

Todo list

Struktur vom Dokument erläutern	3
Definition: Entscheidungsbaum	3
Definition: Schaltkreis	3
Definition: Entstellter Schaltkreis	3
Annahme: ehrliche anwender := "handeln nach protokoll"	3
Vision der Anwendung	4
Festlegen, wie die E-Mails ins Programm kommen	4
Festlegen: rechnen mit rationalen Zahlen zwecks Genauigkeit und ein- facherem Parsen spaeter	4
Festlegen, wie das Programm verteilt wird und die Teile kommunizieren .	4
Kurz die Einzelnen phasen der Anwendung beschreiben	4
Einleitung, Verweisen auf Figure für gemeinsame Wortliste	5
Für section-Titel besseren Begriff für "Vorkommnisse der Worte in eigenen Spam/Nicht Spam E-Mails" finden	5
Content	5
Content	5
Content	5
Einleitung, auf Figure für Schwellwerte verweisen	6
Für section-Titel besseren Begriff für "Vorkommnisse der Worte in eigenen Spam/Nicht Spam E-Mails" finden	6
Content	6
Content	6
Content	6
Einleitung, Figure referenzieren	7
Content	7
verteilt ID3 beschreiben	8
Yaos algorithmus grundlegend zusammenfassen (garbled decision table, garbled gate, garbled circuit)	8
Verschlüsselung für die garbled Circuits festlegen (RSA?)	8
1-2 Oblivious Transfer festlegen (mit RSA?)	8
Feststellung der benoetigten Bytezahl beschreiben	8
Schaltkreis designen: Maximum von Summen von positiven Zahlensequen- zen	8
Schaltkreis designen: Gleichheit.	8
Schaltkreis für $x * \log x$ -Protokoll aus dem Paper zusammenfassen	8
Vorghehen zusammenfassen, Schaltkreis aus dominierender Ausgabe wiederver- wenden	8
Einleitung: Wir brauchen ein Programm, was den Klassifikator auf eine MAil oder Mails anwendet	9
Arbeitsweise des Klassifikators erklären	10

Contents

1	Einleitung	3
1.1	Begriffe	3
1.2	Annahmen	3
2	Grundlagen der Anwendung	4
2.1	Form der Benutzereingabe	4
2.2	Interaktion der verteilten Programme	4
2.3	Phasen der Anwendung	4
3	Finden der gemeinsamen Wortliste	5
3.1	Berechnung der Vorkommnisse	5
3.2	Sortierung der Worte nach Informationsheuristik	5
3.3	Syncronisierung der Wortlisten	5
4	Finden der gemeinsamen Schwellwerte	6
4.1	Berechnung der Vorkommnisse	6
4.2	Bestimmung der eigenen Schwellwerte	6
4.3	Syncronisierung der Schwellwerte	6
5	Diskretisieren der eigenen E-Mails	7
6	Lernen der gesamten E-Mails	8
6.1	Yaos Protokoll	8
6.2	Feststellen der dominierenden Ausgabe	8
6.3	Feststellen ob Ausgabe eindeutig	8
6.4	Das Entropien-Protokoll	8
6.5	Attribut mit maximalem Informationsgewinn finden	8
7	Verwenden des Klassifikators	9
7.1	Eingabe des Klassifikators	9
7.2	Arbeitsweise des Klassifikators	10

1 Einleitung

Struktur vom Dokument erläutern

1.1 Begriffe

Definition: Eigenes, Gesamtes

M sei eine Menge von Elementen, die in zwei Teilmengen M_A und M_B zerfällt, sodass $M = M_A \cup M_B$ ist. Wir nehmen desweiteren an, dass Alice M_A kennt, aber weder M noch M_B und dass Bob M_B kennt, aber weder M noch M_A . Dann bezeichnen wir:

- M als **gesamtes** Wissen
- M_A als das **eigene** Wissen von Alice
- M_B als das **eigene** Wissen von Bob
- M_B als das **andere** Wissen von Alice
- M_A als das **andere** Wissen von Bob

Definition: Gemeinsam

Wenn beide Anwender das gleiche Wissen w haben, dann bezeichnen wir w als **gemeinsames Wissen**.

Definition: Vorwissen

Wenn Anwender verschiedene Phasen hintereinander ausführen, dann bezeichnen wir das Wissen aus den bereits ausgeführten Phasen als **Vorwissen**.

