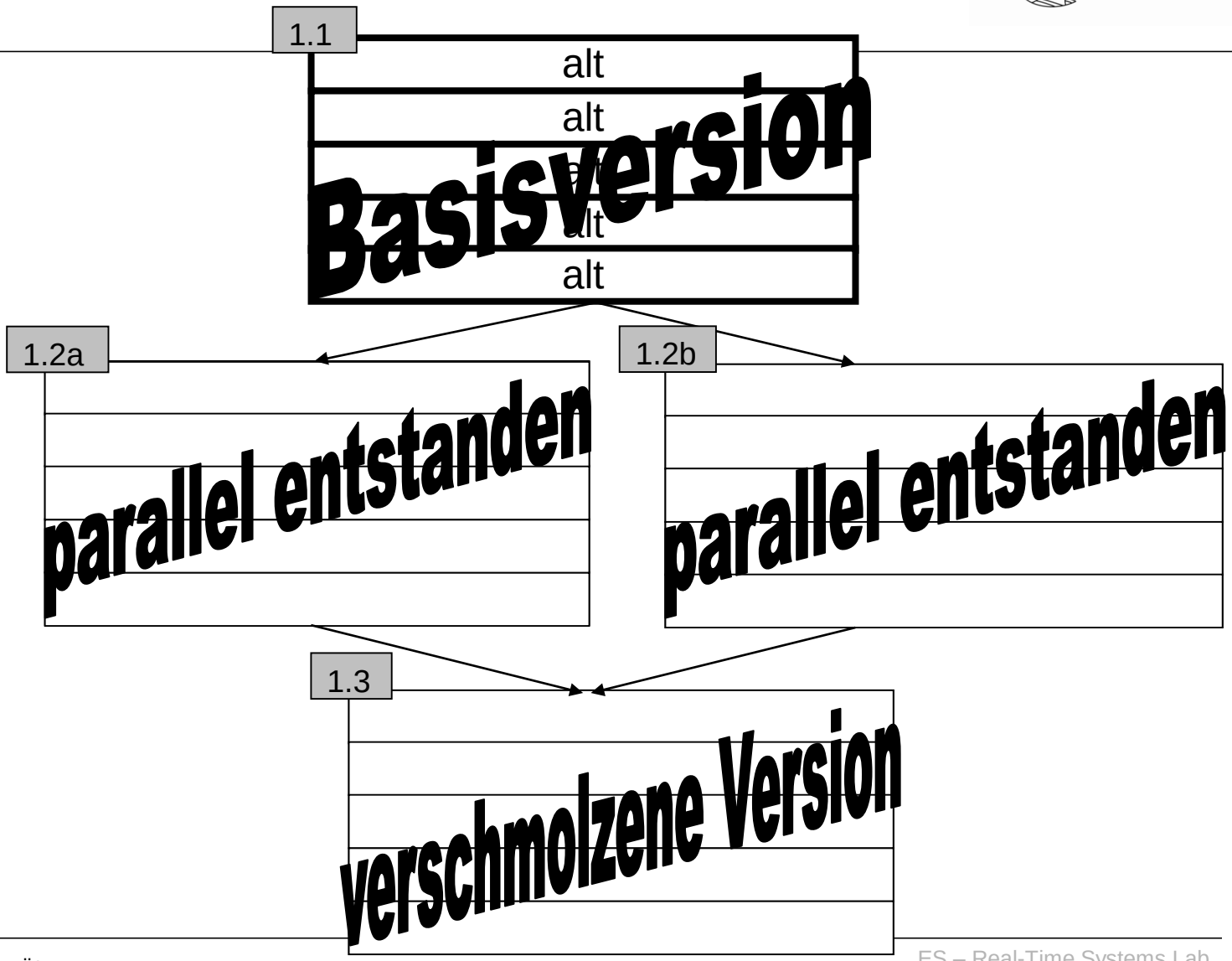
Drei-Wege-Verschmelzung



Drei-Wege-Verschmelzung

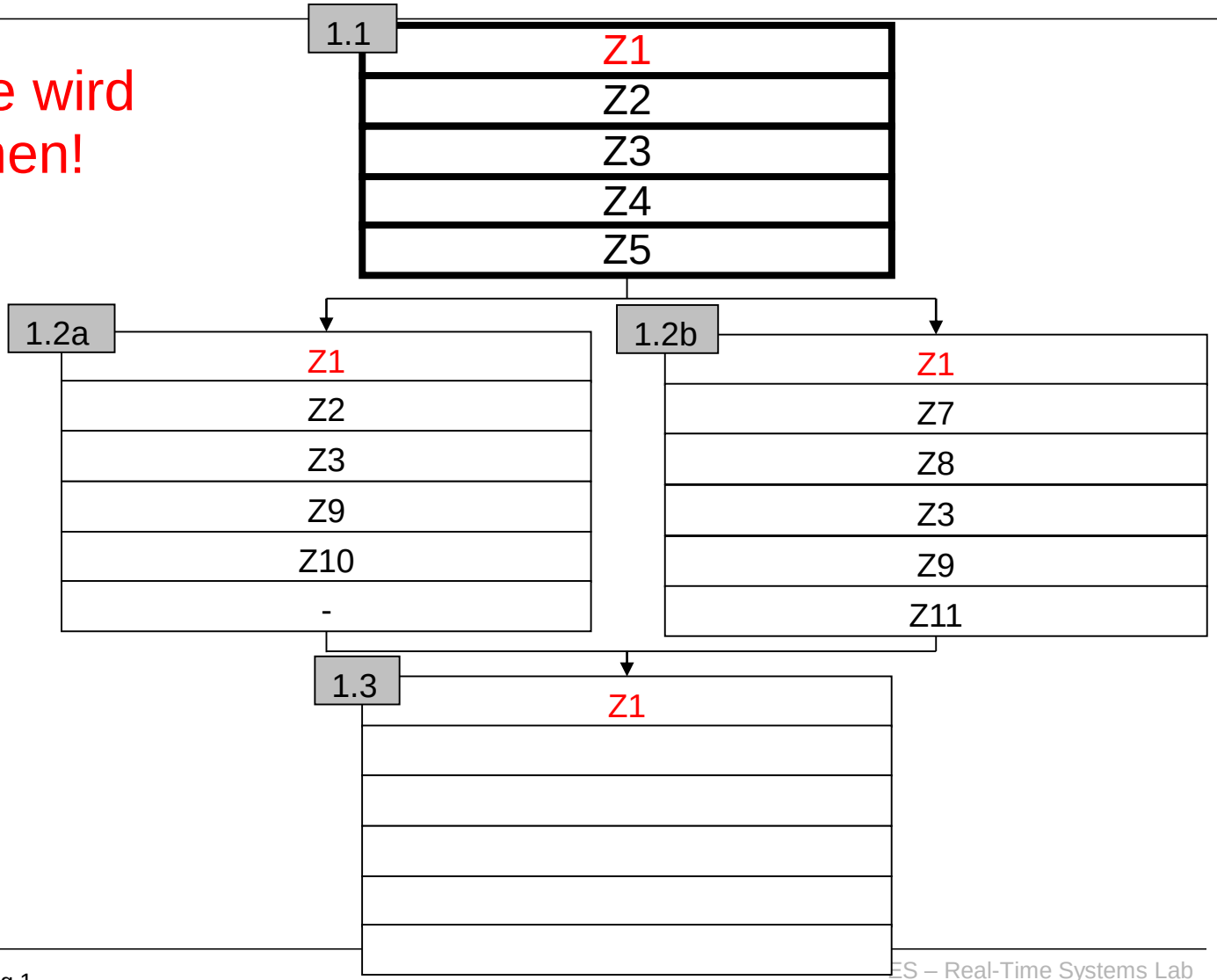


