Mathematisch-Didaktische Analyse: Fehler-Kategorisierung und Päckchen-Generierung

1. Fehler-Klassifikations-System

Das System basiert auf 9 wissenschaftlich fundierte Fehler-Kategorien im Zahlenraum bis 20:

Primäre Fehlertypen (mathematisch-kognitiv):

1. zehnuebergang_addition - Zehnerübergang bei Addition

Erkennung: (num1 + num2 > 10) && (num1 ≤ 10 && num2 ≤ 10)

Beispiel: 8+5=12 (statt 13)

Didaktisches Problem: Fehlende Zerlegungsstrategien

Kognitive Lücke: Kind nutzt 10 nicht als Ankerzahl

2. zehnuebergang_subtraction - Zehnerübergang bei Subtraktion

Erkennung: (num1 > 10) && ((num1 % 10) < num2)

Beispiel: 13-9=5 (statt 4)

Didaktisches Problem: Keine tragfähige Strategie für Zehnerunterschreitung

Kognitive Lücke: Zerlegung des Subtrahenden nicht verstanden

3. complementary_pairs - Partnerzahlen zu 10/20

Erkennung: (num1 + num2 === 10 || num1 + num2 === 20) && incorrectAnswer ≠ correctAnswer

Beispiel: 7+3=9 (statt 10)

Didaktisches Problem: Partnerzahlen nicht automatisiert

Kognitive Lücke: Fehlende Teil-Ganzes-Beziehung

4. calculation_facts - Grundlegende Rechenfertigkeiten

Erkennung: Fehler ohne Zehnerübergang oder Partnerzahl

Beispiel: 5+3=9 (statt 8)

Didaktisches Problem: Zählendes Rechnen statt Strategien

Kognitive Lücke: Keine Mustererkennung

Sekundäre Fehlertypen (strukturell-konzeptionell):

5. number_reversal - Zahlenreihenfolge vertauscht

Erkennung: Bei Subtraktion: incorrectAnswer === (num2 - num1) oder |num2 - num1|

Beispiel: 13-5=0 (gerechnet: 5-13)

Didaktisches Problem: Kommutativgesetz fälschlich übertragen

Kognitive Lücke: Operationsverständnis Subtraktion fehlt

6. digit_reversal - Ziffern vertauscht (Zahlendreher)

Erkennung: 2-stellige Zahl, Ziffern gespiegelt

Beispiel: Ergebnis 21 statt 12

Didaktisches Problem: Unsicheres Stellenwertverständnis

Kognitive Lücke: Räumliche Ziffernfolge nicht gefestigt

7. operation_confusion - Operation verwechselt

Erkennung: Addition gerechnet, aber Subtraktion gefordert (oder umgekehrt)

Beispiel: 8+5=3 (Subtraktion statt Addition)

Didaktisches Problem: Operationszeichen übersehen

Kognitive Lücke: Symbol-Bedeutung nicht verankert

8. subtraction_borrowing - Allgemeine Subtraktionsschwierigkeiten

Erkennung: Subtraktion ohne spezifisches Muster

Didaktisches Problem: Umkehrbeziehung nicht verstanden

Kognitive Lücke: Subtraktion als separate Operation statt Umkehrung

9. pattern_break - Musterbruch (in Päckchen)

Erkennung: Logisch, in Päckchen-Kontext

Didaktisches Problem: Keine systematische Mustererkennung

Kognitive Lücke: Strukturelles Denken fehlt

2. Päckchen-Generierungs-System (12 Typen)

Kategorie A: Grundlegende Operationsbeziehungen

1. constant_sum - Konstanz der Summe

Mathematisches Prinzip: Gegensinnige Veränderung bei konstanter Summe

Pattern: num1↑+1, num2↓-1, sum=konstant

Beispiel:

5 + 8 = 13

6 + 7 = 13

7 + 6 = 13

Ziel-Fehlertypen: zehnuebergang_addition, complementary_pairs

Didaktische Einsicht: Flexibilität bei Zahlzerlegungen

2. opposing_change - Gegensinnige Veränderung (Subtraktion)