Definition: Entscheidungsbaum

Definition: Attribut

Wir definieren eine Menge von Wahrscheinlichkeiten $P = [0, 1] \subset \mathbb{R}$ und eine Menge von Buchstaben $\Sigma = \{a, b, \dots, z, A, B, \dots, Z\}$. Damit definieren wir ein **Attribut** als $\Sigma^+ \times P \times P$. Wenn ein Attribut $A = (w, l, h)$ gegeben ist, bezeichnen wir w als **Wort**, l als **unterer Schwellwert** und h als **oberer Schwellwert**. Es wird desweiteren von allen Attributen gefordert, dass $l \leq h$ ist.

Definition: Schaltkreis

Definition: Entstellter Schaltkreis

1.2 Annahmen

Annahme: ehrliche anwender := "handeln nach protokoll"

2 Grundlagen der Anwendung

Vision der Anwendung

2.1 Form der Benutzereingabe

Festlegen, wie die E-Mails ins Programm kommen

Festlegen: rechnen mit rationalen Zahlen zwecks Genauigkeit und einfacherem Parsen später

2.2 Interaktion der verteilten Programme

Festlegen, wie das Programm verteilt wird und die Teile kommunizieren

2.3 Phasen der Anwendung

Kurz die Einzelnen Phasen der Anwendung beschreiben

3 Finden der gemeinsamen Wortliste

Einleitung, Verweisen auf Figure für gemeinsame Wortliste

Für section-Titel besseren Begriff für "Vorkommnisse der Worte in eigenen Spam/Nicht Spam E-Mails" finden

3.1 Berechnung der Vorkommnisse

Content

3.2 Sortierung der Worte nach Informationsheuristik

Content

3.3 Synchronisierung der Wortlisten

Content

4 Finden der gemeinsamen Schwellwerte

Einleitung, auf Figure für Schwellwerte verweisen

Für section-Titel besseren Begriff für "Vorkommnisse der Worte in eigenen Spam/Nicht Spam E-Mails" finden

4.1 Berechnung der Vorkommnisse

Content

4.2 Bestimmung der eigenen Schwellwerte

Content

4.3 Synchronisierung der Schwellwerte

Content

5 Diskretisieren der eigenen E-Mails

Einleitung, Figure referenzieren

Content

6 Lernen der gesamten E-Mails

verteiltes ID3 beschreiben

6.1 Yaos Protokoll

Yaos algorithmus grundlegend zusammenfassen (garbled decision table, garbled gate, garbled circuit)

Verschlüsselung für die garbled Circuits festlegen (RSA?)

1-2 Oblivious Transfer festlegen (mit RSA?)

Feststellung der benötigten Bytezahl beschreiben

6.2 Feststellen der dominierenden Ausgabe

Schaltkreis designen: Maximum von Summen von positiven Zahlensequenzen

6.3 Feststellen ob Ausgabe eindeutig

Schaltkreis designen: Gleichheit.

6.4 Das Entropien-Protokoll

Schaltkreis für $x * \log x$ -Protokoll aus dem Paper zusammenfassen

6.5 Attribut mit maximalem Informationsgewinn finden

Vorghehen zusammenfassen, Schaltkreis aus dominierender Ausgabe wiederverwenden

7 Verwenden des Klassifikators

Einleitung: Wir brauchen ein Programm, was den Klassifikator auf eine MMail oder Mails anwendet

7.1 Eingabe des Klassifikators

Ausgehend von den Definitionen eines Attributes und eines Entscheidungsbaumes kann leicht eine Grammatik erzeugt werden, welche die Form des eingegebenen Klassifikators eindeutig bestimmt. Wir notieren diese Grammatik in BNF. Diese ist so gewählt, dass sie eine LL(1)-Grammatik ist, d.h., sie ist besonders einfach zu parsen.

```
tree -> 'Decide' '(' attribute '(' ',' tree)+ ')'
      | 'Output' '(' class ')'.
class -> 'Spam' | 'Not Spam'.
attribute -> '(' word ',' probability ',' probability ')'.
word -> ('a' | 'b' | ... | 'z' | 'A' | ... | 'Z')+
probability -> number '/' number .
number -> ('0' | '1' | ... | '9')+.
```

Somit wäre ein kodierter Klassifikationsbaum beispielsweise:

```
Decide((Foo, 1/3, 2/3),
      Output(Spam),
      Decide((Bar, 1/2, 2/3),
            Output(Spam),
            Output(Not Spam)
          )
    )
```

