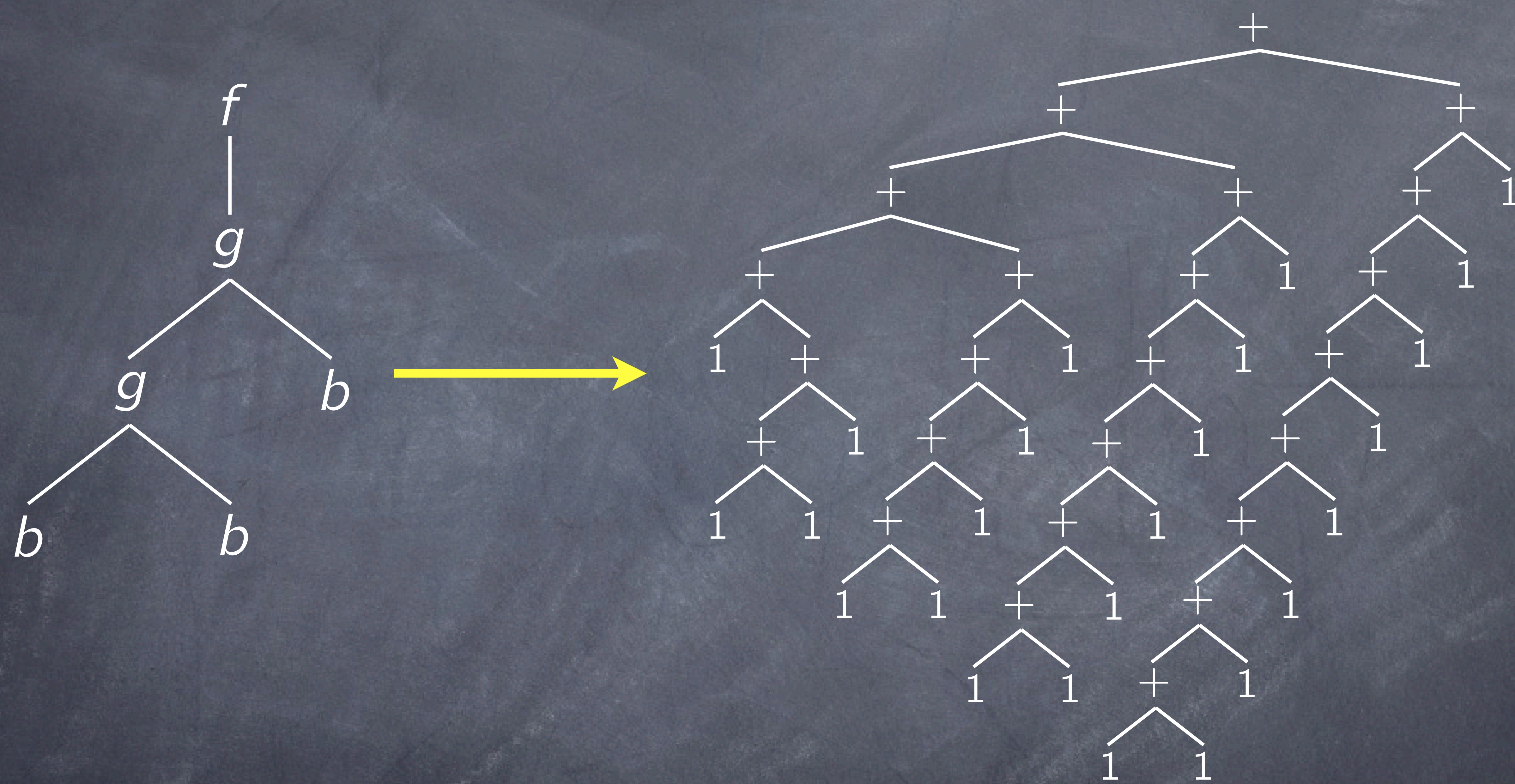
A Lambda Calculus Characterization of MSO Definable Tree Transductions

Makoto Kanazawa
National Institute of Informatics
Tokyo, Japan

Tree transductions



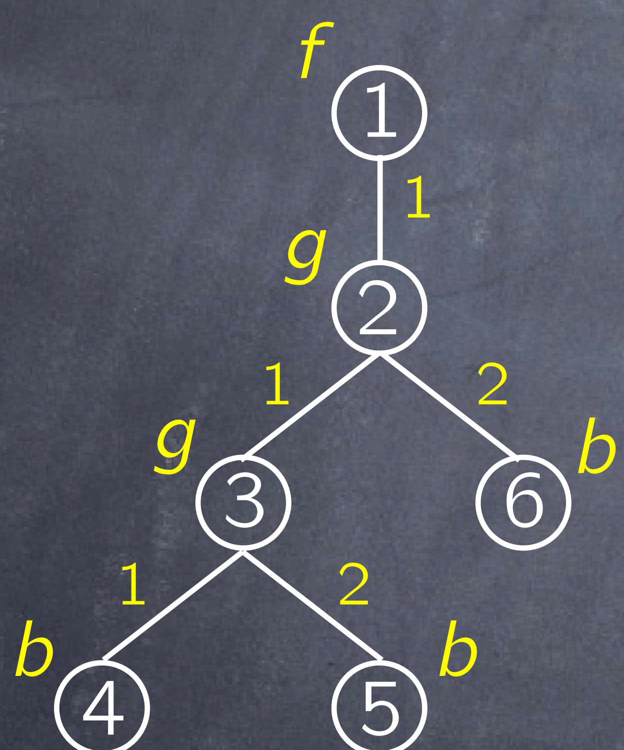
Tree transductions defined by

- MSO term graph transducers
- λ-homomorphisms

Monadic Second-Order Logic

First-order logic + $\forall X$, $\exists X$ set variables

Graphs as relational structures



$lab_f(x)$

$lab_g(x)$ $edg_1(x, y)$

$lab_b(x)$ $edg_2(x, y)$

unary predicates binary predicates

node labels

$\Sigma = \{f, g, b\}$

$\Gamma = \{1, 2\}$

edge labels

$$G = (V_G, lab_f^G, lab_g^G, lab_b^G, edg_1^G, edg_2^G)$$

$$V_G = \{1, 2, 3, 4, 5, 6\}$$

$$lab_f^G = \{1\}, lab_g^G = \{2, 3\}, lab_b^G = \{4, 5, 6\}$$

$$edg_1^G = \{(1, 2), (2, 3), (3, 4)\}, edg_2^G = \{(2, 6), (3, 5)\}$$

MSO definable sets of graphs

- A class $\mathscr{G}$ of graphs over (Σ, Γ) is MSO definable if

$$\mathscr{G} = \{\, G \in \mathrm{GR}(\Sigma, \Gamma) \mid G \models \varphi \,\}$$

for some closed MSO formula φ

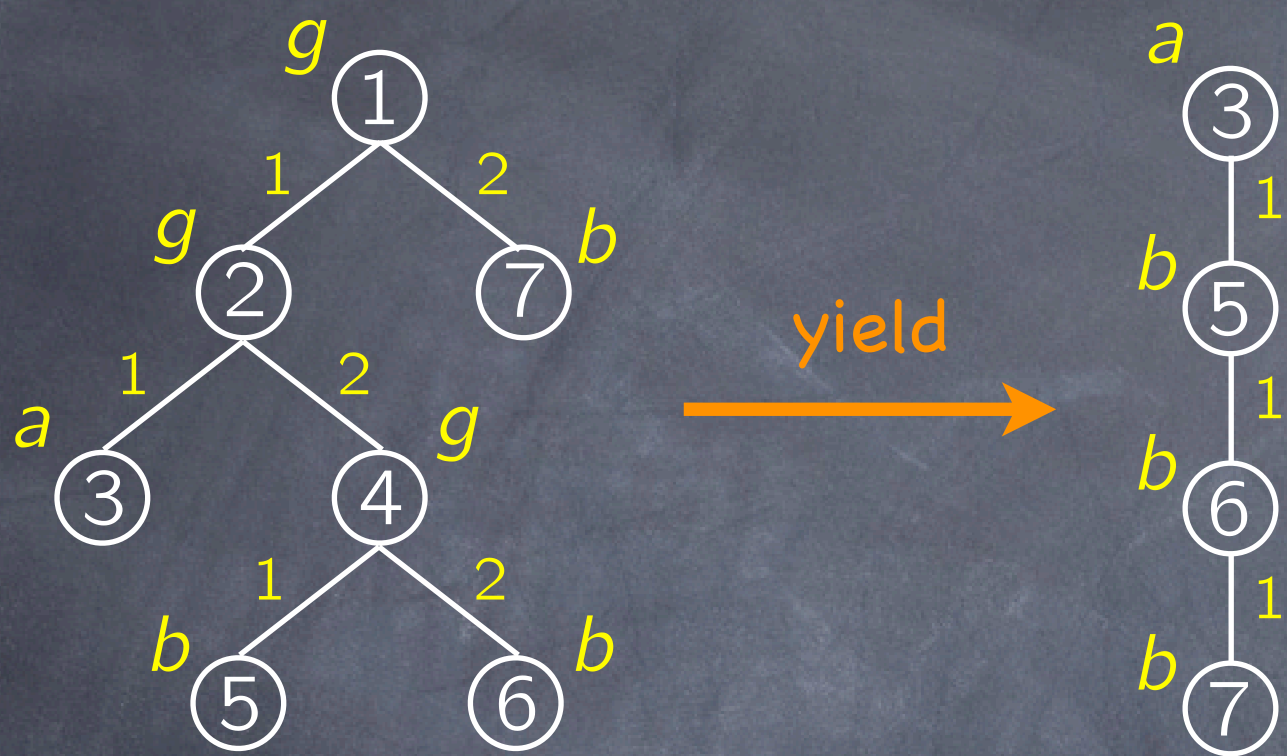
$$\forall xy \forall X (\forall zw((edg(z,w) \vee edg(w,z)) \rightarrow (z \in X \rightarrow w \in X)) \rightarrow (x \in X \rightarrow y \in X))$$

"G is connected"

- (Buchi 1960, Elgot 1961, Trachtenbrot 1962) A set L of strings is MSO definable iff L is regular

- (Thatcher & Wright 1968, Doner 1970) A set L of trees is MSO definable iff L is recognizable (regular)

MSO definable graph transductions

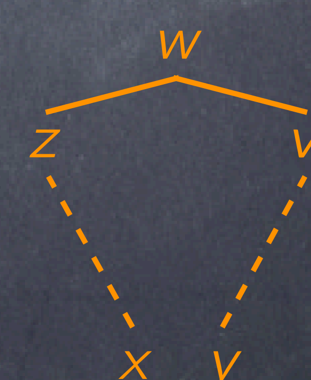


$$
\begin{aligned}
V_{G'} &= lab_a^{G'} \cup lab_b^{G'} \\
lab_a^{G'} &= \{\, v \in V_G \mid G \models \psi_a(v) \,\} \\
lab_b^{G'} &= \{\, v \in V_G \mid G \models \psi_b(v) \,\} \\
edg_1^{G'} &= \{\, (v_1, v_2) \in V_G \times V_G \mid \\
&\qquad\qquad G \models \chi_1(v_1, v_2) \,\}
\end{aligned}
$$

$$\psi_a(x) = lab_a(x)$$

$$\psi_b(x) = lab_b(x)$$

$$
\begin{aligned}
\chi_1(x, y) = \exists z w v (&edg_2^*(z, x) \wedge edg_1(w, z) \wedge \\
&edg_2(w, v) \wedge edg_1^*(v, y))
\end{aligned}
$$



$$
\begin{aligned}
R^*(x, y) = \forall X (\forall z w (R(z, w) \rightarrow (z \in X \rightarrow w \in X)) \\
\rightarrow (x \in X \rightarrow y \in X))
\end{aligned}
$$

