1 Zusammenfassung

1. Induktive Definitionen (z.B. für Terme und Formeln in der Prädikatenlogik) spielen in der Informatik eine zentrale Rolle: Zitate von Prof. Kindler (Semantik, Skript zur Vorlesung Semantik von Programmiersprachen im WS 2003/04 an der Universität Paderborn. Überarbeitet im WS 2004/05 Ekkart Kindler Version: 0.2.8 vom 28. Januar 2005. Kapitel 4 induktive Definitionen und Beweise): "Wenn man genau hinsieht, gibt es in der Informatik fast nichts, was nicht induktiv de?niert wäre"..."Deshalb spielen induktive Definitionen und Beweise in der Informatik eine ganz zentrale Rolle ? ob man sie nun explizit macht oder nicht"... Ëin weiterer Grund ist, ein Bewußtsein dafür zu schaffen, daß in der Informatik fast uberall nur mit Wasser gekocht wird, wobei das Wasser die Konzepte des induktiven Definierens und Beweisens sind."

   Bemerkung: Das komplette Skript (Artikel) zu dieser Vorschau befindet sich unter: umaterialien.de/THEORETISCHEINFORMATIK/theoretische_informatik.html

2. Neben den induktiven Definitionen spielen Rekursionen in der Informatik auch eine Rolle. Während induktiv definierte Mengen "von unten her", von der leeren Menge aus, aufgebaut werden, wird eine Rekursion durch SZurücklaufen", also von öben her"durchgeführt.

3. In dieser Vorschau wird versucht, einen Zusammenhang zwischen einer induktiv definierten Menge und einer rekursiven Funktion herzustellen: Wie kann man eine rekursive Funktion angeben, die ëntscheidet", ob ein vorgegebenes Element zu einer induktiv definierten Menge gehört oder nicht? Diese rekursive Funktion kann dann (nach der induktiven Definition) rekursiv implementiert werden (die Formel gibt änschaulich gesprochen"den Algorithmus vor). Da die rekursive Funktion ein Problem lösen soll, muß sie berechenbar sein. Deshalb werden dazu bestimmte Voraussetzungen gemacht. Man muss sich also keine problemspezifischen Backtrack-Strategien überlegen: spezifisch sind hier nur die Regeln für induktive definierte Mengen. Das was man also informal unter Backtrack-Strategie versteht, wird hier u.a. versucht zu präzisieren (formalisieren).

4. Beispiele

   Bemerkung: Die in den folgenden Beispielen angegebenen Formeln, also e(...)= und N(...)=
   sind Spezialfälle einer allgemeineren Formel, die in dem Artikel entwickelt wird.

   Beispiel 1:
   Man kann mit Hilfe von Regeln Zahlenterme aufbauen (Regeln dazu kommen gleich).
   Unter Zahlenterme verstehen wir hier z.B. die Zeichenfolgen “((3+4)+5)“ bzw. “(2+9)“, wohingegen z.B. die Zeichenfolge "(++++((((" kein Zahlenterm ist.

Formalisieren kann man dies mit Hilfe von Regeln, die hier in einer Regelnkurznotation angegeben werden:

$X = \{\ 0\ ;\ 1\ ;\ 2\ ;\ 3\ ;\ 4\ ;\ 5\ ;\ 6\ ;\ 7\ ;\ 8\ ;\ 9\ ;\ (\ ;\ )\ ;\ +\ \}^*$
$\mathbb{N}$ = Menge der natürlichen Zahlen
$R = \{(\{x_1; x_2\}, (x_1 + x_2)) | x_1 \in X \wedge x_2 \in X\} \bigcup \{(\emptyset; n) | n \in \mathbb{N}\}$

R kann man informal kürzer durch folgende Regelnkurznotation
$(r \in X, s \in X, n \in \mathbb{N})$ darstellen:

$$\frac{\emptyset}{n} \qquad \frac{\{r; s\}}{(r+s)}$$

Mit I(R) bezeichnet man dann die durch R induktiv definierte Menge, hier also die Menge der Zahlenterme.