Da die Wahrscheinlichkeiten in den Attributen berechnet werden als Verhältnis der Anzahl der E-Mails mit einer bestimmten Eigenschaft zu der Gesamtanzahl aller E-Mails, ist die Darstellung einer Wahrscheinlichkeit in einem Attribut durch ein Paar von natürlichen Zahlen ausreichend, um alle möglichen Wahrscheinlichkeiten darzustellen. Da die Buchstabenmenge als die lateinischen Klein- und Grossbuchstaben definiert ist und ein Wort definiert ist als Sequenz dieser Zeichen, ist die Produktion „word” ausreichend, um alle möglichen Worte darzustellen. Damit ergibt sich, dass die Produktion „attribute” in der Lage ist, alle Attribute, die in dieser Anwendung auftreten können, darzustellen.

Es gibt desweiteren bei uns nur die Ausgaben „Spam” und „Nicht Spam”. Damit ist die Produktion „class” ausreichend, um alle möglichen Ausgaben des Baumes darzustellen. Daraus folgt, dass die zweite Produktion von tree in der Lage ist, alle möglichen Blätter darzustellen. Desweiteren folgt induktiv aus der Vollständigkeit der Produktion „attribute” und der Darstellbarkeit der Blätter als Induktionsanfang, dass die erste Produktion von tree in der Lage ist,

alle möglichen auszugebenden Entscheidungsbäume darzustellen. Damit ist die Grammatik mächtig genug für unsere Zwecke.

Es ist weiterhin zu bemerken, dass eine semantische Validierung notwendig ist, da in der Grammatik weder gefordert ist, dass der untere Schwellwert eines Attributes wirklich kleiner ist als der obere Schwellwert eines Attributes, noch dass die Wahrscheinlichkeiten wohldefiniert sind, d.h., dass der Nenner ungleich 0 ist.

