

# Didaktik: Selbstkorrigierende Kodierung: “Wer bin ich?” mit einmal Lügen

## 1 Kontext

Diese Unterrichtssequenz ist für eine Doppellektion in einer 9. Klasse (1. Jahr Informatik-Unterricht, Grundlagenfach) gemeint. Die gewünschten, aber optionalen, Vorkenntnissen sind:

- Das Binärsystem verstehen
- Das Konzept von Kontrollbit kennen
- Mit dem Entscheidungsbaum vertraut sein
- Die binäre Suche kennen

Diese Unterrichtssequenz könnte auch mit wenigen Anpassungen für jüngere SuS verwendet werden.

## 2 Einführung

### 2.1 Ziel

Das Ziel dieser Sequenz ist, die selbstkorrigierende Kodierung einzuführen. Am Ende der Sequenz müssen die SuS den Zweck der selbstkorrigierenden Kodierung verstehen und selbst überlegt/entdeckt haben, wie man die Daten gestalten kann, dass man mit so wenig Redundanz wie möglich einen Fehler entdecken und korrigieren kann.

### 2.2 Konzept der Sequenz

Die SuS entdecken die Regel der selbstkorrigierenden Kodierung mit dem Spiel “Wer bin ich?” mit einmal Lügen. Nachdem sie selbst entdeckt haben, wie man mit so wenig wie möglich Fragen wie möglich zur richtigen Antwort gelangt, wird die Parallele zur selbstkorrigierenden Kodierung gemacht und die Theorie erklärt.

### 2.3 Adversarial Spielen

Alle Spiele, die hier gespielt werden, haben die folgende Struktur: Spieler 1 wählt eine Antwort (Gesicht oder Zahl), Spieler 2 stellt Fragen, die mit *ja* oder *nein* beantwortet werden, um die richtige Antwort zu finden. Damit die Strategie von Spieler 2 nicht von Glück abhängt, wird Spieler 1 adversarial (gegnerisch) spielen. Das heisst, er wird seine Wahl so verändern, dass so viele Züge wie möglich verwendet werden.

### 2.4 Begründung der gewählten Sequenz

Der spielerische Ansatz macht die Problematik sehr konkret und erlaubt den SuS, sich sofort mit der Lösung zu beschäftigen. Die Parallele mit der selbstkorrigierenden Kodierung ist dann relativ selbstverständlich.

### 3 Theorie

Es wird das Spiel von “Wer bin ich?” mit 16 Gesichtern von Abenteuer-Informatik (mit der Genehmigung von Jens Gallenbacher) verwendet. Die Gesichter sind in einem balancierten Binärbaum mit 4 Eigenschaften verteilt. Jede Eigenschaft kann mit einer Frage bestimmt werden. Deshalb können wir immer mit 4 Grundfragen zur richtigen Antwort gelangen.

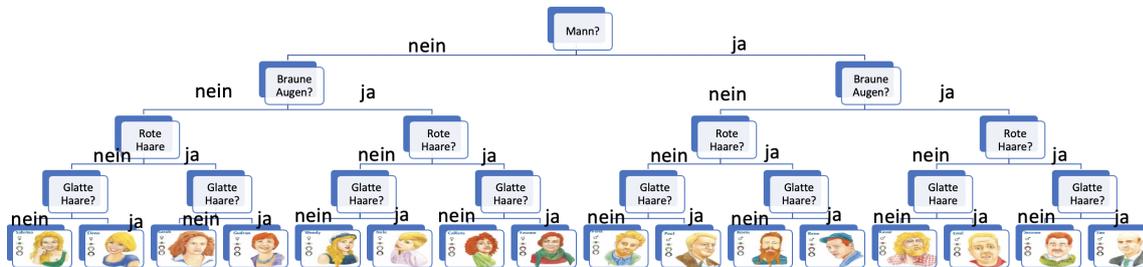


Abbildung 1: Binärbaum

1. Frau oder Mann. Frage: *Bist du ein Mann?*
2. Grüne oder braune Augen. Frage: *Hast du braune Augen?*
3. Blondes oder rotes Haar. Frage: *Hast du rotes Augen?*
4. Lockiges oder glattes Haar. Frage: *Hast du glattes Haar?*

