Lineare Logik

- Intuitionistische Lineare Logik
- Zusammenhang zur intuitionistischen Logik
- Termzuweisung und funktionale Interpretation

Intuitionistische Lineare Logik

- „Sinn und Zweck“:
  - Kontrolle über *Ressourcen*
  - ... im Rahmen der *funktionalen Programmierung*: Informationen bzw. Daten, die als Argumente verwandt werden stellen Ressourcen dar.
  - ... im *Sequenzenkalkül*: Strukturregeln sind verantwortlich für Ressourcenmanagement.

(Contraction) $\dfrac{\Gamma, x : A, y : A \vdash t : B}{\Gamma, z : A \vdash t[z/x, z/y] : B}$	erlaubt das *Kopieren* von Ressouren
(Weakening) $\dfrac{\Gamma \vdash t : B}{\Gamma, z : A \vdash t : B}$	erlaubt das *Entfernen* von Ressouren

Wirkung struktureller Regeln

- (Contraction) erlaubt *mehrfaches Auftreten* derselben Variable im „Beweisterm“ (proof term)

$$\text{(Contraction)}\quad \frac{\Gamma, x : A, y : A \vdash t : B}{\Gamma, z : A \vdash t[z/x, z/y] : B}$$

- (Weakening) erlaubt die Verwendung von Variablen als Prämisse, die *nicht* im Beweisterm vorkommen

$$\text{(Weakening)}\quad \frac{\Gamma \vdash t : B}{\Gamma, z : A \vdash t : B}$$

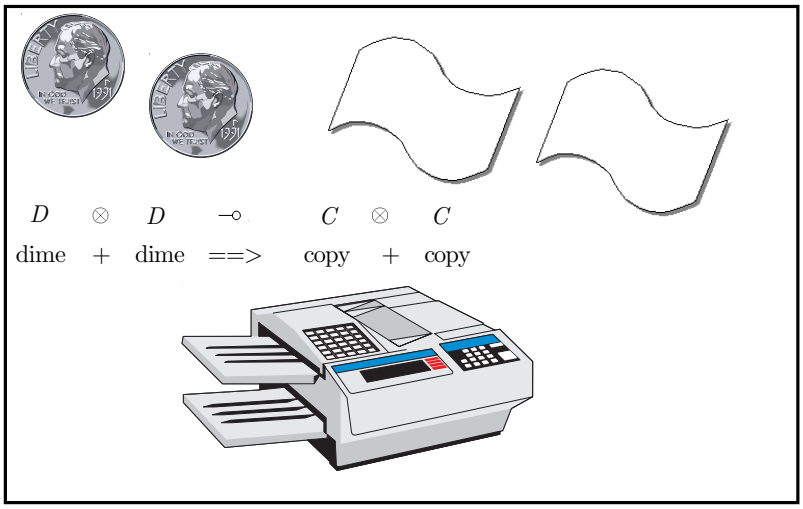
In Linearer Logik werden diese beiden Regeln nicht verwendet! Also muß jedes „Argument“ *genau einmal* verwendet werden, um die gewünschte Ausgabe zu erzeugen.

Auswirkungen des Fehlens struktureller Regeln

- Zwei verschiedene Arten der Konjunktion:
  - Tensorprodukt (multiplikativ): $A \otimes B$
    Beide Komponenten des Paares werden als Eingabe verwendet. Keine Projektion möglich!
  - additive Konjunktion: $A \,\&\, B$
    Beim Verbrauch dieser Eingabe wird *entweder* die erste *oder* die zweite Komponente verwandt. Der andere Teil der Eingabe ist für die „Berechnung“ nicht sichtbar!

    ==> Die additive Konjunktion entspricht „externer Auswahl“, d.h. der Benutzer bestimmt, welche Daten verbraucht werden.