7.2 Arbeitsweise des Klassifikators

Arbeitsweise des Klassifikators erklären

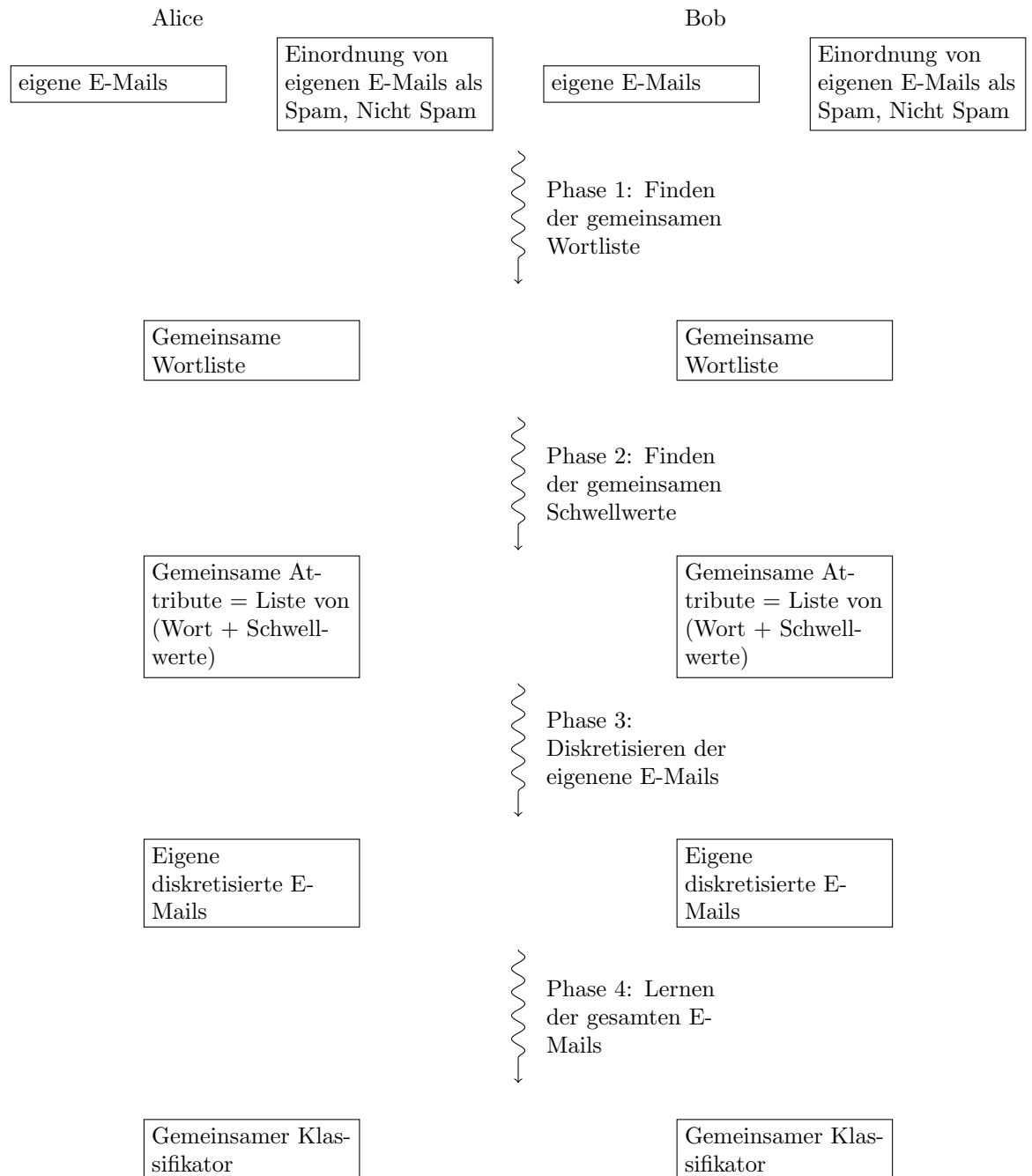


Figure 1: Phasen der Anwendung

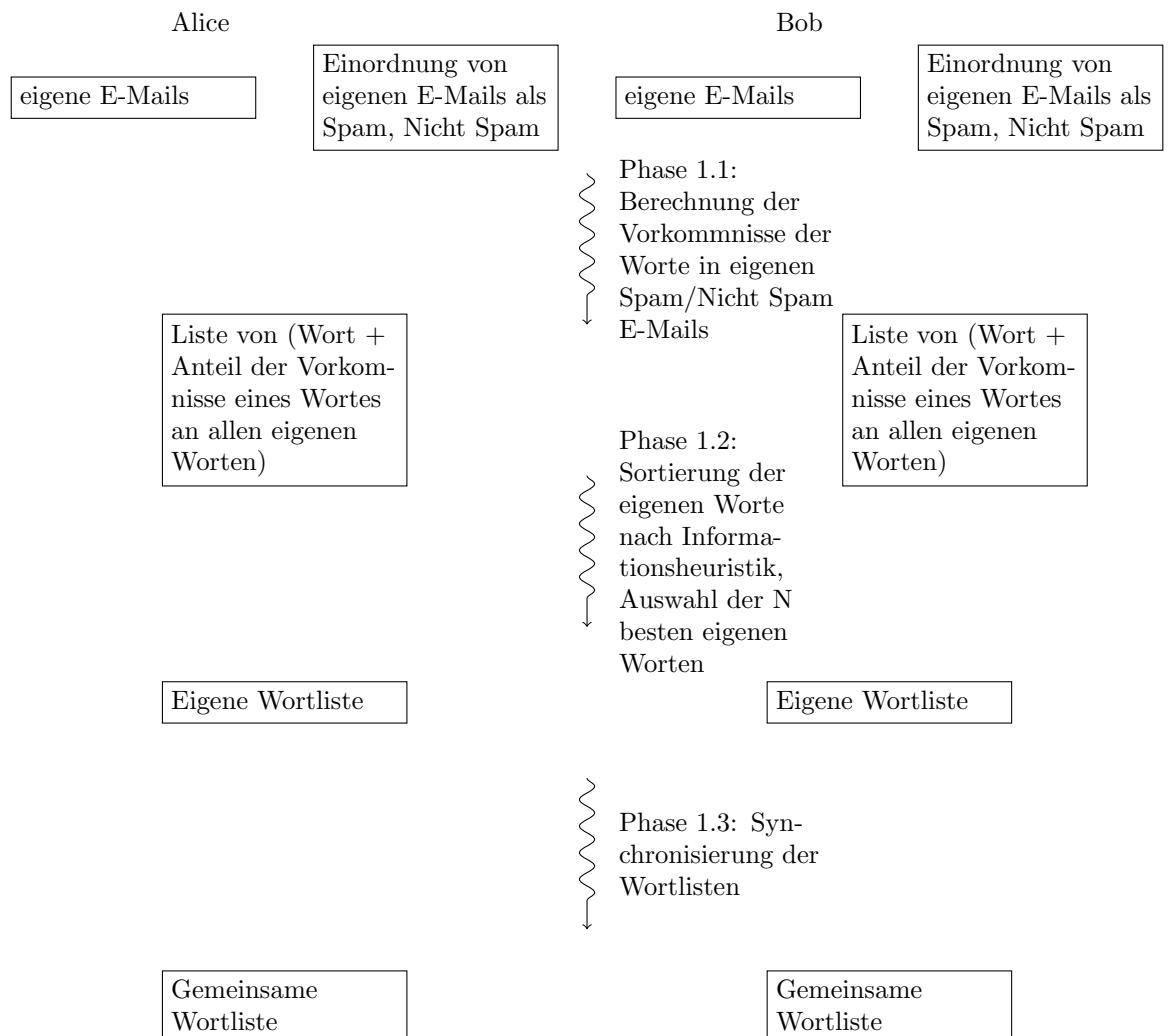
