Second-order ACGs as Hyperedge Replacement Grammars

Makoto Kanazawa
NII, Tokyo

This talk

string generating power

$$STR(\mathbf{A}) = STR(\mathbf{HR})$$

tree generating power

$$TR(\mathbf{A}) = TR(\mathbf{HR})$$

- **A** : second-order abstract categorial grammars (de Groote 2001)
- **HR** : hyperedge replacement graph grammars (Bauderon and Courcelle 1987, Habel and Kreowski 1987)
- Generalize "context-free" grammars on strings/trees

This talk

string generating power

$$STR(\mathbf{A}) = STR(\mathbf{HR})$$

already known (de Groote & Pogodalla, Salvati)

tree generating power

$$TR(\mathbf{A}) = TR(\mathbf{HR})$$

- **A** : second-order abstract categorial grammars (de Groote 2001)
- **HR** : hyperedge replacement graph grammars (Bauderon and Courcelle 1987, Habel and Kreowski 1987)
- Generalize "context-free" grammars on strings/trees

This talk

string generating power

$$STR(\mathbf{A}) = STR(\mathbf{HR})$$

already known (de Groote & Pogodalla, Salvati)

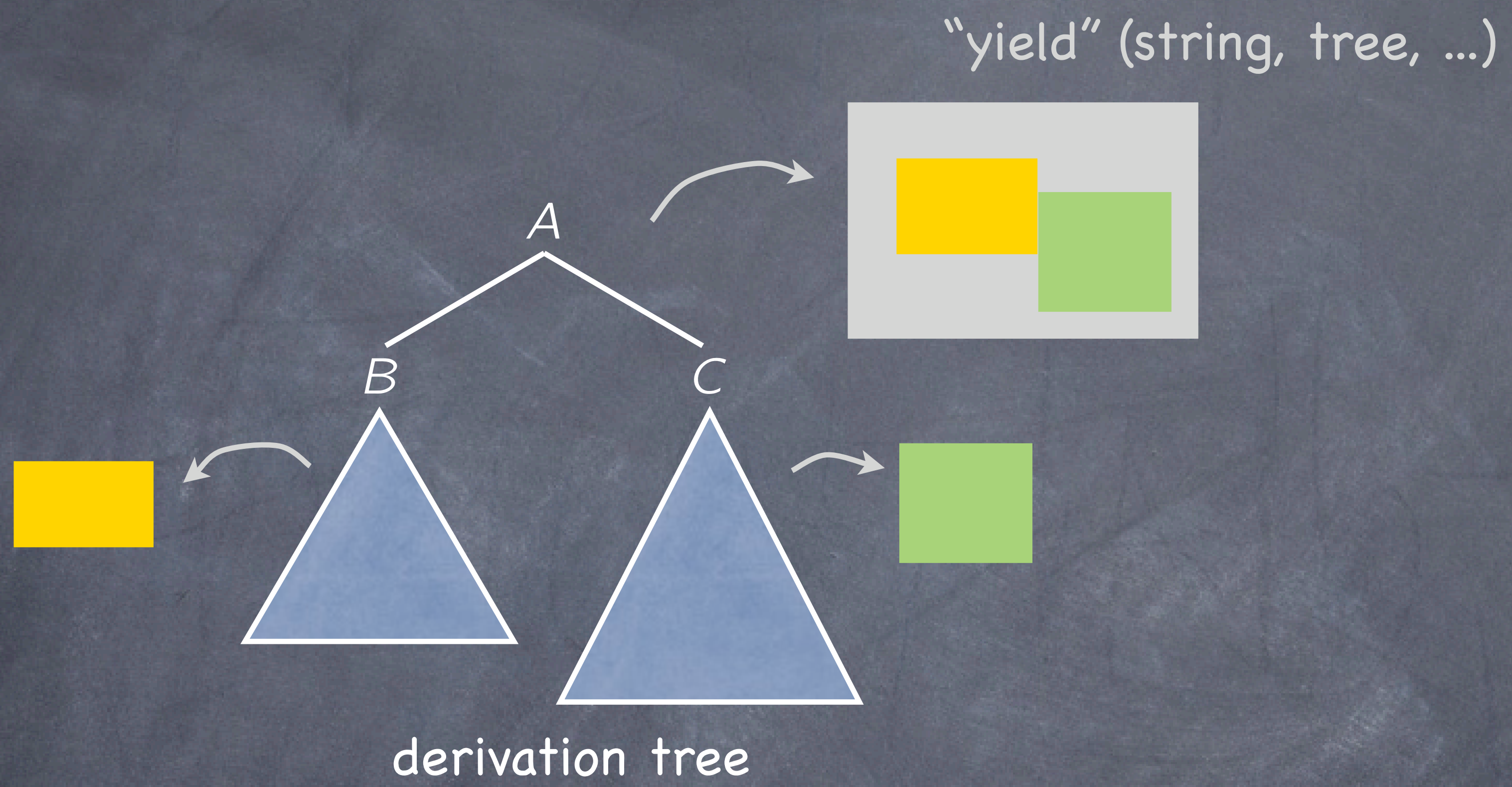
tree generating power

$$TR(\mathbf{A}) = TR(\mathbf{HR})$$

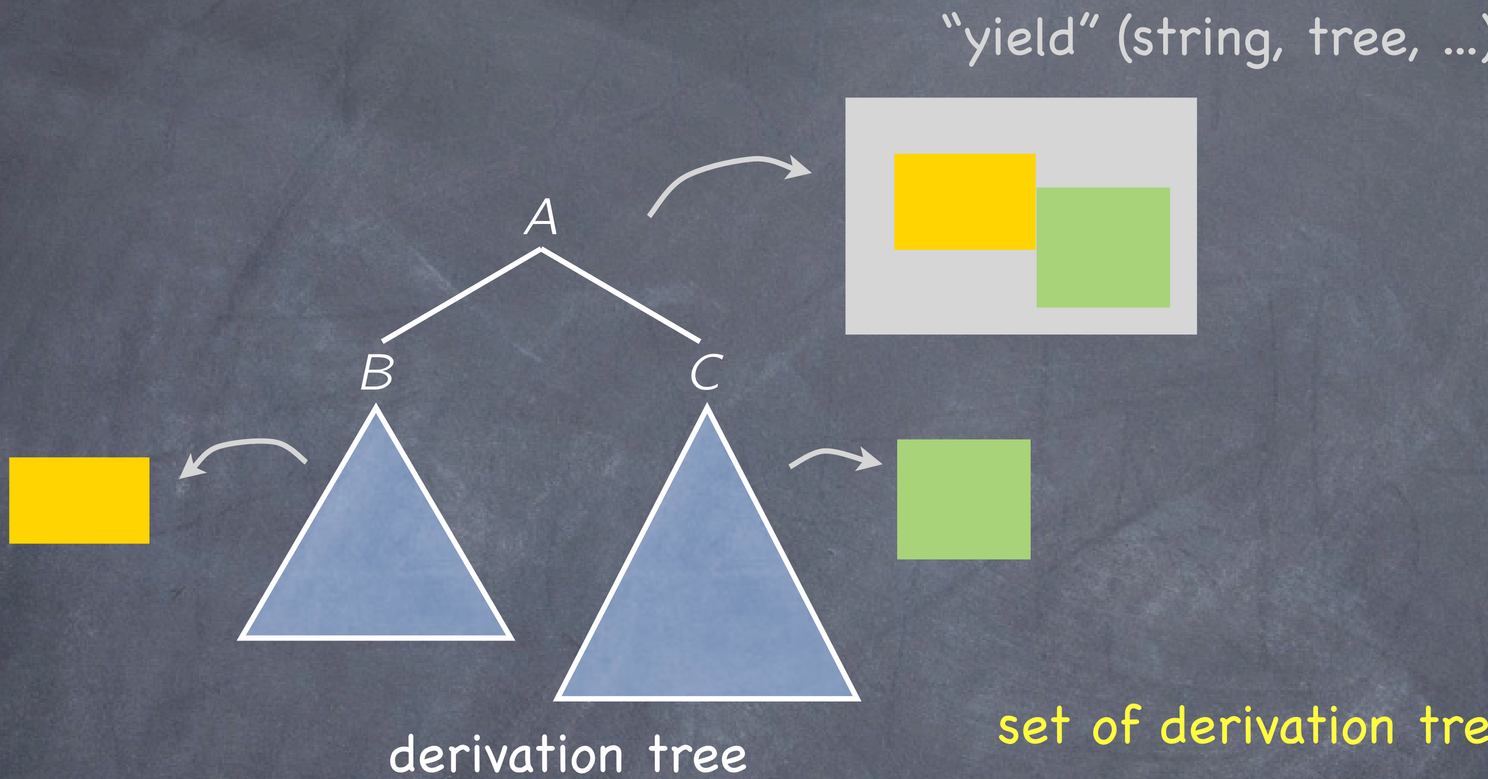
new!

- **A** : second-order abstract categorial grammars (de Groote 2001)
- **HR** : hyperedge replacement graph grammars (Bauderon and Courcelle 1987, Habel and Kreowski 1987)
- Generalize "context-free" grammars on strings/trees