Beispiel: multiplikative Konjunktion



Auswirkungen des Fehlens struktureller Regeln (2)

- ... und auch zwei verschiedene Disjunktionen, aber uns interessiert zunächst nur eine:
  - additive Disjunktion: $A \oplus B$
    entspricht *interner Auswahl*, d.h. der „Produzent“ der Information kann entscheiden, ob A (aus `inl(`t`)`) oder B *aus* `inr(`t`)` gewählt wird.
  - multiplikative Implikation: $A \multimap B$
    - ist die einzige Art der Implikation
    - stellt Funktionen dar, die ihre Argumente *genau einmal* verwenden, d.h. intern der linearen Implikation gehorchen.

Linear Logische Modalitäten

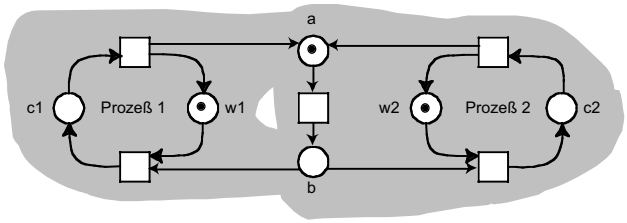
- Zur Wiedererlangung der Ausdruckstärke der klassischen bzw. intuitionistischen Logik:
  - Einführung der Modalitäten ! (of course) und ? (why not)
  - „of course“ stellt soviele Instanzen eines Wertees zur Verfügung, wie gebraucht werden
  - intuitionistische Implikation wird ausgedrückt durch:

$$A \boxed{\rightarrow} B = !A \multimap B$$

  - strukturelle Regeln werden „relativiert“ wieder eingeführt ...

Lineare Logik und Petrinetze

• Beispiel eines Petrinetzes: wechselseitiger Ausschluß



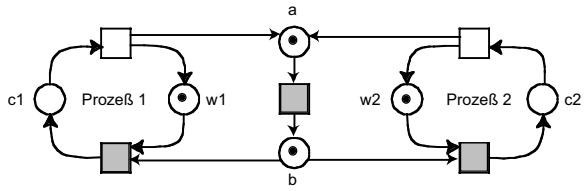
• Konstruktion einer LL-Formel für obiges Netz:

Die Transition wird beim Schalten nicht verbraucht.

... und eine passende Startmarkierung

Lineare Logik und Petrinetze (2)

- Schalten des Netzes:



- Dargestellt als Ableitung in Linearer Logik:

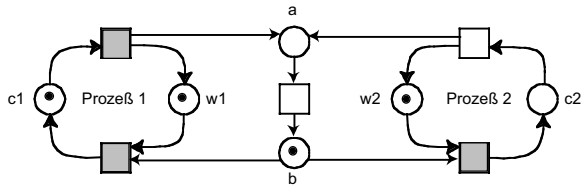
  Wir verwenden die abgeleitete Regel $\dfrac{}{A \multimap B, A \vdash B}$ (Imp)

  D.h. für obiges Beispiel kann z.B. abgeleitet werden:

$$\Gamma, !(A \multimap B), A \vdash \Gamma, !(A \multimap B), B$$

Lineare Logik und Petrinetze (3)

- Schalten des Netzes:



- Dargestellt als Ableitung in Linearer Logik:

  ... weiter kann abgeleitet werden:

$$\Gamma', !(B \otimes W_1 \multimap C_1), B \otimes W_1 \vdash \Gamma', !(B \otimes W_1 \multimap C_1), C_1$$

Linearlogische Codierung von Petrinetzen (1)

Eine linearlogische Repräsentation des Verhaltens von Transition r:

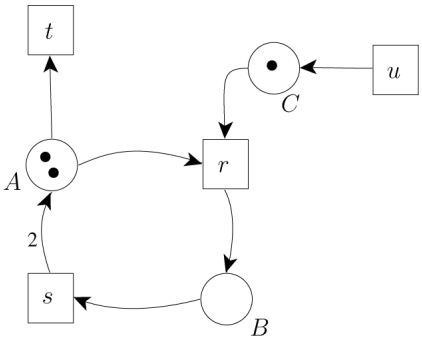
$!(A \otimes C \multimap B)$

... die Modalität stellt die Wiederverwendbarkeit von r dar.

