Lösungen zum Buch: Wissensverarbeitung

Kapitel 4

Evolutionäre Algorithmen

Lösung 4.1 (Evolution)

a) *Beispiel 1: Röcke*

- *Genotyp: Schnittmuster, welches vom Designer an die Schneiderei übergeben wird*
- *Phänotyp: Rock, der getragen wird*
- *genetische Operatoren: 1) Mutation, z. B. Ändern von analogen Parametern, wie Rocklänge, Durchsichtigkeit oder Ändern diskreter Parameter, wie Material, Schlitz, Knopfanzahl 2) Crossover, z. B. Kombination von Ideen anderer Designer*
- *Fitness: Echo in der Modepresse, Verkaufserfolg (bedingt u.a. durch Eigenschaften des Phänotyps, wie Haltbarkeit, Tragekomfort etc.)*

Beispiel 2: Schnittstellen von Festplatten

- *Genotyp: Dokument mit Beschreibung der technischen Ausführung und der Protokolle*
- *Phänotyp: reale Schnittstelle zur Festplatte in einem Computer*
- *genetische Operatoren: 1) Mutation, z. B. Ändern von Befehlssätzen und Taktraten 2) Crossover, z. B. Kombination von Ideen, z.B. seriell und extern*
- *Fitness: Datendurchsatz, Zuverlässigkeit, Alleinstellungsmerkmale, Entwicklungsaufwand der Controller*

Beispiel 3: Tanzmusik

- *Genotyp: Partitur und Interpret*
- *Phänotyp: Schallwellen*
- *genetische Operatoren: 1) Mutation, z. B. Ändern von Texten, Einfügen von Strophen, Covern 2) Crossover, z. B. Kombination von Themen, Hitmixes, Medley*
- *Fitness: Tanzbarkeit, eingängige Melodie, Marketingmöglichkeiten*

b) *Die Partnerwahl beeinflusst die Konkurrenzfähigkeit der eigenen Nachkommen, so dass Individuen, die den Partner bewusst oder unbewusst nach bestimmten Merkmalen auswählen, Evolutionsvorteile erlangen können. Die Vorteile liegen bspw. im Vermeiden von Erbkrankheiten, höherer Überlebenswahrscheinlichkeit durch Gesundheit, Intelligenz u.v.a. Es wird angenommen, dass die Fähigkeit zum Erkennen eines geeigneten Partners so wesentlich ist, dass sie im Laufe der Evolution genetisch assimiliert wurde und damit den Lebewesen als Instinkt zur Verfügung steht. Die Fähigkeit, die Attraktivität eines potentiellen Partners - seine Schönheit - zu beurteilen, ist also evolutionär entstanden.*

c) *Vermehrung, die Vermehrungsrate beeinflussende Veränderungen, Vererbbarkeit dieser Veränderungen, begrenzte Ressourcen*

d) *Lamarckismus: Ursprüngliche Giraffen mit kurzen Hälsen strecken sich nach den oberen Blättern der Bäume und dehnen dabei ihre Muskeln und Knochen. Ihr Hals wird durch die tägliche Anstrengung dauerhaft etwas länger und diese zusätzliche Länge wird an die Nachkommen vererbt. Diese starten nun mit einem längeren Hals und strecken sich ebenfalls. Die kumulierten Verlängerungen führen zum langen Hals der Giraffen.*

Darwinismus: Die einzelnen Giraffen verfügen durch Variationen im Erbmaterial über individuelle Merkmale, z. B. kürzere oder längere Hälse. Die Individuen mit längeren Hälsen haben größere Chancen im Konkurrenzkampf, denn sie erreichen noch Nahrung, wenn Kurzhalsige schon hungern. Sie werden also mehr Nachkommen hervorbringen, so dass der Anteil der längeren Hälse in der nächsten Generation steigt. Treten noch längere Hälse auf, wiederholt sich der Prozess.

Kausalkette Lamarckismus: Bedürfnis → Aktion des Individuums → geeignete Veränderung → Vererbung der Veränderung

Kausalkette Darwinismus: Spontane Veränderungen im Erbgut → Unterschiede → Vorteile bei der Selektion → mehr Nachkommen

Lösung 4.2 (Fitness)

a) *Problematische Eigenschaften sind: Rauschen, häufige lokale Minima, Plateaus.*

b) *Anwendungsbeispiele:*

- *interaktives Komponieren*
- *interaktives Design (Architektur, Möbel, Kleidung, Beleuchtung, 3D-Modelle)*
- *interaktives Mischen (Parfüm)*
- *interaktives Layout (CSS, Poster)*
- *interaktive Phantombilder*

Die Beispiele lassen sich auf die gemeinsame Vergabe von Fitnesspunkten durch mehrere Personen verallgemeinern.

