

4. Evolutionäre Algorithmen

- Idee: Lernverfahren bzw. Optimierungsstrategien aus der Evolutionstheorie nachbilden.
- Hierzu müssen
 - das betrachtete Problem entsprechend kodiert,
 - Bewertungskriterien für gefundene Lösungen definiert,
 - sowie evolutionäre Operatoren wie
 - ◆ Mutation
 - ◆ Crossover
 - ◆ Selektiondefiniert werden.

Genetische Programmierung

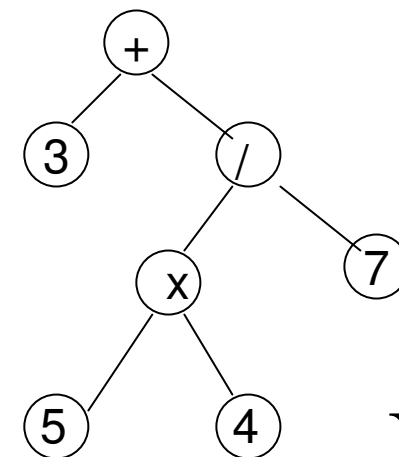
Bei der genetische Programmierung werden funktionale Programme (z.B. basierend auf Lisp-Funktionen) entwickelt.

Beispiel: Lernen algebraische Ausdrücke

■ Kodierung durch Bäume

- Interne Knoten: Funktionen, Prädikate,...
- Blätter: Konstanten, Aktionen,...

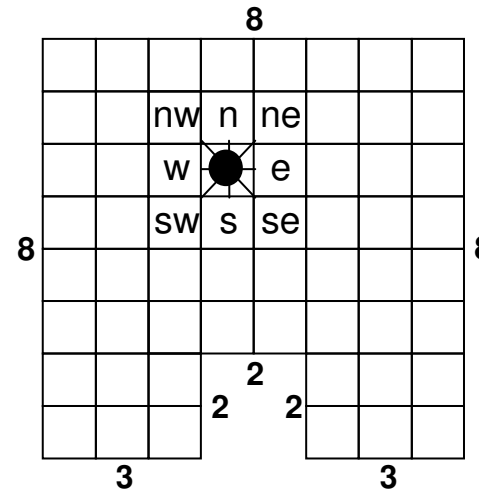
■ Beispiel: $3+(5 \times 4)/7$



Beispiel: Wandverfolgung

$$s_1 = nw, s_2 = n, \dots, s_8 = w$$

$$s_i = 0 \Leftrightarrow \text{Zelle } i \text{ frei}$$



Gesucht:

