On Word and Frontier Languages of Unsafe Higher-Order Grammars

Kazuyuki Asada Naoki Kobayashi

University of Tokyo

This talk

Theorem

the class of order-(n+1) word languages
$=$ the class of order-n frontier languages

(safe case: [Damm, 82])

Higher-order grammar
[Maslov 74, Wand 74]

- extension of CFG where non-terminals may take parameters of higher-order functions on trees
- applied to higher-order program verification
- natural extension of familiar language classes [Wand 74]
  - order-0 word language = regular language
  - order-1 word language = CFL
  - order-2 word language = indexed language

Order-1 Word = Order-0 Frontier (= CFL)

- order-1 word grammar G_1

  $S \to F\ \boldsymbol{e}$ $\quad F\ x \to \mathrm{a}\ (F\ (\mathrm{b}\ x))$ $\quad F\ x \to x$

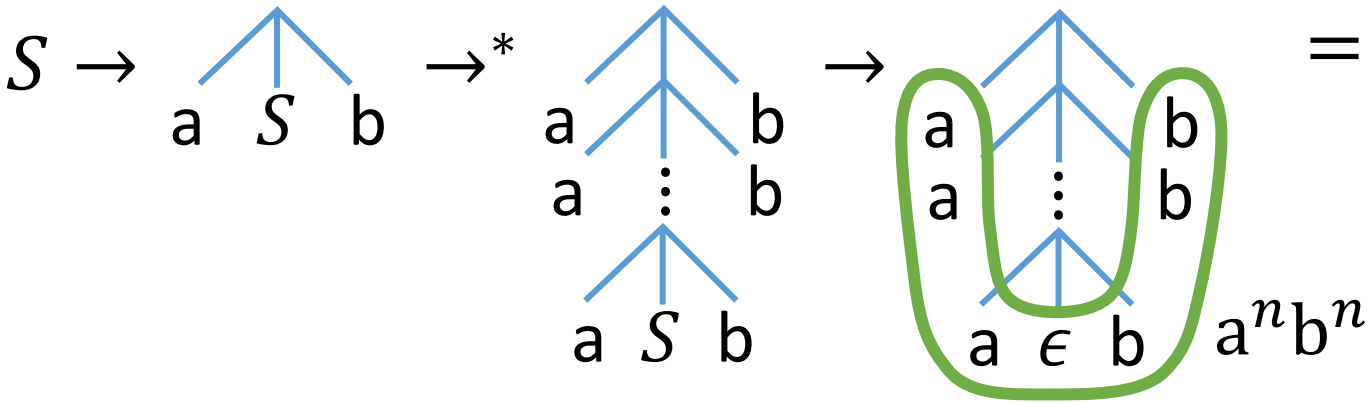
  $S \to F\ \boldsymbol{e} \to \mathrm{a}(F(\mathrm{b}\ \boldsymbol{e})) \to^* \mathrm{a}^n(F(\mathrm{b}^n(\boldsymbol{e}))) \to \mathrm{a}^n(\mathrm{b}^n(\boldsymbol{e}))$

  $L_{\mathrm{word}}(G_1) = \{\mathrm{a}^n\mathrm{b}^n | n \geq 0\}$

- order-0 frontier grammar G_2

  $S \to \mathrm{br\ a}\ S\ \mathrm{b}$ $\quad S \to \epsilon$

$L_{\mathrm{leaf}}(G_2)$
$= \{\mathrm{a}^n\mathrm{b}^n | n \geq 0\}$
$= L_{\mathrm{word}}(G_1)$



Applications of the theorem

- unsafe order-2 word languages
  = safe order-2 word languages [Aehlig+ 05]
- context-sensitivity of order-3 word languages
  - by context-sensitivity of
    order-2 frontier languages [K.+ 14]
- a special case of diagonal problem
  (general case: [Clemente+ 16])
  - by a proof method suggested in [Damm 82]

Outline

- Introduction
- Definition and main theorem
- Application
  - General method for reasoning about higher-order languages
- Proof of main theorem
- Related work and conclusion

Higher-order grammar

- extension of CFG where non-terminals may take parameters of higher-order functions on trees