Anwendungssindizien („Wie es Euch gefällt“):

- *Leicht fassbare Darstellung des Genotypen möglich, z. B. als Bild, Bewegung, Ton*
- *Fitnessfunktion enthält schwierig oder nicht berechenbare Anteile, die von speziellen menschlichen Eigenschaften (z. B. Körperlichkeit, Sinne, Ästhetikempfinden, Intuition) abhängen*

Lösung 4.3 (Rucksackproblem)

a) *Sei M die Menge aller Waren und n die Anzahl ihrer Elemente, dann können wir das Vorhandensein einer Ware a im Rucksack durch ein Bit x_a ausdrücken. Eine konkrete Belegung des Rucksacks ist dann ein Bitvektor $x = (x_1, .., x_n)$ mit $x_a = 1$, wenn die Ware im Rucksack ist und $x_a = 0$ sonst.*

b) *Der Wert der eingepackten Waren stellt die Fitness dar:*

$$f(x) = w(x) = \sum_{a=1}^{n} w(a) \cdot x_a.$$

Die Nebenbedingung des Maximalgewichtes G kann als Fitness mit dem Wert 0 formuliert werden:

$$f(x) = 0, \text{ wenn } g(x) = \sum_{i=1}^{n} g(a) \cdot x_a > G$$

.

c) *Die folgende (logistische) Funktion $f_g(x)$ ist für erlaubte Gewichte $g(x) < G$ ungefähr 1 und geht an der Grenze $g(x) = G$ und darüber hinaus stetig in Werte nahe Null über:*

$$f_g(x) = 1 - \frac{1}{1 + e^{-g(x)+G}}$$

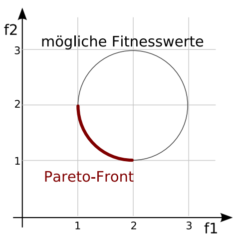
Damit können wir die Nebenbedingung $g(x) \leq G$ als zweites Optimierungsziel in die Fitnessfunktion aufnehmen: $f_g(x)$ und $w(x)$ sind gleichzeitig zu maximieren.

d) *3 Varianten: Pareto-Optimierung, gewichtete Summe von $f_g(x)$ und $w(x)$, zufällige Anwendung von $f_g(x)$ oder $w(x)$.*

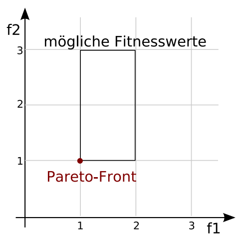
e)
- *Tourenplanung Wachschutz: kurze Wege, Einhalten der Kontrollfenster und Arbeitszeiten*
- *Aufbau einer Regelmenge: wenig Regeln, hohe Genauigkeit*
- *Verstärker-Schaltung: Verstärkung, Stromverbrauch, Bandbreite*
- *Auftragsreihenfolgeplanung: durchschnittliche und maximale Dauer*

Lösung 4.4 (Pareto-Front)

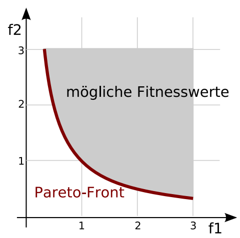
a)



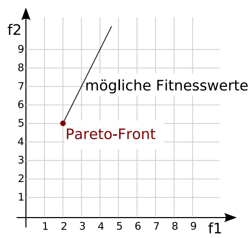
b)



c)



d)



a) $P_{Front} = \{(f_1, f_2) \mid f_2 = 2 - \sqrt{1 - (f_1 - 2)^2},\ 1 \leq f_1 \leq 2\}$

b) $P_{Front} = \{(1, 1)\}$

c) $P_{Front} = \{(f_1, f_2) \mid f_2 = 1/f_1,\ f_1 > 0\}$

d) $P_{Front} = \{(2, 5)\}$

e) *Die Selektion sollte Individuen der Pareto-Front bevorzugen, die sich an dünn besetzten Stellen des Fitnessraumes befinden. Eine Variante ist das Schätzen der Dichte der Individuen, z. B. durch Bestimmen des Abstandes zur benachbarten Lösung im Raum der Fitnesswerte.*