MSO graph transducers

A (non-copying) MSO graph transducer from (Σ_1, Γ_1) to (Σ_2, Γ_2) (without parameters) (Courcelle 1991) consists of:

$$\{\, \psi_\sigma(x) \mid \sigma \in \Sigma_2 \,\}$$
$$\{\, \chi_\gamma(x, y) \mid \gamma \in \Gamma_2 \,\}$$

MSO formulas over (Σ_1, Γ_1)

A graph G is mapped to G′ with

$$V_{G'} = \bigcup_{\sigma \in \Sigma_2} lab_\sigma^{G'}$$

deterministic
size of G′ ≤ size of G

$$lab_\sigma^{G'} = \{\, v \in V_G \mid G \models \psi_\sigma(v) \,\}$$
$$lab_\gamma^{G'} = \{\, (v_1, v_2) \in V_G \times V_G \mid G \models \chi_\gamma(v_1, v_2) \,\}$$

"interpretation"

MSO graph transducers

In general, a (copying) MSO graph transducer from (Σ_1, Γ_1) to (Σ_2, Γ_2) (with parameters) (Courcelle 1993, 1994) consists of:

$X_1, \ldots, X_k$ parameters

$\varphi(X_1, \ldots, X_k)$

C set of copy names

$\{ \psi_{\sigma,c}(x, X_1, \ldots, X_k) \mid \sigma \in \Sigma_2, c \in C \}$

$\{ \chi_{\gamma,c_1,c_2}(x, y, X_1, \ldots, X_k) \mid \gamma \in \Gamma_2, c_1, c_2 \in C \}$

For each choice $U_1, \ldots, U_k$ such that $G \models \varphi(U_1, \ldots, U_k)$

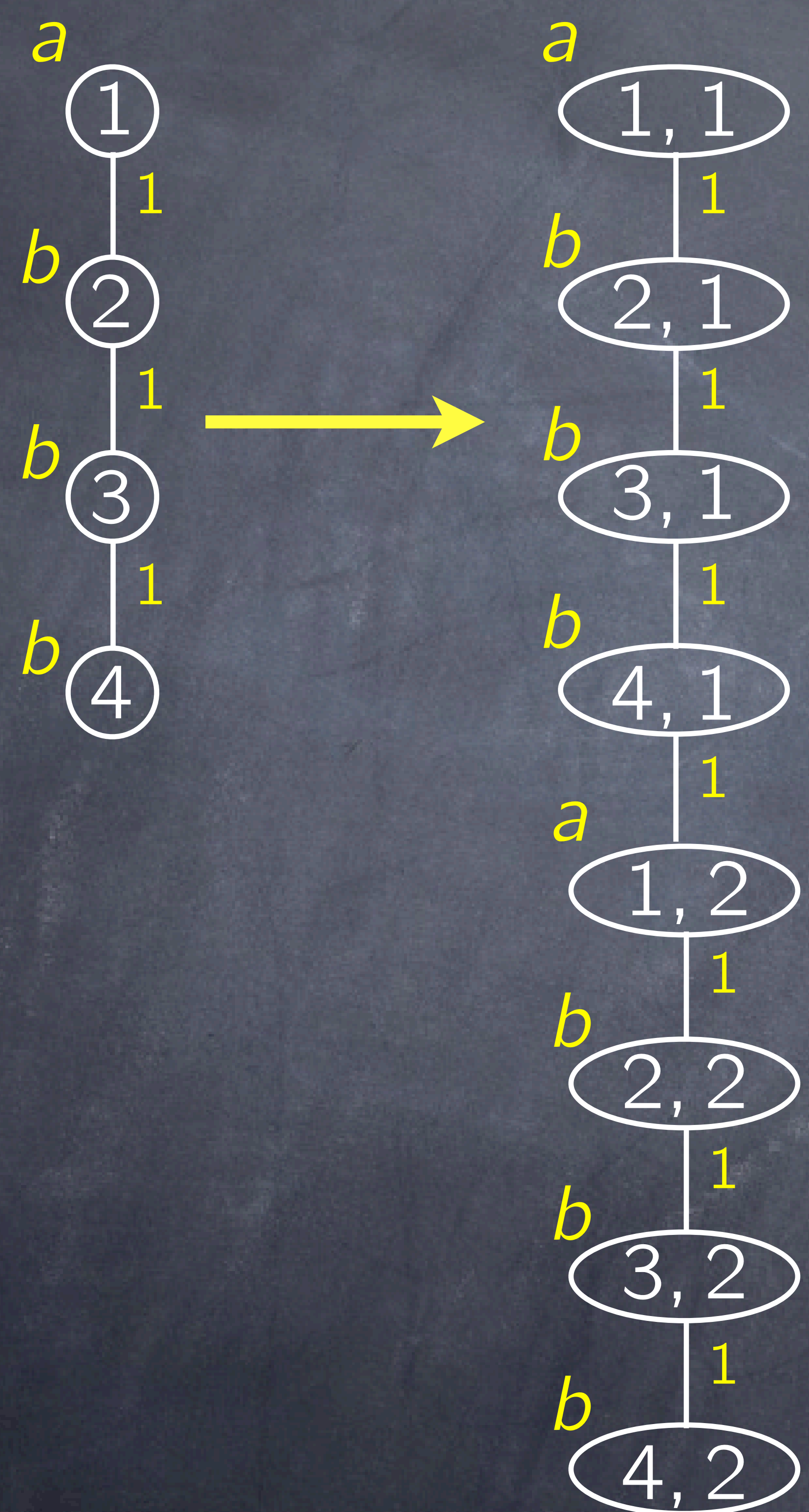
$$V_{G'} = \bigcup_{\sigma \in \Sigma_2} lab_\sigma^{G'}$$

nondeterministic
size of G′ linear in size of G

$$lab_\sigma^{G'} = \{ (v, c) \in V_G \times C \mid G \models \psi_{\sigma,c}(v) \}$$

$$lab_\gamma^{G'} = \{ ((v_1, c_1), (v_2, c_2)) \in (V_G \times C)^2 \mid G \models \chi_{\gamma,c_1,c_2}(v_1, v_2) \}$$

Copying



$$
\begin{aligned}
\varphi &= \text{“}G \text{ is a string graph”} \\
C &= \{1, 2\} \\
\psi_{a,1}(x) &= lab_a(x) \\
\psi_{b,1}(x) &= lab_b(x) \\
\chi_{1,1,1}(x, y) &= edg_1(x, y) \\
\chi_{1,1,2}(x, y) &= \neg \exists z (edg_1(x, z) \vee edg_1(z, y)) \\
\chi_{1,2,1}(x, y) &= false \\
\chi_{1,2,2}(x, y) &= edg_1(x, y)
\end{aligned}
$$

defines a deterministic string transduction

$$\{\, (w, ww) \mid w \in \{a, b\}^* \,\}$$

cf. Engelfriet and Hoogeboom 2001, 2007

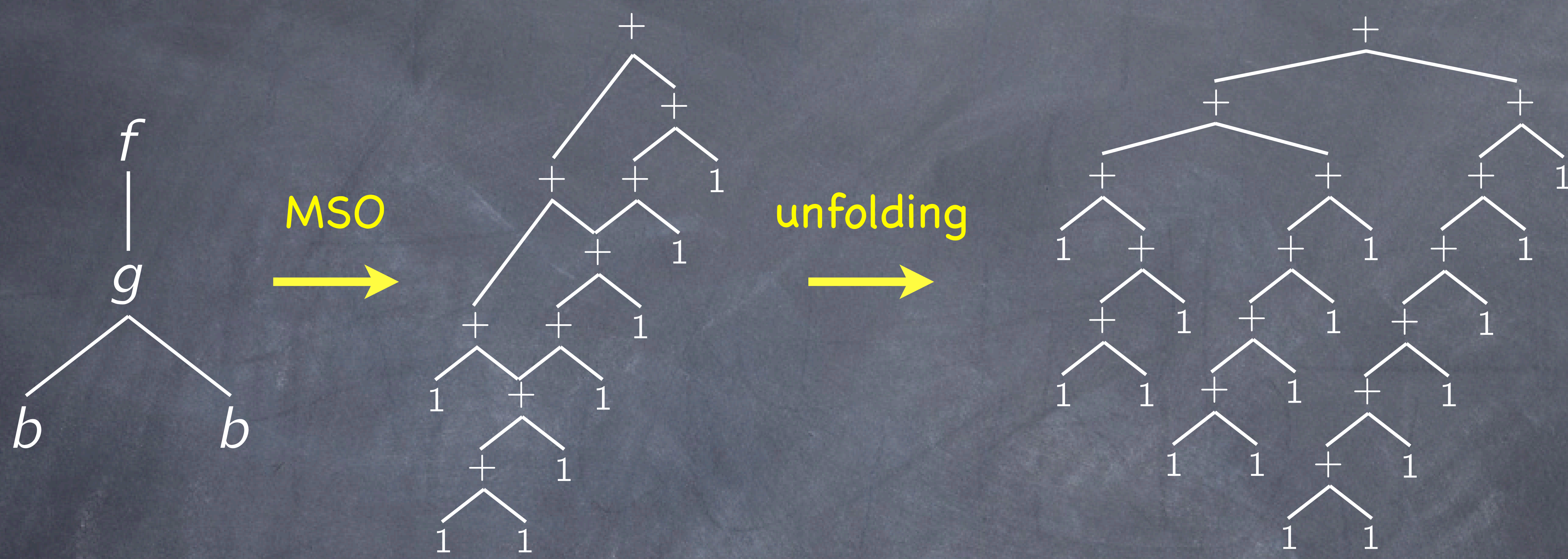
MSO graph transductions

- The class of MSO graph transduction is closed under composition
- The domain of an MSO graph transduction is MSO definable
- (Engelfriet & Hoogeboom 2001)