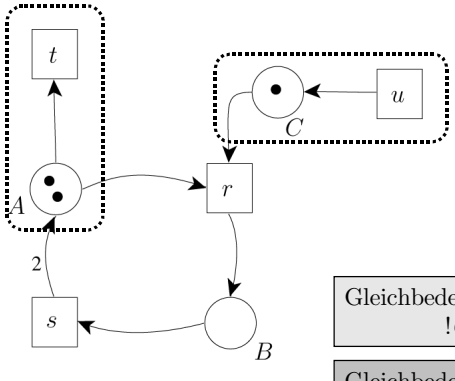
Die aktuelle Markierung:

$A \otimes A \otimes C$

Ein einfaches S/T-Netz:



Linearlogische Codierung von Petrinetzen (2)



... insgesamt ergibt sich für das Netz:

$A^2 \otimes C$

$\otimes!(A \otimes C \multimap B)$

$\otimes!(B \multimap A^2)$

$\otimes!(\mathbf{1} \multimap C)$

$\otimes!(A \multimap \perp)$

Gleichbedeutend mit: $!C$

Gleichbedeutend mit: $!(A^{\perp})$

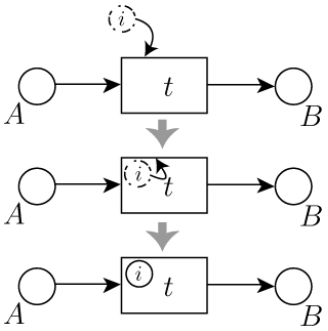
Einweg-Transitionen (1)

Transitionen, die maximal einmal Schalten können und daraufhin aus der Netzstruktur verschwinden.

Zur Darstellung:

Aus traditionellen Transitionen ...

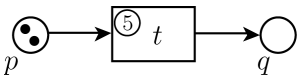
... werden Einweg-Transitionen.



Für i=1 entspricht ihr die Implikation $A \multimap B$

Einweg-Transitionen (2)

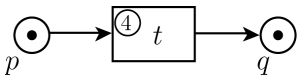
Schalten von Einweg-Transitionen:



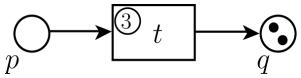
$$p \otimes p \otimes (p \multimap q) \otimes (p \multimap q) \otimes (p \multimap q) \otimes (p \multimap q) \otimes (p \multimap q)$$

oder kurz: $p^2 \otimes (p \multimap q)^5$

1. Schalten



2. Schalten



$$p \otimes q \otimes (p \multimap q)^4 \vdash q^2 \otimes (p \multimap q)^3$$

ist ableitbar

Linearlogische Codierung von erweiterten Petrinetzen (1)

Die *kanonische Formel* $\Psi_{\mathcal{S}(\mathbf{m})}$ für ein erweitertes S/T-Netz $\langle S, T, E, F, W, \mathbf{m}\rangle$ ist das Tensorprodukt aus folgenden Faktoren:

- Für jede *traditionelle Transition* t mit Vorbereich ${}^\bullet t$ und Nachbereich $t^\bullet$:

$$! \left(\bigotimes_{p \in {}^\bullet t} p^{W(p,t)} \multimap \bigotimes_{q \in t^\bullet} q^{W(t,q)} \right),$$

- Für jede *Einweg-Transition* e mit Vorbereich ${}^\bullet e$ und Nachbereich $e^\bullet$:

$$\bigotimes_{p \in {}^\bullet e} p^{W(p,e)} \multimap \bigotimes_{q \in e^\bullet} q^{W(e,q)},$$

- für die *aktuelle Markierung*:

$$\bigotimes_{\substack{p \in S \\ \mathbf{m}[p] \geq 1}} p^{\mathbf{m}[p]}$$

Linearlogische Codierung von erweiterten Petrinetzen (2)

Adäquatheit der Repräsentation:

Der Kalkül $\mathbf{ILL}_{\mathrm{PN}}{}'$ ist ein Fragment des *intuitionistischen linearlogischen Sequenzenkalküls* mit den Konnektoren $\otimes$, $\multimap$ und !.