Lösung 4.5 (Kodierung GA)

a) *Binärcode (Dualzahl):* $47_{10} = 2^5 + 2^3 + 2^2 + 2^1 + 2^0 = 101111_2$

$$\begin{array}{lcccccc} \textit{Binärcode} & 1 & 0 & 1 & 1 & 1 & 1 \\ \textit{zu} & \downarrow\searrow & \downarrow\searrow & \downarrow\searrow & \downarrow\searrow & \downarrow\searrow & \downarrow \\ \textit{Graycode:} & 1 & 1 & 1 & 0 & 0 & 0 \end{array}$$

b) *Als Binärcode (Dualzahl):* $1010101_2 = 2^0 + 2^2 + 2^4 + 2^6 = 85_{10}$

Als Graycode: Schritt 1 - Umwandeln in Binärzahl:

Variante1: Berechnung des i-ten Bits x_i^{bin} des Binärcodes aus den i linken Bits des Graycodes:

$$x_i^{bin} = \bigoplus_{j=1}^{i} x_j^{Gray}, \quad \textit{(i beginnt links mit 1)}$$

$$\begin{array}{lll} x_1^{bin} & & = 1 \\ x_2^{bin} = & 1 \oplus 0 & = 1 \\ x_3^{bin} = & 1 \oplus 0 \oplus 1 & = 0 \\ x_4^{bin} = & 1 \oplus 0 \oplus 1 \oplus 0 & = 0 \\ x_5^{bin} = & 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 & = 1 \\ x_6^{bin} = & 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 & = 1 \\ x_7^{bin} = & 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 & = 0 \end{array}$$

Variante 2: Rekursiv unter Nutzung schon berechneter x_i:

$$x_i^{bin} = \begin{cases} x_i^{Gray} & \textit{, wenn } i = 1, \\ x_{i-1}^{bin} \oplus x_i^{Gray} & \textit{sonst.} \end{cases}$$

$$\begin{array}{lccccccccccccc} \textit{Graycode} & 1 & & 0 & & 1 & & 0 & & 1 & & 0 & & 1 \\ \textit{zu} & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \textit{Binärcode:} & 1 & \rightarrow & 1 & \rightarrow & 0 & \rightarrow & 0 & \rightarrow & 1 & \rightarrow & 1 & \rightarrow & 0 \end{array}$$

Schritt 2 - Umwandeln der Binärzahl in Dezimalzahl:

$1100110_2 = 2^6 + 2^5 + 2^2 + 2^1 = 102_{10}$

c) *1. Tausch der Mittelstücke:*

$$\begin{array}{lcl} v_1 = (1, 2, |3, 4, 5, 6, |7, 8, 9) & \rightarrow & v_1' = (x, x, |1, 5, 9, 3, |x, x, x) \\ v_2 = (8, 4, |1, 5, 9, 3, |6, 2, 7) & \rightarrow & v_2' = (x, x, |3, 4, 5, 6, |x, x, x) \end{array}$$

2. Finden der Ersetzungsregeln: $\{3 \leftrightarrow 1, 4 \leftrightarrow 5, 5 \leftrightarrow 9, 6 \leftrightarrow 3\}$
$\Rightarrow \{1 \leftrightarrow 3 \leftrightarrow 6, 4 \leftrightarrow 5 \leftrightarrow 9\} \Rightarrow \{1 \leftrightarrow 6, 4 \leftrightarrow 9\}$

3. Übernahme der restlichen Werte aus den Eltern unter Anwendung der Ersetzungsregeln:

$$v_1' = (6, 2, |1, 5, 9, 3, |7, 8, 4) v_2' = (8, 9, |3, 4, 5, 6, |1, 2, 7)$$

Lösung 4.6 (Selektion)

a)

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \quad \textit{und} \quad \sum_{j=1}^{n} f_j = 360$$

$$p_1 = \frac{200}{360} = \frac{5}{9} \approx 0.556$$

$$p_2 = \frac{40}{360} = \frac{1}{9} \approx 0.111$$

$$p_9 = \frac{5}{360} = \frac{1}{72} \approx 0.014$$

$$p_{10} = \frac{1}{360} \approx 0.003$$

b) Aus der Vorsortierung

$$r_i = i \tag{1}$$

folgt $p_{best} = p_1$ *und* $p_{worst} = p_{10}$. *Es soll weiter gelten*

$$p_1 = 2 \cdot p_{10} \tag{2}$$

Die Summe aller Auswahlwahrscheinlichkeiten ist stets 1, also

$$\sum_{i=1}^{10} p_i = 1. \tag{3}$$

Die Auswahlwahrscheinlichkeit soll eine lineare Funktion des Ranges sein:

$$p_i = m \cdot r_i + n \overset{(1)}{=} m \cdot i + n \tag{4}$$

Aus (2) und (4) folgt:

$$p_1 \overset{(4)}{=} m \cdot 1 + n \overset{(2)}{=} 2 \cdot p_{10} \overset{(4)}{=} 2(m \cdot 10 + n) = 20m + 2n \quad \Rightarrow n = -19m \tag{5}$$