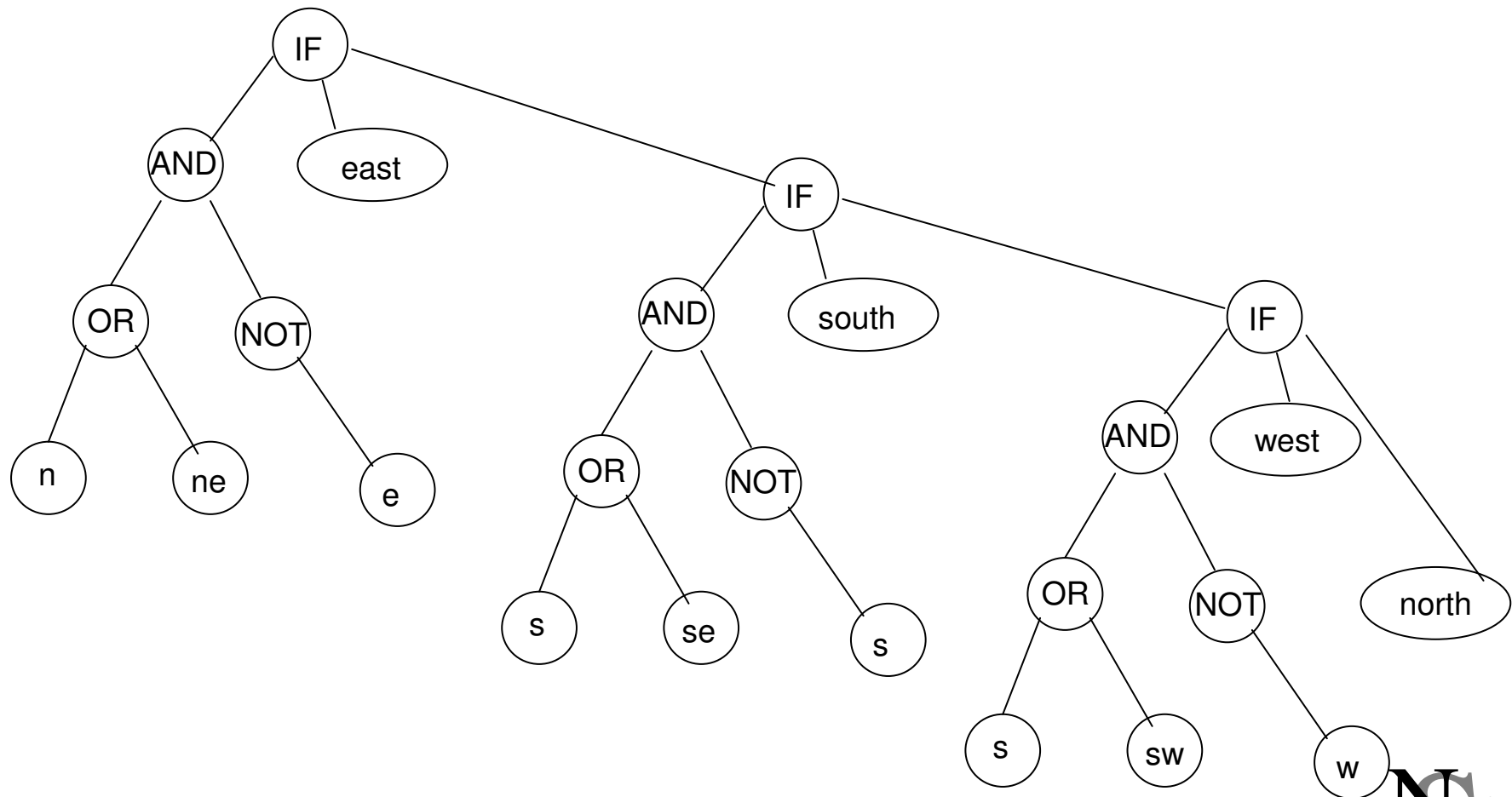
Ein Programm, das die benachbarten Zellen inspiziert und eine Aktion $\{0,1\}^8 \rightarrow \{\text{north}, \dots, \text{west}\}$ durchföhrt. Mit der mehrfachen Ausführung des Programms soll der Roboter zu einer Wand gehen und ihr dann folgen.

Wandverfolgung (II)

- Einfache Funktion im Programm:
 - $\text{AND}(x,y) = 0$ if $x = 0$; else y
 - $\text{OR}(x,y) = 1$ if $x = 1$; else y
 - $\text{NOT}(x) = 0$ if $x = 1$; else 1
 - $\text{IF}(x,y,z) = y$ if $x = 1$; else z
- Aktionen: north, east, south, west.
- Die Aktionen haben keinen Einfluss auf die Baumstruktur des Programms. Eine Aktion wird beendet, wenn der Roboter versucht in eine Wand zu laufen.
- Beim Genetischen Programmieren müssen alle Ausdrücke in dem Programm Werte für alle möglichen Argumente liefern (es sei denn des Programm wird durch die Bearbeitung eines Ausdrucks beendet).