Ein Vollständigkeits- und Korrektheitstheorem:

Theorem. Eine Markierung $\mathbf{m}'$ ist im erweiterten S/T-Netz $\mathcal{S}(\mathbf{m}) = \langle P, T, D, F, W, \mathbf{m}_0\rangle$ erreichbar genau dann, wenn für die kanonische Formel die Sequenz $\Psi_{\mathcal{S}(\mathbf{m})} \vdash \Psi_{\mathcal{S}(\mathbf{m}')}$ in $\mathbf{ILL}_{\mathrm{PN}}{}'$ ableitbar ist.

Der Beweis erfolgt direkt, ohne Hilfskonstruktionen, die keine Petrinetztheoretische Interpretation erlauben.

Ein analoges Theorem ist auch für traditionelle S/T-Netze beweisbar [Brown 1989].

Kombination von Linearer Logik mit Petrinetzen

Grundgedanken:

- Verwendung linearlogischer Formeln als *Marken*,
- eine Formel darf sich gemäß den Regeln des zugrundeliegenden Kalküls verändern, ohne daß eine Transition schaltet (*autonomer Ableitungsschritt*),
- Kanten erhalten *Multimengen von Variablen* als Inschrift,
- beim Schalten einer Transition werden Formeln an die Variablen der Kanteninschriften gebunden,
- Eine Transition kann nur schalten, falls ihr *Guard* erfüllt ist,
- ein *Guard* besteht aus linearlogischen Sequenzen, welche die Variablen der Eingangs- und Ausgangskanten beinhalten.

Transformation für intuitionistische Logik

... außer der Implikation lassen sich auch die anderen Konnektoren leicht mit Linearer Logik darstellen:

$$
\begin{array}{lcl}
(A \boxed{\rightarrow} B)^\circ & = & !A^\circ \multimap B^\circ \\
(A \wedge B)^\circ & = & A^\circ \,\&\, B^\circ \\
(A \vee B)^\circ & = & !A^\circ \oplus !B^\circ
\end{array}
$$

Diese Übersetzung funktioniert nicht nur auf der Ebene von Theoremen, sondern auch auf der Beweisebene!

Interessanter Aspekt des Curry-Howard-Isomorphismus:
Die Übersetzung von ***LJ*** nach ***ILL*** induziert die Existenz eines Compilers von Programmen, die zu den Beweisen von ***LJ*** gehören, zu den Programmen der entsprechenden Beweise in ***ILL***.

Sequenzenkalkül für intuitionistische Lineare Logik *ILL*

... zunächst einige Regeln, die unverändert bleiben:

Axiom:

$$(\text{Id}) \quad \frac{}{A \vdash A}$$ ◀····· beliebige Formel

Structural Rule:

$$(\text{Exchange}) \quad \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}$$

Cut Rule:

$$\frac{\Gamma \vdash A \quad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}$$

intuitionist-
ischer Kalkül

Sequenzenkalkül für intuitionistische Lineare Logik *ILL* (2)

Logical Rules:

$$(\mathbf{1} - R) \quad \frac{}{\vdash \mathbf{1}} \qquad (\mathbf{1} - L) \quad \frac{\Gamma \vdash A}{\Gamma, \mathbf{1} \vdash A}$$

$$(\otimes - R) \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} \qquad (\otimes - L) \quad \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}$$

$$(\multimap - R) \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} \qquad (\multimap - L) \quad \frac{\Gamma \vdash A \quad B, \Delta \vdash C}{\Gamma, A \multimap B, \Delta \vdash C}$$

$$(\& - R) \quad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \,\&\, B} \qquad (\& - L) \quad \frac{\Gamma, A \vdash C}{\Gamma, A \,\&\, B \vdash C} \quad \frac{\Gamma, B \vdash C}{\Gamma, A \,\&\, B \vdash C}$$

... nur eine Art der Implikation, da die Antezedenten und Sukzedenten multiplikativ interpretiert werden.