Eine Antwort von *nein* kann mit 0 kodiert werden und eine Antwort von *ja* kann mit 1 kodiert werden. Infolgedessen kann jedes Gesicht mit einer 4-stelligen Binärzahl kodiert werden. Umgekehrt entspricht jede 4-stellige Binärzahl einem Gesicht.

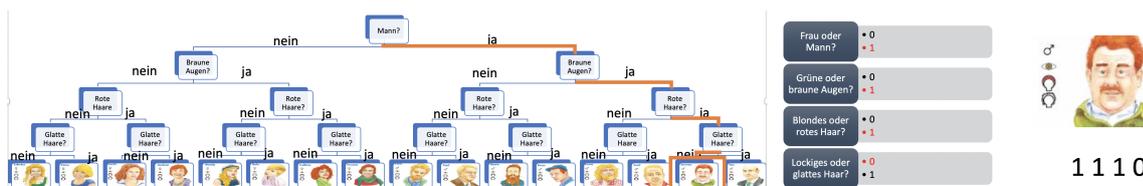


Abbildung 2: Kodierung der Gesichter

#### 3.1 Einmal Lügen

Mit “Einmal Lügen” kann keine Möglichkeit mit einer Frage entfernt werden. Wir können erst Gesichter entfernen, wenn sie 2 Lügen entsprechen würden. Abbildung 3 stellt die Möglichkeiten, welche nach den folgenden 2 Fragen übrig bleiben, dar:

1. Bist du ein Mann? *ja*
2. Hast du braune Augen? *ja*

Die ausgegrauten Bilder stellen die Gesichter dar, die nicht mehr erlaubt sind, d.h. die Gesichter von Frauen mit grünen Augen. Die leicht schattierten Figuren sind die Gesichter, die mit einer Lüge immer noch möglich sind, d.h. Frauen mit braunen Augen oder Männer mit grünen Augen.



Abbildung 3: Kodierung der Gesichter

### 3.2 Strategie 1: Fragen wiederholen

Um zu bestimmen, ob gelogen wurde, können wir die Fragen wiederholen. Wenn die Antworten der ursprünglichen Frage und der Wiederholung nicht übereinstimmen, wissen wir zweifellos, dass es eine Lüge gab. Da es nur eine Lüge geben kann, können wir anschließend diese Frage noch einmal wiederholen, um zur richtigen Antwort zu gelangen. Mit dieser Strategie braucht man höchstens  $2 \cdot 4 + 1$  Fragen, um die Antwort mit 100%-iger Wahrscheinlichkeit zu finden.

Die 4 Grundfragen über die Eigenschaften und deren Wiederholungen stellen 8 Fragen dar, die wir systematisch fragen können. Die letzte Frage ist eine weitere Wiederholung einer Frage, deren Antworten sich widersprechen. Somit ist sie eine adaptive Frage, das heißt, sie hängt von den vorherigen Antworten ab. Wir werden diese letzte Frage die *Kontrollfrage* nennen.

In der Abbildung 4 haben wir senkrecht die gestellten Fragen und waagrecht die entsprechenden Eigenschaften. Die Fragen  $f_1$  bis  $f_4$  beziehen sich auf die 4 Eigenschaften  $e_1$  bis  $e_4$ . Die Kontrollfrage  $k$  bezieht sich auf die Eigenschaft  $e_i$ , bei welcher die Frage  $f_i$  widersprechende Antworten erhalten hat.