Wandverfolgung: Repräsentation

■ Baumdiagramm:



Wandverfolgung: Repräsentation

■ Funktional:

```
(IF (AND (OR (n) (ne)) (NOT (e)))  
    (east)  
    (IF (AND (OR (e) (se)) (NOT (s)))  
        (south)  
        (IF (AND (OR (s) (sw)) (NOT (W)))  
            (west)  
            (north))))))
```

Wandverfolgung: Repräsentation

■ Imperativ:

```
if (n or ne) and not e
  then east
  else if (e or se) and not s
    then south
    else if (s or sw) and not w
      then west
      else north
    fi
  fi
fi
```

Genetische Programmierung

Grundstruktur eines Programms:

```
begin
  t:=0;
  initialize(P(t)); // Bestimme Anfangspopulation
  evaluate(P(t));   // Bewerte die Population
  while (not Abbruchkriterium(P(t),t)) do
    t:=t+1;
    select P(t) from P(t-1);
    crossover(P(t));
    mutate(P(t));
    evaluate(P(t));
  end;
end;
```

Beispiel: Wandverfolgung

■ Generation 0:

- Population von zufällig erzeugten Programmen mit Funktionen, Konstanten, Sensoreingaben

- Hier:

5000 Programme, die aus

- ♦ and, or, not, if,
- ♦ nw,..., w,
- ♦ north,..., west,
- ♦ 0, 1

zusammengesetzt werden.

Beispiel eines initialen Programms

```
(IF (OR (NOT (AND (IF (sw) (north) (ne))
                  (AND (south) (1)))))
  (OR (OR (NOT (s))
          (NOR (e) (e)))
    (AND (IF (west) (ne) (se))
          (IF (1) (e) (e)))))
(OR (NOT (AND (NOT (ne)) (IF (east) (s) (n))))
  (OR (NOT (IF (ne) (east) (s)))
    (AND (IF (w) (sw) (1))
          (OR (sw) (nw)))))
(OR (NOT (IF (OR (n) (w))
              (OR (0) (se))
              (OR (1) (east)))))
  (OR (AND (OR (1) (ne))
          (AND (nw) (east)))
    (IF (NOT (west))
      (AND (west) (east))
      (IF (1) (north) (w))))))
```

Beispiel: Wandverfolgung

Die Programme werden nach ihrer Güte bewertet

■ Hier:

- Jedes Programm läuft 60 mal (d.h. es kann maximal 60 Schritte zurücklegen)
- Die Zahl der direkt neben einer Wand liegenden vom Programm besuchten Zellen wird gezählt (jede Zelle nur einmal): $0 \leq \textit{Güte} \leq 32$
- Jedes Programm wird an 10 zufällig ausgesuchten Orten gestartet.
- Die *Fitness* eines Programms ist die Summe der jeweiligen Güten: $0 < \textit{Fitness} < 320$

Konstruktion neuer Generationen

- Konstruktion der Generation $i+1$ aus der Generation i durch:
 - a) Direkte Übernahme von x % der Programme aus Generation i unter Berücksichtigung der Fitness.
 - b) Die restlichen Kindprogramme werden durch *Crossover* aus zwei Elternprogrammen der Generation i erzeugt.
 - c) z % der Programme werden zufällig verändert (*Mutation*)

Beispiel: Wandverfolgung

Konstruktion der Generation $i+1$:

- a) **500 Programme (10%) aus der Generation i werden direkt in Generation $i+1$ durch einen Wettbewerbsprozess übernommen:**

