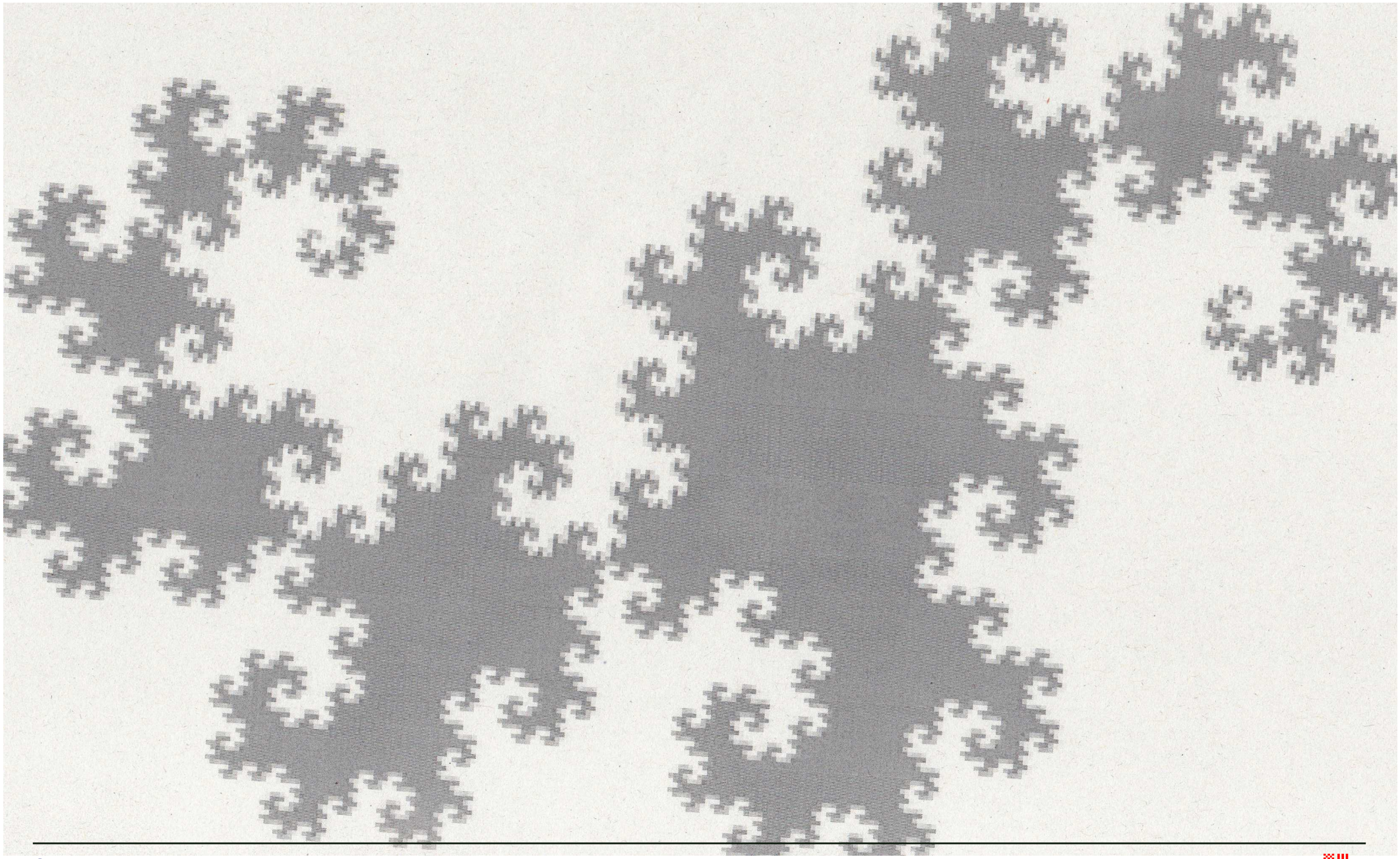