Mathematisches Prinzip: Konstanter Minuend, variabler Subtrahend

Pattern: minuend=konstant, subtrahend↑+1, differenz↓-1

Beispiel:

15 - 5 = 10

15 - 6 = 9

15 - 7 = 8

Ziel-Fehlertypen: zehnuebergang_subtraction, subtraction_borrowing

Didaktische Einsicht: Zusammenhang Subtrahend ↔️ Differenz

3. inverse_tasks - Umkehraufgaben

Mathematisches Prinzip: Addition ↔️ Subtraktion Umkehrbeziehung

Pattern: a+b=c, c-b=a, c-a=b

Beispiel:

8 + 5 = 13

13 - 5 = 8

13 - 8 = 5

Ziel-Fehlertypen: zehnuebergang_addition/subtraction, operation_confusion

Didaktische Einsicht: Operationen sind verbunden, nicht isoliert

4. exchange_tasks - Tauschaufgaben (Kommutativgesetz)

Mathematisches Prinzip: a+b = b+a

Pattern: Operanden vertauschen

Beispiel:

3 + 7 = 10

7 + 3 = 10

Ziel-Fehlertypen: calculation_facts, complementary_pairs

Didaktische Einsicht: Bei Addition ist Reihenfolge irrelevant

Kategorie B: Strukturelle Muster

5. same_direction - Gleichsinnige Veränderung

Mathematisches Prinzip: Beide Summanden wachsen

Pattern: num1↑+1, num2↑+1, sum↑+2

Beispiel:

3 + 4 = 7

4 + 5 = 9

5 + 6 = 11

Ziel-Fehlertypen: calculation_facts

Didaktische Einsicht: Veränderung beider Summanden → doppelte Auswirkung

6. crossing_tens - Systematischer Zehnerübergang

Mathematisches Prinzip: Eine Zahl konstant (8 oder 9), andere wächst

Pattern: 8=konstant, addend↑+1, sum↑+1 (über 10)

Beispiel:

8 + 2 = 10

8 + 3 = 11

8 + 4 = 12

Ziel-Fehlertypen: zehnuebergang_addition

Didaktische Einsicht: Routine beim Zehnersprung aufbauen

Kategorie C: Zerlegungsstrategien

7. decomposition_steps - Schrittweise Zerlegung

Mathematisches Prinzip: Mehrstufige Rechenwege explizit

Pattern: Zur 10, dann weiter

Beispiel (13-9):

Schritt 1: 13 - 3 = 10 (zur 10)

Schritt 2: 9 - 3 = 6 (Rest)

Schritt 3: 10 - 6 = 4 (Ergebnis)

Ziel-Fehlertypen: zehnuebergang_subtraction, subtraction_borrowing

Didaktische Einsicht: Transparenz der Zerlegungsstrategie

Kategorie D: Metakognitive Päckchen

8. error_research - Fehlerforschungs-Päckchen

Mathematisches Prinzip: Absichtlicher Fehler im Muster

Pattern: 4 korrekte + 1 falsche Aufgabe

Ziel-Fehlertypen: Alle (Vertiefung)

Didaktische Einsicht: Musterverständnis durch Fehleranalyse

9. pattern_analysis - Muster-Ja/Nein-Analyse

Mathematisches Prinzip: Ist das ein schönes Päckchen?