### 3.3 Darstellung von “Einmal Lügen”

Die ersten vier Fragen entsprechen den 4 Eigenschaften und die Antworten bezeichnen ein Gesicht, das das gewählte Gesicht ist, wenn es keine Lüge gab. Wir können dieses Gesicht und die Gesichter, die möglich wären, wenn einmal gelogen wurde, im Entscheidungsbaum darstellen. Man muss zwischen den Zweigen ohne Lügen und den Zweigen mit Lügen unterscheiden.

Die Kodierung eines Gesichtes mit einmal Lügen kann höchstens an einer Stelle von der

	$e_1$	$e_2$	$e_3$	$e_4$
$f_1$	x			
$f_2$		x		
$f_3$			x	
$f_4$				x
$f_1$	x			
$f_2$		x		
$f_3$			x	
$f_4$				x
$k$	?	?	?	?

Abbildung 4: Fragen und entsprechende Eigenschaften

Kodierung des Gesichtes ohne Lügen abweichen. Es gibt 4 mögliche Stellen, an welchen die Lüge stattfinden könnte und ein Gesicht ohne Lüge: insgesamt gibt es 5 Möglichkeiten.

Abbildung 4 zeigt die übrigbleibenden Möglichkeiten nach den folgenden 4 Fragen: Mann? *ja*, braune Augen? *ja*, rotes Haar? *ja*, glattes Haar? *nein*, was der Kodierung 1110 entspricht.

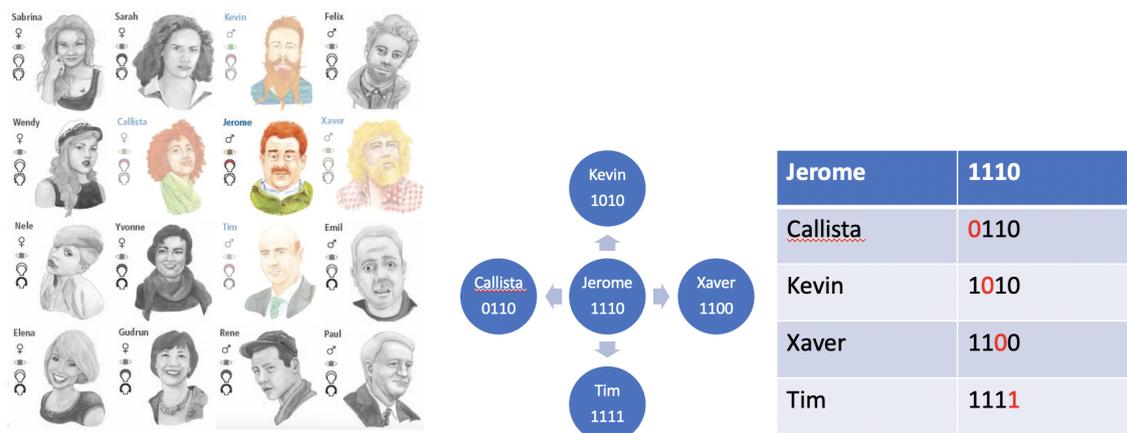


Abbildung 5: Übrig bleibende Möglichkeiten nach 4 Fragen

Die Kodierung der übriggebliebenen Gesichter haben höchstens eine Entfernung von 1 zur Kodierung 1110, das heisst, sie können sich nur an einer Stelle unterscheiden.

Der Entscheidungsbaum mit einmal Lügen ist in Abbildung 5 dargestellt.

### 3.4 Zusammenhang mit der selbstkorrigierenden Kodierung

Mit der selbstkorrigierenden Kodierung haben wir eine 4-stellige Binärzahl, die wir übermitteln möchten. Wenn es einen Übermittlungsfehler gibt, kann sich der übermittelte Code an einer Stelle von dem ursprünglichen Code unterscheiden. Ähnlich zum Spiel "Einmal Lügen" können wir die Binärzahl wiederholen. Wenn beide Binärzahlen nicht übereinstimmen, können wir feststellen, dass es einen Übermittlungsfehler gab, aber es erlaubt uns nicht, ihn zu korrigieren. Dafür brauchen wir noch ein Kontrollbit.