Drei-Wege-Verschmelzung



Drei-Wege-Verschmelzung

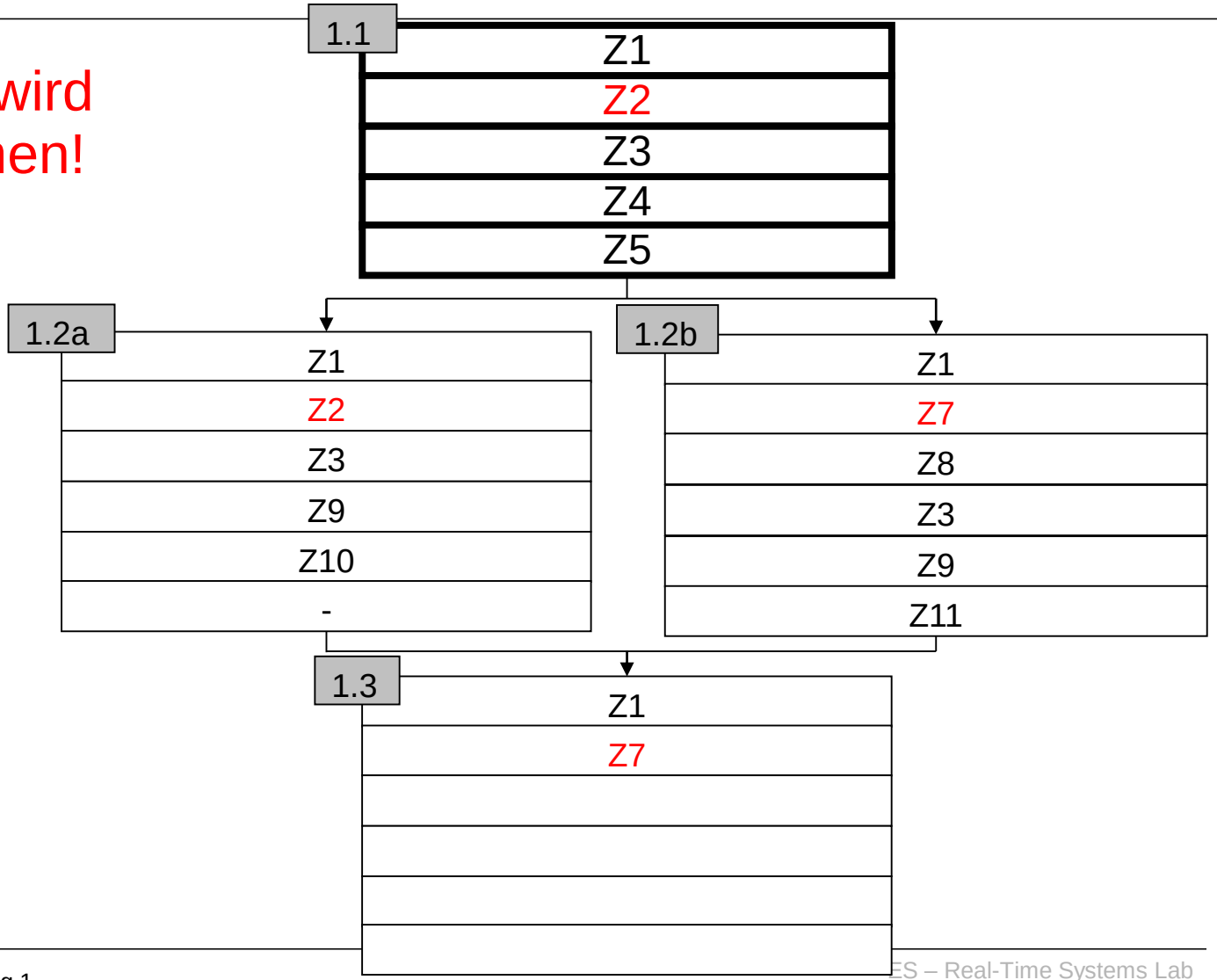
Gleiche Zeile wird übernommen!