Wiederhole 500 mal

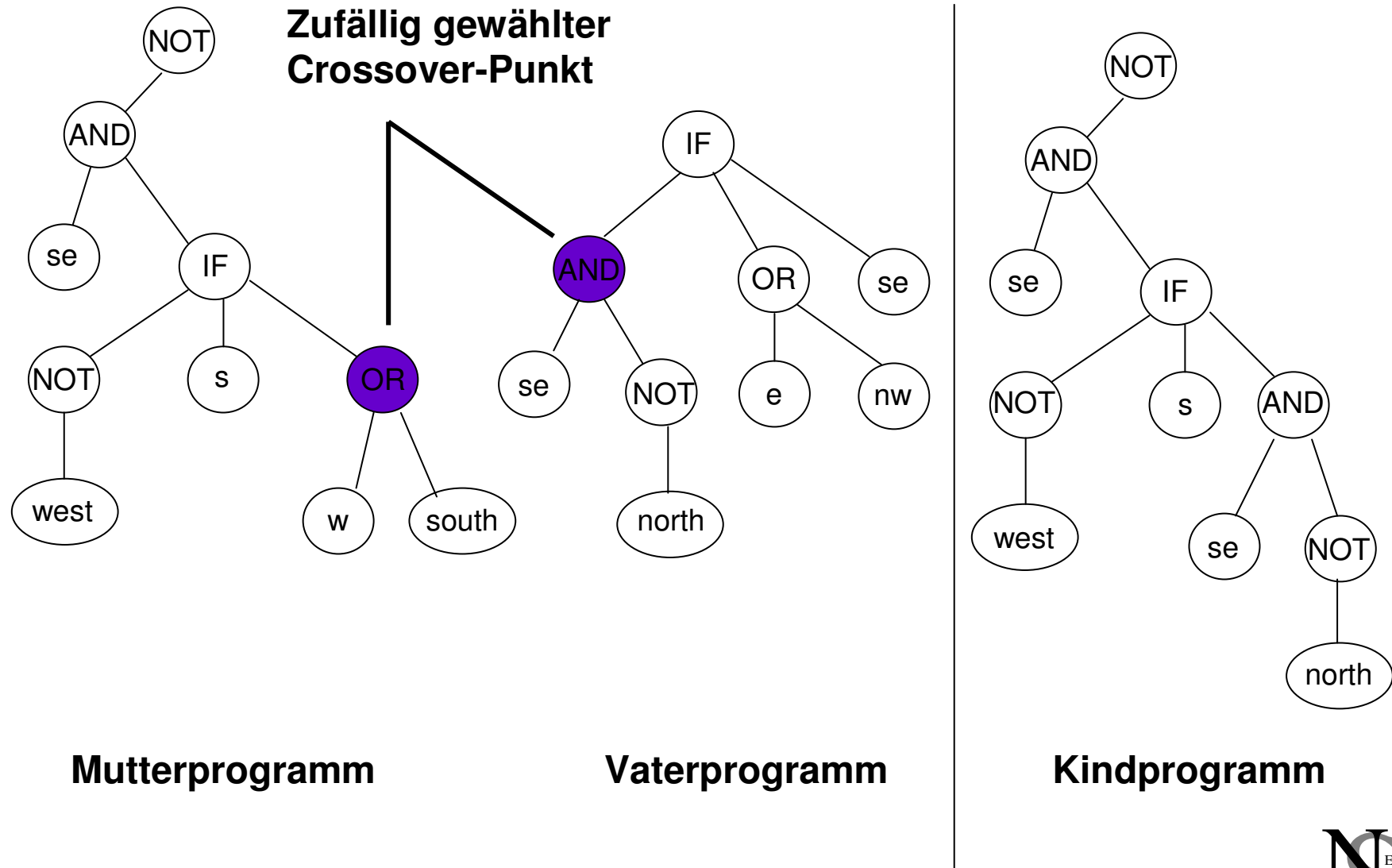
- 7 Programme werden zufällig ausgewählt (Ziehen mit zurücklegen)
- das „beste“ Programm wird übernommen („survival of the fittest“)

Beispiel: Wandverfolgung

Konstruktion der Generation $i+1$:

- b) **4500 Kind-Programme werden durch *Crossover* aus zwei Elternprogrammen der Generation i erzeugt:**
- **Zufällige Wettbewerbsauswahl des Vater und Mutter Programms (wie in a)**
 - **Zufällig ausgewählter Teilbaum des Vaters wird durch einen zufällig ausgewählten Teilbaum der Mutter ausgetauscht.**

Crossover



Beispiel: Wandverfolgung

Konstruktion der Generation $i+1$:

- c) **Manchmal wird noch ein *Mutations*-Operator eingeführt, der ca. 1% der Nachkommen betrifft. Dabei wird ein Teilgraph weggelassen und durch einen zufällig erzeugten Teilgraph ersetzt (in unserem Beispiel nicht!)**

Beispiel: Wandverfolgung

Fitness der Generation 0:

Die meisten der 5000 Programme sind nutzlos, z.B.