Pattern: Gegeben → bewerten → begründen

Ziel-Fehlertypen: Alle (Transferleistung)

Didaktische Einsicht: Kritisches, argumentatives Denken

10. continuation_challenge - Fortsetzungs-Challenge

Mathematisches Prinzip: Muster selbst fortsetzen

Pattern: 3 Aufgaben gegeben → Kind erfindet 2-3 weitere

Ziel-Fehlertypen: Alle (Kreativität)

Didaktische Einsicht: Aktive Mustererkennung und Transfer

Kategorie E: Spezialprobleme

11. number_reversal_demo - Reihenfolge demonstrieren

Mathematisches Prinzip: Bei Subtraktion gilt NICHT: a-b = b-a

Pattern: Direkter Vergleich unmöglicher Aufgaben

Beispiel:

15 - 8 = 7 ✓

8 - 15 = ??? (nicht möglich in ZR 20)

Ziel-Fehlertypen: number_reversal

Didaktische Einsicht: Operationsverständnis schärfen

12. digit_detective - Zahlendreher-Detektiv

Mathematisches Prinzip: Stellenwertverständnis Zehner/Einer

Pattern: Vergleich gespiegelter Zahlen (12 vs. 21)

Ziel-Fehlertypen: digit_reversal

Didaktische Einsicht: Bewusste Ziffernfolge

13. operation_contrast - Addition vs. Subtraktion

Mathematisches Prinzip: Direkte Gegenüberstellung

Pattern: Gleiche Zahlen, unterschiedliche Operationen

Beispiel:

12 + 7 = 19 (größer)

12 - 7 = 5 (kleiner)

Ziel-Fehlertypen: operation_confusion

Didaktische Einsicht: Operationsbedeutung verdeutlichen

3. Mathematisch-Didaktische Kernprinzipien

Algorithmus-Logik:

1. Fehler klassifizieren (klassifySingleError)

→ Pattern-Matching anhand numerischer Eigenschaften

2. Fehlertyp → Päckchen-Typ Mapping (selectPaeckchenTypeForError)

→ Jeder Fehlertyp hat 1-3 optimale Päckchen-Typen

3. Päckchen generieren (generate*Paeckchen Funktionen)

→ Deterministische Regel-basierte Aufgabengenerierung

4. Didaktische Begründung (whyThisHelps)

→ Explizite Verbindung: Fehler → Strategie → Einsicht

Mathematische Qualitätskriterien:

Strukturtreue: Jedes Päckchen folgt einer klaren mathematischen Regel

Sichtbarkeit: Muster sind visuell und numerisch erkennbar (↑+1, ↓-1, =)

Transferierbarkeit: Erkannte Muster gelten für viele Aufgaben

Fehlerprävention: Päckchen adressieren die kognitive Ursache, nicht nur Symptom

Didaktische Differenzierung:

Easy: constant_sum, exchange_tasks, same_direction, operation_contrast

Medium: opposing_change, crossing_tens, inverse_tasks, number_reversal_demo, digit_detective

Hard: decomposition_steps, error_research, pattern_analysis, continuation_challenge

4. Forschungs- und Optimierungspotenzial

A) Datenbasierte Verfeinerung:

Tracking: Welche Päckchen führen zu messbarer Fehlerreduktion?

A/B-Testing verschiedener Päckchen für denselben Fehlertyp

B) Erweiterte Klassifikation:

Systematische Fehler (z.B. immer +1 zu viel) vs. zufällige Fehler

Schweregrad-Bewertung (konsistenter Fehler vs. einmalig)

C) Adaptive Sequenzierung:

Wenn crossing_tens erfolgreich → automatisch constant_sum als Vertiefung

Wenn error_research zu schwer → Rückfall auf same_direction

D) Linguistische Ebene:

Sentence Stems validieren: Fördern sie mathematische Argumentation?

Reflexionsfragen optimieren basierend auf Schülerantworten

E) Visualisierungsforschung:

Welche Visualisierungshints (Plättchen, Pfeile, Zahlenstrahl) sind am effektivsten?

Integration von dynamischen Visualisierungen

Fazit: Das System kombiniert fehlerdiagnostische Präzision mit strukturorientierter Päckchen-Didaktik. Die mathematische Fundierung liegt in erkennbaren Zahlenbeziehungen (Teil-Ganzes, Umkehrung, Zerlegung), die didaktische Kraft in der expliziten Mustererkennung als Weg zur Strategieentwicklung.