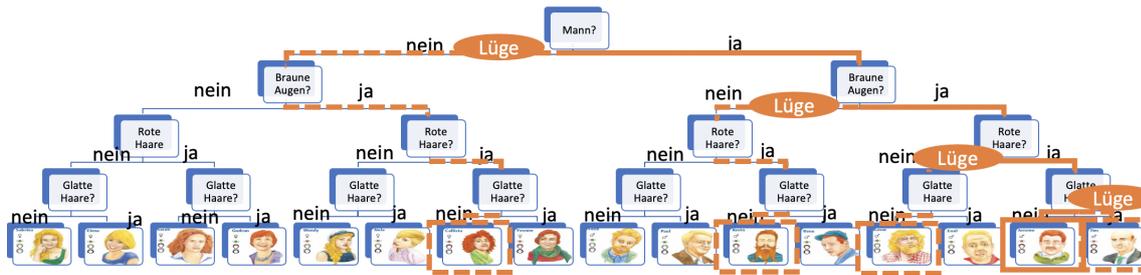


Abbildung 6: Entscheidungsbaum mit einmal Lügen

Der Unterschied mit dem Spiel “Einmal Lügen” ist, dass das Kontrollbit nicht von den empfangenen Bits abhängig sein kann, da der Sender nur weiss, was geschickt wird, aber nicht was empfangen wird. Das heisst, die Bits einer selbstkorrigierenden Kodierung dürfen nicht adaptiv sein, sondern müssen immer die gleiche Information darstellen.

Anstatt von der Kontrollfrage, die sich auf Antworten zu anderen Fragen bezieht, verwendet man ein Kontrollbit, das die Parität der anderen Bits überprüft. Die Parität der übermittelten Binärzahl genügt, um einen eventuellen Fehler korrigieren zu können.

### 3.5 Strategie 2: 3 Kontrollfragen verwenden

Um zwischen den 5 übriggebliebenen Gesichtern zu entscheiden, können wir Kontrollbits verwenden: Wir gruppieren die 4 Eigenschaften in 3 Dreiergruppen, damit jede Eigenschaft mindestens in zwei Gruppen vorkommt und keine Eigenschaft genau in den gleichen Gruppen wie eine andere Eigenschaft vorkommt. Wir fragen für jede Gruppe von Eigenschaften, ob alle drei Eigenschaften stimmen. Wenn eine Antwort über die Eigenschaften falsch war, werden mindestens zwei Antworten zu den Kontrollfragen *nein* sein. Wenn alle Eigenschaften stimmen, wird höchstens eine Antwort zu den Kontrollfragen *nein* sein.

Wenn zum Beispiel die Antworten über die Eigenschaften darauf deuten, dass die gewählte Person ein Mann mit braunen Augen, rotem Haar und lockigem Haar ist, könnten die 3 Kontrollfragen sein:

1. Bist du ein Mann mit braunen Augen und rotem Haar?
2. Bist du ein Mann mit braunen Augen und lockigem Haar?
3. Bist du ein Mann mit rotem und lockigem Haar?

Es gibt die folgenden Möglichkeiten:

- Wenn die gewählte Person eine Frau ist, werden alle drei Kontrollfragen mit *nein* beantwortet.
- Wenn die gewählte Person grüne Augen hat, werden nur die ersten zwei Kontrollfragen mit *nein* beantwortet.
- Wenn die gewählte Person blondes Haar hat, werden nur die erste und dritte Kontrollfragen mit *nein* beantwortet.

- Wenn die gewählte Person glattes Haar hat, werden die zweite und dritte Kontrollfragen mit *nein* beantwortet.
- Wenn alle vier Eigenschaften stimmen, werden entweder alle Kontrollfragen mit *ja* beantwortet oder es wird einmal gelogen und eine Kontrollfrage wird mit *nein* beantwortet.

Deshalb können wir immer mit höchstens 7 Fragen zur richtigen Antwort gelangen.