- **(AND (sw) (ne)) wertet nur aus und terminiert dann: Fitness 0**
- **(OR (e) (west)) liefert manchmal west und geht somit einen Schritt nach Westen, landet manchmal neben einer Wand: Fitness 5**
- **Das beste Programm hat die Fitness 92. Es ist schwer zu lesen, hat redundante Operatoren.
(Es läuft nach Osten bis zu einer Wand, dann nach Norden bis es nach Osten oder Westen kann und dann in Ecke oben links gefangen ist.)**

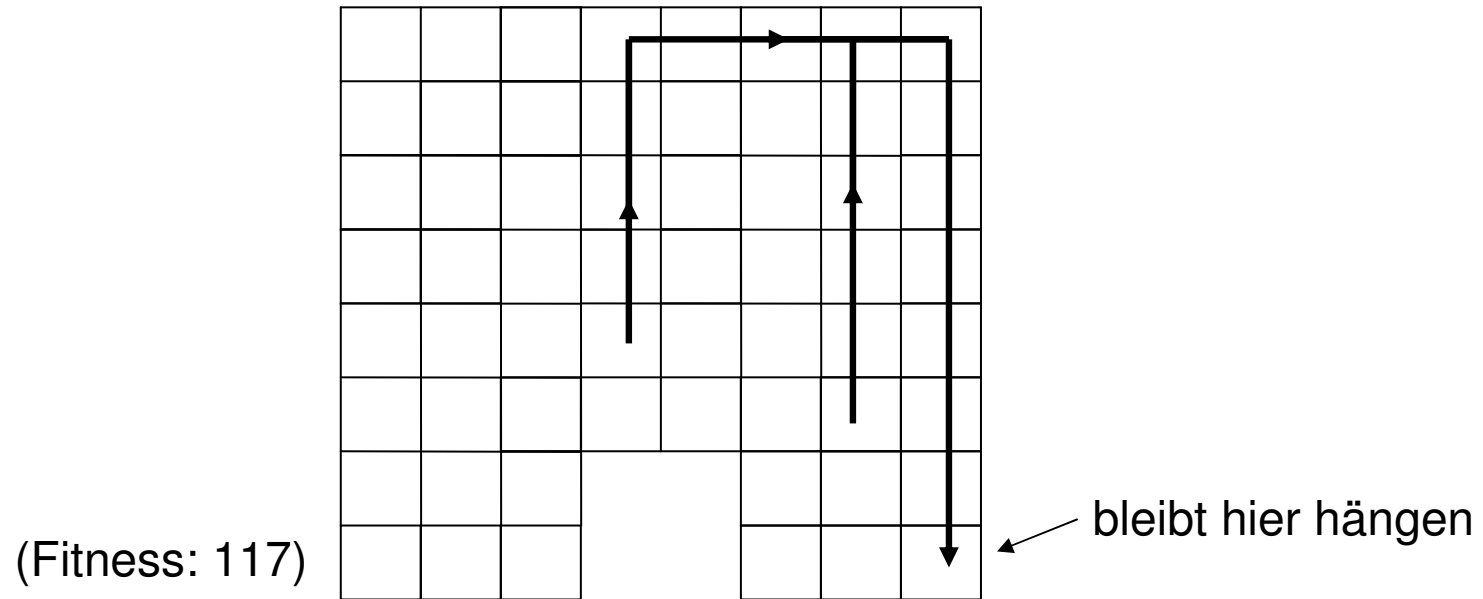


...

