

Übung 4

Niclas Kusenbach, 360227

5th January 2026

Aufgabe 1

Aufgabe 1a: Formalisierung als CSP

Variablen: Die Menge der Variablen ist $V = \{A, B, C, U\}$.

- A, B, C : Repräsentieren die Ziffern der Gleichung.
- U : Repräsentiert den Übertrag (Carry) von der Einer- in die Zehnerstelle.

Wertebereiche (Domains):

- $D_A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $D_B = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (Da B führende Ziffer im Ergebnis ist, gilt $B \neq 0$)
- $D_C = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $D_U = \{0, 1\}$ (Maximaler Übertrag bei $9 + 9$ ist 1)

Constraints:

1. **Alldiff-Constraint:** Jeder Buchstabe steht für eine unterschiedliche Ziffer.

$$A \neq B \wedge A \neq C \wedge B \neq C$$

2. **Spalten-Constraints (Addition):**

- *Einerstelle (Rechts):* $A + B = 10 \cdot U + C$
- *Zehnerstelle (Links):* $U = B$

Aufgabe 1b: Backtracking-Search

Annahmen:

- Variablenreihenfolge: C, B, A, U .
- Werteauswahl: Aufsteigend ($0 \rightarrow 9$).

Ablauf:

1. **Variable C:** Wähle $C = 0$.
 - Konsistenzprüfung: OK.
2. **Variable B:** Wähle $B = 1$ (0 nicht in D_B).
 - Konsistenzprüfung (*alldiff*): $1 \neq 0 \rightarrow$ OK.

3. **Variable A:** Iteration durch $D_A = \{0, \dots, 9\}$.
- Versuch $A = 0$: Konflikt mit C (*alldiff*). → Backtrack.
 - Versuch $A = 1$: Konflikt mit B (*alldiff*). → Backtrack.
 - Versuch $A = 2$: *alldiff* OK. → Weiter zu U .
 - (a) **Variable U:**
 - (b) Versuch $U = 0$: Constraint $U = B$ ($0 = 1$) verletzt.
 - (c) Versuch $U = 1$: Constraint $U = B$ ($1 = 1$) OK.
Aber Addition: $A+B = 10U+C \Rightarrow 2+1 \neq 10(1)+0 \Rightarrow 3 \neq 10$.
Verletzt.
 - (d) Keine Werte mehr für U . → Backtrack zu A .
 - Versuche $A = 3$ bis $A = 8$: Scheitern analog im Schritt U an der Summenbedingung.
 - Versuch $A = 9$: *alldiff* OK. → Weiter zu U .
 - (a) **Variable U:**
 - (b) Versuch $U = 0$: Verletzt $U = B$.
 - (c) Versuch $U = 1$: Erfüllt $U = B$ ($1 = 1$).
Prüfung Addition: $9 + 1 = 10(1) + 0 \Rightarrow 10 = 10$. OK.
- Ergebnis:** Gefundene Belegung $\sigma = \{C = 0, B = 1, A = 9, U = 1\}$.

Aufgabe 1c: Forward-Checking und Constraint-Propagation

Einsatz von Forward-Checking (FC): Forward-Checking eliminiert inkonsistente Werte aus den Domains der *noch nicht zugewiesenen* Variablen.

1. Zuweisung $C = 0$: Entfernt 0 aus D_A und D_B .
2. Zuweisung $B = 1$:
 - Entfernt 1 aus D_A .
 - Durch Constraint $U = B$ reduziert sich D_U auf $\{1\}$.
3. Zuweisung A :
 - Bei Wahl von $A = 2$: FC prüft Konsistenz mit verbleibender Variable U .
 - Gleichung $A + B = 10U + C$ ergibt $2 + 1 = 10U \Rightarrow U = 0.3$.
 - Der Wert 0.3 (bzw. 0) ist nicht in $D_U = \{1\}$. Die Domain D_U läuft leer (Domain Wipeout).
 - **Effekt:** Der Algorithmus bricht für $A = 2$ (und analog $A = 3 \dots 8$) sofort ab, ohne die Variable U überhaupt zu instanziiieren.

Potenzial von Constraint-Propagation höherer Ordnung (Arc Consistency): Forward-Checking prüft nur die Konsistenz zwischen der *aktuellen* Variable und den *zukünftigen*. Es prüft nicht die Beziehungen zwischen zwei zukünftigen Variablen (hier A und U).

- Nach der Zuweisung $C = 0$ und $B = 1$ sind A und U noch offen.
- $D_A = \{2, \dots, 9\}$ und $D_U = \{1\}$.
- Eine Prüfung der Kantenkonsistenz (Arc Consistency, z.B. AC-3) hätte

erkannt:

$$A + 1 = 10 \cdot U + 0$$

Da U zwingend 1 ist (wegen $B = 1$), muss $A + 1 = 10$ sein, also $A = 9$.

- **Reduktion:** Alle Werte außer 9 wären aus D_A entfernt worden, **bevor** die Suche für A überhaupt begonnen hätte. Dies hätte die Fehlversuche für $A \in \{2..8\}$ komplett vermieden (Backtrack-free search in diesem Zweig).