$$\mathrm{MSO} = \mathrm{REL} \circ \mathrm{DMSO}$$

node relabeling
determined by $R \subseteq \Sigma_1 \times \Sigma_2$

MSO term graph transducers



input tree "term graph" output tree

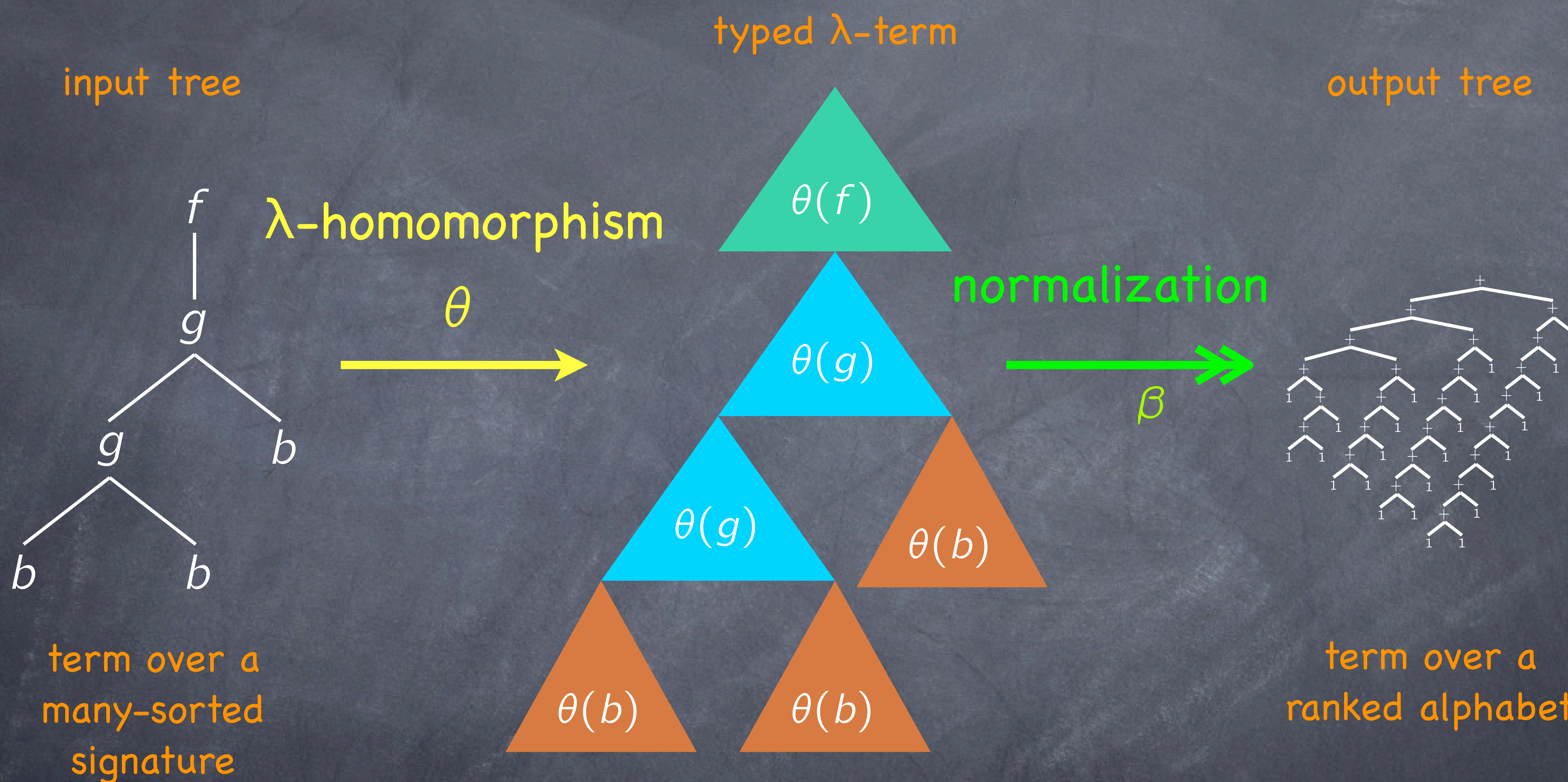
size at most exponential

(Bloem & Engelfriet 2000)

$$\text{DMSO-TGT} = \text{DMSO-REL} \circ \text{ATT}$$

DMSO term graph transducer — DMSO-definable relabeling — attributed tree transducer

λ-homomorphism (de Groote 2001)



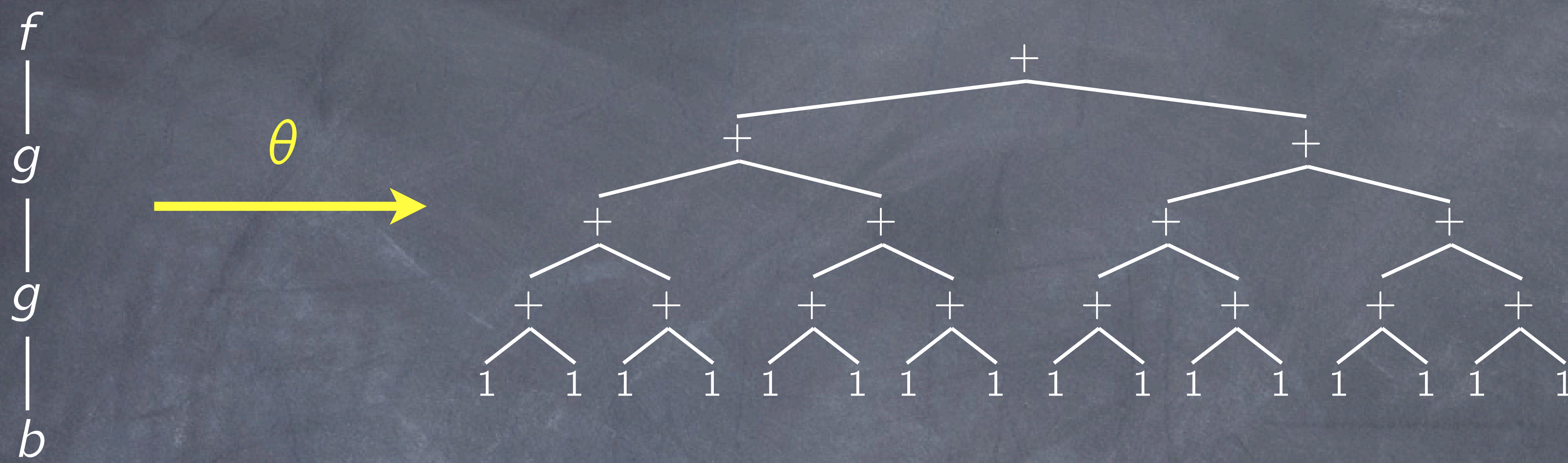
$$\theta : f \mapsto \lambda x.x((1+1)+1)1(\lambda st.t+s)$$

$$g \mapsto \lambda x_1 x_2 ijw.x_1 ij(\lambda s_1 t_1.x_2 s_1 t_1(\lambda s_2 t_2.ws_2 t_2))$$

$$b \mapsto \lambda ijw.w((i+1)+1)(j+i)$$

typed λ-terms

The power of λ



$$\theta : f \mapsto \lambda x.x1$$
$$g \mapsto \lambda xy.x(xy)$$
$$b \mapsto \lambda y.y + y$$

size doubly exponential

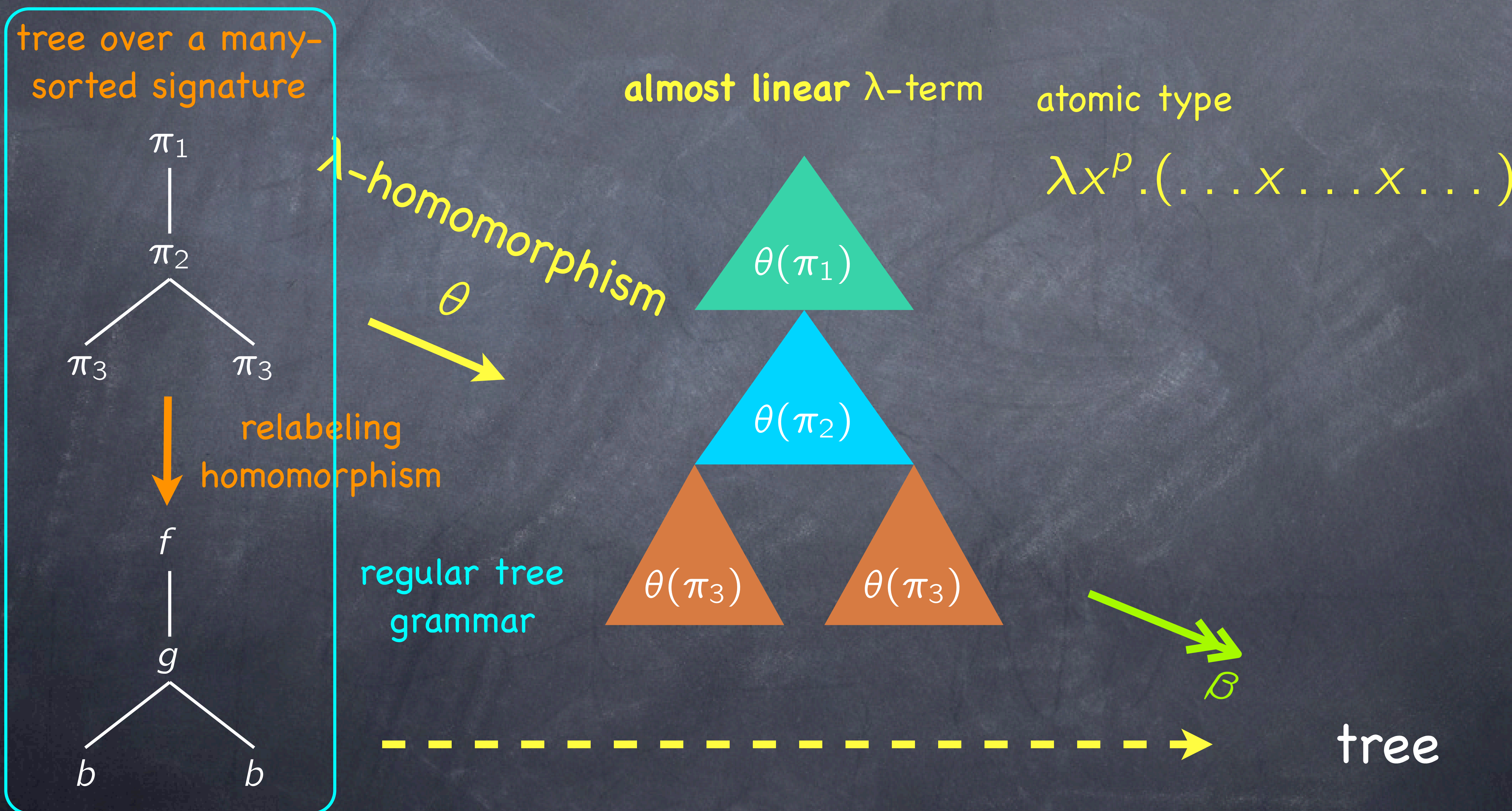
The λ-homomorphisms are closed under composition

(Salvati) $\mathsf{DMSO\text{-}TGT} \subset \mathsf{DMTT} \subset \mathsf{DMTT}^* \subseteq \lambda\text{-}\mathsf{H}$

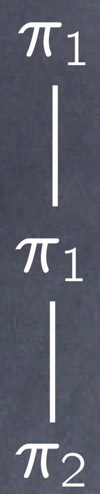
composition closure of deterministic macro tree transductions

Nondeterminism

Characterize MSO-TGT in terms of a subclass of λ-homomorphisms using almost linear λ-terms

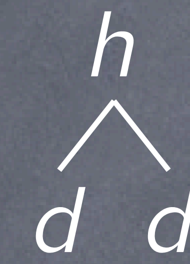


Almost linear λ-homomorphisms

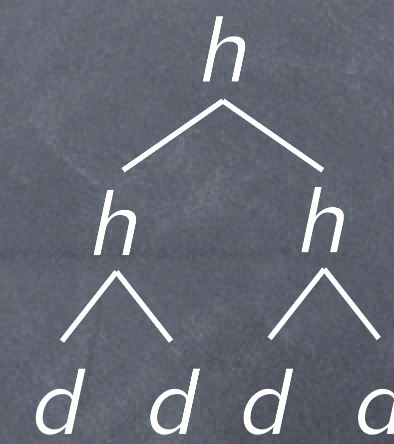


$$\theta\colon \pi_1 \mapsto \lambda x^o.h^{o\to o\to o}xx$$
$$\pi_2 \mapsto d^o$$