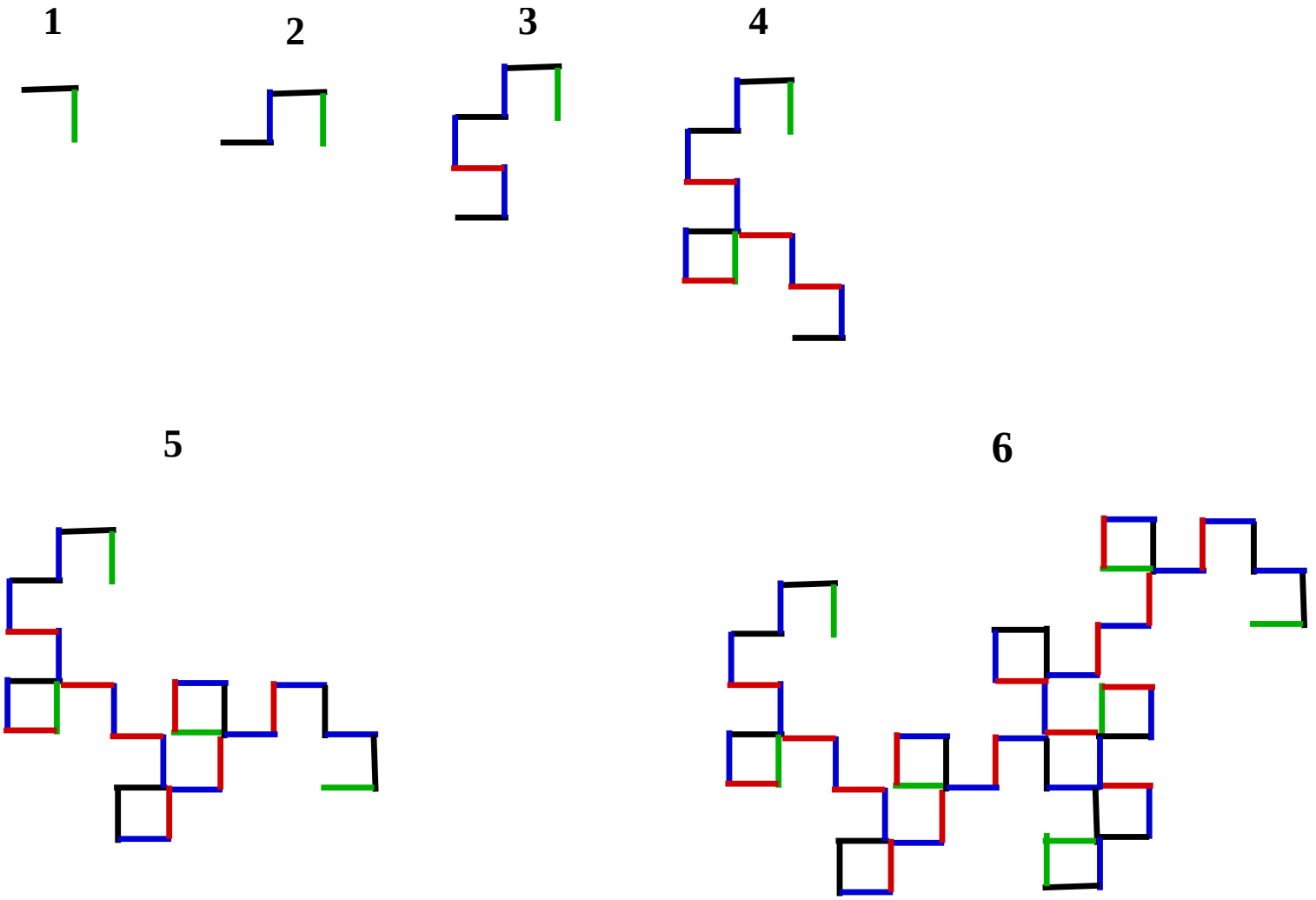
Fallstudie "Drachenkurve"