"Context-free" grammar formalisms

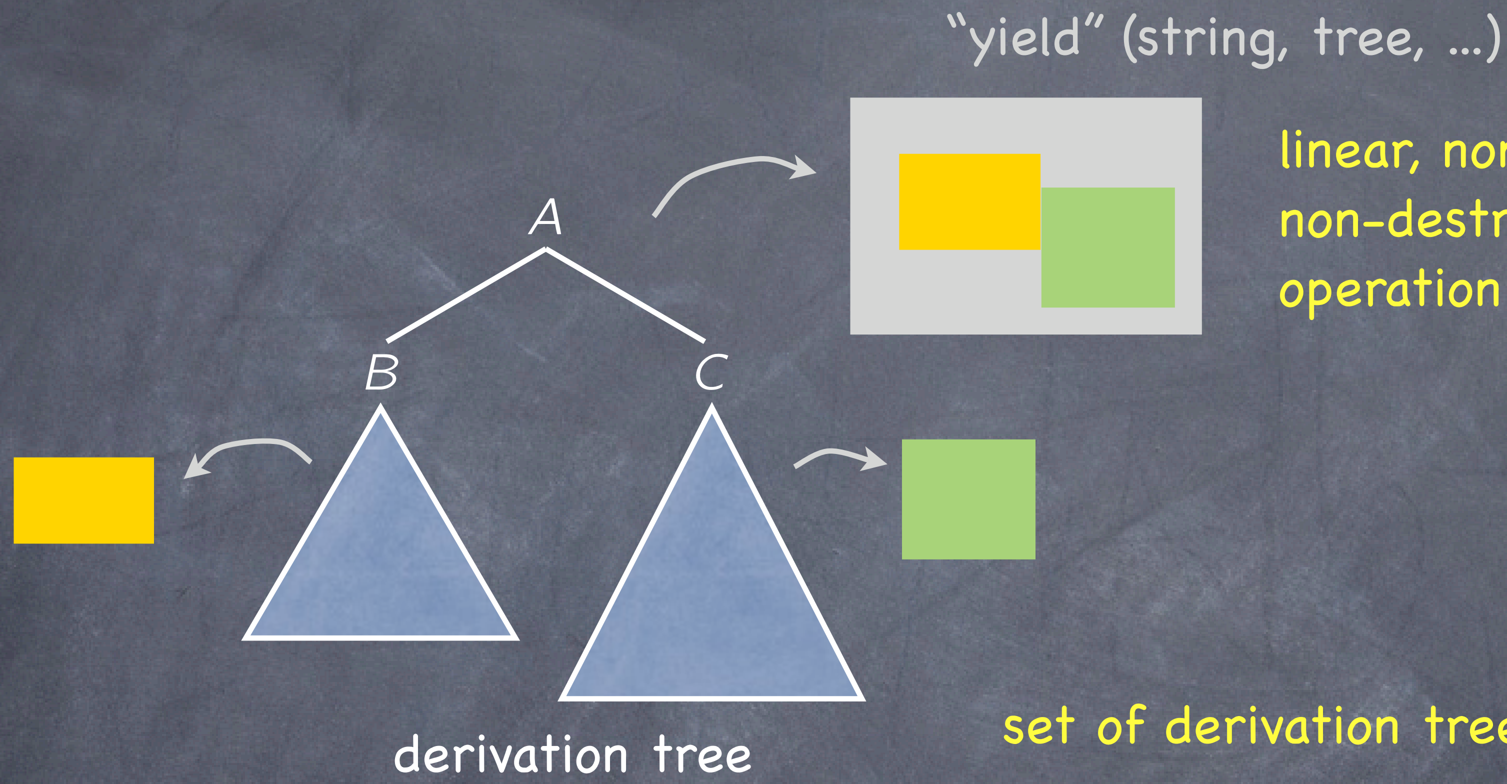


“Context-free” grammar formalisms



set of derivation tree = local set

"Context-free" grammar formalisms

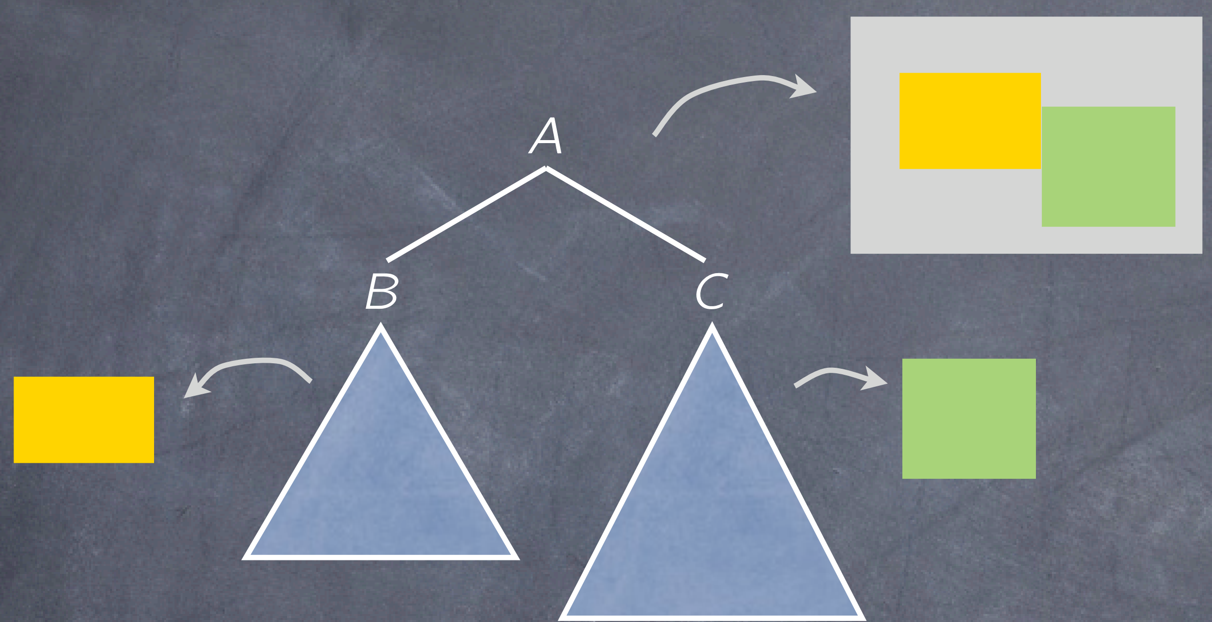


linear, non-deleting,
non-destructive
operation

set of derivation tree = local set

“Context-free” grammar formalisms

“yield” (string, tree, ...)



linear, non-deleting, non-destructive operation

derivation tree

set of derivation tree = local set

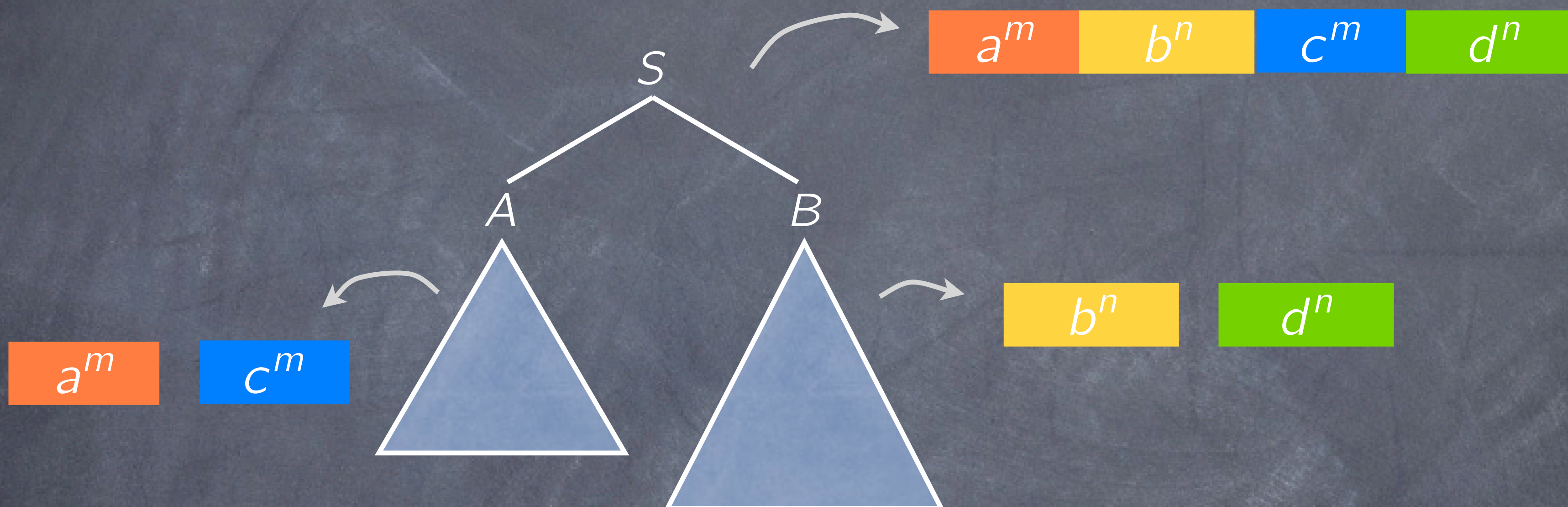
“context-free” production

top-down view

bottom-up view

Multiple context-free grammars

$$\{ a^m b^n c^m d^n \mid m, n \geq 0 \}$$



$$S(x_1 y_1 x_2 y_2) :- A(x_1, x_2), B(y_1, y_2).$$

$$A(a x_1, c x_2) :- A(x_1, x_2).$$

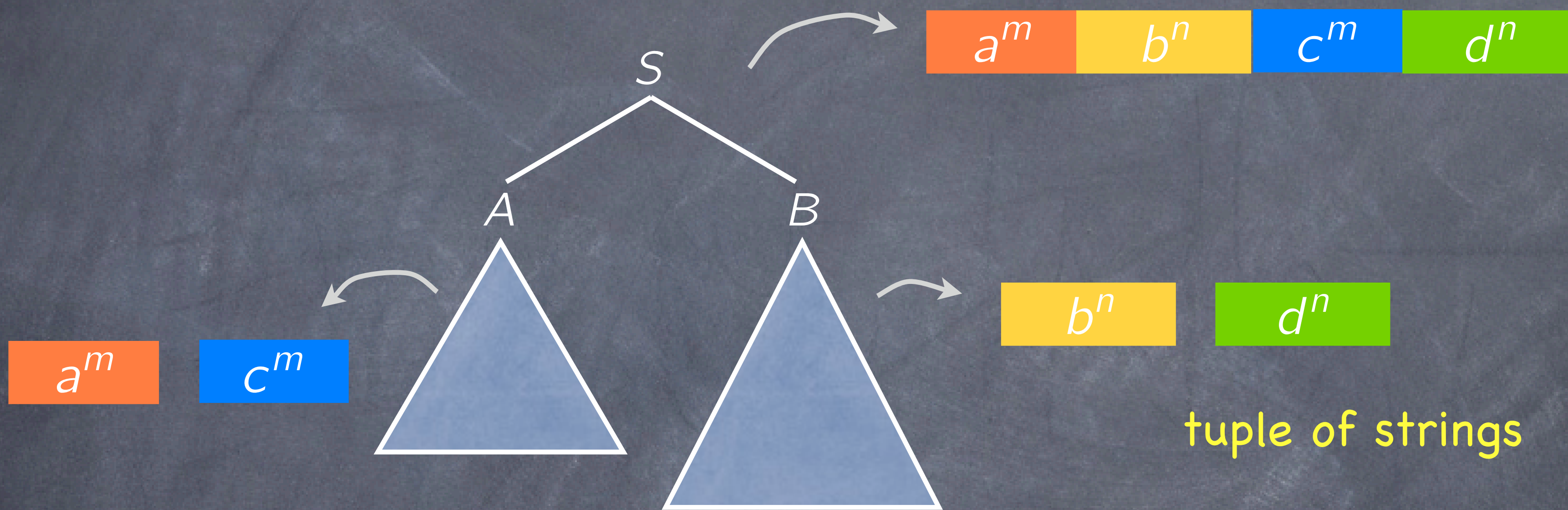
$$A(\epsilon, \epsilon).$$

$$B(b y_1, d y_2) :- B(y_1, y_2).$$

$$B(\epsilon, \epsilon).$$

Multiple context-free grammars

$$\{ a^m b^n c^m d^n \mid m, n \geq 0 \}$$



$$S(x_1 y_1 x_2 y_2) :- A(x_1, x_2), B(y_1, y_2).$$

$$A(a x_1, c x_2) :- A(x_1, x_2).$$

$$A(\epsilon, \epsilon).$$

$$B(b y_1, d y_2) :- B(y_1, y_2).$$

$$B(\epsilon, \epsilon).$$

"Context-free" grammars on ...

	strings	trees	tree contexts $C[x_1]$	n-ary tree contexts $C[x_1,\dots,x_n]$
single	CFG	RTG	TAG (monadic LCFTG)	LCFTG
multiple	MCFG	MRTG	MCTAG	

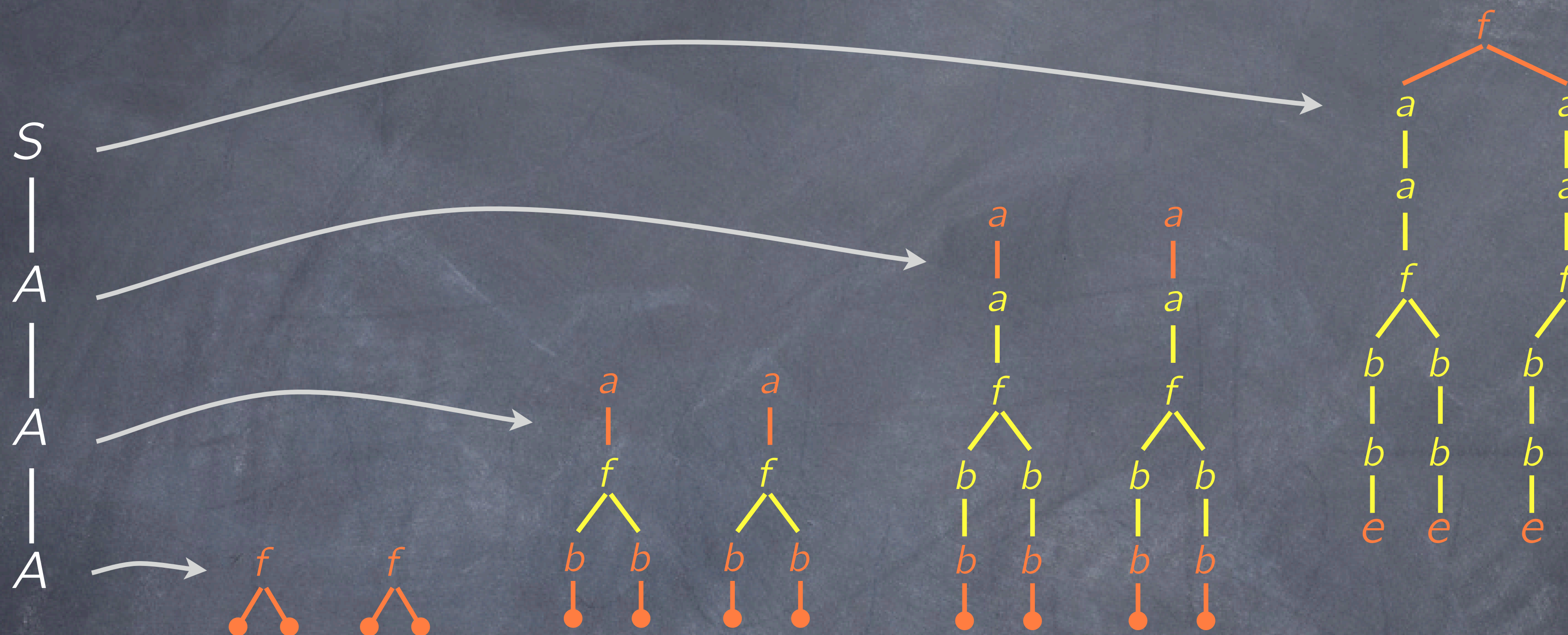
- RTG: regular tree grammar
- LCFTG: linear non-deleting context-free tree grammar
- MCFG: multiple context-free grammar
- MRTG: multiple regular tree grammar
- MCTAG: multi-component tree-adjoining grammar