$\theta(\pi_1\,\pi_2) \twoheadrightarrow_\beta$



$\theta(\pi_1(\pi_1\,\pi_2)) \twoheadrightarrow_\beta$

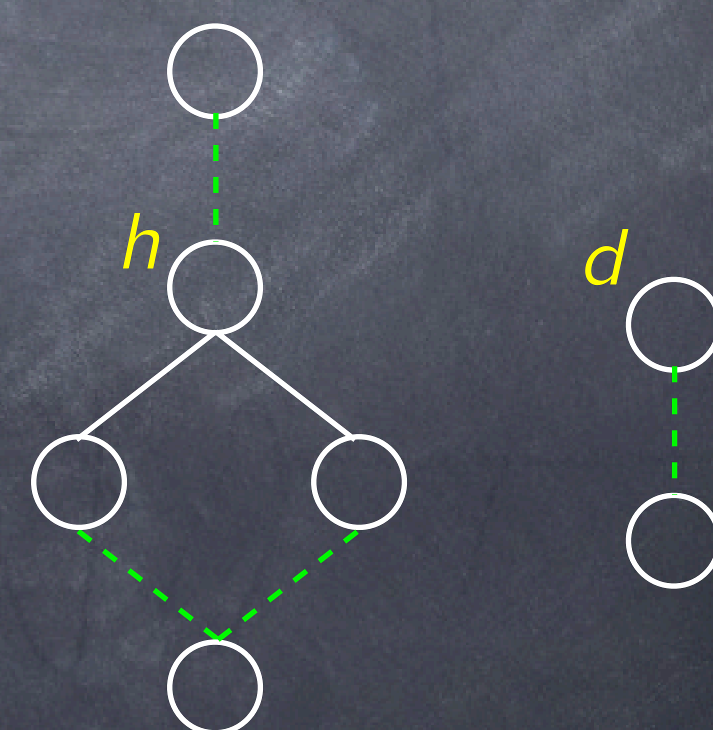


typing

$$h : o \to o \to o \vdash \lambda x.hxx : o \to o$$

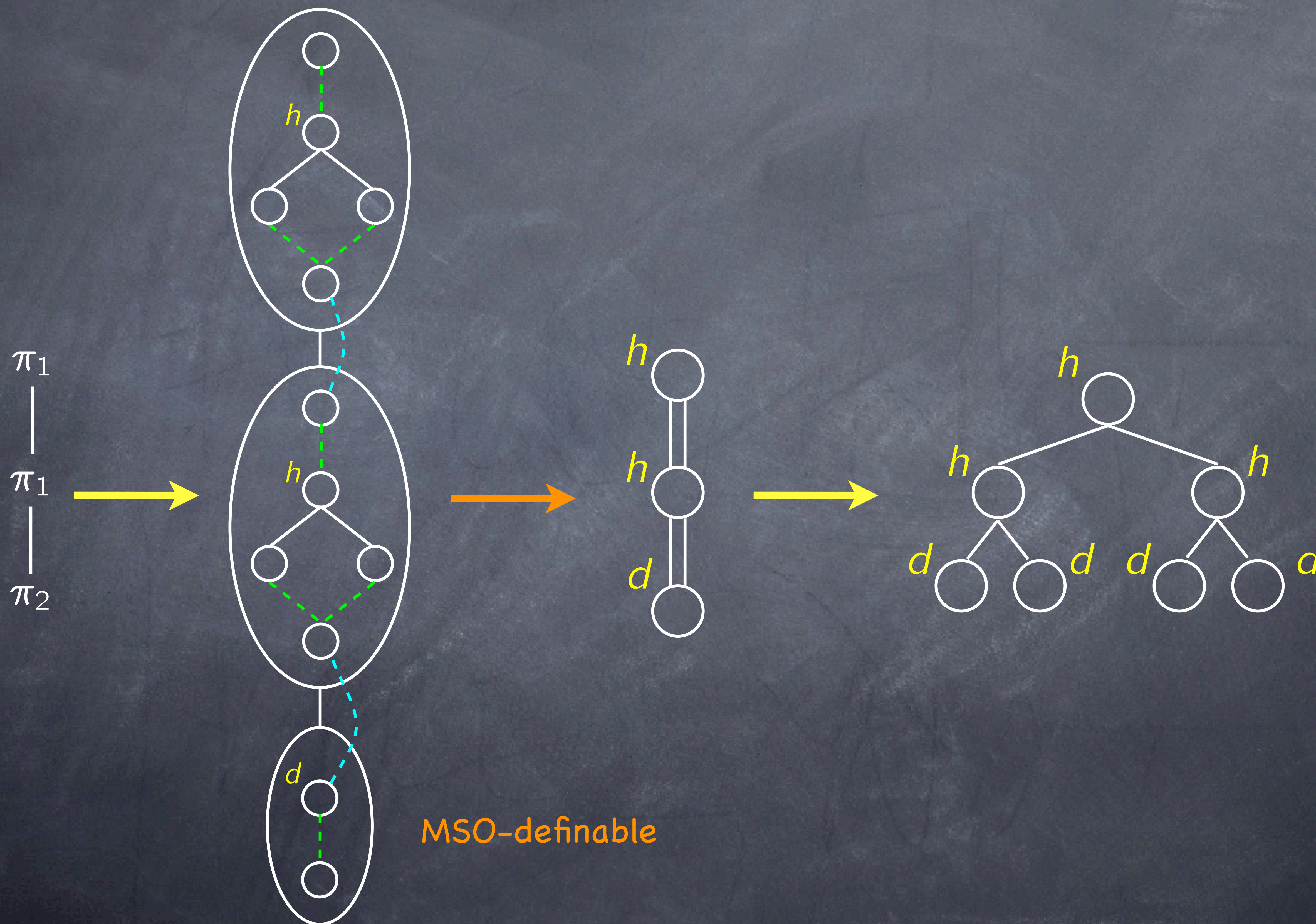
$$d : o \vdash d : o$$

Each positive occurrence of an atomic type has a unique negative counterpart (cf. Aoto 1999)

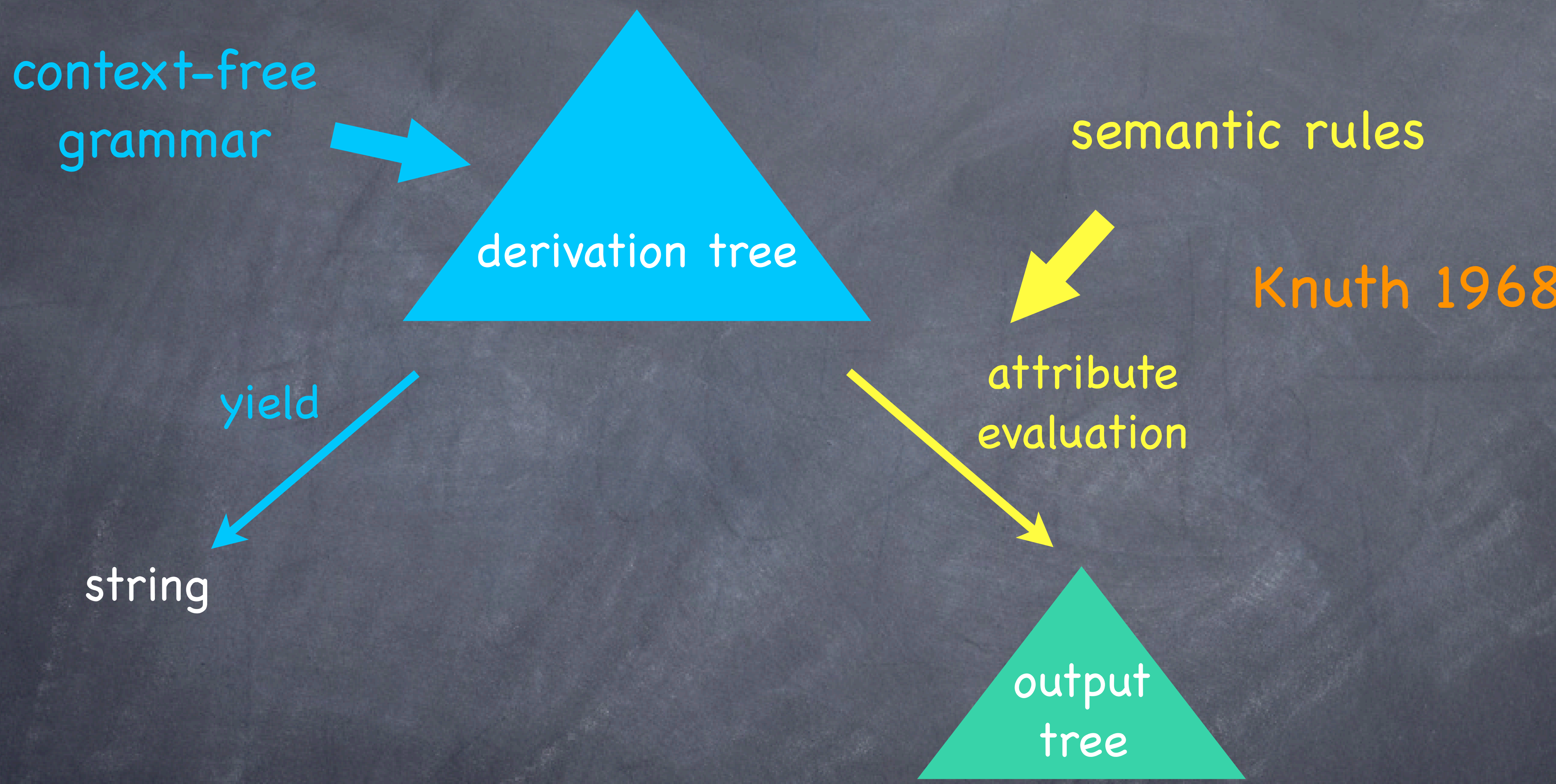


term graphs

Almost linear λ-H ⊆ MSO-TGT



Attribute grammars (outputting trees)

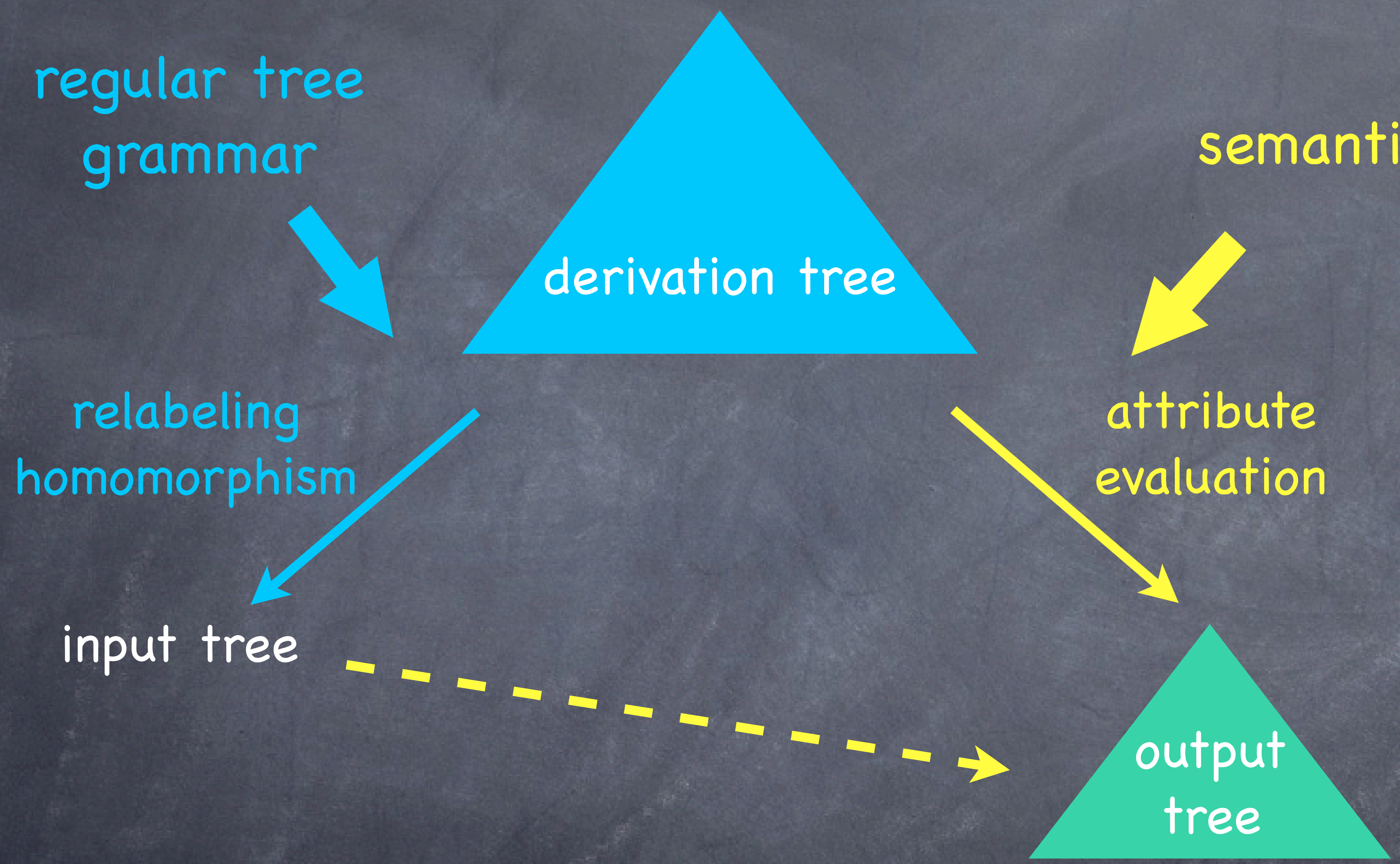


$\cap$

output language $\in OUT(\mathbf{AG})$

cf. Engelfriet and Heyker 1992

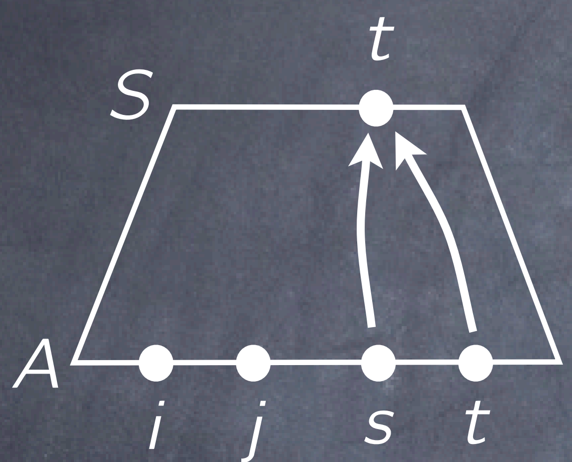
Tree-to-tree attribute grammars



MSO-TGT = TT-AG (from Bloem & Engelfriet 2000)