Es soll dann "berechnet"werden, ob eine Zeichenfolge ein Zahlenterm ist.
Damit kann man "berechnen", ob z.B. “(2+9)“ oder “(++++((((“ ein bzw. kein Zahlenterm ist.
Dazu wurde die folgende rekursive Formel (e wie evaluate) entwickelt:

Für alle $x \in X$ gilt ($\cdot \mathrel{\widehat{=}}$ logische UND und $+ \mathrel{\widehat{=}}$ logische ODER):

$$e(x) = \begin{cases} 1 & falls\ (\emptyset, x) \in R \\ 0 & falls\ R^{-1}[\{x\}] = \emptyset \\ \sum\limits_{\{x_1,\ldots,x_m\} \in R^{-1}[\{x\}]} e(x_1) \cdot \ldots \cdot e(x_m) & sonst \end{cases}$$

Damit kann man "berechnen", ob z.B. “(2+9)“ ein Zahlenterm ist.
Man bildet dazu e((2+9)) .
Um die Formel zu verwenden müssen vorher die Elemente der Umkehrrelation $R^{-1}[\{(2+9)\}]$ ermittelt werden, also alle Mengen $\{r; s\}$ mit $(\{r; s\}; (2+9)) \in R$.
Das ist nur die Menge $\{2; 9\}$
Mit Hilfe obiger rekursiven Formel folgt dann:
$e((2+9)) = e(2) \cdot e(9)$
Um weiter zu rechnen, müssen jetzt noch e(2) und e(9) "berechnet"werden.
Dazu müssen die Elemente der Umkehrrelation $R^{-1}[\{2\}]$ und $R^{-1}[\{9\}]$ ermittelt werden:
$R^{-1}[\{2\}] = \{\emptyset\}$ und $R^{-1}[\{9\}] = \{\emptyset\}$
Mit Hilfe obiger rekursiven Formel folgt dann:
e(2) = 1 und e(9) = 1 also insgesamt:
$e((2+9)) = e(2) \cdot e(9) = 1 \cdot 1 = 1$,
also ist (2+9) ein Zahlenterm.
Wäre e((2+9)) = 0, dann wäre “(2+9)“ kein Zahlenterm.

Beispiel 2:
Man kann mit Hilfe von Regeln Zahlenterme mit ihren Werten aufbauen (Regeln dazu kommen gleich).
Das 1. Element im Tupel ist der Zahlenterm und das 2. Element (= der Nachbar des Zahlenterm) ist der Wert des Zahlenterms.
Da die mathematische Addition nicht mehr mit + bezeichnet werden kann (da dieses Zeichnen schon bei den Zahlentermen verwendet wird, nimmt man hier dafür die Bezeichnung add.
Die in dem vorigen Beispiel definierte Menge aller Zahlenterme wird hier mit MAZ abgekürzt.
$R = \{(\{(T_1; w_1); (T_2; w_2)\}; ((T_1 + T_2); add(w_1; w_2))) \mid$
$\quad T_1 \in MAZ \wedge T_2 \in MAZ \wedge w_1 \in \mathbb{N} \wedge w_2 \in \mathbb{N}\} \bigcup \{(\emptyset; (n; n)) \mid n \in \mathbb{N}\}$

R kann man informal kürzer durch folgende Regelnkurznotation ($w_1 \in \mathbb{N}, w_2 \in \mathbb{N}, n \in \mathbb{N}$, $T_1 \in MAZ$, $T_2 \in MAZ$) darstellen:

$$\frac{\emptyset}{(n;n)} \qquad \frac{(T_1; w_1); (T_2; w_2)}{((T_1 + T_2); add(w_1, w_2))}$$

Es soll dann der Wert eines Zahlenterms (siehe letztes Beispiel) "berechnet"werden.
Damit kann man z.B. den Wert des Zahlenterms “(2+9)“ "berechnen".
Dazu wurde die folgende rekursive Formel (N wie Nachbar) entwickelt:

Für alle $a \in MAZ$ gilt:

$$N(a) = \begin{cases} a & falls\ a \in \mathbb{N} \\ add(N(a_1), N(a_2)) & sonst\ d.h.\ a = (a_1 + a_2) \end{cases}$$

Damit kann man dann z.B. den Wert des Zahlenterms “(2+9)“ "berechnen"
Man bildet dazu N((2+9)) und "berechnet":
N((2+9)) = add((N(2), N(9)) = add(2,9) = 11

5. Der Zusammenhang zwischen induktiver Definition und Rekursion wird durch 4 Sätze (Rekindsatz 1- Rekindsatz 4) beschrieben.
   Diese Sätze werden auf konkrete Beispiele angewendet:
   - Syntaxchecker (Terme, Formeln, Anweisungen),
   - Wert von (eindeutigen bzw. mehrdeutigen) Termen "berechnen",
   - reguläre Ausdrucke prüfen (bzw. Semantik davon bestimmen),
   - optimale Spielstrategien "berechnen"von z.B:
     Nimm-Spiel,
     Acht-Damen-Problem,
     magisches Quadrat,
     Sudokulöser,
     Würfel kippen
   - Erfüllbarkeitsprobelm der Aussagenlogik,
   - operationale Semantik, usw.