- Bekannte flächenfüllende Kurve, bekanntes Fraktal
- Algorithmische Modellierung als Approximation durch wiederholtes Falten eines Papierstreifens (Linie)
- Genauer: als Liniensequenzen in Abhängigkeit von der Zahl der Faltungen durch Angabe der Himmelsrichtungen, in die Einheitslinien verlaufen



Die Approximationen 1 bis 6

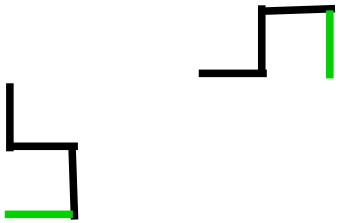


Erzeugung der 4. Approximation aus der 3.

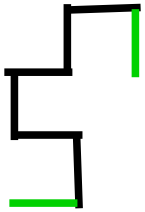
- 3. Aproximation

- Kopieren

- Im Uhrzeigersinn drehen (90 Grad)

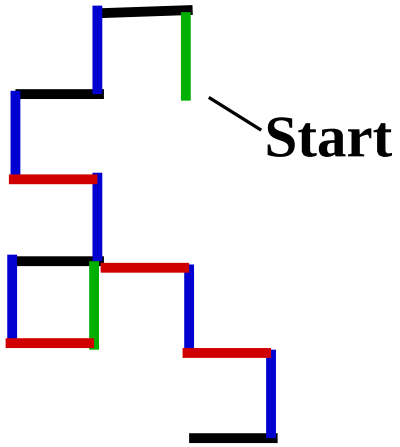


- Endpunkte zusammenfügen



Formale Beschreibung der Drachenkurve

- Alphabet: $Comp = \{N, O, S, W\}$
- sogenannte Kettencodes zur Beschreibung der Liniensequenzen: $u \in Comp^*$
- z.B. $NWSWSOSWSONOSOSW$



dragoncurve — Name der Spezifikation

opns: $dc: \mathbb{N} \rightarrow Comp^*$
$ctb: Comp^* \rightarrow Comp^*$

Deklaration von Operationen (Name, Argument- und Wertebereiche)

vars: $n \in \mathbb{N}$
$u \in Comp^*$

Deklaration von Variablen (Name, Typ)

eqns:

$$dc(0) = N$$
$$dc(n+1) = ctb(dc(n))$$
$$ctb(\lambda) = \lambda$$
$$ctb(Nu) = Nctb(u)W$$
$$ctb(Ou) = Octb(u)N$$
$$ctb(Su) = Sctb(u)O$$
$$ctb(Wu) = Wctb(u)S$$

Definition der Operationen durch Gleichungen; Festlegung des Effekts von einzelnen Rechenschritten mit Rekursion über den induktiven Aufbau natürlicher Zahlen und Zeichenketten

Auswertung (Beispiel)

$$dc(3) = ctb(dc(2)) = ctb(ctb(dc(1))) = ctb(ctb(ctb(dc(0))))$$
$$= ctb(ctb(ctb(N))) = ctb(ctb(Nctb(\lambda)W)) = ctb(ctb(NW))$$
$$= ctb(Nctb(W)W) = ctb(NWctb(\lambda)SW) = ctb(NWSW)$$
$$= Nctb(WSW)W = NWctb(SW)SW$$
$$= NWSctb(W)OSW = NWSWctb(\lambda)SOSW$$
$$= NWSWSOSW$$

Aufwand: 14 Gleichungsanwendungen

Lehr- und Lernziel

Ermittlung des Zeitaufwands bei der Auswertung von Operationen auf Zeichenketten

Wesentliche Elemente zur Ermittlung des Aufwands von Algorithmen

- Modellierung von Algorithmen
- einschließlich ihrer Berechnung (Ausführung)
- Quantitative Erfassung als Zahlen der Berechnungsschritte
- Nachweisbarkeit der dafür erforderlichen Eigenschaften
- Geeigneter Ansatz: CE-S