Aus (3) und (4) folgt:

$$1 \overset{(3)}{=} \sum_{i=1}^{10} p_i \overset{(4)}{=} \sum_{i=1}^{10} m \cdot i + n = m \cdot \sum_{i=1}^{10} i + 10n = 55m + 10n \tag{6}$$

(5) in (6) eingesetzt:

$$1 = 55m + 10n = 55m - 190m = -135m \Rightarrow m = \frac{-1}{135} \Rightarrow n \overset{(5)}{=} \frac{19}{135} \tag{7}$$

$$p_1 = \frac{-1}{135} \cdot 1 + \frac{19}{135} = \frac{2}{15} \approx 0.133$$

$$p_2 = \frac{-1}{135} \cdot 2 + \frac{19}{135} = \frac{17}{135} \approx 0.126$$

$$p_9 = \frac{-1}{135} \cdot 9 + \frac{19}{135} = \frac{2}{27} \approx 0.074$$

$$p_{10} = \frac{-1}{135} \cdot 10 + \frac{19}{135} = \frac{1}{15} \approx 0.067$$

c) *Da sich die Fitnesswerte des besten und zweitbesten Individuums stark unterscheiden, wird bei der Rouletteselektion das beste Individuum sehr viel häufiger ($p_1 = 0.556$) als das zweitbeste gewählt ($p_2 = 0.111$). Schlechte Individuen haben fast keine Chance sich zu reproduzieren - es herrscht ein starker Selektionsdruck mit der Chance auf schnelle gute Lösungen, aber bei dieser kleinen Population (M=10) auch der Gefahr einer vorzeitigen Konvergenz und Stagnation. Bei der rangbasierten Selektion haben alle Individuen gute Chancen sich zu reproduzieren (kleiner Selektionsdruck), so dass die Diversität des Genpools erhalten bleibt.*

Lösung 4.7 (Baustoffe)

a) *Wir kodieren die Routen der LKW als dreiteilige Permutationskodierung:* $(x_1, ..., x_m, Z, x_{m+1}, ..., x_n, Z, x_{n+1}, ..., x_{20})$,
wobei die x_i die Städte darstellen: $x_i \in \{A, ..., T\}$ und paarweise verschieden sind: $x_i \neq x_j$, wenn $i \neq j$. Z sei das Trennzeichen zwischen den Routen der einzelnen LKW, so dass die Städte $(x_1, ..., x_m)$ vom LKW a in genau dieser Reihenfolge abgefahren werden, die Städte $(x_{m+1}, ..., x_n)$ vom LKW b und die restlichen vom LKW c. Der Genotyp ist also eine Zeichenkette aus 22 Zeichen.

b)
- *permutationserhaltende Mutation, z. B. Tausch, Verschieben, Inversion oder Permutation*
- *permutationserhaltendes Crossover, z. B. PMX-, OX- oder CX-Crossover*

c) *Mutation (Tausch):*

$(A, B, C, D, E, Z, F, G, H, I, J, Z, K, L, M, N, O, P, Q, R, S, T)$
$Tausch\ von\ B\ und\ S \Longrightarrow$
$(A, S, C, D, E, Z, F, G, H, I, J, Z, K, L, M, N, O, P, Q, R, B, T)$

PMX-Crossover:

$v_1 = (A, B, C, D, |E, Z, F, G|, H, I, J, Z, K, L, M, N, O, P, Q, R, S, T)$
$v_2 = (T, S, R, Q, |P, O, Z, G|, A, B, C, D, E, F, H, I, J, Z, K, L, M, N)$
$PMX - Crossover \Longrightarrow$
$v_1' = (A, B, C, D, P, O, Z, G, H, I, J, Z, K, L, M, N, F, E, Q, R, S, T)$
$v_2' = (T, S, R, Q, E, Z, F, G, A, B, C, D, P, O, H, I, J, Z, K, L, M, N)$