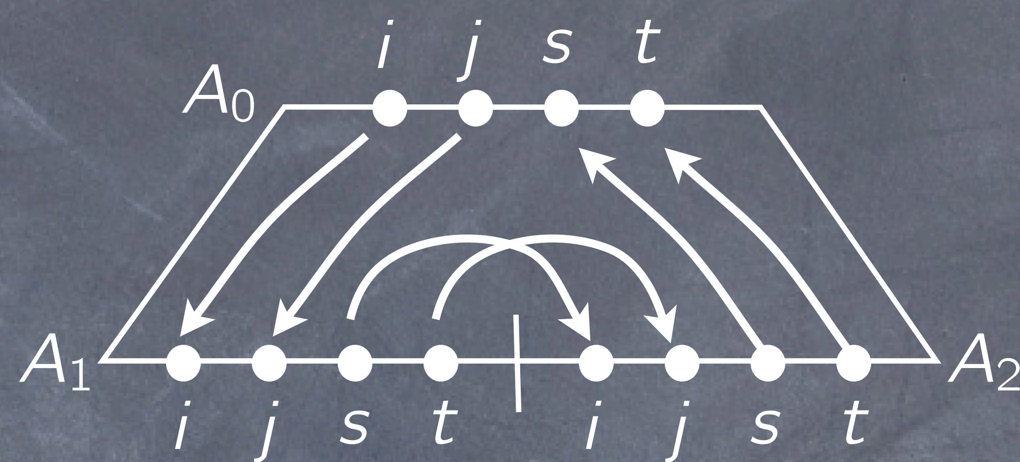
Attribute grammar: example

$\pi_1 : S \to f\ A$



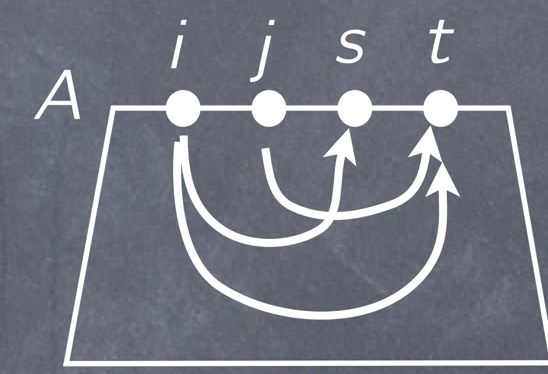
$$S.t := A.t + A.s$$
$$A.i := (1+1)+1$$
$$A.j := 1$$

$\pi_2 : A \to g\ A\ A$



$$A_0.s := A_2.s$$
$$A_0.t := A_2.t$$
$$A_1.i := A_0.i$$
$$A_1.j := A_0.j$$
$$A_2.i := A_1.s$$
$$A_2.j := A_1.t$$

$\pi_3 : A \to b$



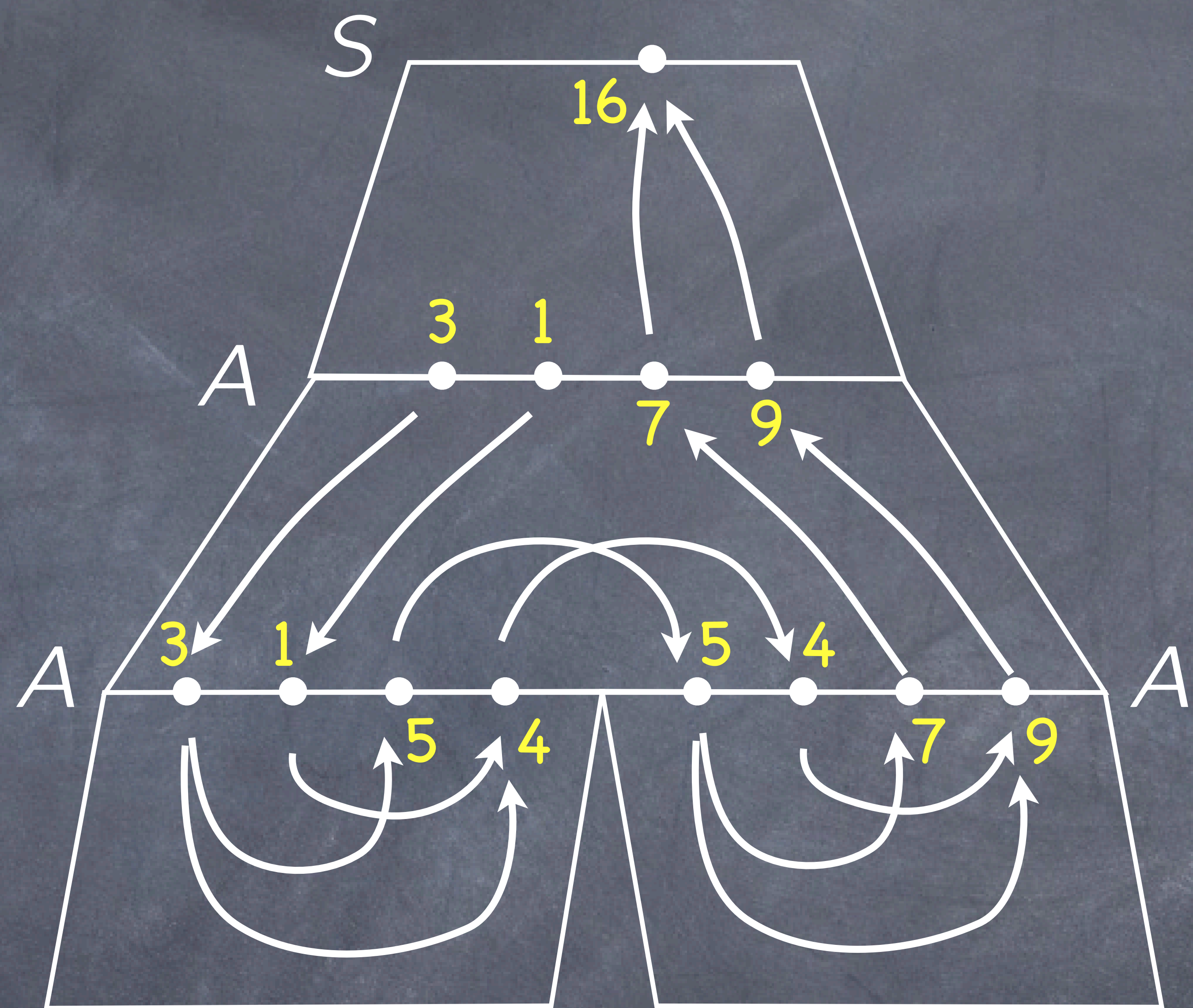
$$A.s := (A.i+1)+1$$
$$A.t := A.j + A.i$$