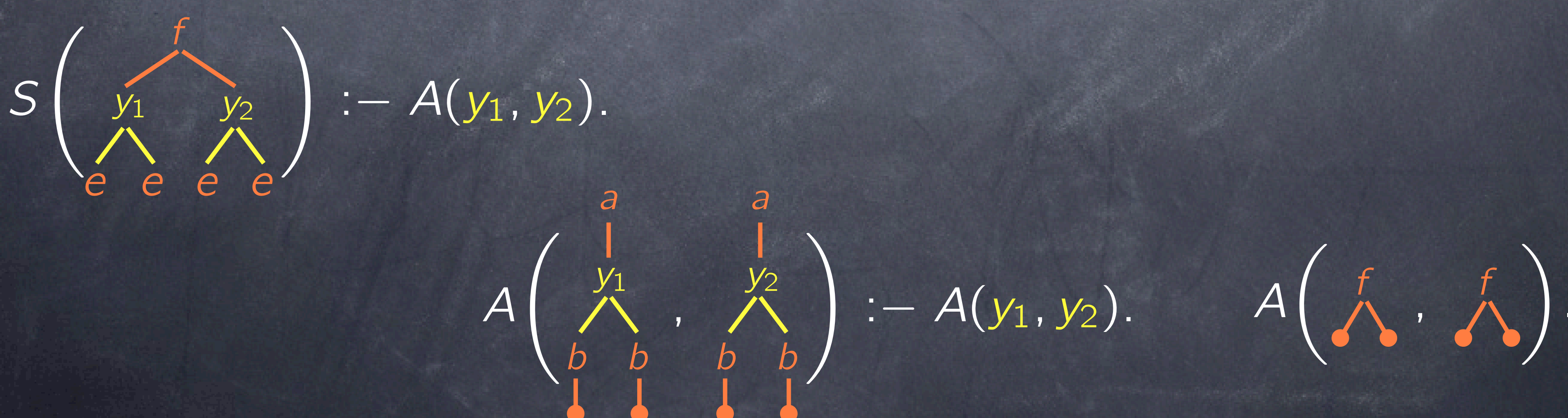
“Context-free” grammars on ...

	strings	trees	tree contexts $C[x_1]$	n-ary tree contexts $C[x_1,\ldots,x_n]$
single	CFG	RTG	TAG (monadic LCFTG)	LCFTG
multiple	MCFG	MRTG	MCTAG	MLCFTG

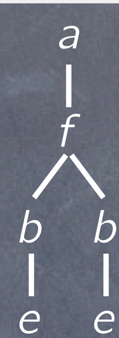
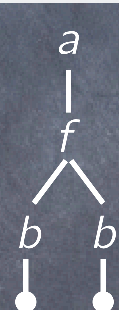
- RTG: regular tree grammar
- LCFTG: linear non-deleting context-free tree grammar
- MCFG: multiple context-free grammar
- MRTG: multiple regular tree grammar
- MCTAG: multi-component tree-adjoining grammar
- MLCFTG: multiple linear non-deleting context-free tree grammar

MLCFTG



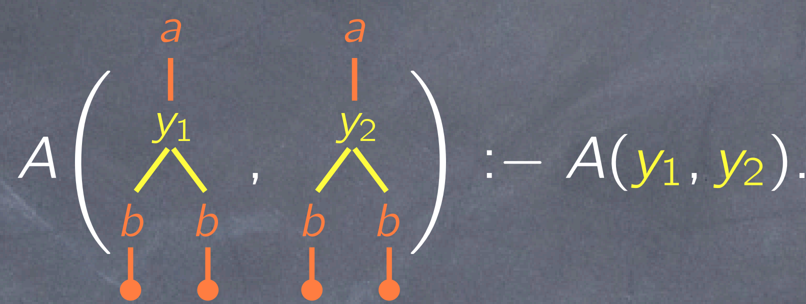


ACG: Everything is a linear λ-term

string	$abbac$	$\lambda z.a(b(b(a(cz))))$
tree		$a(f(be)(be))$
n-ary tree context		$\lambda x_1 x_2.a(f(bx_1)(bx_2))$
tuple	(N_1, N_2, N_3)	$\lambda w.wN_1N_2N_3$
logical formula	$\forall x \exists y(Rxy)$	$\forall(\lambda x.\exists(\lambda y.Rxy))$

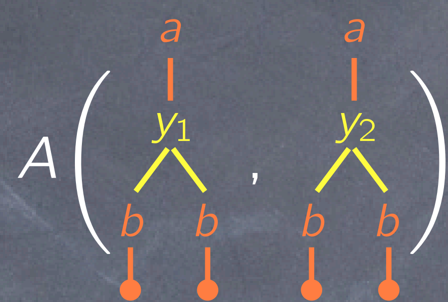
MLCFTGs as second-order ACGs

MLCFTG:



MLCFTGs as second-order ACGs

MLCFTG:



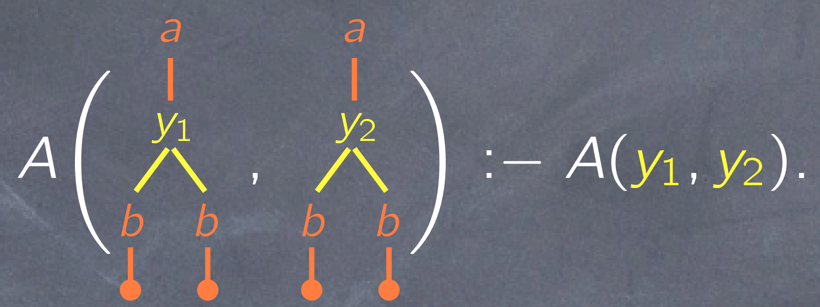
$:-\ A(y_1, y_2).$

2nd-order ACG: linear λ-term

$$A\big(\lambda w.X(\lambda y_1 y_2.w(\lambda x_1 x_2.a(y_1(bx_1)(bx_2)))(\lambda x_1 x_2.a(y_2(bx_1)(bx_2))))\big)\ :-\ A(X).$$

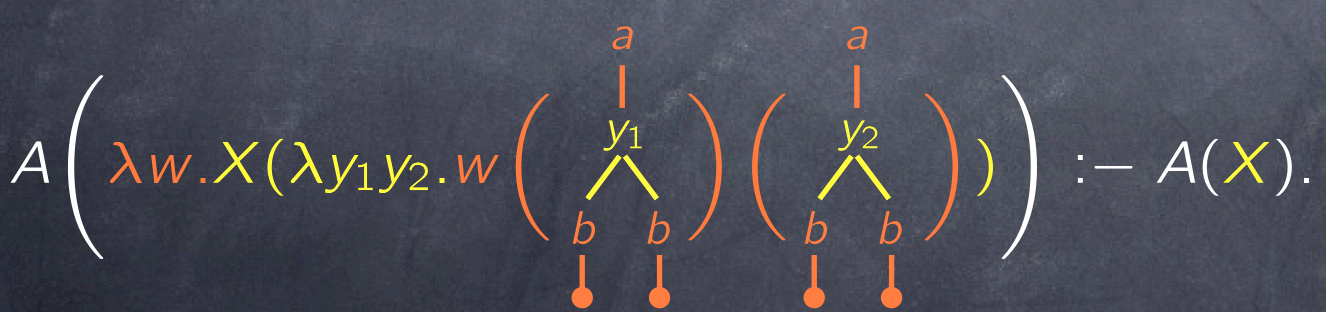
MLCFTGs as second-order ACGs

MLCFTG:



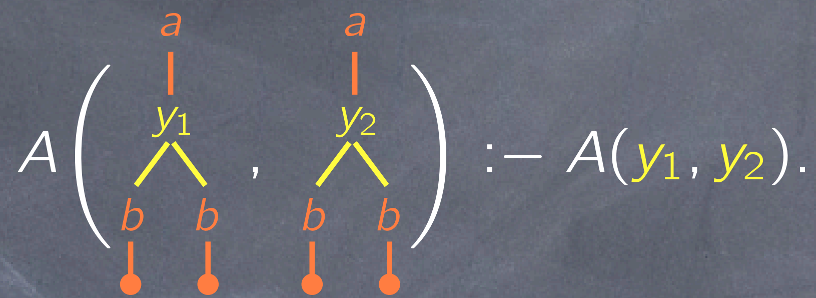
2nd-order ACG: linear λ-term

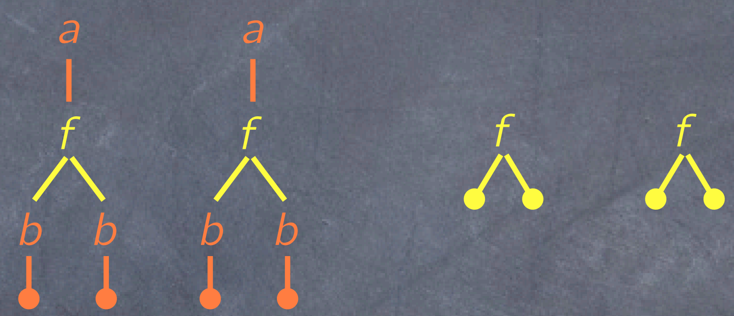
$$A(\lambda w.X(\lambda y_1 y_2.w(\lambda x_1 x_2.a(y_1(bx_1)(bx_2)))(\lambda x_1 x_2.a(y_2(bx_1)(bx_2))))) :- A(X).$$



MLCFTGs as second-order ACGs

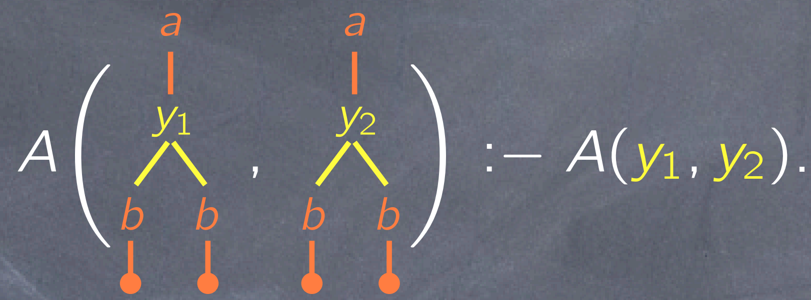
MLCFTG:

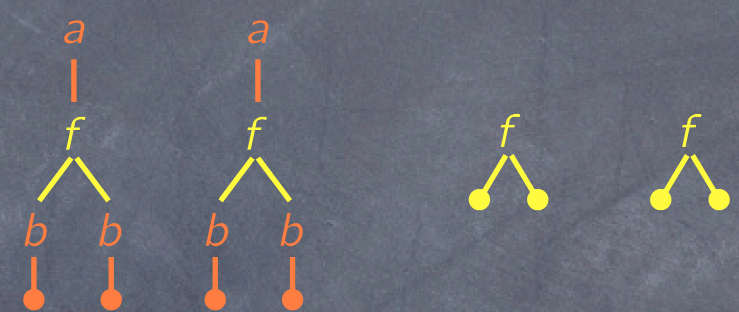




MLCFTGs as second-order ACGs

MLCFTG:



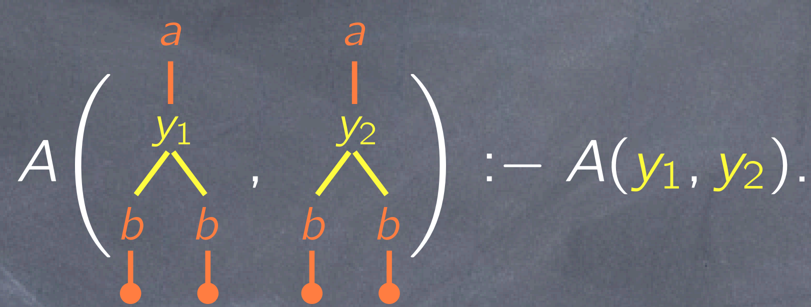


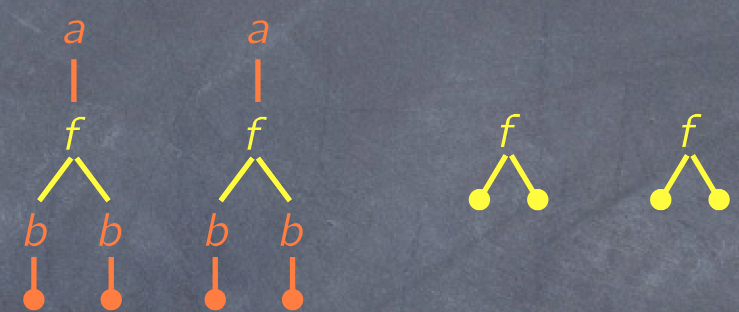
$$\lambda w.\underline{(\lambda v.v(\lambda x_1x_2.f\,x_1x_2)(\lambda x_1x_2.f\,x_1x_2))}(\lambda y_1y_2.w(\lambda x_1x_2.a(y_1(bx_1)(bx_2)))(\lambda x_1x_2.a(y_2(bx_1)(bx_2)))$$

$$\twoheadrightarrow_\beta \lambda w.w(\lambda x_1x_2.a(f(bx_1)(bx_2)))(\lambda x_1x_2.a(f(bx_1)(bx_2)))$$

MLCFTGs as second-order ACGs

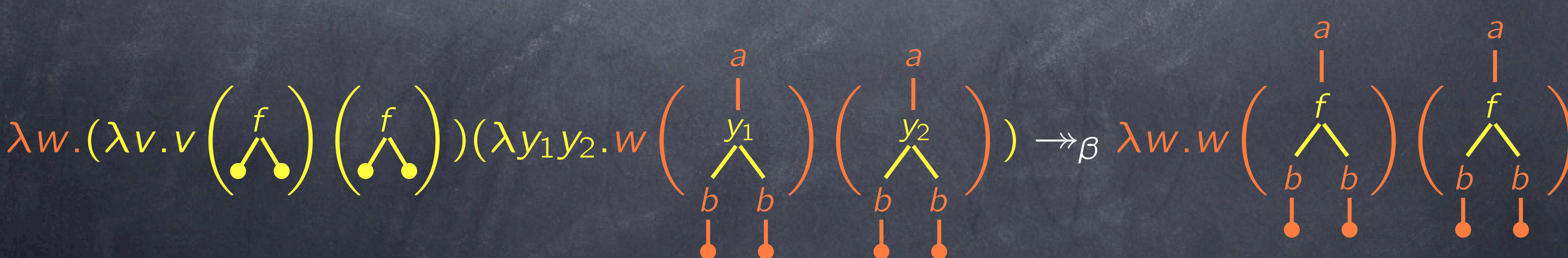
MLCFTG:





$$\lambda w.\underline{(\lambda v.v(\lambda x_1 x_2.f\, x_1 x_2)(\lambda x_1 x_2.f\, x_1 x_2))}(\lambda y_1 y_2.w(\lambda x_1 x_2.a(y_1(bx_1)(bx_2)))(\lambda x_1 x_2.a(y_2(bx_1)(bx_2))))$$

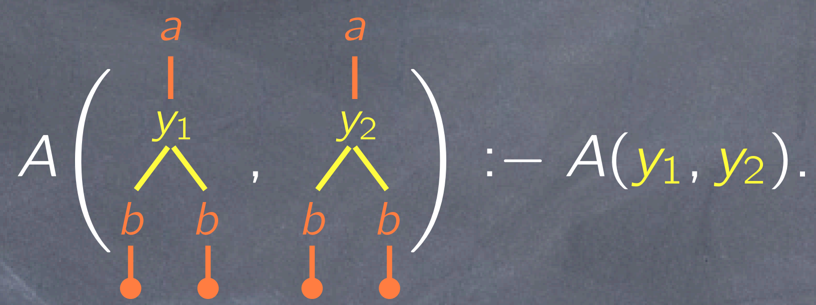
$$\twoheadrightarrow_\beta \lambda w.w(\lambda x_1 x_2.a(f(bx_1)(bx_2)))(\lambda x_1 x_2.a(f(bx_1)(bx_2)))$$

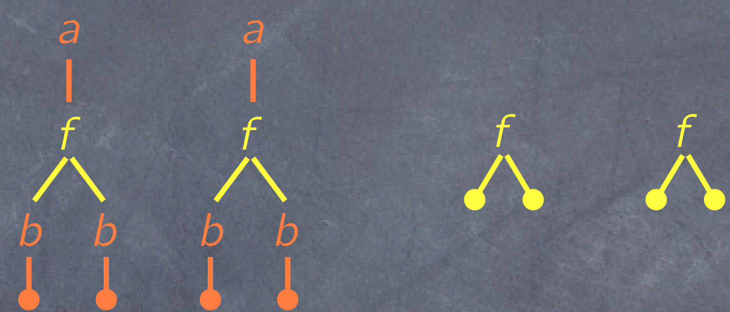


β-reduction

MLCFTGs as second-order ACGs

MLCFTG:



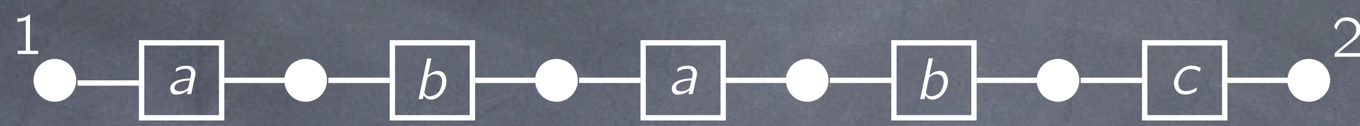


$\lambda w.\underline{(\lambda v.v(\lambda x_1 x_2.f\,x_1 x_2)(\lambda x_1 x_2.f\,x_1 x_2))}(\lambda y_1 y_2.w(\lambda x_1 x_2.a(y_1(bx_1)(bx_2)))(\lambda x_1 x_2.a(y_2(bx_1)(bx_2))))$

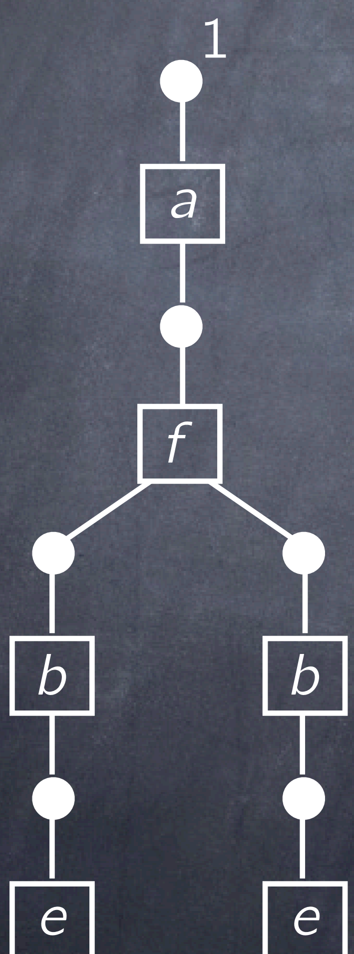
$\twoheadrightarrow_\beta \lambda w.w(\lambda x_1 x_2.a(f(bx_1)(bx_2)))(\lambda x_1 x_2.a(f(bx_1)(bx_2)))$