CE-S (Conditional Equations on Strings)

- Algorithmenmodellierungssprache
- Algorithmen als Operationen mit Argument- und Wertebereichen
- Modellierung (Spezifikation, Definition) durch bedingte Gleichungen

Beispiel

dragoncurve

opns: $dc\colon \mathbb{N} \to Comp$

$ctb\colon Comp^* \to Comp^*$

vars: $n \in \mathbb{N}\ u \in Comp^*$

eqns: $dc(0) = N$

$dc(n+1) = ctb(dc(n))$

$ctb(\lambda) = \lambda$

$ctb(Nu) = Nctb(u)W$

$ctb(Ou) = Octb(u)N$

$ctb(Su) = Sctb(u)O$

$ctb(Wu) = Wctb(u)S$

Syntaxschema

spec	Name
opns: $decl_1, \ldots, decl_k$	Operationsdeklarationen
vars: $tv_1, \ldots tv_p$	Variablendeklarationen
eqns: $ce_1, \ldots, ce_l$	Bedingte Gleichungen

Operationsdeklaration

$$f\colon D_1 \times \cdots \times D_m \to D$$

- f: Name
- $D_1, \ldots, D_m$: Argumenttypen
- D: Wertetyp

Spezialfall Konstantendeklaration: $c\colon \to D$

Beispiele

- $count\colon A \times A^* \to \mathbb{N}$
- $5\colon \to \mathbb{N}$

Variablendeklaration

$$x \in D$$

- x: Name
- D: Typ

Beispiel

$$x \in A^*$$

Bedingte Gleichung

$$L = R \text{ falls } b$$

- L, R: Terme desselben Typs
- b: Term vom Typ $BOOL$

Term

Sei $DECL$ eine Menge von Operationsdeklarationen und X eine Menge von Variablendeklarationen.

Menge T_D aller Terme des Typs D

1. $(c\colon \to D) \in DECL$ impliziert $c \in T_D$;
2. $(x \in D) \in X$ impliziert $x \in T_D$;
3. $(f\colon D_1 \times \cdots \times D_k \to D) \in DECL$ und $t_i \in T_{D_i}$ für $i = 1, \ldots, k$ implizieren $f(t_1, \ldots, t_k) \in T_D$.

CE-S- Datentypen

Typen	Grundoperationen
Wahrheitswerte $BOOL$	$T, F: \ \rightarrow BOOL$ $\neg: BOOL \rightarrow BOOL$ $\wedge, \vee, \ldots : \ BOOL \times BOOL \rightarrow BOOL$
Natürliche und ganze Zahlen $\mathbb{N}$,$\mathbb{Z}$	übliche Konstanten, arithm. Operationen und Vergleiche
Alphabete $A, B, \ldots$	Elemente $a: \ \rightarrow A$, $\ldots$ $\equiv: A \times A \rightarrow BOOL$
Sprachen $A^*, B^*, \ldots, \mathbb{N}^*, \ldots, (A^*)^*, \ldots$	λ, Konkatenation, $length$, $\equiv$, $\ldots$

Wertzuweisung und Substitution

Variablen sind Platzhalter für Terme desselben Typs

Wertzuweisung: $a(x) \in T_D$ für alle $(x \in D) \in X$

Substitution (von x durch $a(x)$ in Term)

1. $c[a] = c$ für $(c\colon \ \rightarrow D) \in DECL$
2. $x[a] = a(x)$ für $(x \in D) \in X$
3. $f(t_1, \ldots, t_k)[a] = f(t_1[a], \ldots, t_k[a])$ für $(f\colon D_1 \times \cdots \times D_k \rightarrow D) \in DECL$, $t_i \in T_{D_i}$, $i = 1, \ldots, k$

Auswertungsschritt (Gleichungsanwendung)

1. Zuweisung aktueller Werte $a(x) \in T_D$ an die Variablen $x \in D$ und Substitution in Gleichung $L = R$:

$$L[a] \rightarrow R[a]$$

2. Dasselbe im Argumentterm:[1]

$$f(t_1, \ldots t_{i-1}, t_i, t_{i+1}, \ldots, t_k) \rightarrow f(t_1, \ldots, t_{i-1}, t'_i, t_{i+1}, \ldots, t_k)$$

falls $t_i \rightarrow t'_i$

[1] Wegen Rekursion in 2. kann direkte Gleichungsanwendung gemäß 1. beliebig weit innen im Term stattfinden.