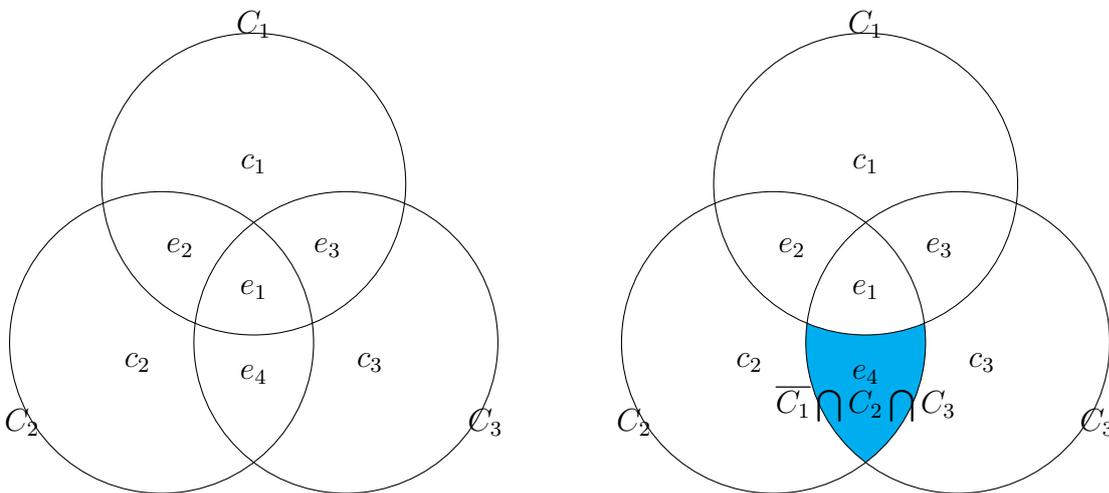
Die Strategie kann mit einer Fehlermeldungstabelle dargestellt werden (Abbildung 7). Die  $x$  entsprechen den Eigenschaften, auf welche die Kontrollfragen sich beziehen und kennzeichnen, in welche Frage die Möglichkeit besteht, dass eine Lüge stattfand, wenn *nein* zu den Kontrollfragen geantwortet wurde.

Wenn zum Beispiel die Kontrollfrage  $c_2$  eine *nein*-Antwort erhalten hat, könnte es eine Lüge entweder in Eigenschaft  $e_1$ , in Eigenschaft  $e_2$ , in Eigenschaft  $e_4$  oder in der Kontrollfrage  $c_2$  selbst geben.

	$e_1$	$e_2$	$e_3$	$e_4$	$c_1$	$c_2$	$c_3$
$c_1$	x	x	x		x		
$c_2$	x	x		x		x	
$c_3$	x		x	x			x

**Abbildung 7: Fehlermeldungstabelle**

Die Kontrollfragen und die Eigenschaften, auf welche sie sich beziehen, können mit einem Venn-Diagramm dargestellt werden (Abbildung 8, links).



**Abbildung 8: Darstellung der Kontrollfragen als Venn-Diagramm**

Mit dem Venn-Diagramm können wir wie folgt die falsche Antwort bestimmen: Wir berechnen den Schnitt der Mengen der Kontrollfragen mit Antwort *nein* und der Komplementärmenge der Kontrollfragen mit Antwort *ja*. Wenn zu den Kontrollfragen  $c_2$  und  $c_3$  *nein* beantwortet wurde und *ja* zu der Kontrollfrage  $c_1$ , können wir auf dem Venn-Diagramm die Eigenschaft mit der falschen Antwort als den Bereich  $\overline{C_1} \cap C_2 \cap C_3$  herauslesen (Abbildung 8, rechts).

Damit das Gesicht gefunden werden kann, kann sich höchstens eine Eigenschaft oder Kontrollfrage in einem Bereich des Venn-Diagramms befinden.