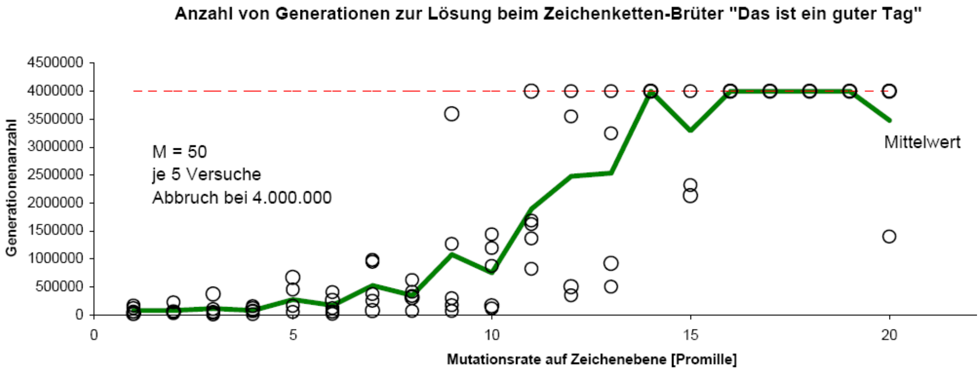
d) *Bergsteiger-Verfahren, Sintflut-Algorithmus, Toleranzschwellenverfahren, Hopfield-Netze, simuliertes Ausglühen, Ameisenkolonie-Optimierung, Constraintpropagierung, Branch and Bound*

Lösung 4.8 (Zeichenketten-Brüter)

a) *Ein entsprechendes C++-Programm findet sich unter*
`http://www.intelligente-systeme.de/mediapool/52/521977/data/Lsg-4.8a.zip`

b) *5 Versuche mit Mutationsrate $0,5\,\%$ benötigen die folgende Anzahl von Generationen bis zur optimalen Lösung: 25937, 45805, 340957, 78317 und 10300. Der Mittelwert beträgt 100263 (die realisierten Zufallswerte ändern sich natürlich in jedem Experiment).*

c) *Ein entsprechendes C++-Programm inklusive Versuchsdaten findet sich unter*
`http://www.intelligente-systeme.de/mediapool/52/521977/data/Lsg-4.8c.zip`.
Als Ergebnis ergibt sich eine Diagramm, das einen günstigen Bereich der Mutationsrate unter 5 Promille zeigt:



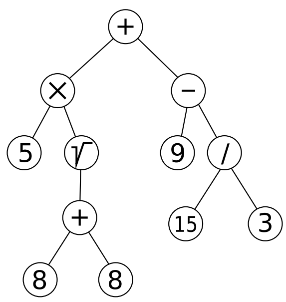
d) *Zeichenketten mit maximaler Fitness stimmen in jedem Zeichen mit einem positionsgleichen Zeichen in mindestens einem Zielstring überein. Die beiden Zielstrings haben s übereinstimmende und damit $l - s$ verschiedene Zeichen. Die s übereinstimmenden Zeichen müssen in Zeichenketten maximaler Fitness an der richtigen Stelle enthalten sein, die verbleibenden $l - s$ Zeichen können aus dem einen oder anderen Zielstring stammen, also lassen sich $M_F = 2^{l-s}$ verschiedene Zeichenketten mit maximaler Fitness l konstruieren.*

Die Anzahl M möglicher Zeichenketten der Länge l und einem Zeichenvorrat von k Zeichen beträt $M = k^l$. Bei $k = s = 2$ ergibt sich somit der Quotient $\frac{M_F}{M} = \frac{2^{l-2}}{2^l} = 25\,\%$.

Lösung 4.9 (GP-Strukturen)

a) *24*

b) *Der Ausdruck als Baum:*



c) *Abb. 4.11a: (+ + – 7 5 sin X * + Y 1 / 2 9)*
*Abb. 4.11b: (+ + 4 / X 1 * 2 lg X)*

d) *Alle*

e) *(+ X 0) oder (+ .. (- X X)) oder (NOT (NOT X))*

f)
- *Symbolischer Ausdruck (Baumstruktur),*
- *Zustandsautomat (Graphstruktur),*
- *Befehlssequenz (Lineare Struktur)*

Lösung 4.10 (GP-Ablauf)

a) *Zufällige Erzeugung mit der Ramped Half-and-Half-Methode*

b) *Der Interpreter führt die Individuen als eine Anweisungsfolge aus.*

c) *Bloat-Effekt: Individuen mit Introns verfügen bei gleicher Fitness über einen Selektionsvorteil gegenüber Individuen ohne Introns, da sie besser vor der zerstörenden Wirkung genetischer Operatoren geschützt sind, d. h. ihre Nachkommen sind im Mittel fitter. Dies gilt insbesondere, wenn bessere Fitnesswerte nicht ohne Weiteres zu erreichen sind. Als Folge wächst die durchschnittliche Individuengröße bei gleichbleibender Fitness.*

Gegenmaßnahmen: Ressourcenbeschränkung, z. B. durch Verwerfen oder Bestrafen (Fitness) zu großer Individuen