$TR(\mathbf{A}) \supseteq \mathrm{MLCFTL}$

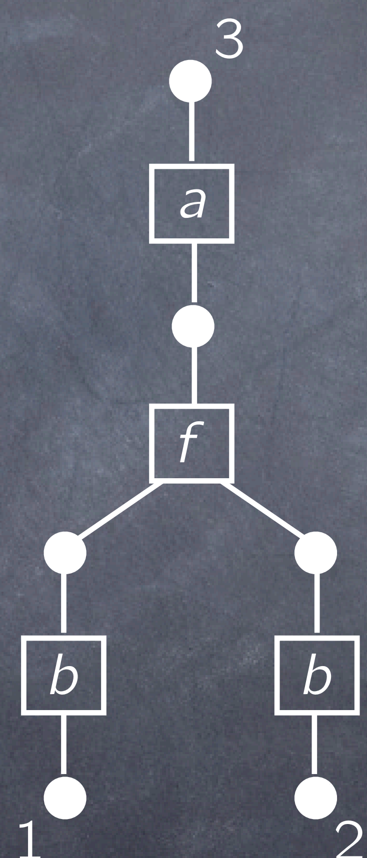
HR: Everything is a hypergraph



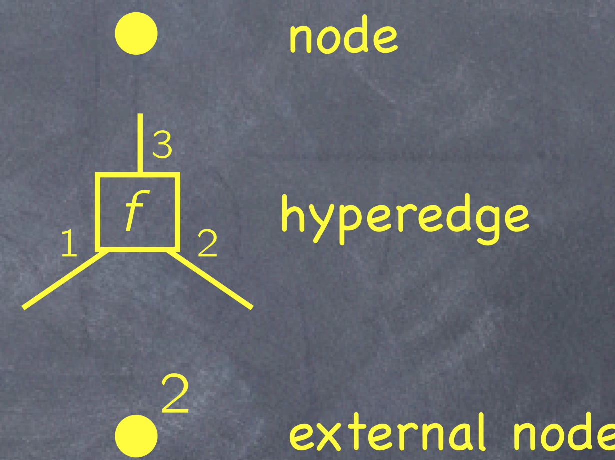
string graph



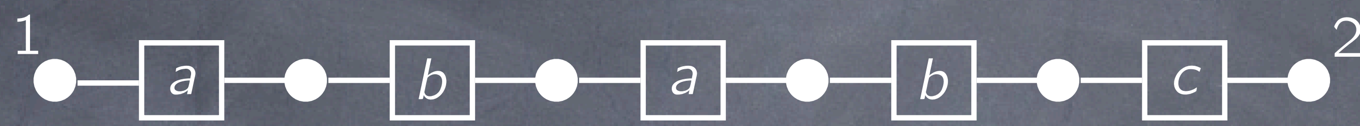
tree graph of type 1



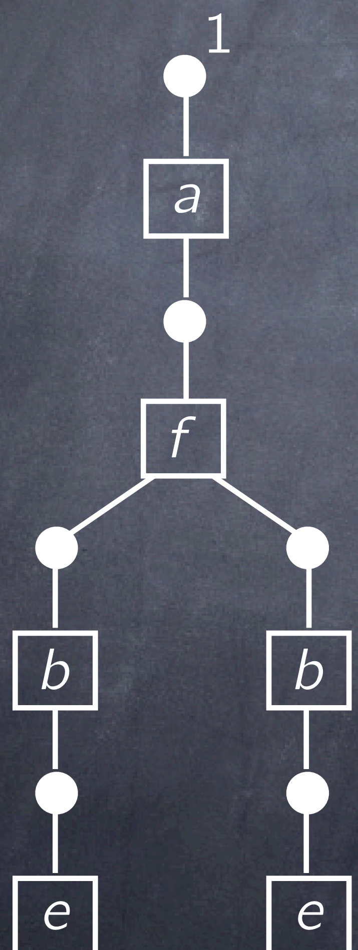
tree graph of type 3



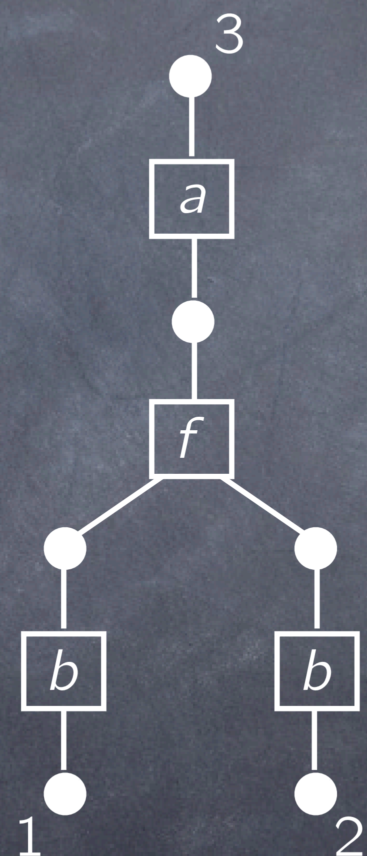
HR: Everything is a hypergraph



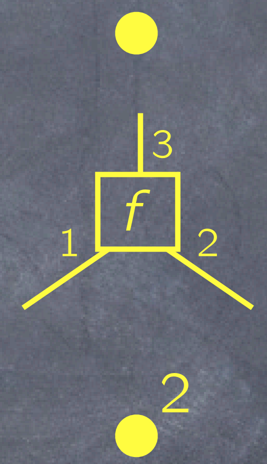
string graph



tree graph of type 1



tree graph of type 3



node

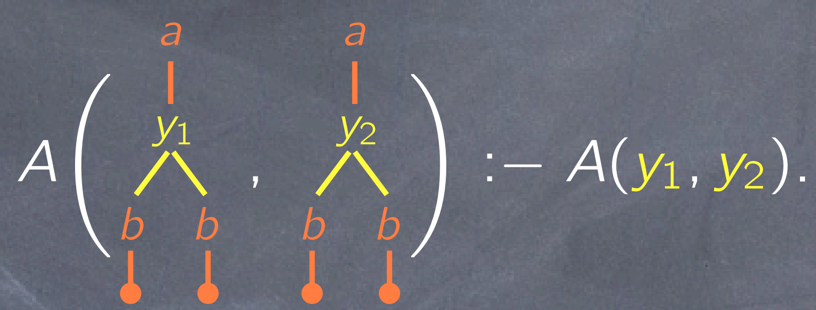
hyperedge

external node

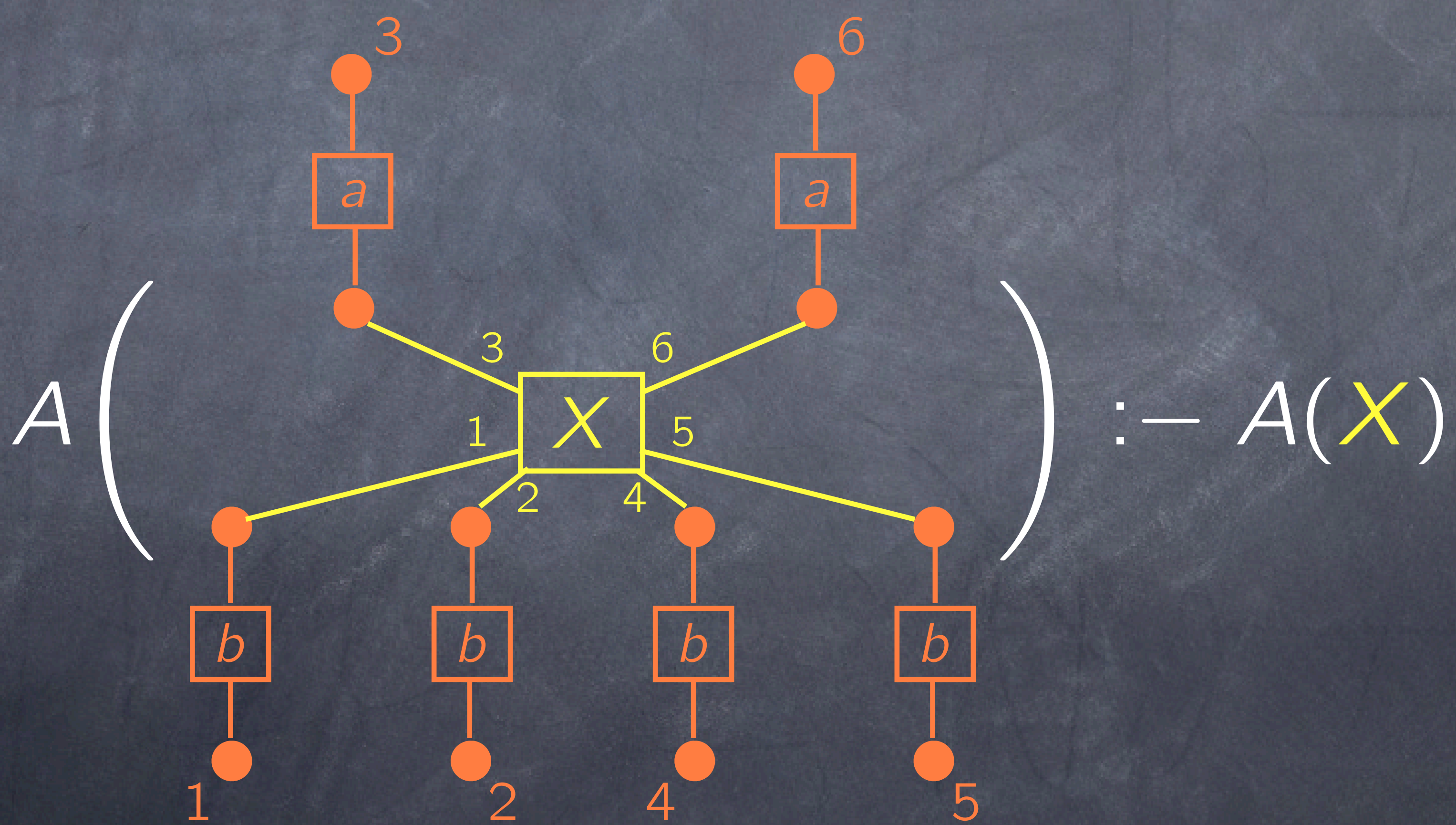
tuple = disjoint union

MLCFTGs as HRs

MLCFTG:

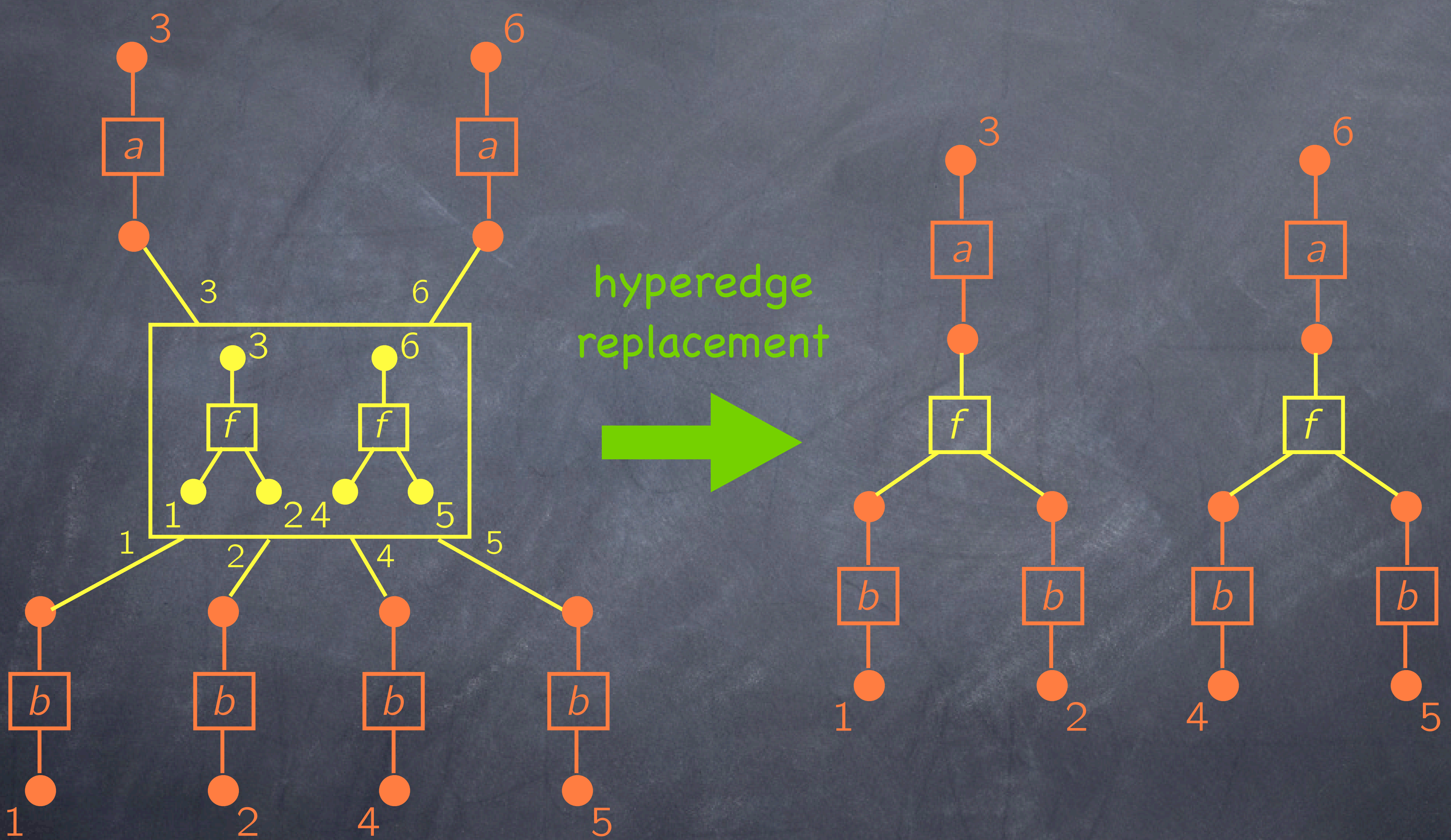


HR:

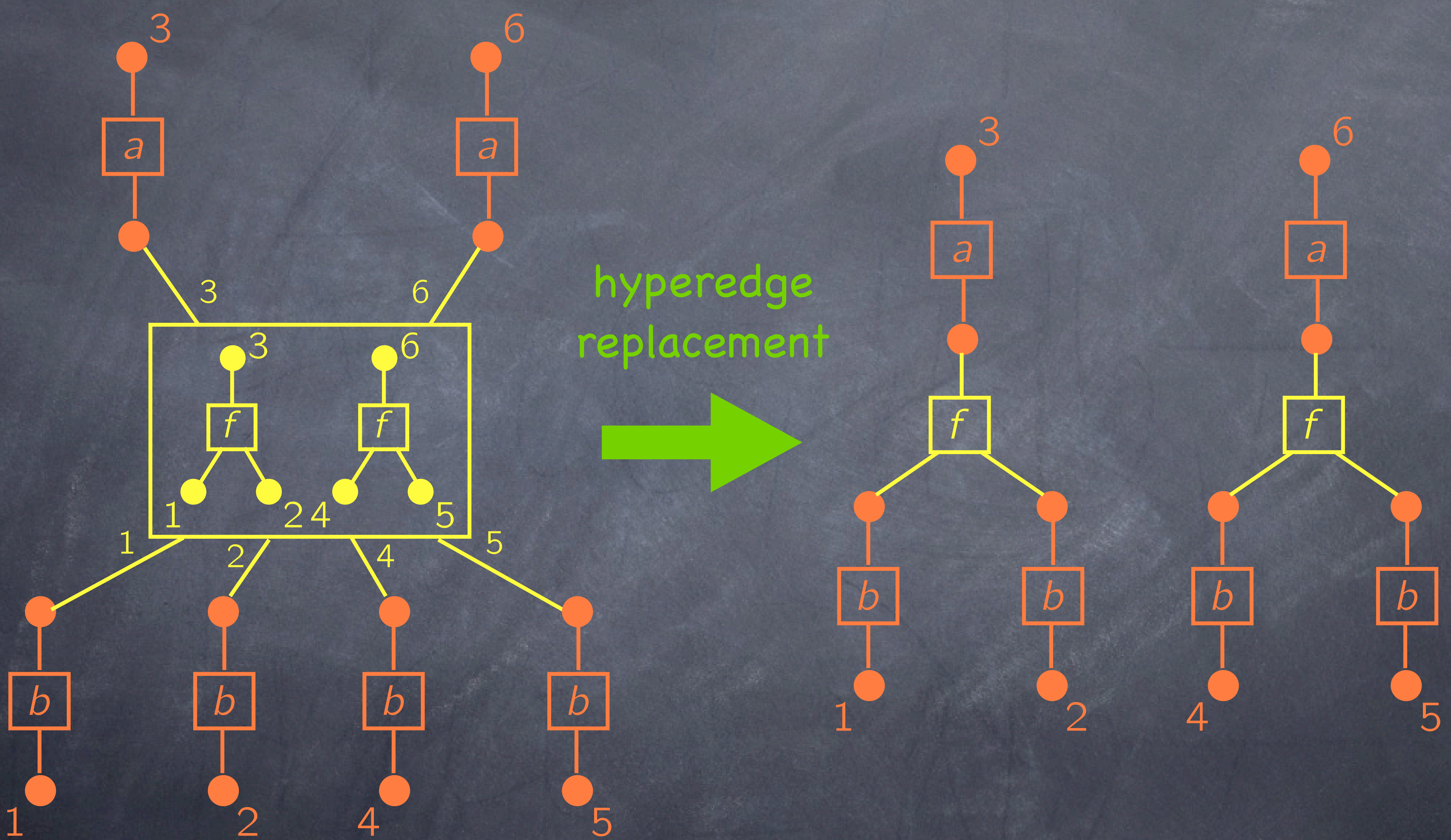


hypergraph

Hyperedge replacement

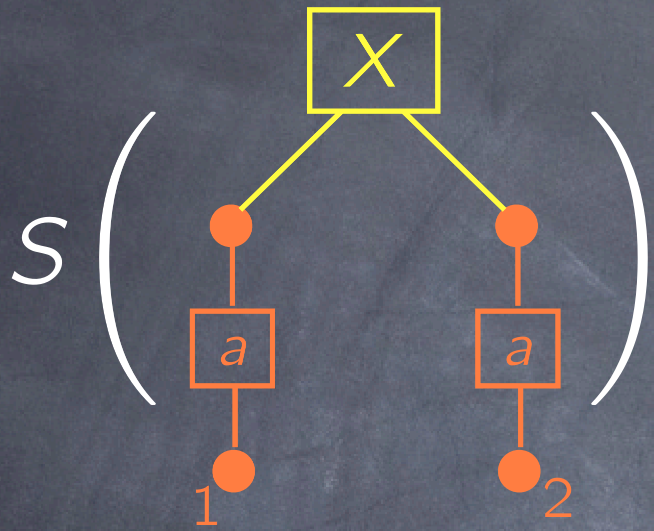


Hyperedge replacement

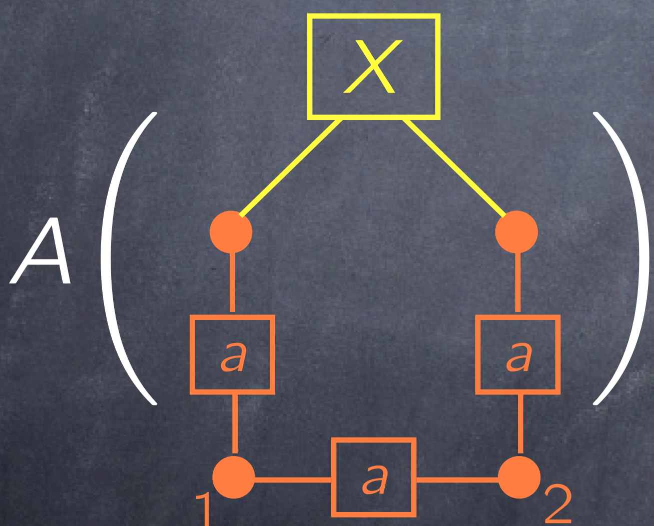


$TR(\mathbf{HR}) \supseteq \mathrm{MLCFTL}$

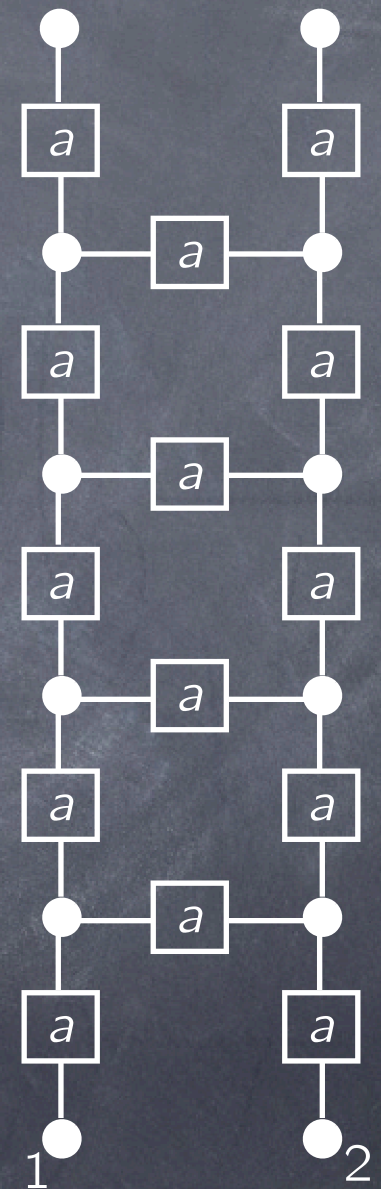
Example: Ladder



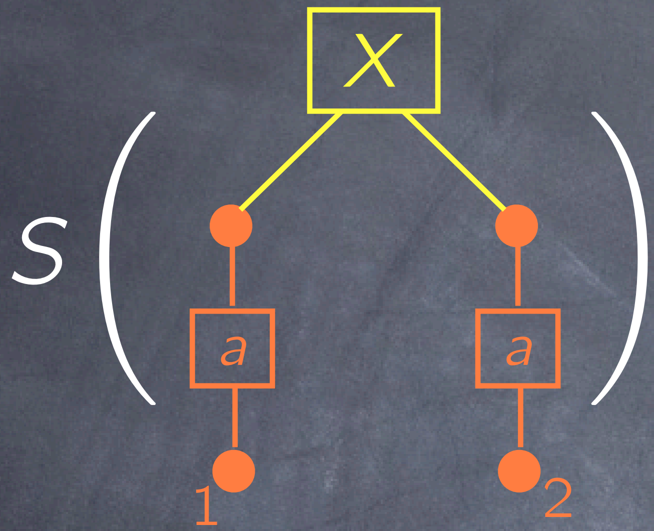
$:- A(X).$



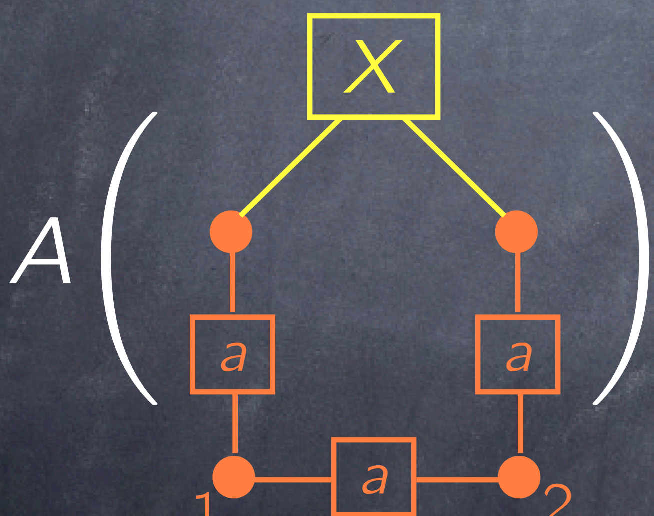
$:- A(X).$



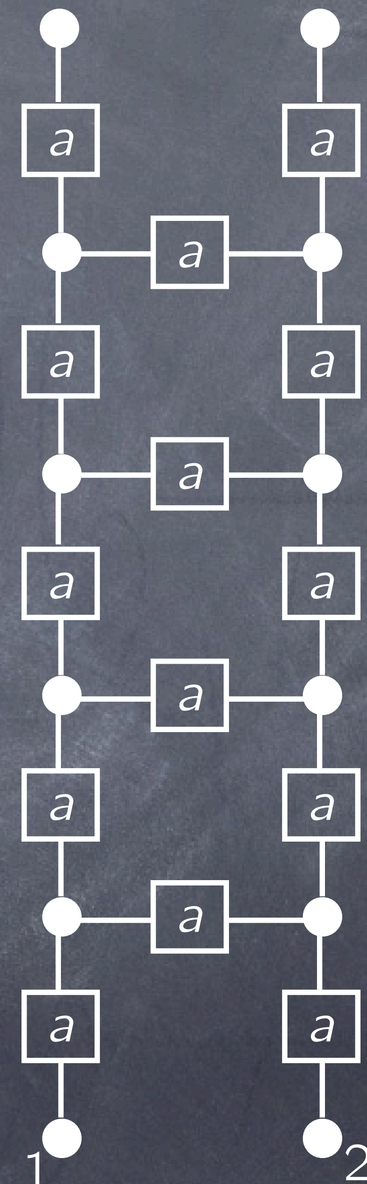
Example: Ladder



:– $A(X)$.

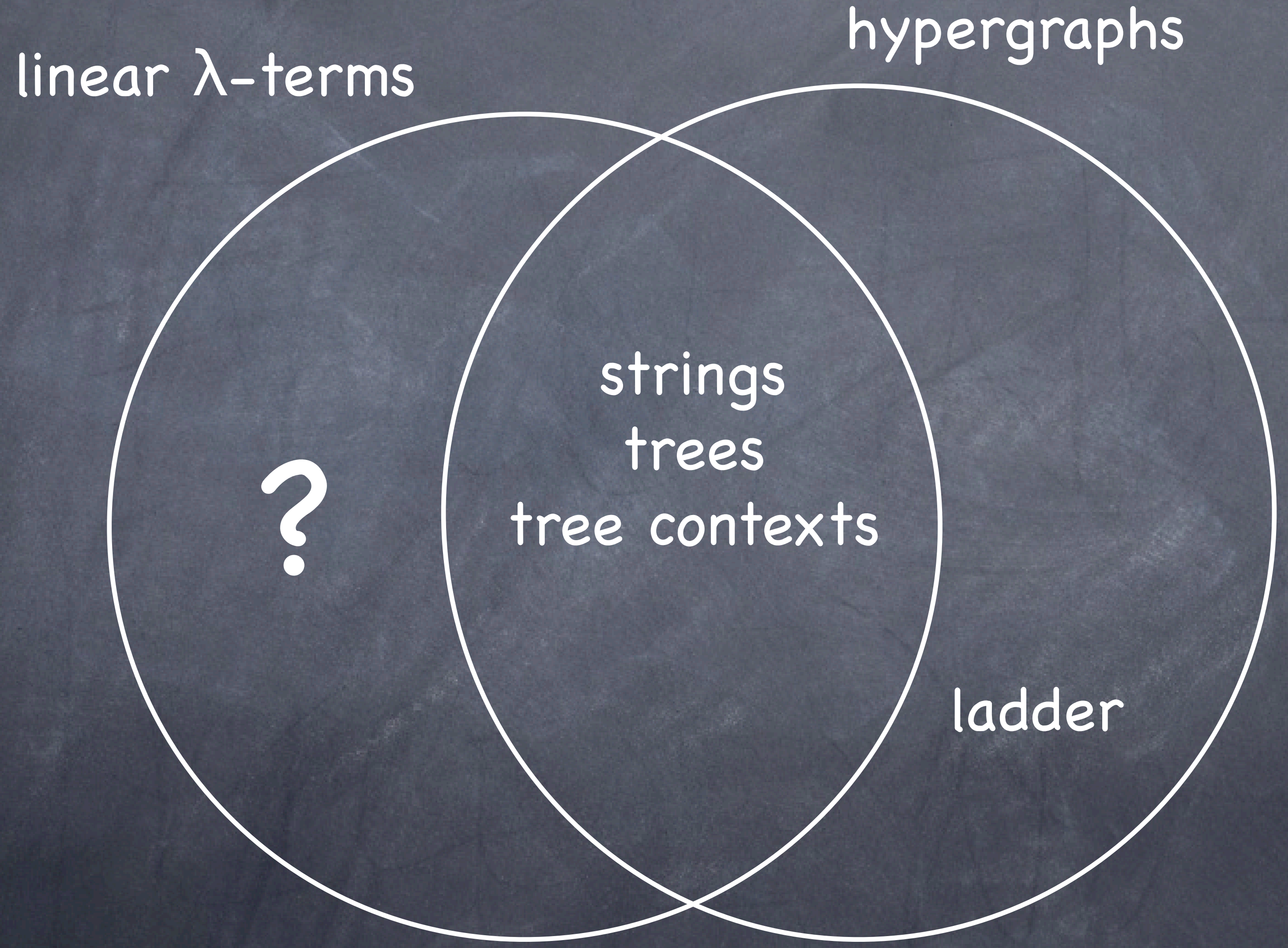


:– $A(X)$.