Berechnung

- Berechnung als Auswertungsschrittfolge:

$$t \equiv t_0 \longrightarrow t_1 \longrightarrow \cdots \longrightarrow t_n \equiv t' \qquad (n = 0 : t = t')$$

dafür kurz: $t \overset{*}{\longrightarrow} t'$

Gleichwertigkeit

- Gleichwertigkeit wie Berechnung mit Rückwärtsschritt ($R = L$ statt $L = R$)

$$\boxed{t \longleftrightarrow t' \text{ falls } t \longrightarrow t' \text{ oder } t \leftarrow t'}$$

$\overset{*}{\longleftrightarrow}$ analog zu $\overset{*}{\longrightarrow}$

- für $t \overset{*}{\longleftrightarrow} t'$ auch $t = t'$
- Beachte: $\longrightarrow \subseteq \overset{*}{\longrightarrow} \subseteq \overset{*}{\longleftrightarrow} \supseteq \longleftrightarrow$

Fallunterscheidung

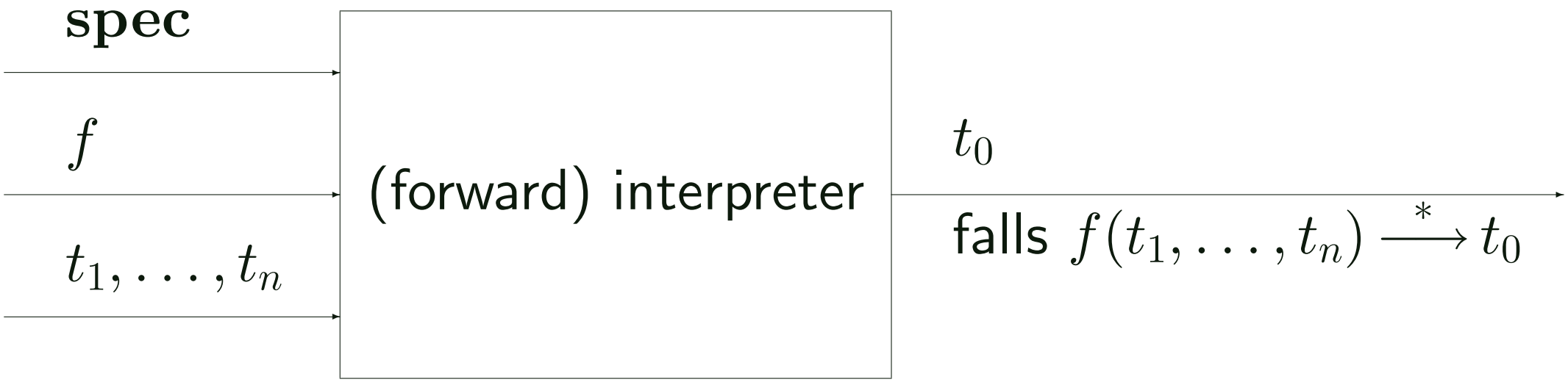
- if-then-else- als Abkürzung für 2 bedingte Gleichungen:

$$t = \text{ if } b \text{ then } t_1 \text{ else } t_2 \text{ kurz für } \begin{cases} t = t_1 & \text{falls } b \\ t = t_2 & \text{falls } \neg b \end{cases}$$