Sequenzenkalkül für intuitionistische Lineare Logik *ILL* (3)

$$(\oplus - R)\quad \frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} \qquad (\oplus - L)\quad \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C}$$

$$(! - R)\quad \frac{!\Gamma \vdash A}{!\Gamma \vdash !A} \qquad (\text{Dereliction})\quad \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}$$

... für Sequenzen Γ von Formeln wird definiert:

$$!\Gamma := !A_1, \ldots, !A_k$$

... und nun das Wichtigste, die veränderten Strukturregeln:

$$(\text{Contraction})\quad \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B} \qquad (\text{Weakening})\quad \frac{\Gamma \vdash B}{\Gamma, !A \vdash B}$$

Nur anwendbar, wenn die Formel „klassischen Charakter“ hat!

Ein Ausflug zu den natürlichen Sprachen

Lambda-Kalkül

- Der Lambda-Kalkül ist eine Kalkül über Funktionen.
- Hauptbestandteile:
  - *Abstraktion* zur Definition von Funktionen
  - *Anwendung* (Applikation) zur Berechnung von Funktionswerten
- Typisierung:
  - Sei M ein Ausdruck, der (möglicherweise) die Variable x enthält. Dann ist $\lambda\, x.M$ die Funktion, die x als Parameter hat und deren Funktionswert durch Auswertung des Ausdruckes M bestimmt wird.
  - Dem *formalen Parameter* der Funktion wird der *Vorbereich* (domain) als Typ zugewiesen: z.B. $\lambda\, x{:}\mathrm{int}.x$ stellt die Identitätsfunktion auf den ganzen Zahlen dar.
  - Anwendung: $(\lambda\, x{:}\mathrm{int}.x)5 = 5$

. . .nochmal zurück zur Termzuweisung für LJ

Terme und Typen: (Formeln als Typen)

Cut Rule:

konkreter Term vom Typ A — Variable vom Typ A — Term vom Typ B

$$\frac{\Gamma \vdash t : A \quad x : A, \Delta \vdash u : B}{\Gamma, \Delta \vdash u[t/x] : B}$$

Variable x in u ersetzt durch t. Der Typ von u bleibt unverändert!

Logical Rules:

$$(\wedge - R) \quad \frac{\Gamma \vdash t : A \quad \Delta \vdash u : B}{\Gamma, \Delta \vdash \langle t, u \rangle : A \wedge B}$$

$$(\wedge - L) \quad \frac{\Gamma, x : A \vdash t : C}{\Gamma, z : A \wedge B \vdash t[\mathtt{fst}(z)/x] : C} \qquad \frac{\Gamma, y : B \vdash u : C}{\Gamma, z : A \wedge B \vdash u[\mathtt{snd}(z)/y] : C}$$

$$(\boxed{\rightarrow} - R) \quad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \boxed{\rightarrow} B} \qquad (\boxed{\rightarrow} - L) \quad \frac{\Gamma \vdash t : A \quad x : B, \Delta \vdash u : C}{\Gamma, f : A \boxed{\rightarrow} B, \Delta \vdash u[(ft)/x] : C}$$

Operationale Semantik

- Intuitionistische Logik (***LJ***) verhielt sich *völlig neutral* hinsichtlich der Evaluationsstrategie für den Lambdakalkül.
- In Linearer Logik ist dies fundamental anders:
  Die logische Struktur der Typen gibt vor, welche Art der Evaluation verwendet werden muß.