Lösung 4.11 (GP-Seiteneffekte)

a) *(+ (PRINT 4) (PRINT 5))*
Je nachdem, ob das linke oder rechte Argument der Addition zuerst ausgewertet wird, resultiert eine andere Ausgabe (Wirkung auf die Umwelt).

b) *(+ (+ (ASSIGN_TO_A 1) (ASSIGN_TO_A 4)) (READ_A))*
Mögliche Ergebnisse: 6, 9 oder undefiniert, falls READ_A vor den Zuweisungen ausgeführt wird

Erläuterung: (ASSIGN_TO_A x) weist einer globalen Variablen A den Wert x zu, (READ_A) liefert den Wert der Variablen A

Lösung 4.12 (Konservativität)

a) *Alle genannten Crossover-Operatoren sind konservativ.*

b) *Nicht konservativ, die Kinder identischer Eltern stimmen in der Regel nicht mit den Eltern überein.*

Lösung 4.13 (GP-Entwurf)

F sei die Funktionsmenge, T die Terminalmenge

a) $F = \{AND, OR, NOT\} \qquad T = \{x_1, x_2, TRUE, FALSE\}$
Bsp: (OR (AND x_1 (NOT x_2)) (AND x_2 (NOT x_1)))

b)

$$\begin{aligned} F = \{ \quad & IF, LESS, AND, OR, NOT, +, -, *\} \\ T = \{ \quad & \text{Felldichte, Kopfbreite, Schwanzdicke, Rückenstreifen,} \\ & \text{Aggressivität, WILD, HAUS}\} \cup \mathbb{R} \end{aligned}$$

Bsp: (IF (LESS Rückenstreifen 4) HAUS WILD)

c) $F = \{+, *\} \qquad T = \{X\} \cup \mathbb{R}$
Bsp: (+ (X (+ -3 (* (+ X 4.2) X))) 7)*

d) $F = \{IF\_WAND\_VORN, PROGN2\}$ $T = \{LEFT, RIGHT, MOVE\}$

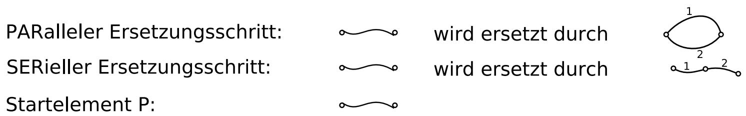
Bsp: (IF_WAND_VORN (PROGN2 LEFT MOVE) MOVE)

e) $F = \{\cup, \cap, /, ZYLINDER\}$ $\quad T = \mathbb{R}$,
allerdings ohne Closure: ZYLINDER erlaubt nur reelle Parameter $(x_1, y_1, z_1, x_2, y_2, z_2, r)$, *die die beiden Achsendpunkte und den Radius darstellen. Die anderen drei Funktionen haben die Stelligkeit 2 und gestatten als Argumente die Ergebnisse aller Funktionen.*

Bsp „T-Stück“:
(∪ (ZYLINDER 5 10 0 13 10 0 12) (ZYLINDER 8 0 0 8 10 0 12))

f) *Die Individuen beschreiben den Graphen indirekt über eine Konstruktionsvorschrift.*

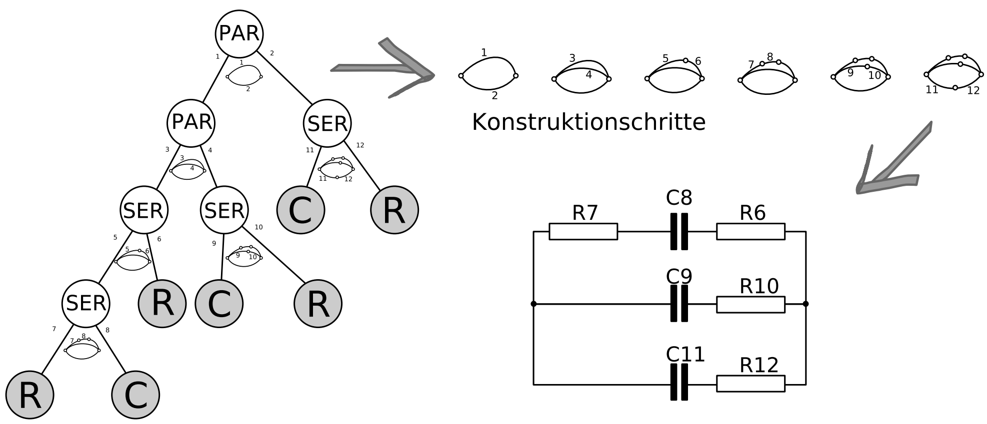
Grundidee zur Schaltungssynthese: Ein Startelement P mit 2 Endpunkten wird durch eine neue Struktur mit zwei Endpunkten ersetzt. Die neue Struktur kann ein terminales Element oder eine Anordnung sein, die zwei Platzhalter P_i *enthält. Terminale Elemente können hierbei Kondensatoren (C) oder Widerstände (R) sein, Anordnungen sind eine serielle oder parallele 2er-Struktur. Ein Individuum beschreibt die Ersetzungsschritte der Platzhalter:*