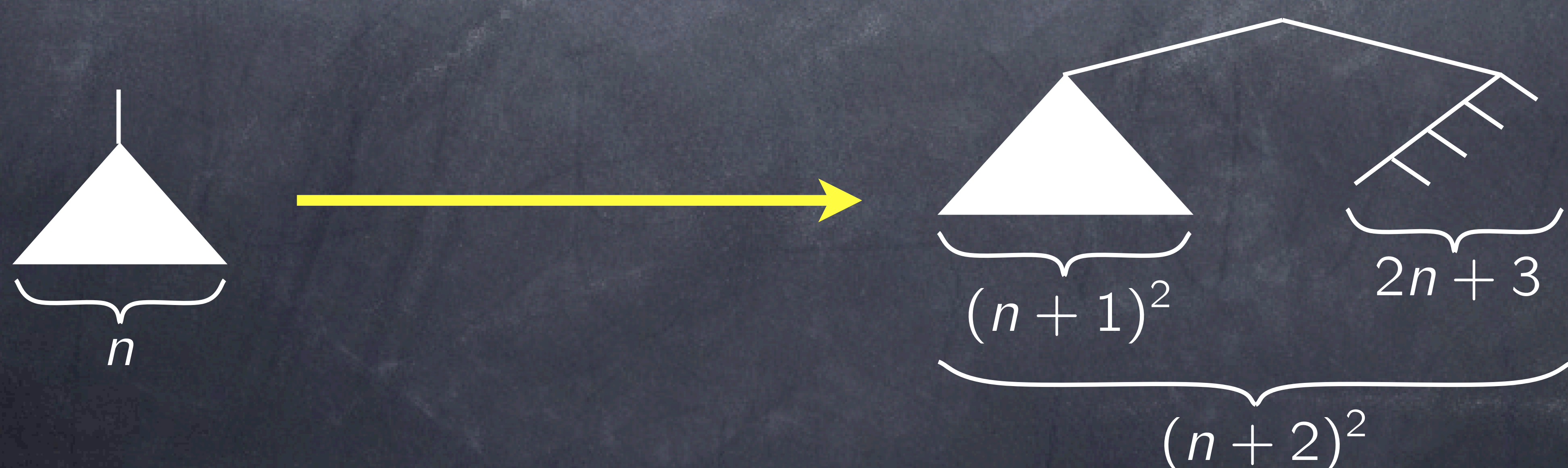
i,j: inherited attributes

s,t: synthesized attributes

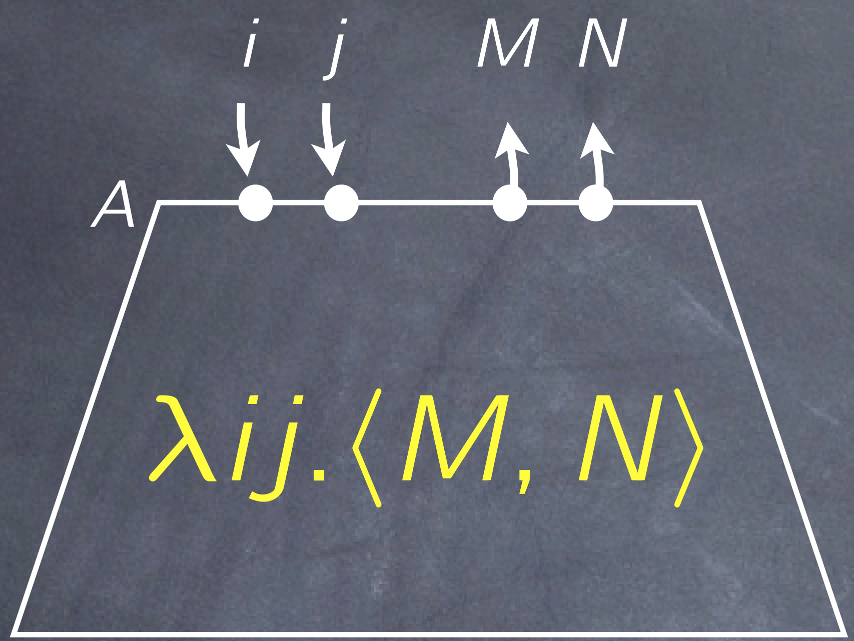
rank(1) = 0, rank(+) = 2

Attribute evaluation





Encoding in almost linear λH



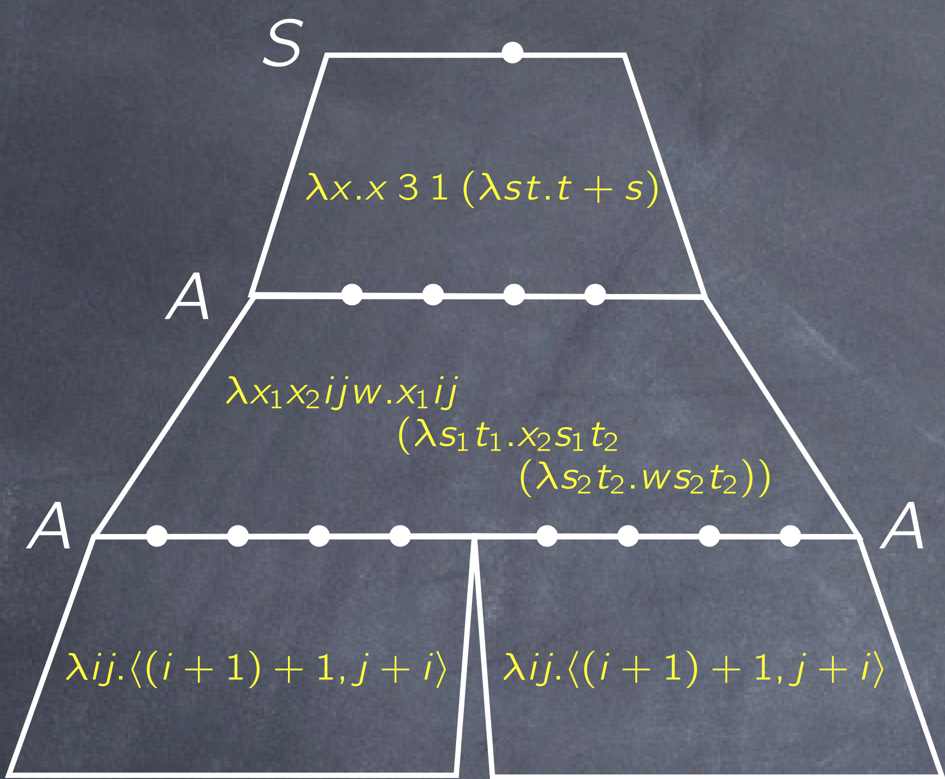
$$\langle M, N\rangle \stackrel{\text{def}}{=} \lambda w.wMN$$
$$\alpha \otimes \beta \stackrel{\text{def}}{=} (\alpha \to \beta \to o) \to o$$

$$\langle M, N\rangle(\lambda yz.P[y,z]) \twoheadrightarrow_\beta P[M,N]$$

$$\begin{aligned}
\pi_1 &:= \lambda x.x((1+1)+1)1(\lambda st.t+s) && : A \to S\\
\pi_2 &:= \lambda x_1x_2ijw.x_1ij(\lambda s_1t_1.x_2s_1t_1(\lambda s_2t_2.ws_2t_2)) && : A \to A \to A\\
\pi_3 &:= \lambda ij.\langle((i+1)+1), j+i\rangle && : A
\end{aligned}$$

$$S := o$$
$$A := o \to o \to (o \otimes o)$$

Attribute evaluation by β-reduction



$$\twoheadrightarrow_\beta \begin{pmatrix} \lambda ijw.(\lambda ij.\langle(i+1)+1, j+i\rangle)ij \\ \quad (\lambda s_1t_1.(\lambda ij.\langle(i+1)+1, j+i\rangle)s_1t_1 \\ \quad\quad (\lambda s_2t_2.ws_2t_2)) \end{pmatrix} 3\,1\,(\lambda st.t+s)$$

$$\twoheadrightarrow_\beta \begin{pmatrix} \lambda w.(\lambda ij.\langle(i+1)+1, j+i\rangle)\,3\,1 \\ \quad (\lambda s_1t_1.(\lambda ij.\langle(i+1)+1, j+i\rangle)s_1t_1 \\ \quad\quad (\lambda s_2t_2.ws_2t_2)) \end{pmatrix} (\lambda st.t+s)$$

$$\twoheadrightarrow_\beta \begin{pmatrix} \lambda w.\langle 5, 4\rangle \\ \quad (\lambda s_1t_1.(\lambda ij.\langle(i+1)+1, j+i\rangle)s_1t_1 \\ \quad\quad (\lambda s_2t_2.ws_2t_2)) \end{pmatrix} (\lambda st.t+s)$$

$$\twoheadrightarrow_\beta \begin{pmatrix} \lambda w.(\lambda ij.\langle(i+1)+1, j+i\rangle)\,5\,4 \\ \quad (\lambda s_2t_2.ws_2t_2)) \end{pmatrix} (\lambda st.t+s)$$

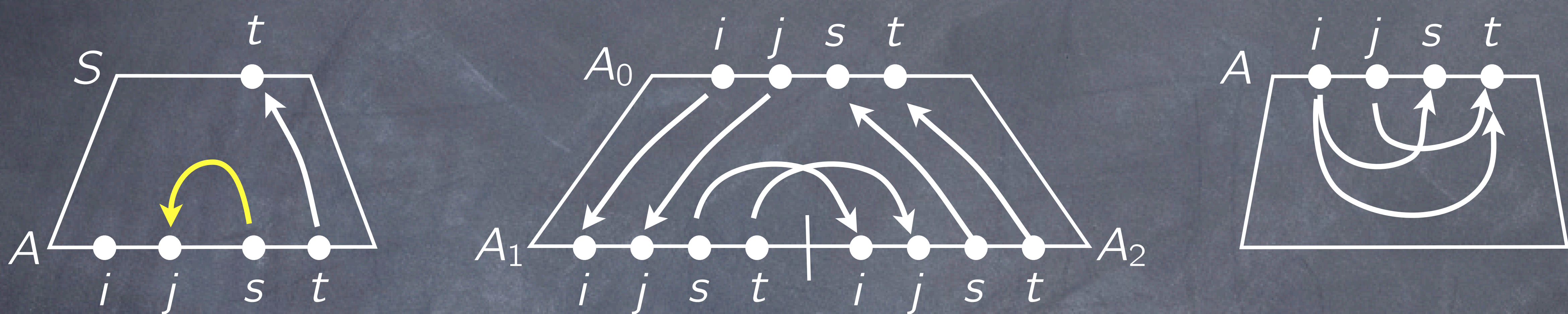
$$\twoheadrightarrow_\beta \begin{pmatrix} \lambda w.\langle 7, 9\rangle \\ \quad (\lambda s_2t_2.ws_2t_2)) \end{pmatrix} (\lambda st.t+s)$$

$$\twoheadrightarrow_\beta \langle 7, 9\rangle(\lambda st.t+s)$$

$$\twoheadrightarrow_\beta 16$$

A second example

$\pi_1 : S \to f\ A$ $\pi_2 : A \to g\ A\ A$ $\pi_3 : A \to b$



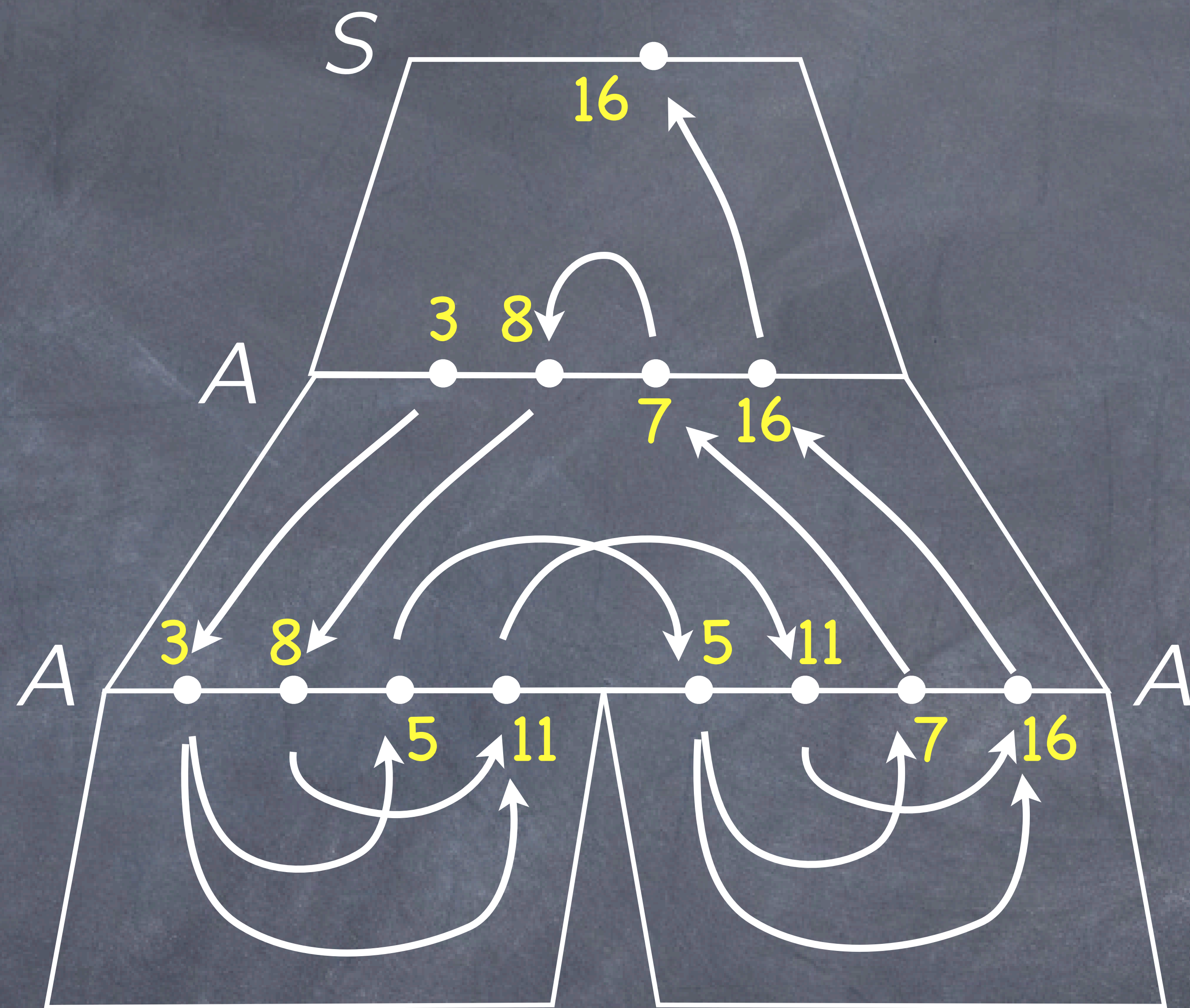
$S.t := A.t$
$A.i := (1+1)+1$
$A.j := A.s + 1$