- Berechnungsschritte wie in Gleichungen aber bedingt:

$$\begin{aligned} t[a] &\longrightarrow t_1[a] \quad \text{falls } b[a] \overset{*}{\longleftrightarrow} T \\ t[a] &\longrightarrow t_2[a] \quad \text{falls } b[a] \overset{*}{\longleftrightarrow} F \end{aligned}$$

(Vorwärts-)Interpreter

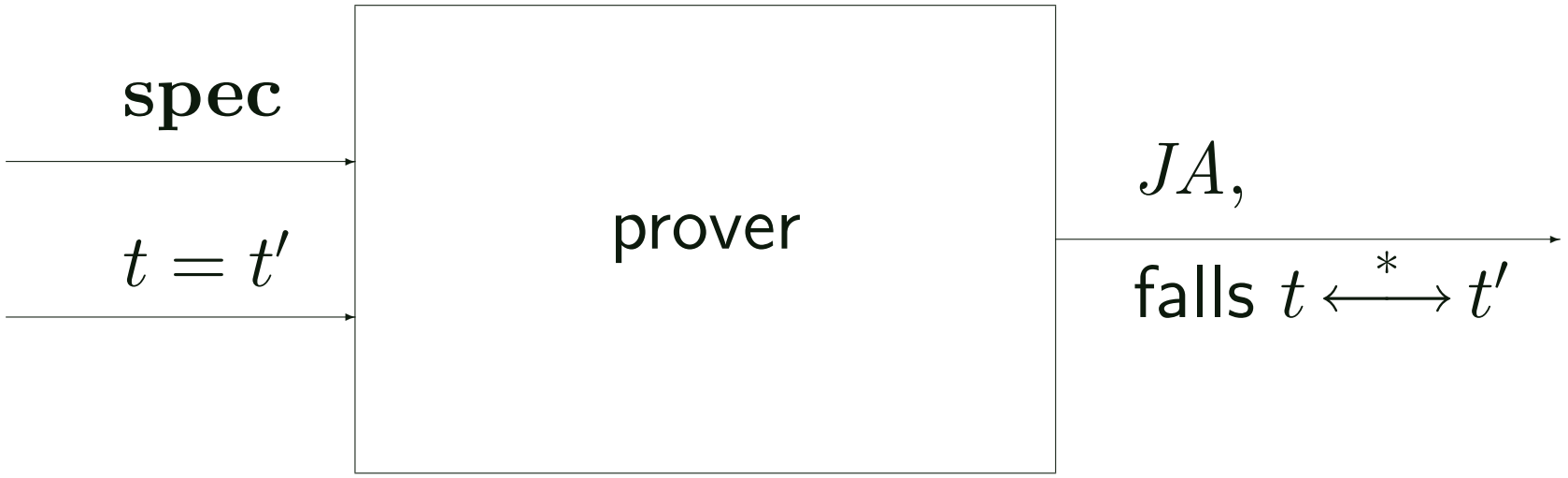


- $t_1, \ldots t_n$: Werteterme, d.h. Terme, die nur aus Grundoperationen und ohne Variablen aufgebaut sind

Beispiele für Werteterme

- $T, F, T \vee T, a \equiv b, abc \equiv abc, \ldots \in T_{BOOL}$
- $0, 1, 2, \ldots, 5 - 4, 3 + 2, length(abc), \ldots \in T_{\mathbb{N}}$
- $a, b, c, \ldots \in T_A$
- $\lambda, abc, abc \cdot abc, \ldots \in T_{A^*}$

Gleichungsbeweiser



Beispiel für eine Spezifikation in CE-S

insort

opns: $insort : A \times A^* \to A^*,\ sort : A^* \to A^*$

vars: $x, y \in A,\ u, v \in A^*$

eqns: $insort(x, \lambda) = x$

$insort(x, yv) = \text{if } x \leq y \text{ then } xyv \text{ else } y\, insort(x, v)$

$sort(\lambda) = \lambda$

$sort(xu) = insort(x, sort(u))$