Drei-Wege-Verschmelzung

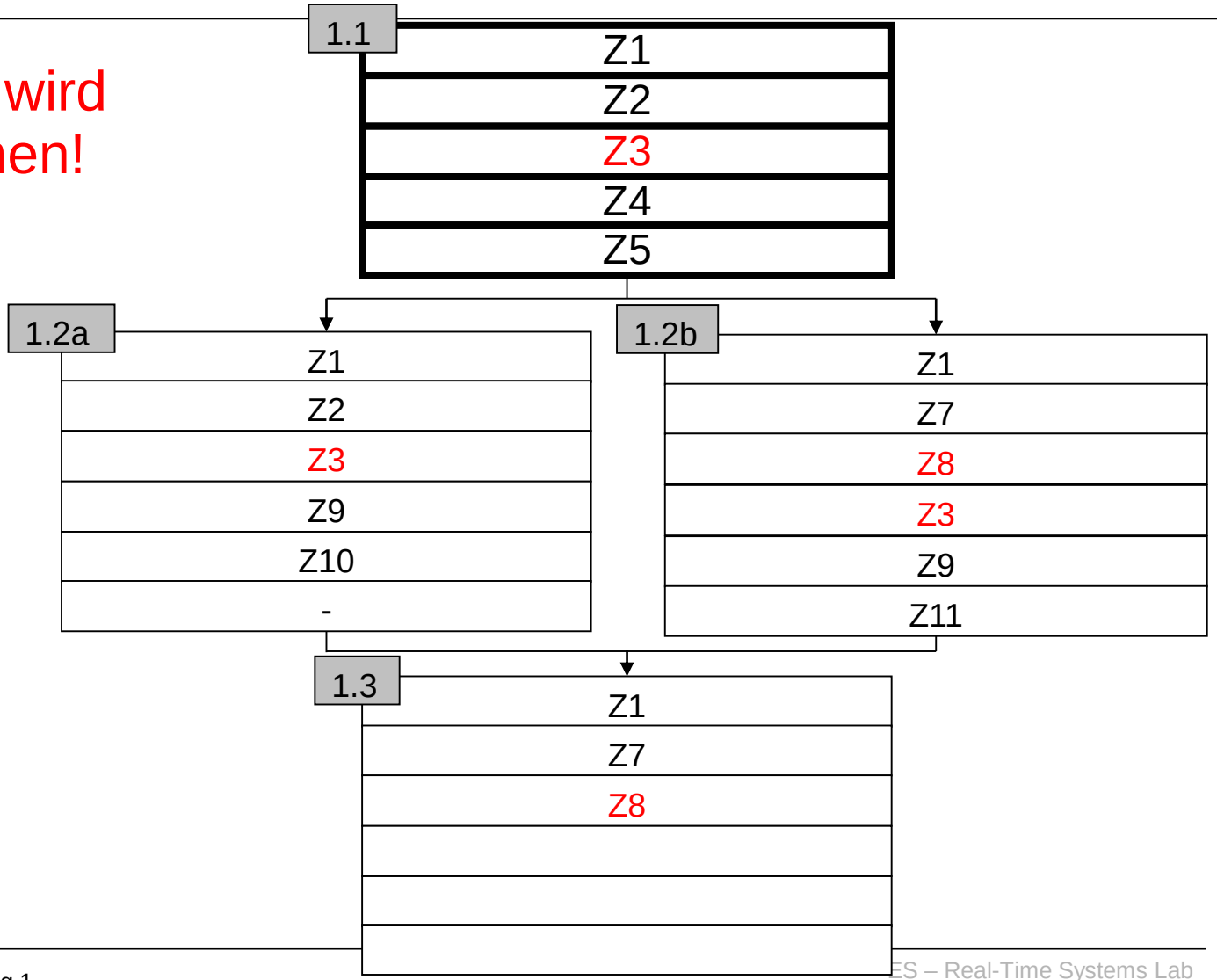
TECHNISCHE UNIVERSITÄT DARMSTADT