### 3.6 Selbstkorrigierende Kodierung mit 7 Bits

Wie mit “Einmal Lügen” wird zuerst die Binärzahl (4 Bits) geschickt. Die Bits werden in drei Gruppen verteilt, wie oben geschrieben. Die 3 Kontrollbits entsprechen der Parität der entsprechenden Bits. Zum Beispiel können die 3 Kontrollbits wie folgt definiert werden:

1. Parität von Bit 1, 2 und 3
2. Parität von Bit 1, 2 und 4
3. Parität von Bit 1, 3 und 4

Wenn die Parität der Kontrollbits mit der Parität der übermittelten Binärzahl übereinstimmen, entspricht es einer Antwort von *ja* an die Kontrollfrage. Wenn die Parität der Kontrollbits mit der Parität der übermittelten Binärzahl nicht übereinstimmt, entspricht es einer Antwort von *nein*. Die Interpretation der Kontrollbits erfolgt nachher ähnlich zu der Interpretation der Kontrollfragen.

### 3.7 Für die Interessierten: Erweiterung zu mehr als 16 Gesichtern

Wenn wir  $n$  Eigenschaften haben und  $2^n$  mögliche Gesichter haben, wie viele Kontrollfragen benötigen wir, um die Lösung mit 100 %-iger Wahrscheinlichkeit zu bestimmen?

#### 3.7.1 Einschätzung der Anzahl Kontrollfragen

Wir können erst eine untere und eine obere Grenze der Anzahl Kontrollfragen finden. Nachdem wir  $n$  Fragen zu allen  $n$  Eigenschaften gestellt haben, haben wir  $n + 1$  mögliche Gesichter übrig: das Gesicht  $G$ , das den gewählten Eigenschaften entspricht, und  $n$  Gesichter, dessen Codes sich an einer Stelle von diesem Code unterscheiden. Wir können noch eine Frage stellen, um zu überprüfen, ob das Gesicht  $G$  richtig ist: Wenn die Antwort *ja* ist, dann wissen wir, dass  $G$  die Lösung ist. Wenn die Antwort *nein* ist, dann haben wir immer noch  $n + 1$  Möglichkeiten. Wir wissen aber, dass schon einmal gelogen wurde. Wir können deshalb mit der binären Suche die richtige Antwort finden. Es werden noch  $\lceil \log_2(n + 1) \rceil$  Fragen benötigt. Deshalb können wir immer in höchstens  $n + 1 + \lceil \log_2(n + 1) \rceil$  Fragen die richtige Antwort finden. Die Frage, um zu bestimmen, ob  $G$  die richtige Lösung ist, ist nicht immer optimal und für bestimmte Werte von  $n$  kann man die Antwort mit einer Kontrollfrage weniger finden, das heisst in nur  $n + \lceil \log_2(n + 1) \rceil$  Fragen.

#### 3.7.2 Genaue Berechnung der Anzahl Kontrollfragen

Wir können die genaue Anzahl  $m$  von Kontrollfragen wie folgt bestimmen:

Seien  $m$  Kontrollfragen. Die Antworten zu den Kontrollfragen können mit einer  $m$ -stelligen Binärzahl ausgedrückt werden, wobei eine 1 an Stelle  $i$  einer Antwort von *nein*

an die Kontrollfrage  $c_i$  entspricht und eine 0 einer Antwort von *ja* entspricht. Mit  $m$  Kontrollfragen können wir höchstens  $2^m$  verschiedene Zahlen haben, das heisst  $2^m$  Gruppen von Antworten an den  $m$  Kontrollfragen. Die Zahl 0 entsteht, wenn die Antworten zu allen Kontrollfragen *ja* waren. In diesem Fall wurde nicht gelogen und alle Antworten zu den Grundfragen waren wahr. Eine Zahl der Form  $2^i$  entspricht dem Fall, dass nur bei der Kontrollfrage  $c_i$  eine Lüge aufgetreten ist. Auch in diesem Fall waren die Antworten an den Grundfragen wahr. Die anderen Zahlen entstehen, wenn bei mehreren Kontrollfragen *nein* beantwortet wurde. Um eine Lüge in den Grundfragen bestimmen zu können, muss jede der  $n$  Eigenschaften einer eindeutigen Kombination von *nein*-Antworten auf die Kontrollfragen entsprechen.  $m$  Kontrollfragen können deshalb höchstens  $2^m - m - 1$  Eigenschaften überprüfen. Zum Beispiel kann man mit 4 Kontrollfragen höchstens 11 Eigenschaften überprüfen, was 2048 Gesichtern entspricht. Mit 5 Kontrollfragen kann man höchstens 26 Eigenschaften überprüfen, was 67'108'864 Gesichtern entspricht.