Bestes Individuum der Generation 0

```
...      (OR (OR (NOT (s))
                (NOR (e) (e)))
          (AND (IF (west) (ne) (se)
                  (IF (1) (e) (e))))))
(OR (NOT (AND (NOT (ne)) (IF (east) (s) (n))))
    (OR (NOT (IF (ne) (east) (s)))
        (AND (IF (w) (sw) (1))
              (OR (sw) (nw))))))
(OR (NOT (IF (OR (n) (w))
              (OR (0) (se))
              (OR (1) (east)))))
(OR (AND (OR (1) (ne))
        (AND (nw) (east)))
    (IF (NOT (west))
        (AND (west) (east))
        (IF (1) (north) (w))))))
```

Bestes Individuum der Generation 2



(NOT (AND (IF (ne)

(IF (se) (south) (east))

(north))

(NOT (NOT (e))))

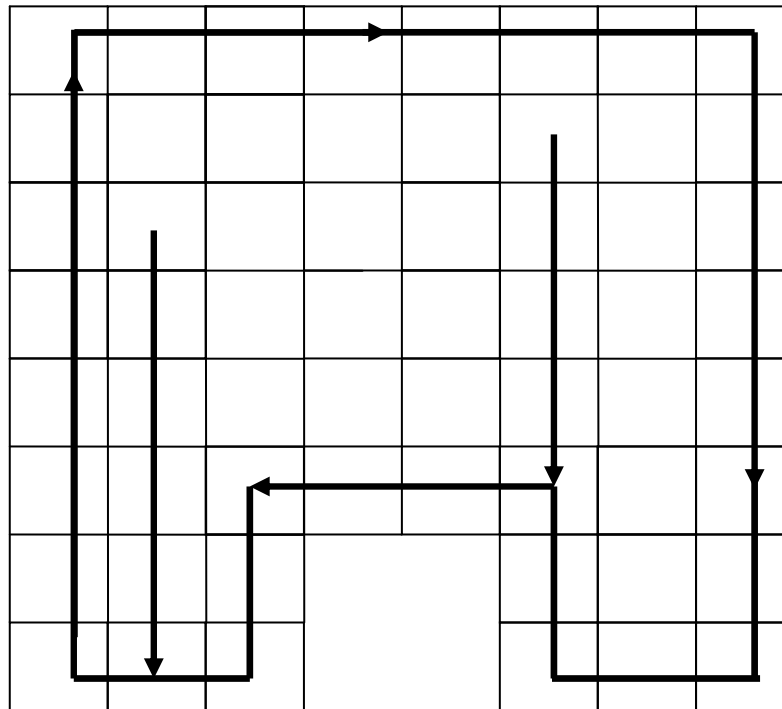


Bestes Individuum der Generation 6

```
(IF (AND (NOT (ne))
          (IF (e) (s) (sw))))
(OR (IF (1) (e) (south))
     (IF (north) (east) (nw)))
(IF (OR (AND (0) (north))
        (AND (e) (IF (e)
                      (IF (se) (south) (east))
                      (north)))))
(AND (e)
      (NOT (IF (s) (sw) (e)))))
(OR (OR (AND (nw) (east))
        (west))
     (nw))))
```

Bestes Individuum der Generation 10

(Fitness: 320)



Bestes Individuum der Generation 10

```
(IF (IF (IF (se) (0) (ne))
      (OR (se) (east))
      (IF (OR (AND (e) (0))
              (sw))
          (OR (sw) (0))
          (AND (NOT (NOT (AND (s) (se))))
              (se))))
  (IF (w)
      (OR (north)
          (NOT (NOT (s))))
      (west))
  (NOT (NOT (NOT (AND (IF (NOT (south))
                        (se)
                        (w))
                    (NOT (n)))))))
```

Beispiel: Wandverfolgung

- Entwicklung der Fitness im Laufe des Lernvorgangs (bestes Individuum der jeweiligen Generation)