Änderung wird übernommen!



Drei-Wege-Verschmelzung

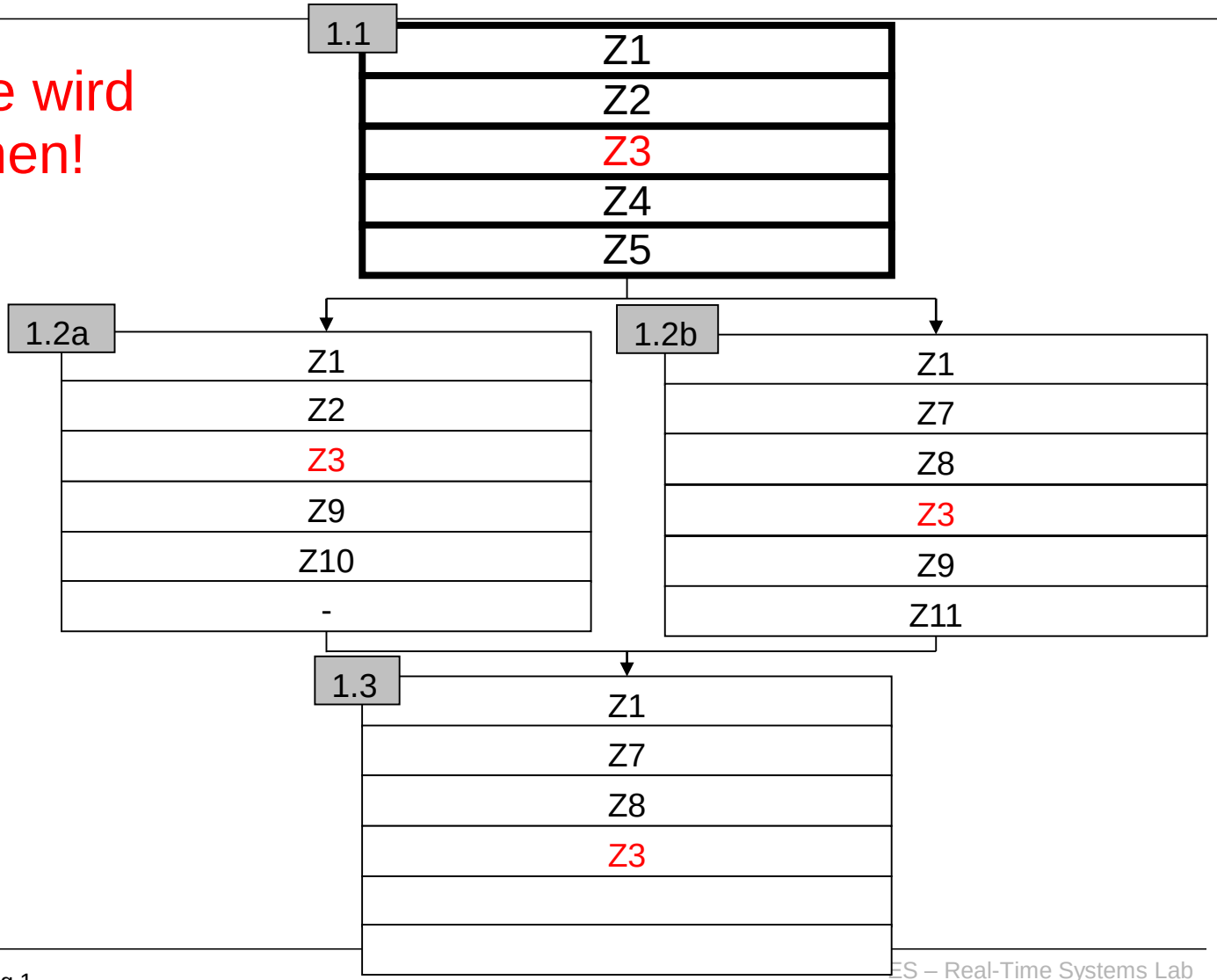
Neue Zeile wird übernommen!