Eager Evaluation

- z.B. multiplikative Konjunktion: Terme vom Typ $A \otimes B$ werden beim Verbrauch zu einem Paar evaluiert. Es werden beide Komponenten verbraucht => eager evaluation.
- Lolli: Ein Term vom Typ $A \multimap B$ evaluiert zu einer Abstraktion, Linearitätsbed. => call-by-value => eager evaluation.
- additive Konjunktion: Ein Term vom Typ $A \oplus B$ wird evaluiert zu $\mathtt{inl}(t)$ oder $\mathtt{inr}(t)$ und t wird dann verwendet, um den rechten oder linken Zweig der `case`-Anweisung zu evaluieren.

Operationale Semantik (2)

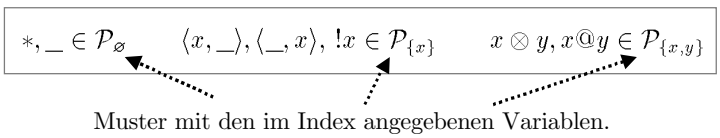
- Anders verhalten sich:

Lazy Evaluation

- additive Konjunktion:
  Terme vom Typ $A\&B$ werden auch zu einem Paar evaluiert, bei dem jedoch genau eine Komponente verwandt wird.
  Es ist nicht vorhersehbar, welche der Komponenten dies sein wird !!!
  Wird also vorschnell evaluiert, so macht man im allgemeinen unnötige Berechnungsschritte => lazy evaluation ist angebrachter.
- „of course“:
  Ein Term vom Typ $!A$ kann ganz verschwinden. (Weakening)
  Vorschnelle Evaluation führt wieder zu vermeidbarem Rechenaufwand => lazy evaluation.

Termzuweisung für *ILL*

- Zu beachten: Die Syntax der Terme ist nicht kontextfrei!
  Linearitätsbedingung an die Instanzen von Variablen: Jede Eingabe muß *genau* einmal verbraucht werden.
- Zur Formalisierung verwenden wir syntaktische Kategorien von Mustern.
  X, Y, Z seien endliche Mengen von Variablen.



Termzuweisung für *ILL* (2)

... lineare Terme mit freien Variablen gemäß der Indexmange:

- $x \in \mathcal{T}_{\{x\}}$
- $* \in \mathcal{T}_{\varnothing}$
- $t \in \mathcal{T}_X, u \in \mathcal{T}_Y, X \cap Y = \varnothing \implies t \otimes u, tu \in \mathcal{T}_{X \cup Y}$
- $t, u \in \mathcal{T}_X \implies \langle t, u \rangle \in \mathcal{T}_X$
- $t \in \mathcal{T}_X \implies \mathtt{inl}(t), \mathtt{inr}(t), !t \in \mathcal{T}_X$
- $t \in \mathcal{T}_{X \cup \{x\}}, x \notin X \implies \lambda x.t \in \mathcal{T}_X$
- $t \in \mathcal{T}_X, p \in \mathcal{P}_Y, u \in \mathcal{T}_{Y \cup Z}, X \cap Z = Y \cap Z = \varnothing$
  $\implies \mathtt{let}\ t\ \mathtt{be}\ p\ \mathtt{in}\ u \in \mathcal{T}_{X \cup Z}$
- $t \in \mathcal{T}_X, u \in \mathcal{T}_{Z \cup \{x\}}, v \in \mathcal{T}_{Z \cup \{y\}}, X \cap Z = \{x, y\} \cap Z = \varnothing$
  $\implies \mathtt{case}\ t\ \mathtt{of}\ \mathtt{inl}(x) \Rightarrow u \mid \mathtt{inr}(y) \Rightarrow v \in \mathcal{T}_{X \cup Z}$

Termzuweisung für *ILL* (3)

... nun weisen wir den Sequenzen lineare Terme zu:

$x_1 : A_1, \ldots, x_k : A_k \vdash t : A$

LL-Formeln über den Konnektoren $\mathbf{1}, \otimes, \multimap, \&, \oplus, !$

x_i sind paarweise unterschiedliche Variablen und $\mathcal{T}_X,\ X = \{x_1, \ldots, x_k\}$

Folgende Regeln ergeben sich für den Termkalkül:

Axiom:

$$\text{(Id)}\quad \frac{}{x : A \vdash x : A}$$

Structural Rule:

$$\text{(Exchange)}\quad \frac{\Gamma, x : A, y : B, \Delta \vdash t : C}{\Gamma, y : B, x : A, \Delta \vdash t : C}$$

Termzuweisung für *ILL* (4)

Cut Rule:

$$\frac{\Gamma \vdash t : A \quad x : A, \Delta \vdash u : B}{\Gamma, \Delta \vdash u[t/x] : B}$$

Logical Rules:

$$(\mathbf{1} - R) \quad \frac{}{\vdash * : \mathbf{1}} \qquad (\mathbf{1} - L) \quad \frac{\Gamma \vdash t : A}{\Gamma, z : \mathbf{1} \vdash \mathtt{let}\ z\ \mathtt{be}\ *\ \mathtt{in}\ t : A}$$

$$(\otimes - R) \quad \frac{\Gamma \vdash t : A \quad \Delta \vdash u : B}{\Gamma, \Delta \vdash t \otimes u : A \otimes B} \qquad (\otimes - L) \quad \frac{\Gamma, x : A, y : B \vdash t : C}{\Gamma, z : A \otimes B \vdash \mathtt{let}\ z\ \mathtt{be}\ x \otimes y\ \mathtt{in}\ t : C}$$

$$(\multimap - R) \quad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \multimap B} \qquad (\multimap - L) \quad \frac{\Gamma \vdash t : A \quad x : B, \Delta \vdash u : C}{\Gamma, f : A \multimap B, \Delta \vdash u[(ft)/x] : C}$$

Termzuweisung für *ILL* (5)

$$(\mathbin{\&} - R) \quad \frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B}{\Gamma \vdash \langle t, u \rangle : A \mathbin{\&} B}$$

$$(\mathbin{\&} - L) \quad \frac{\Gamma, x : A \vdash t : C}{\Gamma, z : A \mathbin{\&} B \vdash \mathtt{let}\ z\ \mathtt{be}\ \langle x, \_ \rangle\ \mathtt{in}\ t : C} \qquad \frac{\Gamma, y : B \vdash t : C}{\Gamma, z : A \mathbin{\&} B \vdash \mathtt{let}\ z\ \mathtt{be}\ \langle \_, y \rangle\ \mathtt{in}\ t : C}$$

$$(\oplus - R) \quad \frac{\Gamma \vdash t : A}{\Gamma \vdash \mathtt{inl}(t) : A \oplus B} \qquad \frac{\Gamma \vdash u : B}{\Gamma \vdash \mathtt{inr}(u) : A \oplus B}$$

$$(\oplus - L) \quad \frac{\Gamma, x : A \vdash u : C \quad \Gamma, y : B \vdash v : C}{\Gamma, z : A \oplus B \vdash \mathtt{case}\ z\ \mathtt{of}\ \mathtt{inl}(x)\ \Rightarrow\ u \mid \mathtt{inr}(y)\ \Rightarrow\ v : C}$$

$$(! - R) \quad \frac{!\Gamma \vdash t : A}{!\Gamma \vdash !t : !A}$$

Termzuweisung für *ILL* (6)

... und zum Schluß noch die neuen „Strukturregeln“:

$$\text{(Dereliction)}\quad \frac{\Gamma, x : A \vdash t : B}{\Gamma, z : !A \vdash \texttt{let}\ z\ \texttt{be}\ !x\ \texttt{in}\ t : B}$$

$$\text{(Contraction)}\quad \frac{\Gamma, x : !A, y : !A \vdash t : B}{\Gamma, z : !A \vdash \texttt{let}\ z\ \texttt{be}\ x@y\ \texttt{in}\ t : B}$$

$$\text{(Weakening)}\quad \frac{\Gamma \vdash t : B}{\Gamma, z : !A \vdash \texttt{let}\ z\ \texttt{be}\ \_\ \texttt{in}\ t : B}$$

Jetzt soll hierfür eine operationale Semantik angegeben werden.