## 4 Didaktische Überlegungen

### 4.1 Einführung

Als Wiederholung der binären Suche, Erklärung einer adversarialen Antwortstrategie und zur Einführung des Spiels “Einmal Lügen” stelle ich das folgende Video zur Verfügung:

<https://www.youtube.com/watch?v=g-uK9dJwPLY>

Über das adversariale Spielen genügt es den SuS zu verstehen, dass sie beliebig das gewählte Gesicht verändern können, solange das neue Gesicht mit den schon gegebenen Antworten vereinbar ist.

### 4.2 “Einmal Lügen” mit Wiederholung der Fragen

Damit alle SuS die Fragestellung gut verstehen und damit sie zuerst den Fall eines einzelnen Kontrollbits behandeln, bevor sie die optimalere Version betrachten, ist es wichtig, die SuS zu einer Lösung mit der Wiederholung der Frage zu steuern. Sonst werden wahrscheinlich einige SuS zuerst die kürzeste Lösung erreichen und es wird schwieriger sein, über die einfachere, aber weniger effiziente Lösung und deren verbundenden Konzepte zu sprechen.

Zu diesem Zweck kann die LP erwähnen, dass die SuS probieren können, die Fragen zu wiederholen, bevor sie zum ersten Mal das Spiel versuchen (Auftrag 2).

Bei der Zusammenfassung von Auftrag 2 kann die LP betonen, dass die Lüge sowohl in den ersten 4 Fragen, als auch in den wiederholten Fragen oder in der Kontrollfrage stattfinden können.

### 4.3 Verbindung mit der selbstkorrigierenden Kodierung

Bei der Parallele zur selbstkorrigierenden Kodierung müssen die SuS den Unterschied zwischen adaptiven Fragen und nichtadaptiven Fragen verstehen. Die LP kann eine Parallele mit dem Spiel “Einmal Lügen” machen, indem alle Fragen gleichzeitig gestellt werden müssen, ohne sich auf die Antworten der anderen Fragen beziehen zu können. Die SuS

müssen auch begreifen, dass die gleiche Information in einer Paritätsbit enthalten wird wie in der adaptiven Kontrollfrage.

#### 4.4 Suche der besten Strategie

Auftrag 6 a) macht eine Parallele zur Binärsuche und stellt eine Strategie dar, die nur funktioniert, wenn nicht in den Kontrollfragen gelogen werden darf. Er bringt auch die Idee, nicht einfach die Grundfragen zu wiederholen, aber sie zu kombinieren, um mehr Informationen zu erhalten. In Auftrag 6 b) sind die SuS eingeladen, diese Idee weiter zu forschen und ihre eigenen Strategien zu entwickeln.

Es ist schwierig für eine einzelne LP, mit allen Gruppen über die Strategie, die sie entwickeln haben, zu diskutieren. Dafür wurde eine Webseite erstellt, wo die SuS “Einmal Lügen” spielen und die Strategien bewerten können. Der Skript der Seite verwendet eine adversariale Strategie, um die Fragen zu beantworten : Die Antworten werden jeweils so gewählt, dass die grösste mögliche Anzahl Fragen benötigt wird.

Wenn die SuS auf der Webseite in 7 Zügen zur Lösung gelangen, können sie sicher sein, dass sie eine optimale Strategie gefunden haben.