Drei-Wege-Verschmelzung

TECHNISCHE UNIVERSITÄT DARMSTADT

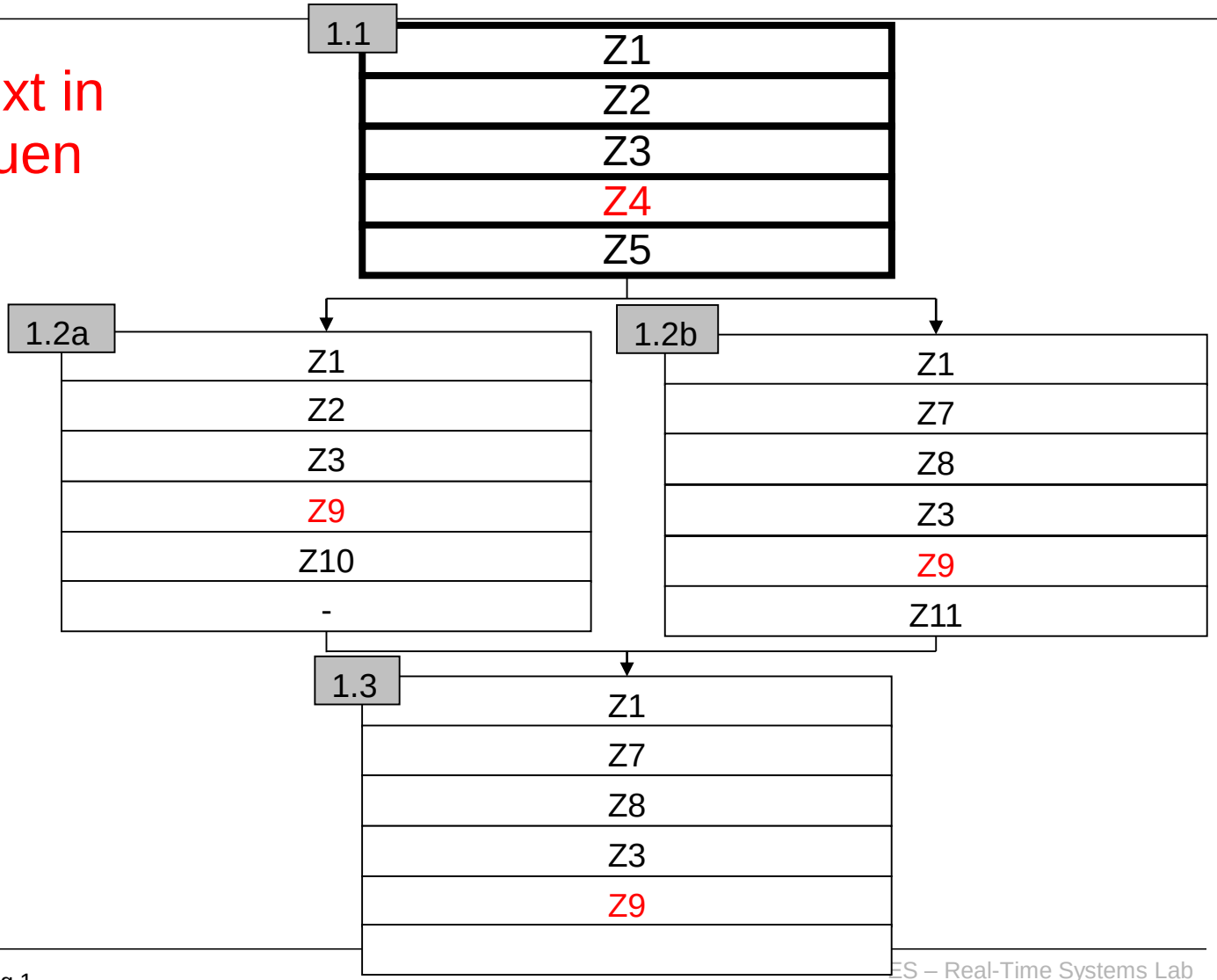
Gleiche Zeile wird übernommen!