hypergraphs $\not\subseteq$ λ-terms

Linear λ-terms vs. hypergraphs

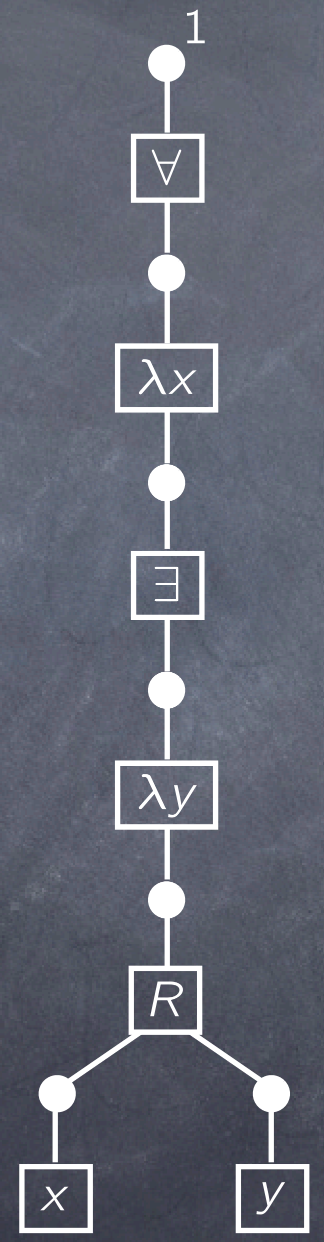


λ-terms as hypergraphs

$\forall(\lambda x.\exists(\lambda y.Rxy))$

λ-terms as hypergraphs

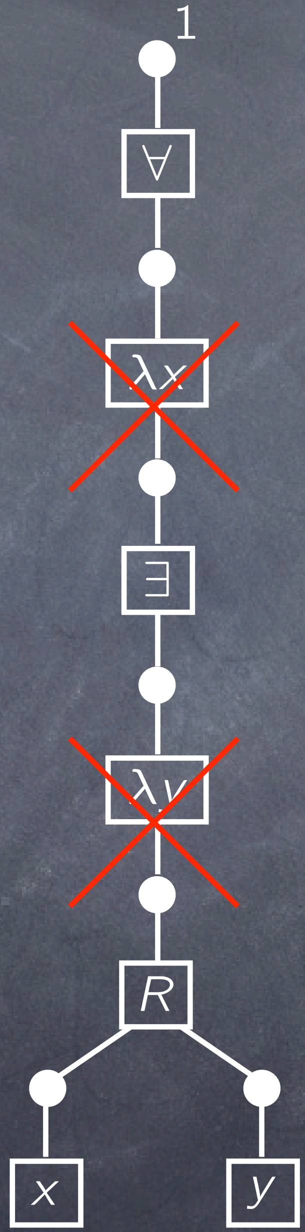
$\forall(\lambda x.\exists(\lambda y.Rxy))$



tree graph

λ-terms as hypergraphs

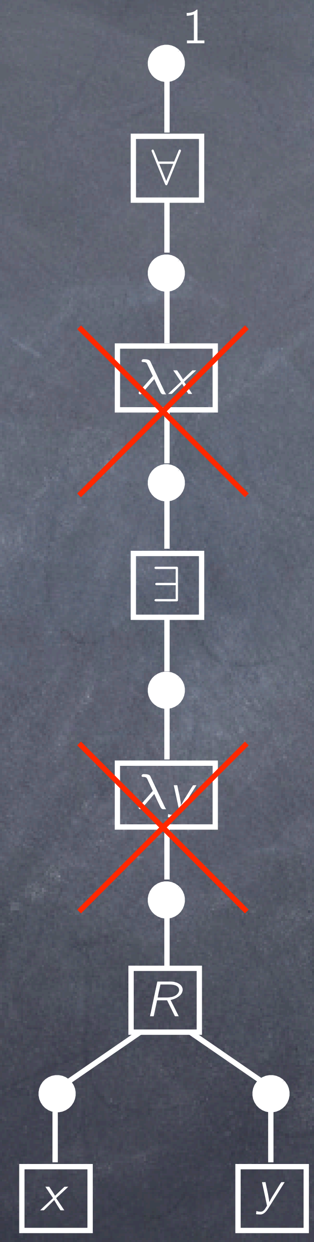
$\forall(\lambda x.\exists(\lambda y.Rxy))$