$F = \{SER, PAR\} \quad T = \{C, R\}$

Symbol	*Ersetzt aktuellen Platzhalter durch*	*Platzhalter*
SER	*Sequenz zweier Platzhalter*	P_1, P_2
PAR	*Parallelschaltung zweier Platzhalter*	P_1, P_2
C	*Kondensator C*	
R	*Widerstand R*	

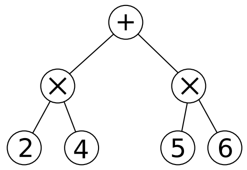
Bsp: (PAR (PAR (SER (SER R C) R) (SER C R)) (SER C R))



Die konkreten Werte der Bauelemente werden hierbei vernachlässigt, lassen sich aber problemlos als Parameter an C und R anfügen. Andere Ansätze verwenden Bondgraphen zur Kodierung der Schaltung.

Lösung 4.14 (GP-Interpreter)

a) $W = 2 \cdot 4 + 5 \cdot 6 = 38$



b) *Die Befehle in der Variablen chrome werden nacheinander rekursiv abgearbeitet. Die globale Variable idx enthält den Index des gerade vergangenen Befehls. Um das nächste Argument einer Operation zu berechnen (Aufruf von EvalNextArg) wird der Index inkrementiert und die Berechnung des Knotens ausgeführt. Falls diese Berechnung Argumente (z. B. bei Addition, Multiplikation) benötigt, kann dazu wieder die Funktion EvalNextArg gerufen werden. Es können nur einstellige Zahlen verarbeitet werden.*

c) *Die Subtraktion ist notwendig, um aus den im Array chrome gespeicherten Zeichencodes die Ganzzahlen für die eigentliche Berechnung zu erhalten.*

Der implizite cast von char (Differenz) nach int (Returnwert der Funktion) ist unproblematisch, da eine Bereichserweiterung erfolgt.

d) *Auf dem Stack des Programms*

e) *Die Ausführungsreihenfolge ist bei der Präfix-Notation eindeutig, so dass keine Klammern benötigt werden. Einzig zur Trennung der Knoten bei mehrstelligen Zahlen wären Klammern sinnvoll, es genügte aber auch ein einfaches Trennzeichen.*

f) *(+11) und (*21)*

g) *Ja, alle Funktionen akzeptieren alle möglichen Terminale und Funktionswerte als Argumente.*

Dennoch sollte der mögliche Überlauf der Rechenoperationen berücksichtigt werden.

h) *Zwischen Zeile 6 und 7 einfügen:*

```
else if (chrome[idx] == '-') return EvalNextArg() - EvalNextArg();
else if (chrome[idx] == '/') {
      int k=EvalNextArg(); int d=EvalNextArg();
      if (d==0) return 0; else return k/d;
}
```

Lösung 4.15 (Crossover in Präfixnotation)

a) *S sei die Stelligkeit der Knoten:*
$S_7 = S_5 = S_X = S_Y = S_1 = S_2 = S_9 = 0, \quad S_{sin} = 1,$
$S_+ = S_- = S_/ = S_. = 2$

M sei die Stelligkeit -1:
$M_7 = M_5 = M_X = M_Y = M_1 = M_2 = M_9 = -1, \quad M_{sin} = 0,$
$M_+ = M_- = M_/ = M_. = 1$