Drei-Wege-Verschmelzung

TECHNISCHE UNIVERSITÄT DARMSTADT

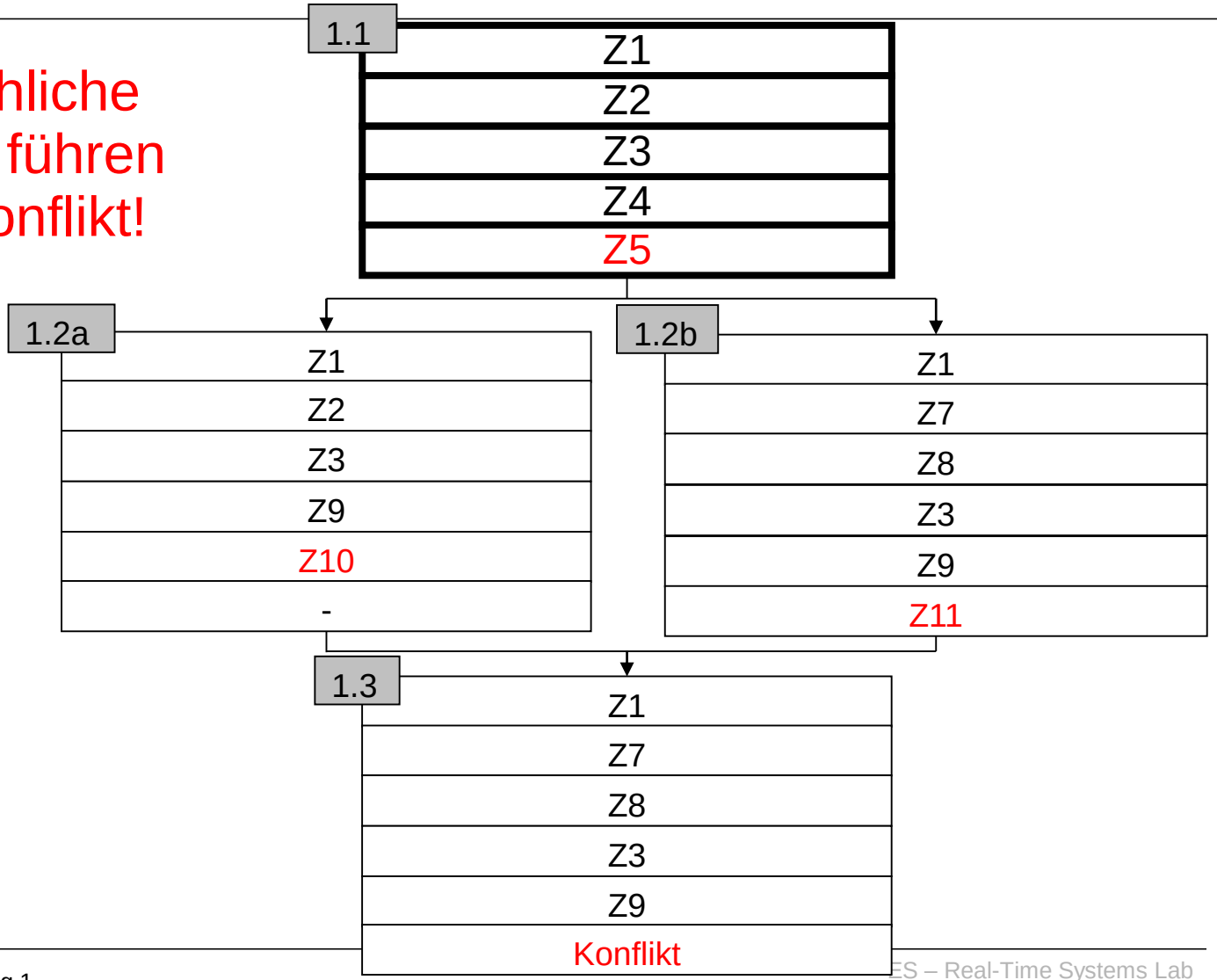
Gleicher Text in beiden neuen Zeilen!