Operationale Semantik (3)

Klassifikation der Konnektoren:

- $\otimes, \multimap, \oplus$ Eager Evaluation
- $\&$, ! Lazy Evaluation

... insbesondere ist für die Interpretation der intuitionistischen Datentypen

$$\begin{aligned} A \times B &= A \,\&\, B \\ A \Rightarrow B &= !A \multimap B \end{aligned}$$

Der intuitionistische Funktionstyp ist also operational *call-by-name* (lazy), da das Argument durch die lazy-Evaluation von $!A$ „eingefroren“ wird.

Operationale Semantik (4)

kanonische Formen:

c, d sind kanonische Formen

$$\langle t, u\rangle \quad !t$$

$$* \quad c \otimes d \quad \lambda x.t \quad \texttt{inl}(c) \quad \texttt{inr}(d)$$

Evaluationsrelation:

$$\frac{}{*\Downarrow *} \qquad \frac{t\Downarrow * \quad u\Downarrow c}{\texttt{let}\ t\ \texttt{be}\ *\ \texttt{in}\ u\Downarrow c}$$

$$\frac{t\Downarrow c \quad u\Downarrow d}{t \otimes u\Downarrow c \otimes d} \qquad \frac{t\Downarrow c \otimes d \quad u[c/x, d/y]\Downarrow e}{\texttt{let}\ t\ \texttt{be}\ x \otimes y\ \texttt{in}\ u\Downarrow e}$$

$$\frac{}{\lambda x.t\Downarrow\lambda x.t} \qquad \frac{t\Downarrow\lambda x.v \quad u\Downarrow c \quad v[c/x]\Downarrow d}{tu\Downarrow d}$$

Operationale Semantik (5)

Evaluationsrelation (Forts.):

$$\frac{}{\langle t, u\rangle\Downarrow\langle t, u\rangle} \qquad \frac{t\Downarrow\langle v, w\rangle \quad v\Downarrow c \quad u[c/x]\Downarrow d}{\texttt{let}\ t\ \texttt{be}\ \langle x, \_\rangle\ \texttt{in}\ u\Downarrow d} \qquad \frac{t\Downarrow\langle v, w\rangle \quad w\Downarrow c \quad u[c/y]\Downarrow d}{\texttt{let}\ t\ \texttt{be}\ \langle \_, y\rangle\ \texttt{in}\ u\Downarrow d}$$

$$\frac{t\Downarrow c}{\texttt{inl}(t)\Downarrow\texttt{inl}(c)} \qquad \frac{u\Downarrow d}{\texttt{inr}(u)\Downarrow\texttt{inr}(d)}$$

$$\frac{t\Downarrow\texttt{inl}(c) \quad u[c/x]\Downarrow d}{\texttt{case}\ t\ \texttt{of}\ \texttt{inl}(x)\ \Rightarrow\ u \mid \texttt{inr}(y)\ \Rightarrow\ v\Downarrow d} \qquad \frac{t\Downarrow\texttt{inr}(c) \quad v[c/y]\Downarrow d}{\texttt{case}\ t\ \texttt{of}\ \texttt{inl}(x)\ \Rightarrow\ u \mid \texttt{inr}(y)\ \Rightarrow\ v\Downarrow d}$$

$$\frac{}{!t\Downarrow !t} \qquad \frac{t\Downarrow !v \quad v\Downarrow c \quad u[c/x]\Downarrow d}{\texttt{let}\ t\ \texttt{be}\ !x\ \texttt{in}\ u\Downarrow d}$$

$$\frac{t\Downarrow !v \quad u\Downarrow c}{\texttt{let}\ t\ \texttt{be}\ \_\ \texttt{in}\ u\Downarrow c} \qquad \frac{t\Downarrow !v \quad u[!v/x, !v/y]\Downarrow c}{\texttt{let}\ t\ \texttt{be}\ x@y\ \texttt{in}\ u\Downarrow c}$$