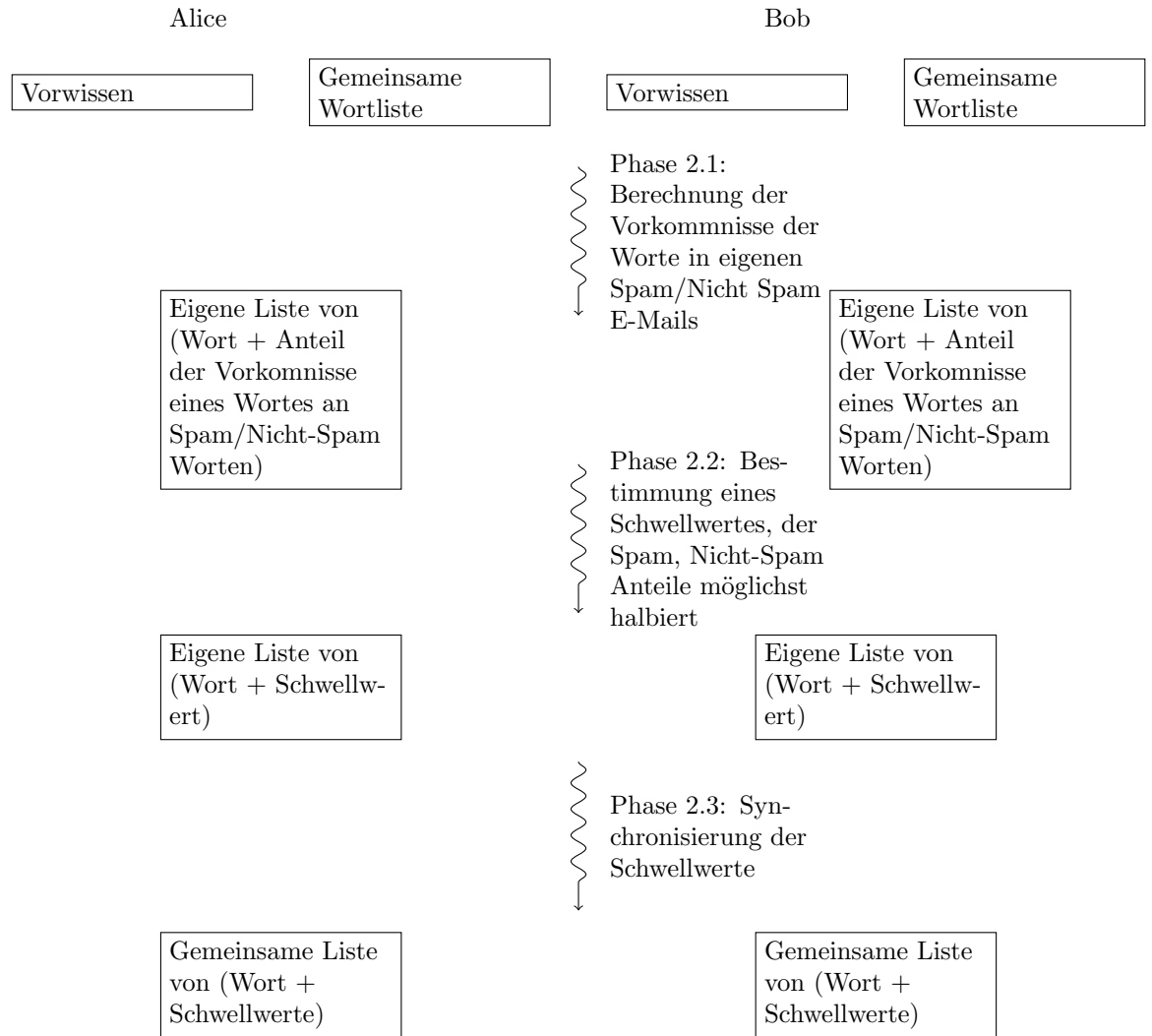
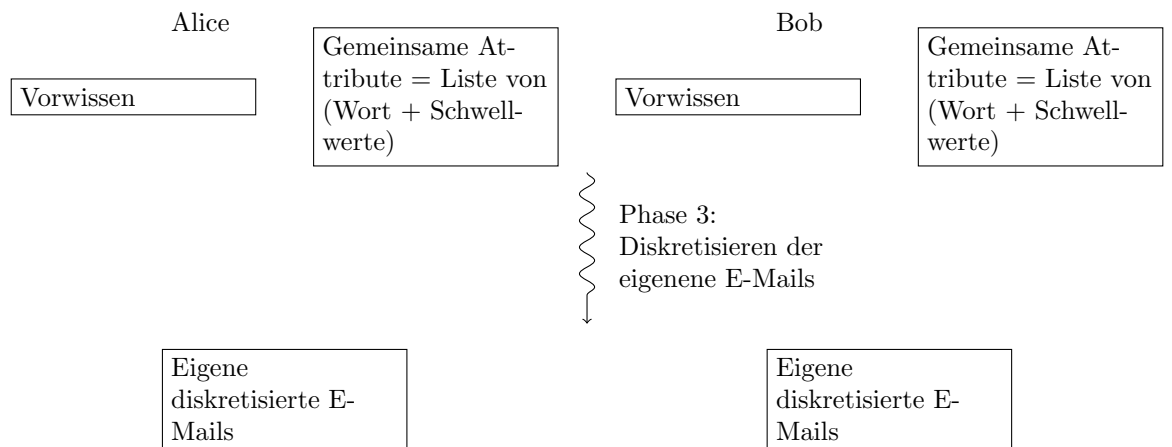