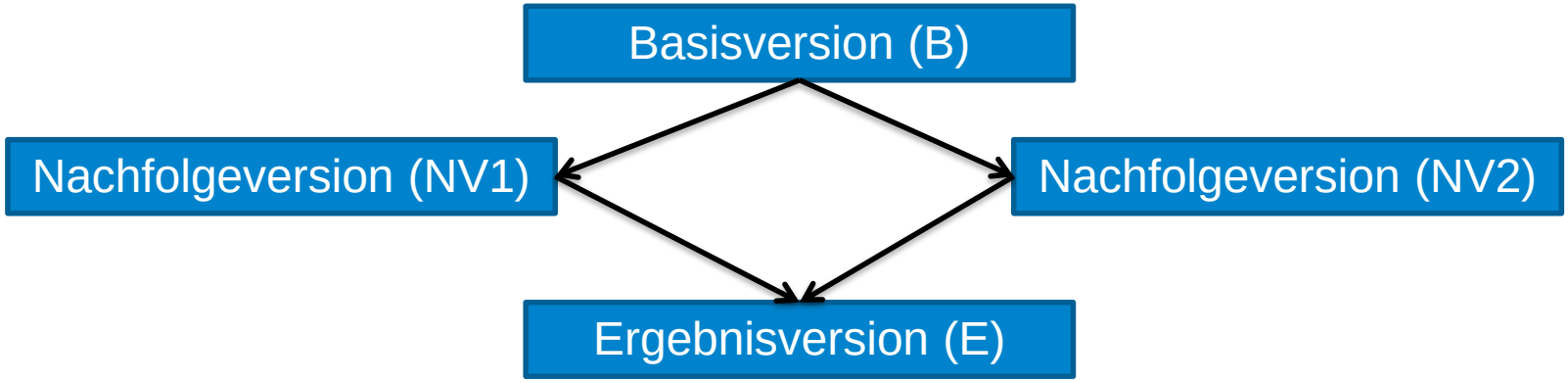
Drei-Wege-Verschmelzung

TECHNISCHE UNIVERSITÄT DARMSTADT

Widersprüchliche Änderungen führen zu einem Konflikt!



Zusammenfassung: Drei-Wege-Verschmelzung



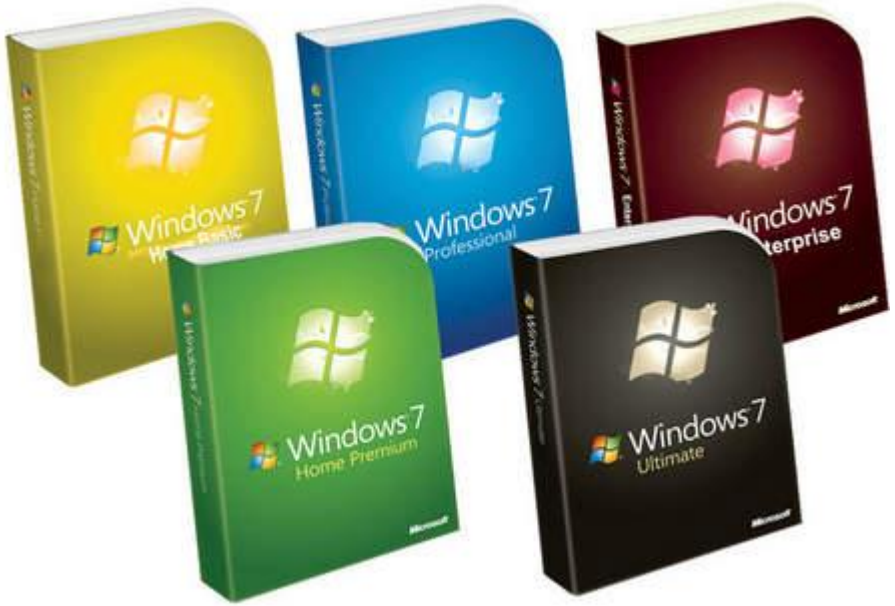
- Textzeile in B, NV1 und NV2 gleich → Textzeile in E
- Textzeile in B aber nicht in NV1 oder/und NV2 → Textzeile nicht in E
- Textzeile in NV1 oder/und NV2 aber nicht in B → Textzeile in E
- Textzeile aus B in NV1 und NV2 geändert → manuelle Konfliktbehebung (gilt auch für neue Textzeilen in NV1 und NV2 an gleicher Stelle)

Variantenmanagement

TECHNISCHE UNIVERSITÄT DARMSTADT

- Eine Software wird als eine Produktlinie betrachtet
  → Diverse Produktinstanzen können in ihrer Funktionalität konfiguriert werden

- Produktkonfiguration kann somit den Buildprozess steuern, um unterschiedliche Varianten einer Software zu erstellen



Variantenmanagement

TECHNISCHE UNIVERSITÄT DARMSTADT

- Feature-Model:
  - Modell aller Eigenschaftskriterien einer Produktlinie (z.B. eines Autos – 5er BMW)
  - Auswahl bestimmter Eigenschaften definieren eine konkrete Variante / Konfiguration der Produktlinie (z.B. BMW 530i – Comfort Edition)

- Features:
  - Konkrete Eigenschaften / Funktionalitäten des Produktes
  - Können voneinander abhängig (*require*) sein oder sich ausschließen (*exclude*)
  - Features sind
    - Optional (*optional*)
    - Benötigt (*mandatory*)
  - Features können in Gruppen eingeordnet werden
    - Oder-Gruppe (*or*), mindestens ein Feature muss ausgewählt werden
    - Alternativ-Gruppe (*alternative*), genau ein Feature muss ausgewählt werden