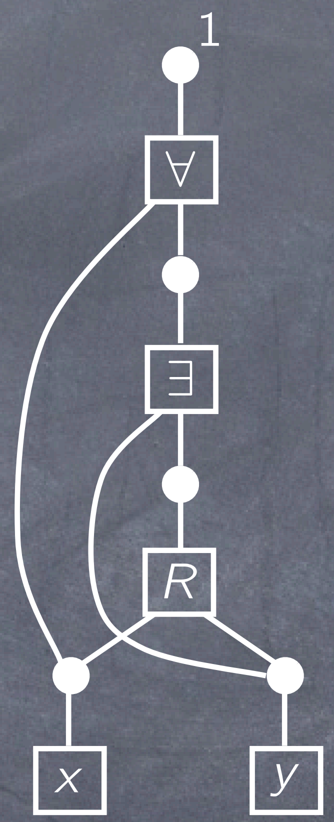
tree graph

λ-terms as hypergraphs

$\forall(\lambda x.\exists(\lambda y.Rxy))$



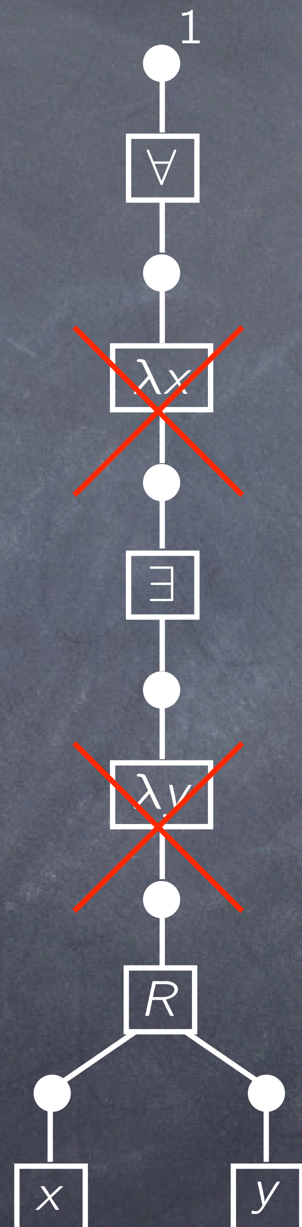
tree graph



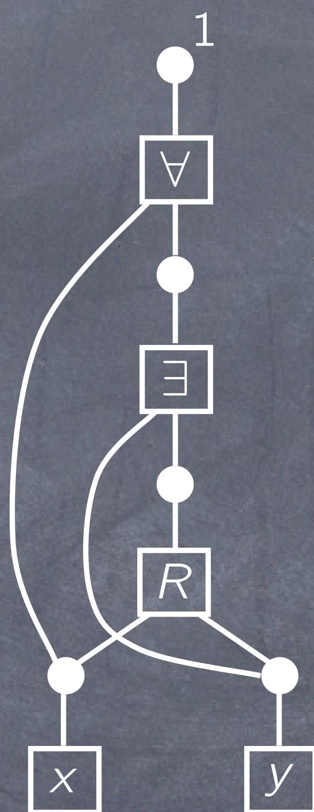
term graph

λ-terms as hypergraphs

$\forall(\lambda x.\exists(\lambda y.Rxy))$



tree graph



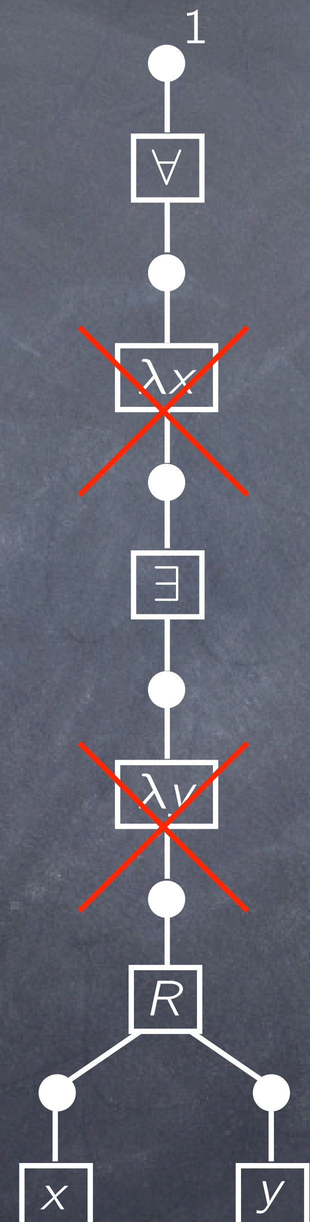
term graph

$\text{rank}(c) = |\text{type}(c)|$

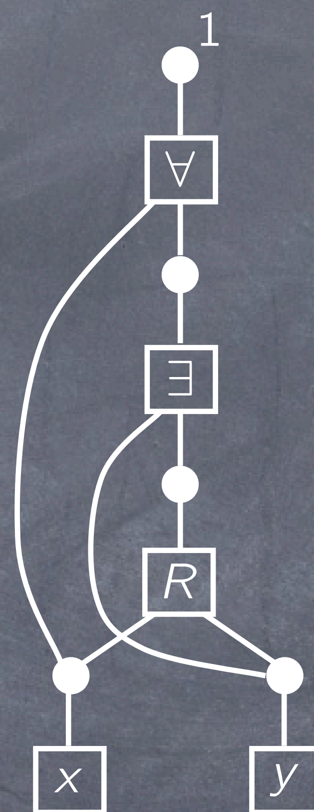
$3 = |(e \to t) \to t|$

λ-terms as hypergraphs

$\forall(\lambda x.\exists(\lambda y.Rxy))$



tree graph



term graph

$\text{rank}(c) = |\text{type}(c)|$

$3 = |(e \to t) \to t|$

linear λ-terms $\subsetneq$ hypergraphs

2nd-order ACGs as HRs

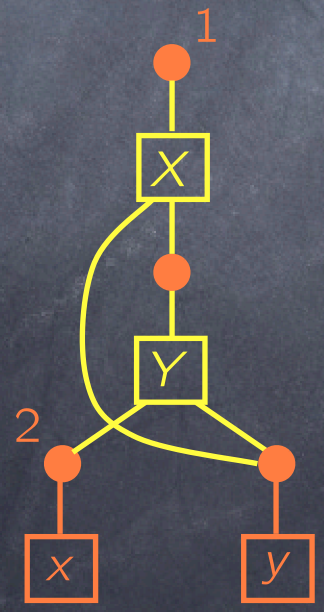
$$S(X(\lambda x.Yx)) :- NP(X), VP(Y).$$

$$VP(\lambda x.Y(\lambda y.Xxy)) :- V(X), NP(Y).$$

$$V(\lambda xy.Rxy).$$

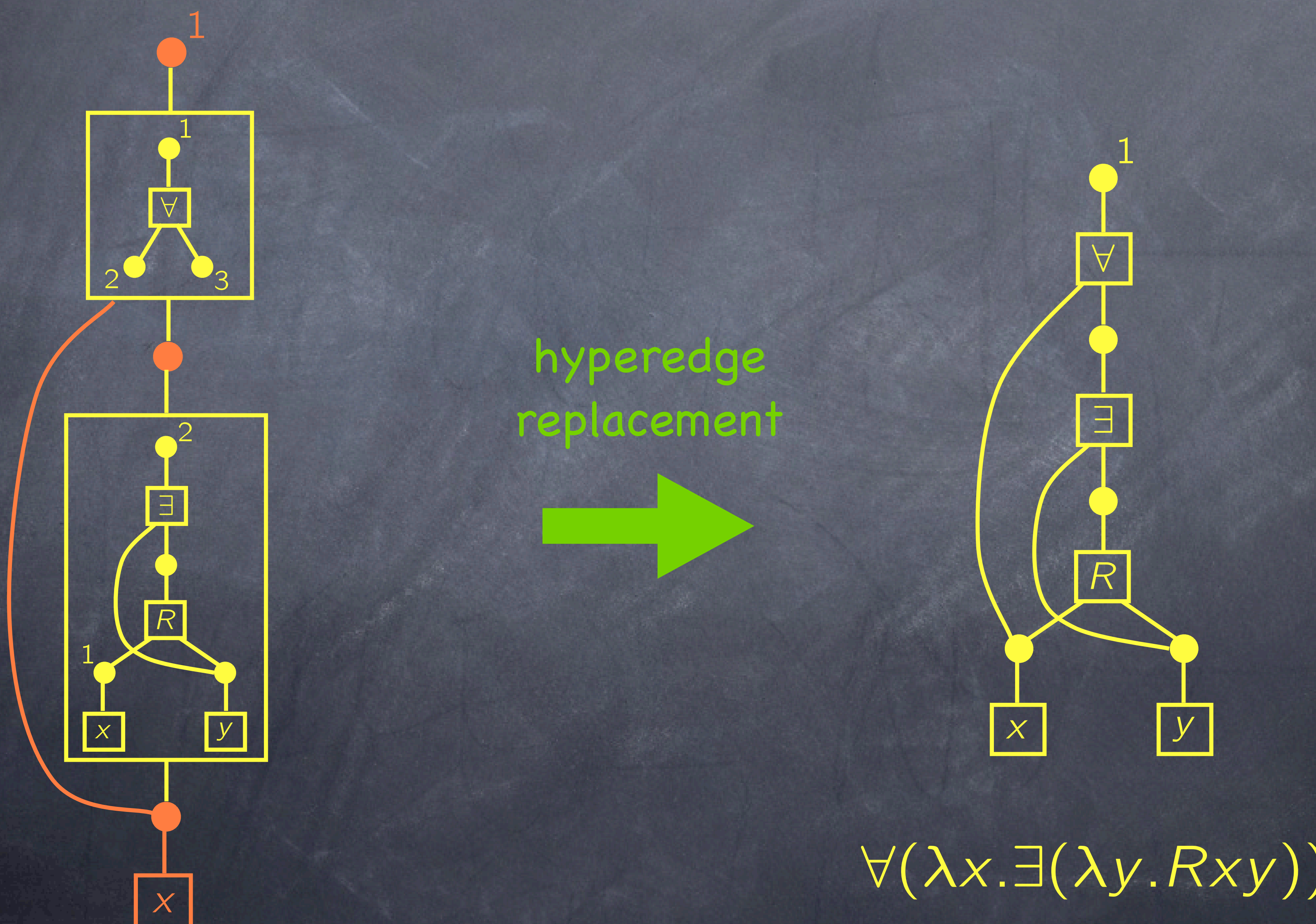
$$NP(\lambda u.\forall(\lambda x.ux)).$$

$$NP(\lambda u.\exists(\lambda x.ux)).$$

VP (

) $:- V(X), NP(Y).$

Linear β-reduction as hyperedge replacement



Second-order ACGs vs. hyperedge replacement grammars

$\mathrm{MLCFTL} \subseteq \mathit{TR}(\mathbf{A})$

$\mathrm{MLCFTL} \subseteq \mathit{TR}(\mathbf{HR})$

$\mathit{LT}(\mathbf{A}) \subseteq \mathit{LT}(\mathbf{HR})$ λ-term generating power

Second-order ACGs vs. hyperedge replacement grammars

$\mathrm{MLCFTL} \subseteq \mathit{TR}(\mathbf{A})$

$\mathrm{MLCFTL} \subseteq \mathit{TR}(\mathbf{HR})$

$\mathit{LT}(\mathbf{A}) \subseteq \mathit{LT}(\mathbf{HR})$ λ-term generating power

$\mathit{TR}(\mathbf{HR}) \subseteq \mathrm{MLCFTL}$ (Engelfriet and Maneth 2000)

Second-order ACGs vs. hyperedge replacement grammars

$\mathrm{MLCFTL} \subseteq TR(\mathbf{A})$

$\mathrm{MLCFTL} \subseteq TR(\mathbf{HR})$

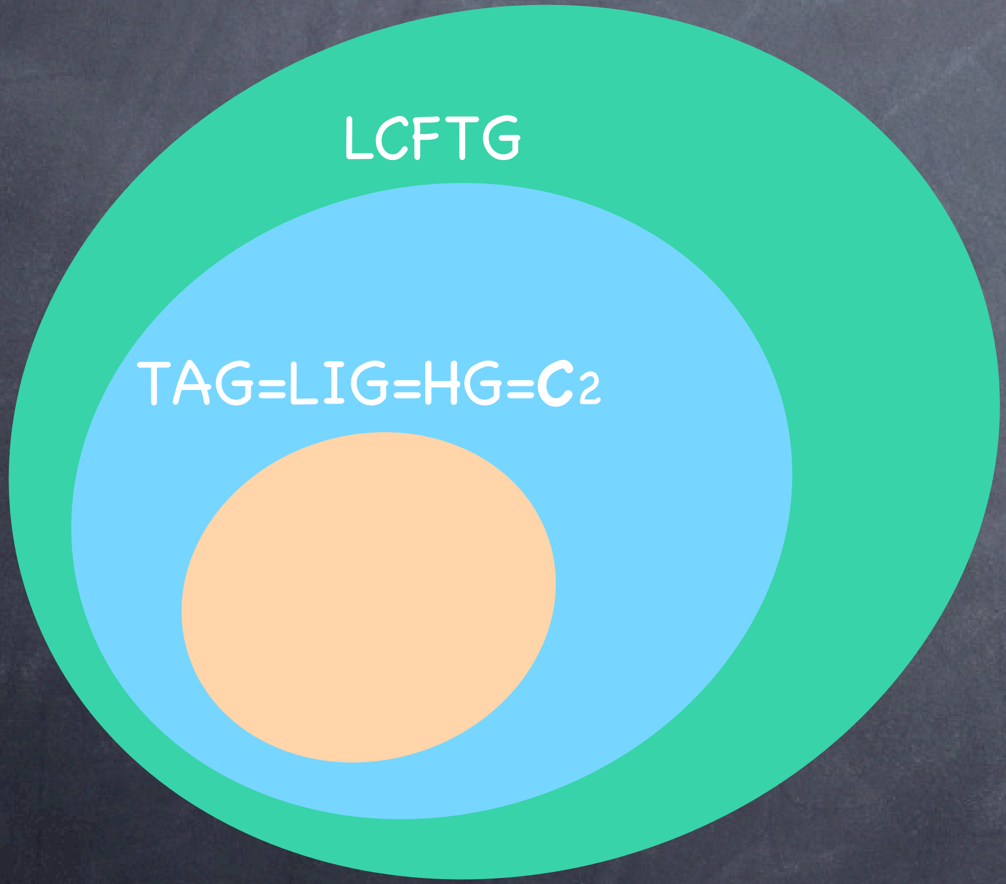
$LT(\mathbf{A}) \subseteq LT(\mathbf{HR})$ λ-term generating power

$TR(\mathbf{HR}) \subseteq \mathrm{MLCFTL}$ (Engelfriet and Maneth 2000)

$$TR(\mathbf{A}) = TR(\mathbf{HR})$$

Convergence of mildly context-sensitive grammar formalisms

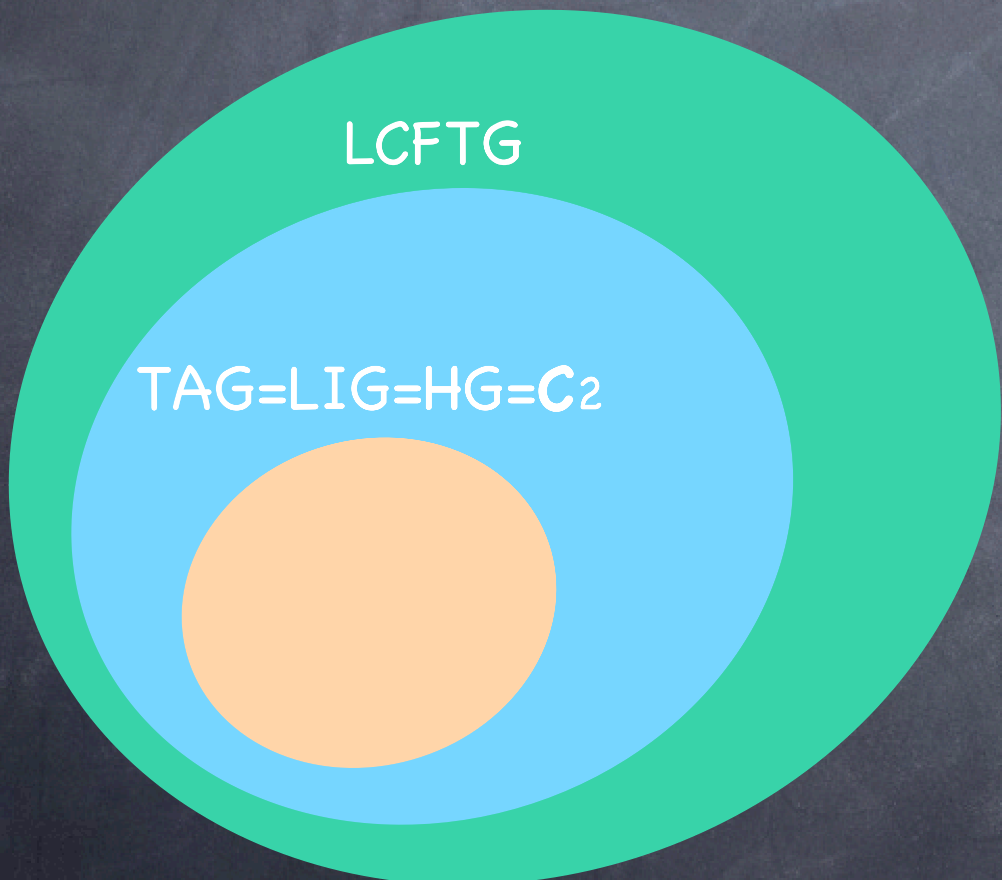
MCFG=MCTAG=DTWT=MG=HR=**A**



string languages

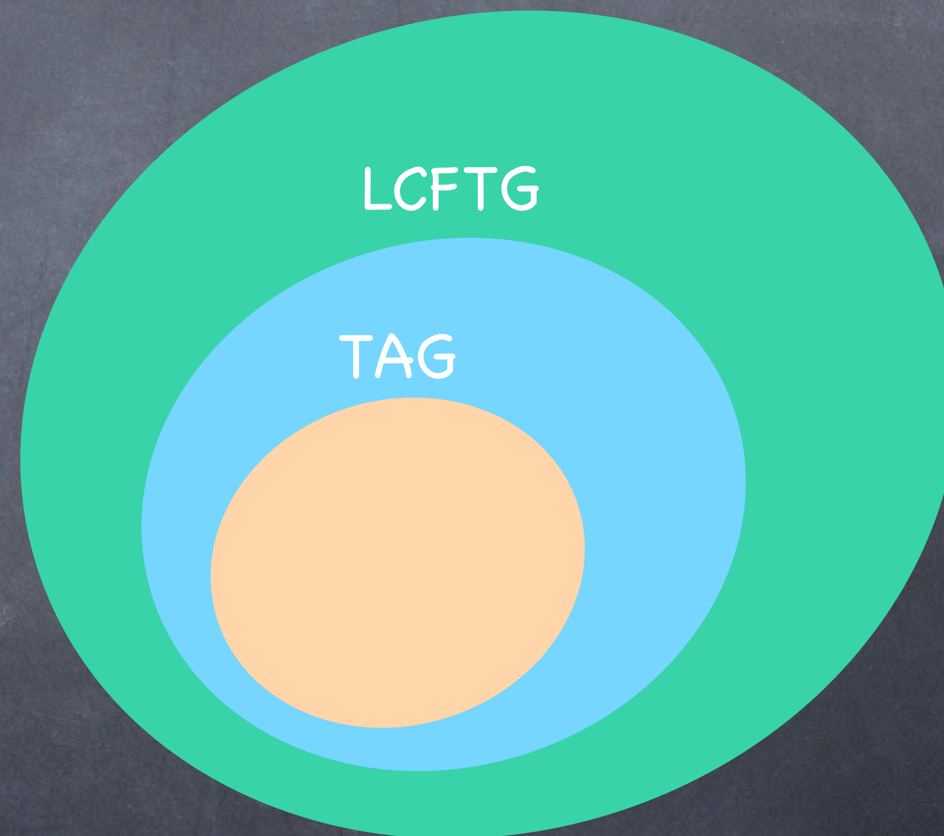
Convergence of mildly context-sensitive grammar formalisms

MCFG=MCTAG=DTWT=MG=HR=A



string languages

sur-AT=HR=A=MLCFTG



tree languages