Figure 2: Schritte zum Berechnen der gemeinsamen Wortliste





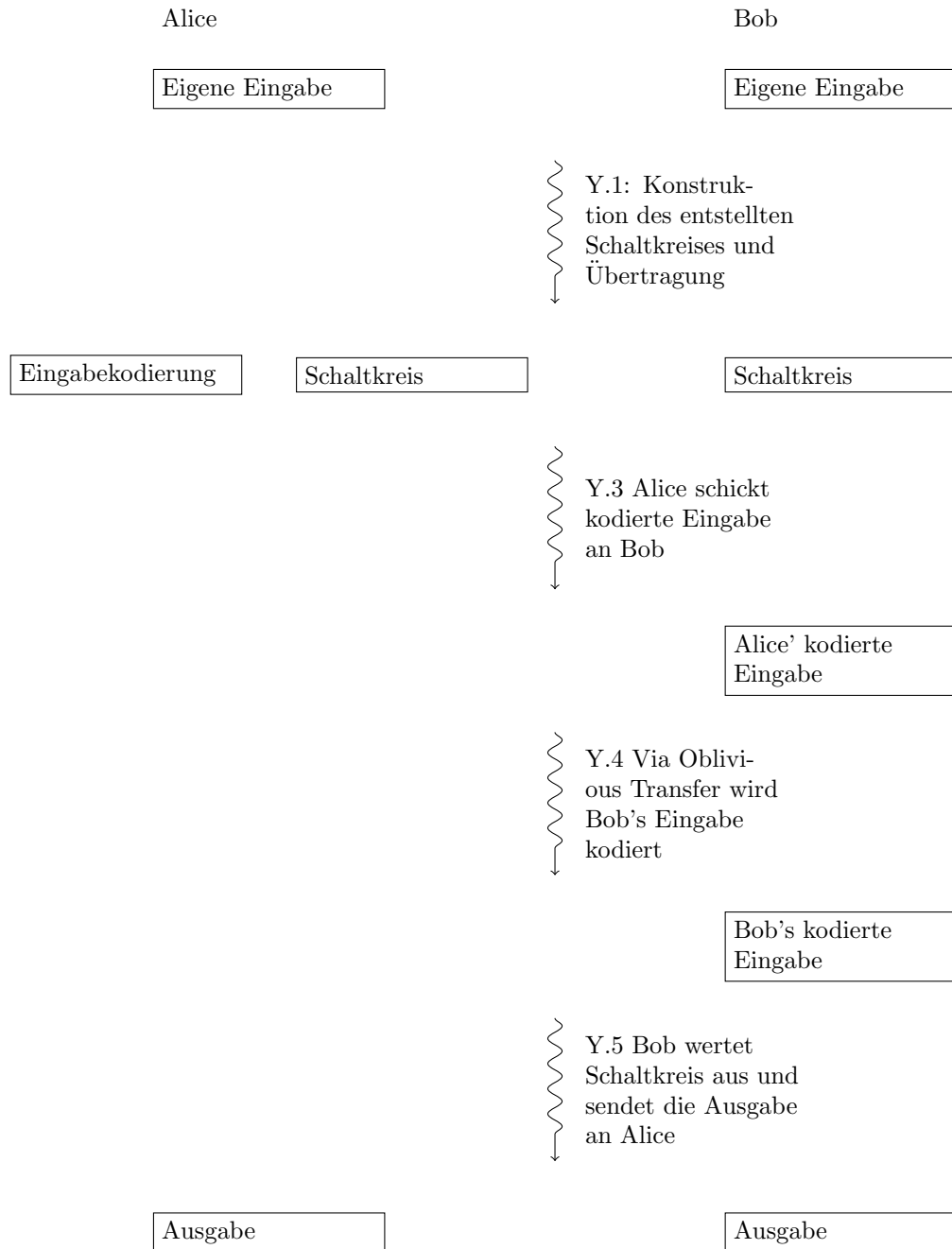


Figure 3: Yao's Algorithmus

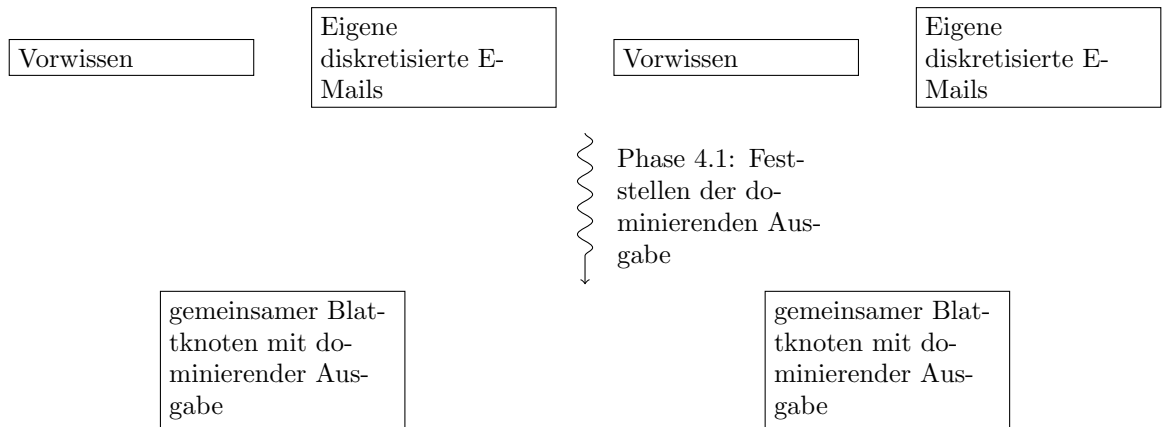


Figure 4: ID3-Algorithmus, Fall 1: Keine Attribute mehr vorhanden