Variantenmanagement

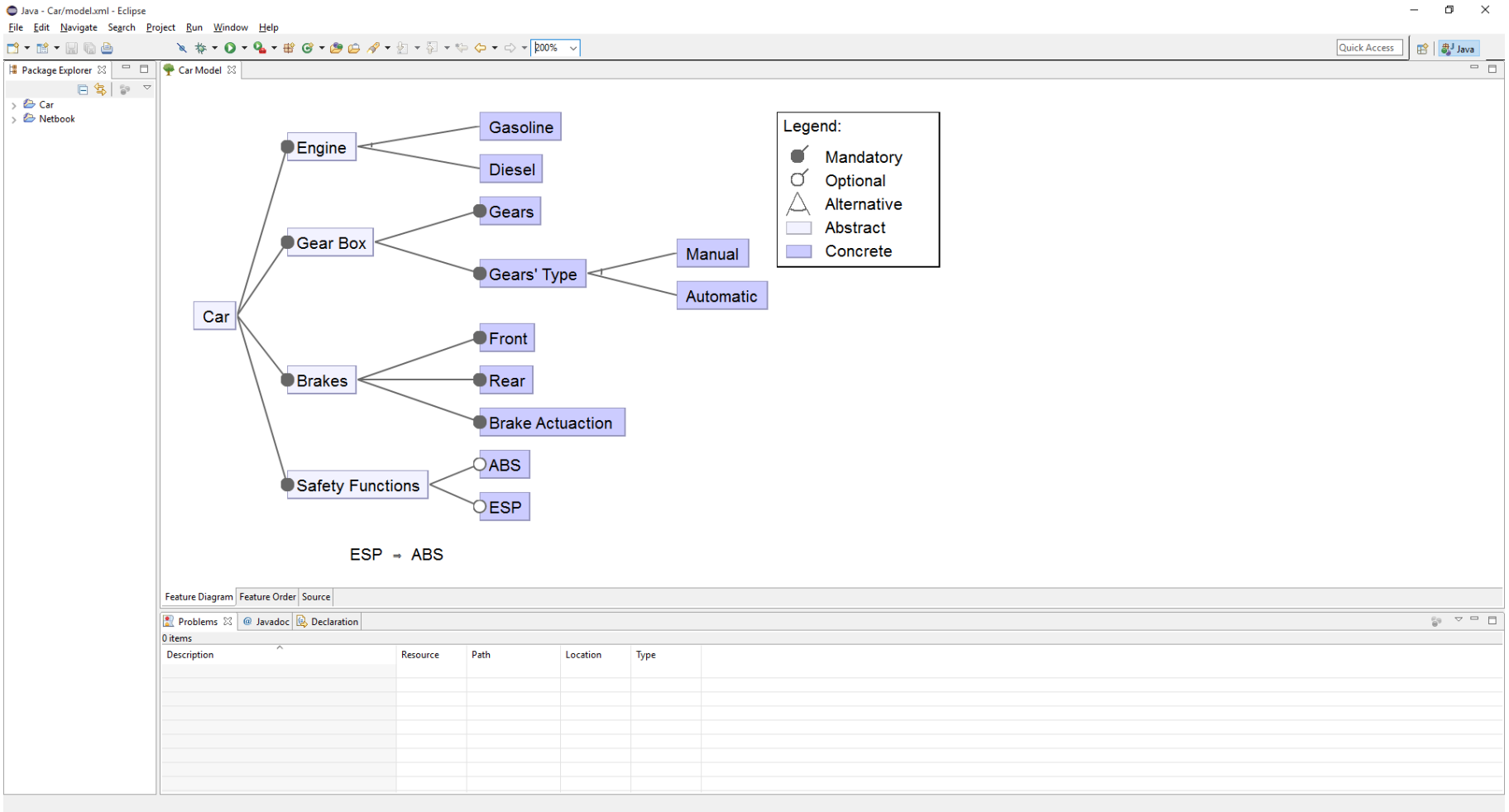
TECHNISCHE UNIVERSITÄT DARMSTADT

FeatureIDE: Modellierungstool für Feature-Modelle

- Eclipse-Plugin
- OpenSource

Feature-IDE DEMO

TECHNISCHE UNIVERSITÄT DARMSTADT



Zertifizierung zum Tester

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zertifizierung zum Software-Tester (Foundation Level nach ISTQB)
- Kosten: 127,93€ pro Person (Erstattung teilweise möglich)
- **Anmeldung ist verpflichtend!**
- **Prüfungsdatum:** 13.02.2020
- **Uhrzeit:** 08:00
- **Raum:** S306/051 + S306|053
- **Was ist mitzubringen?**
  - Studentenausweis
  - Personalausweis
  - Dokumentenechter Stift (blauer/schwarzer Kuli)

Allgemeines Certified Tester

Vorbereitung:

- SE-II Script
- Certified Tester Lehrplan
- Buch: Basiswissen Softwaretest
  (ausleihbar für 1-2 Tage am
  FG Echtzeitsysteme Sekretariat)
  Noch nicht ausleihbar, da momentan neue Auflage bestellt wird. News dazu auf Moodle
- Musterprüfung des Certified Tester
  (Datum wird auf Moodle bekannt gegeben)