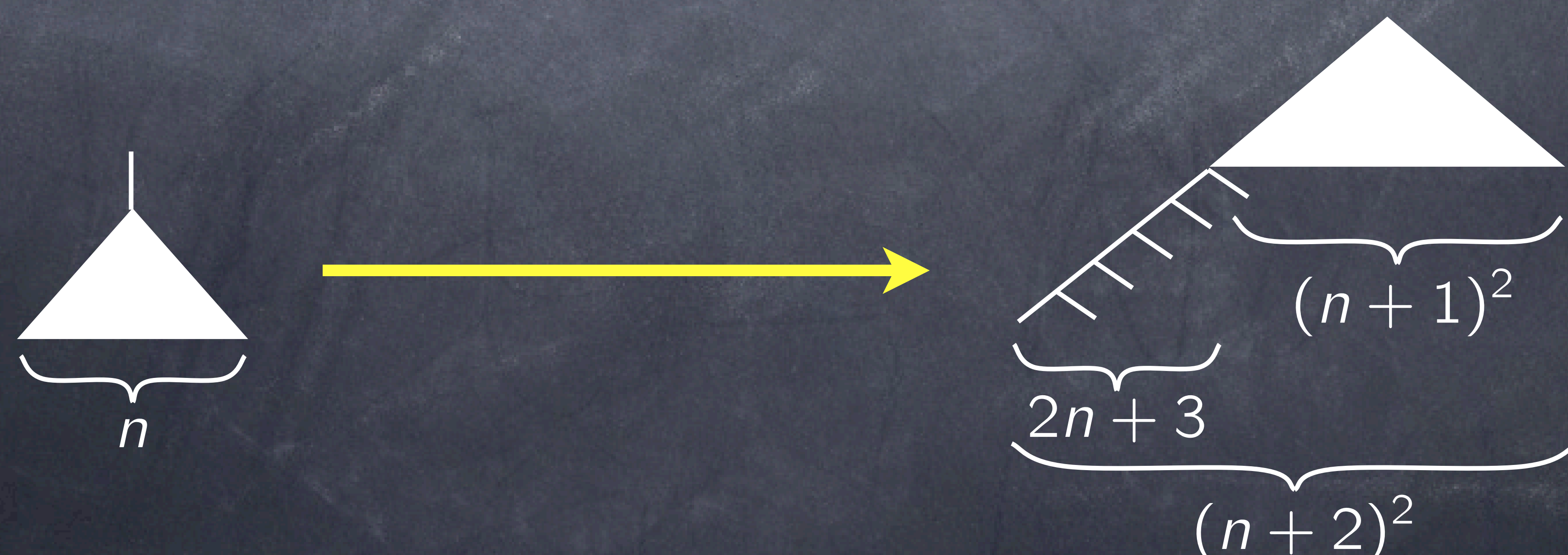
$A_0.s := A_2.s$
$A_0.t := A_2.t$
$A_1.i := A_0.i$
$A_1.j := A_0.j$
$A_2.i := A_1.s$
$A_2.j := A_1.t$

$A.s := (A.i + 1) + 1$
$A.t := A.j + A.i$

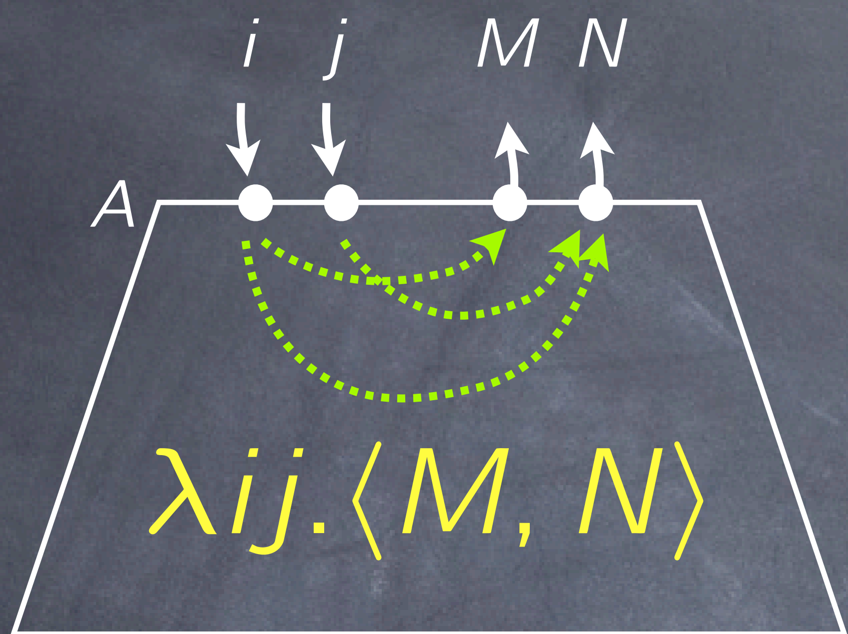
i,j: inherited attributes
s,t: synthesized attributes
rank(1) = 0, rank(+) = 2

Attribute evaluation

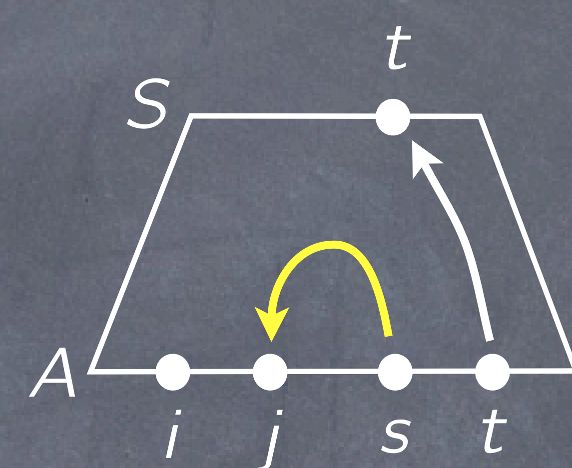




Encoding in almost linear λH?



$\pi_1 : S \to f\ A$



$$S.t := A.t$$
$$A.i := (1+1)+1$$
$$A.j := A.s + 1$$

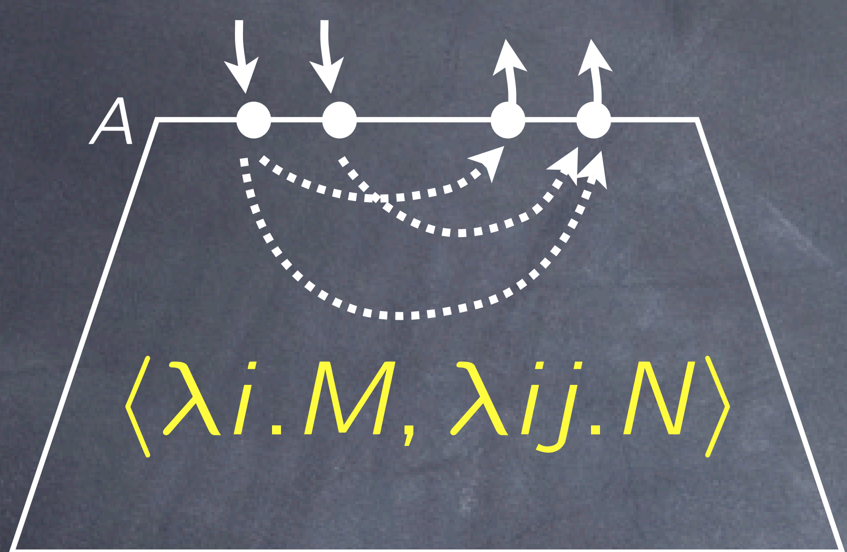
$$\pi_1 := \lambda x.x((1+1)+1)?(\lambda st.t) \qquad : A \to S$$

$$\pi_1 := \lambda x.x((1+1)+1)(x((1+1)+1)?(\lambda st.s+1))(\lambda st.t) \qquad : A \to S$$

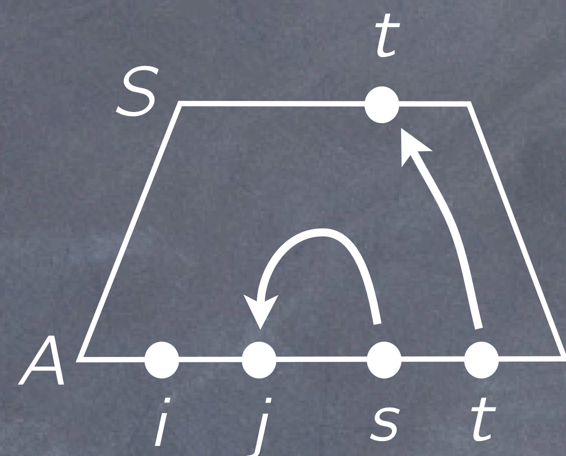
$$\pi_1 := \lambda x.x((1+1)+1)(x((1+1)+1)1(\lambda st.s+1))(\lambda st.t) \qquad : A \to S$$

$$S := o$$
$$A := o \to o \to (o \otimes o)$$

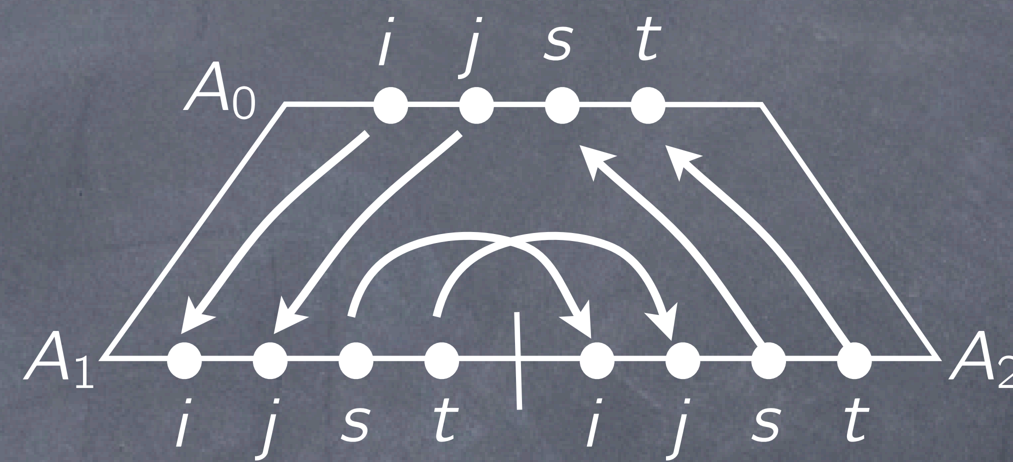
Encoding in almost linear λH?



$\pi_1 : S \to f\ A$



$\pi_2 : A \to g\ A\ A$



$$S.t := A.t$$
$$A.i := (1+1)+1$$
$$A.j := A.s + 1$$

$$\pi_1 := \lambda x.x\Big(\lambda st.t\big((1+1)+1\big)\big(s((1+1)+1)+1\big)\Big) \quad : A \to S$$

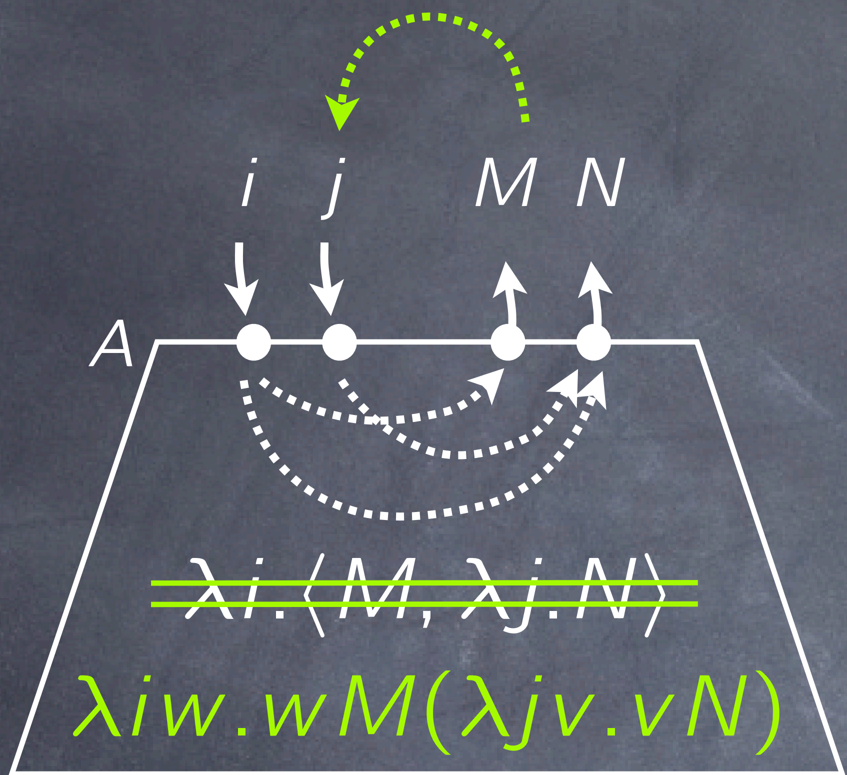
$$\pi_2 := \lambda x_1 x_2 w.x_1(\lambda s_1 t_1.x_2(\lambda s_2 t_2. \quad : A \to A \to A$$
$$w(\lambda i.s_2(s_1 i))(\lambda ij.t_2(s_1 i)(t_1 ij))))$$