Die Webseite lautet <http://uebepplan.ch/gyminf/WerBinIch2.html> .

## 5 Ablauf der Sequenz

Hier ist ein möglicher Ablauf der Sequenz:

Zeit	SF	Inhalte	Kommentar	Material
10'	F	Einführung des Spieles “Wer bin ich?” mit einmal Lügen	Das Video dient als Wiederholung der binären Suche und als Einführung des Spieles. Es wird betont von der LP, dass das Spiel adversarial gespielt wird.	Video Teil 1
10'	PA	Auftrag 1: Die SuS spielen “Finde meine Zahl” mit einer Zahl zwischen 1 und 65 und eine adversariale Strategie	Die Aufgabe hat 2 Ziele: Die SuS müssen wahrnehmen, was mit der Binärsuche geschieht, wenn die Anzahl Möglichkeiten ungerade ist und die SuS müssen verstehen, wie genau man diese Art Spiele adversarial spielt.	Blatt Aufträge
5'	PA	Auftrag 2: Die SuS spielen “Einmal lügen” zu zweit, probieren verschiedene Strategien und diskutieren.	Es wird hier nur verlangt, dass die SuS zur richtigen Lösung gelangen, ohne über die Anzahl Fragen zu sprechen.	“Wer bin ich?” Spiel von Abenteuer-Informatik (von Jens Gallenbacher zur Verfügung gestellt), Blatt Aufträge

10'	KG	Anschauen des Videos bis zum Ende. Diskussion der Anzahl Fragen, die benötigt werden, wenn die Fragen wiederholt werden	Ziel ist, dass die SuS bemerken, dass es genügt, jede Frage einmal zu wiederholen, um zu finden wo einen Fehler stattgefunden hat, aber dass wir eine Kontrollfrage benötigen, um ihn zu korrigieren.	Video Teil 2
5'	F	Darstellung der 4 ersten Fragen und übriggebliebenen Möglichkeiten mit einem Entscheidungsbaum und einer Tabelle.	Die SuS müssen merken, dass nach den 4 Grundfragen 5 Möglichkeiten übrigbleiben	Powerpoint
5'	EA	Auftrag 3: Die SuS stellen einen Fall in einem Entscheidungsbaum und in einer Tabelle dar.		Blatt Aufträge
15'		PAUSE		
5'	F KG	Einführung zur selbstkorrigierenden Kodierung. Zu diskutieren: Wenn jede Grundfrage einmal wiederholt wird, wie kann man die Kontrollfrage gestalten, ohne die Antworten zu den anderen Fragen zu kennen?	Der Unterschied zwischen adaptiven und nichtadaptiven Fragen muss erklärt werden. Die Vorkenntnisse über Kontrollbits sollten den SuS mit der Kontrollfrage helfen.	PowerPoint
5'	EA	Auftrag 4: Die SuS wenden das über die selbstkorrigierende Kodierung Gelernte an.		Blatt Aufträge
10'	GA	Auftrag 6: 4-er Gruppen. Suche der besten Strategie. Testen der Strategien mit der Webseite.		Webseite
5'	KG	Eine Gruppe präsentiert ihre Strategie für "Wer bin ich" im Plenum.		

10'	F	Theoretische Grundlage: Fehlermeldungstabelle, Venn-Diagramm, Zusammenhang mit der Binärsuche, Kontrollfragen für die selbstkorrigierende Kodierung	Wenn Zeit vorhanden ist, kann über die Anzahl Fragen im Allgemeinen gesprochen werden und über direkte Anwendungen von "Einmal Lügen"	PowerPoint
5'	F	Auftrag 7: 7-Bits selbstkorrigierende Kodierungen entziffern		
5'	F	Zusammenfassung.		

Sozialformen:

F	Frontal	LV	Lehrervortrag	KG	Klassengespräch
GA	Gruppenarbeit	PA	Partnerarbeit	EA	Einzelarbeit