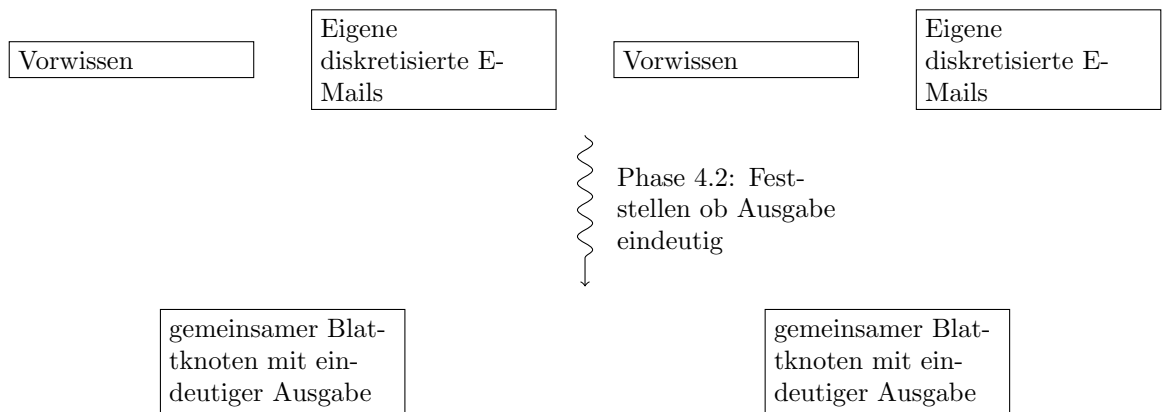


Figure 5: ID3-Algorithmus, Fall 2: Ausgabe eindeutig

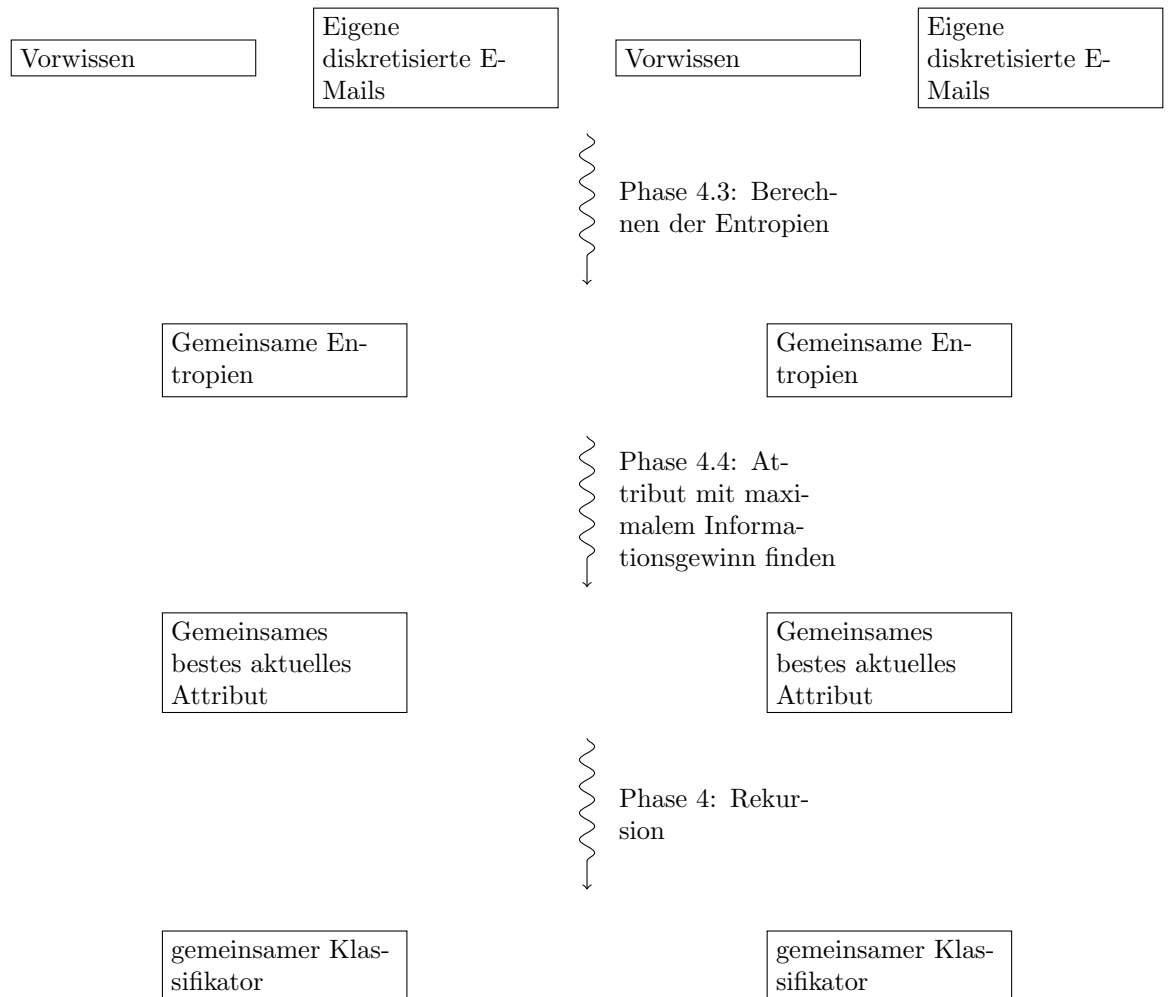


Figure 6: ID3-Algorithmus, Fall 3: Erzeugung eines Astes