$$\pi_3 := \langle \lambda i.(i+1)+1), \lambda ij.j+i \rangle \quad : A$$

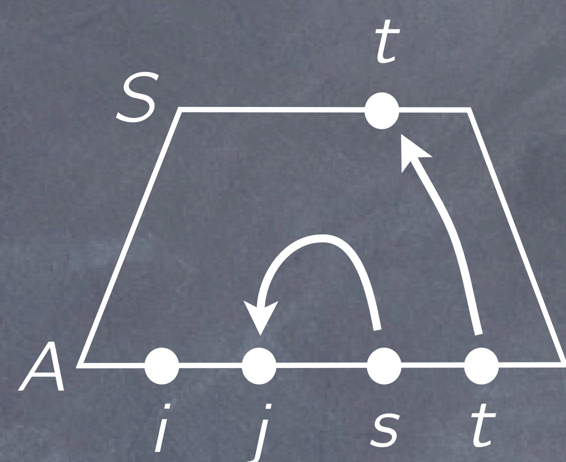
$$S := o$$

$$A := (o \to o) \otimes (o \to o \to o)$$

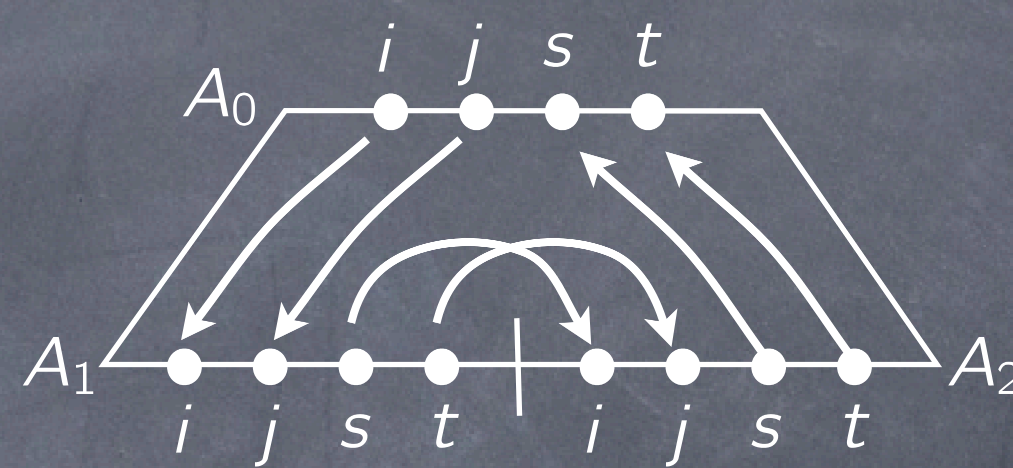
Encoding in almost linear λH



$\pi_1 : S \to f\ A$



$\pi_2 : A \to g\ A\ A$



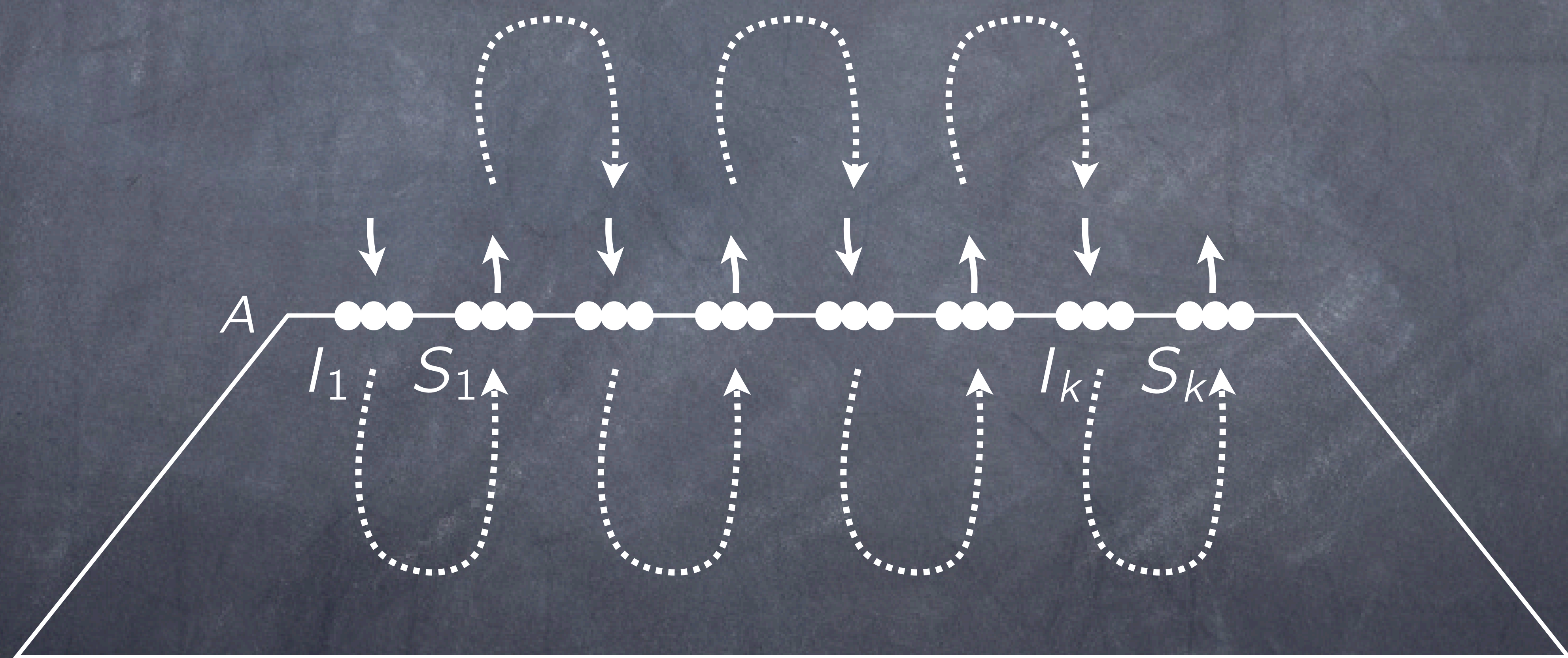
w, v: variables for visits

$$\underbrace{\lambda iw}_{\text{enter with input}}.\underbrace{wM}_{\text{exit with output}}(\underbrace{\lambda jv}_{\text{enter with input}}.\underbrace{vN}_{\text{exit with output}})\underbrace{P}_{\text{give input}}(\underbrace{\lambda s}_{\text{get output}}y.y\underbrace{Q}_{\text{give input}}(\underbrace{\lambda t}_{\text{get output}}.R))$$

$$\pi_1 := \lambda x.x((1+1)+1)(\lambda sy.y(s+1)(\lambda t.t))$$
$$\pi_2 := \lambda x_1 x_2 iw.x_1 i(\lambda s_1 y_1.x_2 s_1(\lambda s_2 y_2.ws_2(\lambda jv.y_1 j(\lambda t_1.y_2 t_1(\lambda t_2.vt_2)))))$$
$$\pi_3 := \lambda iw.w((i+1)+1)(\lambda jv.v(j+i))$$

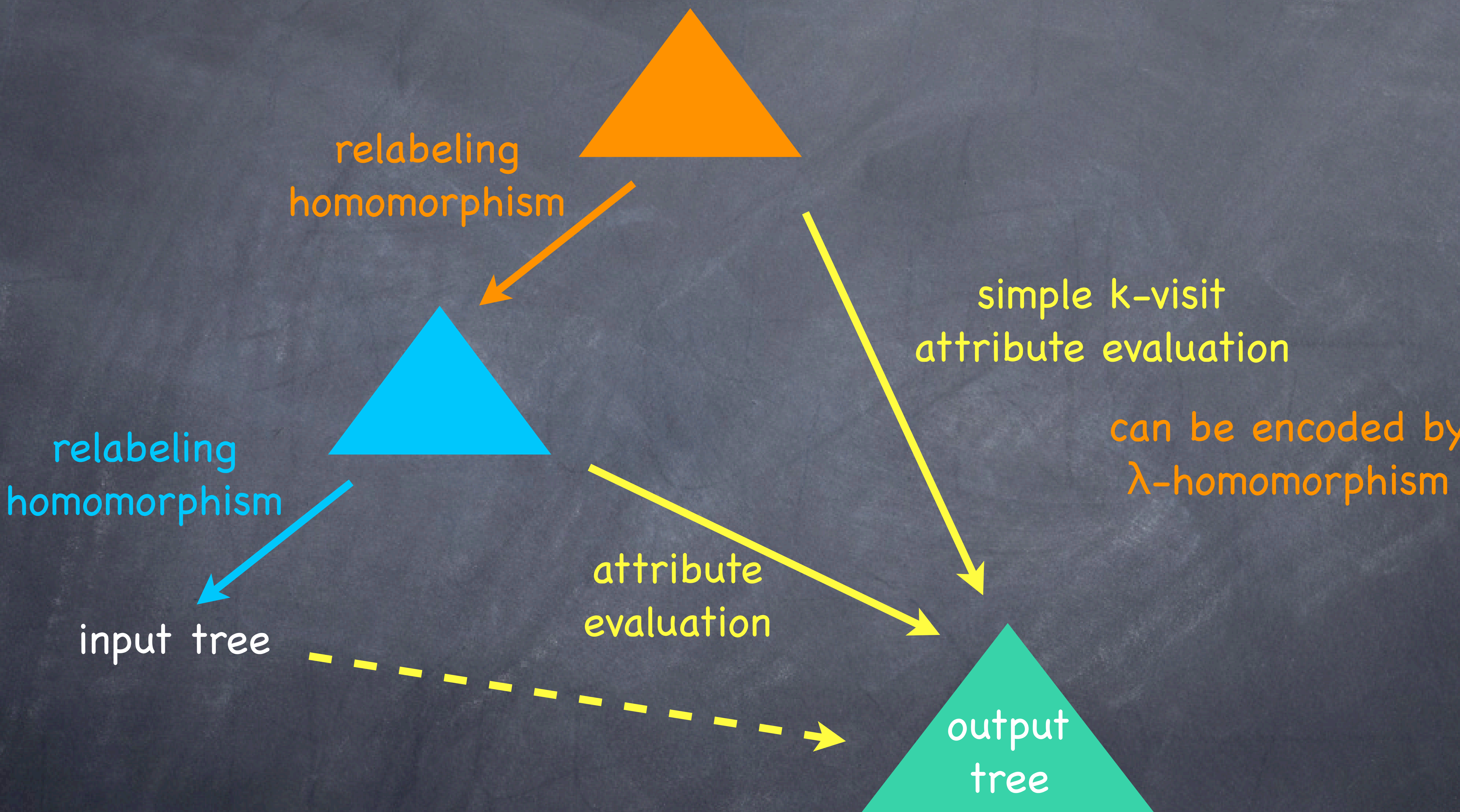
Simple k-visit attribute grammars

- An attribute grammar is simple k-visit if each attribute of a nonterminal A has a fixed visit-number ≤ k such that the attributes with visit-number j are computed at the j-th visit to A.



- Every non-circular attribute grammar can be made simple multi-visit by splitting nonterminals.

TT-AG ⊆ almost linear λ-H



Conclusion

- MSO-TGT = TT-AG = almost linear λ-H
- Linear λ-homomorphisms characterize MSO-TT (MSO transductions from tree graphs to tree graphs)