Summe $\sum$ *der M-Werte für die einzelnen Teilbäume:*

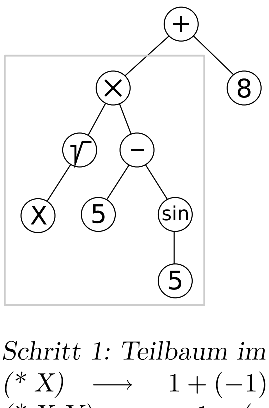
$$
\begin{aligned}
\textstyle\sum_{-} &= M_7 + M_5 + M_{-} = (-1) + (-1) + 1 = -1\\
\textstyle\sum_{sin} &= M_X + M_{sin} = (-1) + 0 = -1\\
\textstyle\sum_{+} &= M_Y + M_1 + M_{+} = (-1) + (-1) + 1 = -1\\
\textstyle\sum_{/} &= M_2 + M_9 + M_{/} = (-1) + (-1) + 1 = -1\\
\textstyle\sum_{+} &= M_{-} + M_7 + M_5 + M_{sin} + M_X + M_{+}\\
&= 1 + (-1) + (-1) + 0 + (-1) + 1 = -1\\
\textstyle\sum_{.} &= M_{+} + M_Y + M_1 + M_{/} + M_2 + M_9 + M.\\
&= 1 + (-1) + (-1) + 1 + (-1) + (-1) + 1 = -1\\
\textstyle\sum_{+} &= M_{-} + M_7 + M_5 + M_{sin} + M_X + M_{+}\\
&\quad +M_{+} + M_Y + M_1 + M_{/} + M_2 + M_9 + M. + M_{+}\\
&= 1 + (-1) + (-1) + 0 + (-1) + 1+\\
&\quad 1 + (-1) + (-1) + 1 + (-1) + (-1) + 1 + 1 = -1
\end{aligned}
$$

oder rekursiv berechnet:

$$
\begin{array}{lcll l}
\sum_{-} & = & M_7 + M_5 + M_{-} & = (-1) + (-1) + 1 & = -1\\
\sum_{sin} & = & M_X + M_{sin} & = (-1) + 0 & = -1\\
\sum_{+} & = & M_Y + M_1 + M_{+} & = (-1) + (-1) + 1 & = -1\\
\sum_{/} & = & M_2 + M_9 + M_{/} & = (-1) + (-1) + 1 & = -1\\
\sum_{+} & = & \sum_{-} + \sum_{sin} + M_{+} & = (-1) + (-1) + 1 & = -1\\
\sum_{.} & = & \sum_{+} * \sum_{/} + M. & = (-1) + (-1) + 1 & = -1\\
\sum_{+} & = & \sum_{+} \sum_{.} + M_{+} & = (-1) + (-1) + 1 & = -1
\end{array}
$$

b) *(*)* $\longrightarrow$ 1
(SQRT)* $\longrightarrow$ $1 + 0 = 1$
(SQRT X)* $\longrightarrow$ $1 + 0 - 1 = 0$
(SQRT X –)* $\longrightarrow$ $1 + 0 - 1 + 1 = 1$
(SQRT X – 5)* $\longrightarrow$ $1 + 0 - 1 + 1 - 1 = 0$
(SQRT X – 5 sin)* $\longrightarrow$ $1 + 0 - 1 + 1 - 1 + 0 = 0$
(SQRT X – 5 sin 5)* $\longrightarrow$ $1 + 0 - 1 + 1 - 1 + 0 - 1 = -1$
*Der bei * beginnende Teilbaum ist (* SQRT X – 5 sin 5).*

c) *Der Teilbaum ab * entspricht dem Ausdruck (* SQRT X – 5 sin 5):*



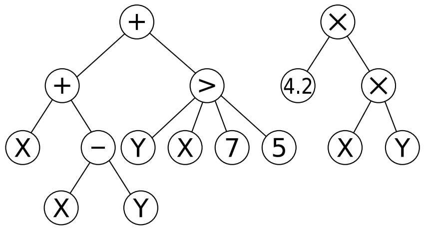
d) *Schritt 1: Teilbaum im ersten Elter bestimmen: (*)* $\longrightarrow$ 1
(X)* $\longrightarrow$ $1 + (-1) = 0$
(X Y)* $\longrightarrow$ $1 + (-1) + (-1) = -1$
*Der bei * beginnende Teilbaum ist (* X Y).*

Schritt 2: Teilbaum im zweiten Elter bestimmen: (+) $\longrightarrow$ 1
(+ X) $\longrightarrow$ $1 + (-1) = 0$
(+ X –) $\longrightarrow$ $1 + (-1) + 1 = 1$
(+ X – X) $\longrightarrow$ $1 + (-1) + 1 + (-1) = 0$

(+ X – X Y) $\longrightarrow$ $1 + (-1) + 1 + (-1) + (-1) = -1$
Der bei + beginnende Teilbaum ist (+ X – X Y).

Schritt 3: Tauschen der Teilbäume
Nachkomme1 = (+ + X – X Y > Y X 7 5)
*Nachkomme2 = und (* 4.2 * X Y)*

e) *Das Ergebnis stimmt mit der Abbildung 4.12b überein:*



Für Fragen und Hinweise kontaktieren Sie mich unter
boersch@fh-brandenburg.de.