Example of order-1 grammar:

$S : o$ $\quad S \to F\ \mathrm{a}$ $\quad F\ x \to \mathrm{br}\ x\ x$

$F : o \to o$ $\quad S \to F\ \mathrm{b}$ $\quad F\ x \to F(\mathrm{br}\ \mathrm{a}\ x)$

$F\ x \to F(\mathrm{br}\ \mathrm{b}\ x)$

Higher-order grammar

- extension of CFG where non-terminals may take parameters of higher-order functions on trees

Example of order-1 grammar:

$S : o$

$F : o \to o$

$S \to F\ \mathrm{a}$

$S \to F\ \mathrm{b}$

$F\ x \to \mathrm{br}\ x\ x$

$F\ x \to F(\mathrm{br}\ \mathrm{a}\ x)$

$F\ x \to F(\mathrm{br}\ \mathrm{b}\ x)$

simply typed

Higher-order grammar

- extension of CFG where non-terminals may take parameters of higher-order functions on trees

Example of order-1 grammar:

$S : o$ $\quad S \to F\ \mathrm{a}$ $\quad F\ x \to \mathrm{br}\ x\ x$

$F : o \to o$ $\quad S \to F\ \mathrm{b}$ $\quad F\ x \to F(\mathrm{br}\ \mathrm{a}\ x)$

$F\ x \to F(\mathrm{br}\ \mathrm{b}\ x)$

$S \to F\ \mathrm{b}$

Higher-order grammar

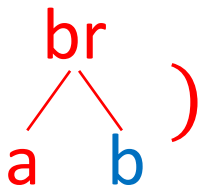
- extension of CFG where non-terminals may take parameters of higher-order functions on trees

Example of order-1 grammar:

$S : o$ $\quad S \to F\ \mathrm{a}$ $\quad F\ x \to \mathrm{br}\ x\ x$

$F : o \to o$ $\quad S \to F\ \mathrm{b}$ $\quad F\ x \to F(\mathrm{br}\ \mathrm{a}\ x)$

$F\ x \to F(\mathrm{br}\ \mathrm{b}\ x)$

$S \to F\ \mathrm{b} \to F($ 

$)$

Higher-order grammar

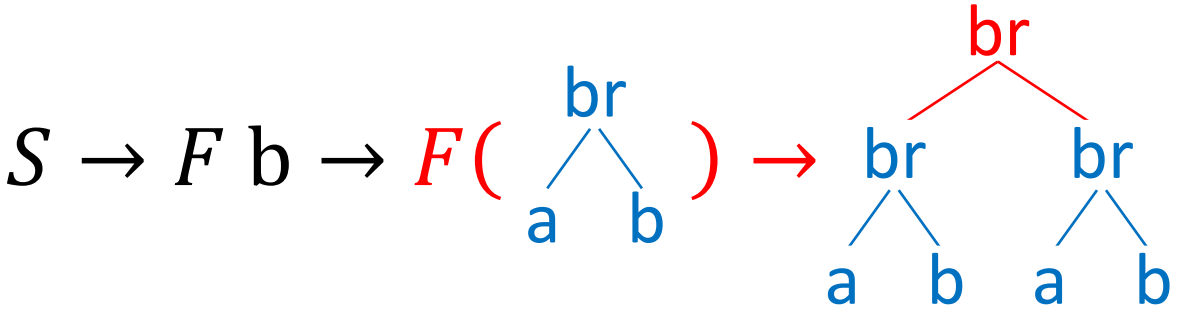
- extension of CFG where non-terminals may take parameters of higher-order functions on trees

Example of order-1 grammar:

$S : o$ $\quad S \to F\ \mathrm{a}$ $\quad F\ x \to \mathrm{br}\ x\ x$

$F : o \to o$ $\quad S \to F\ \mathrm{b}$ $\quad F\ x \to F(\mathrm{br}\ \mathrm{a}\ x)$

$F\ x \to F(\mathrm{br}\ \mathrm{b}\ x)$



Higher-order grammar

- extension of CFG where non-terminals may take parameters of higher-order functions on trees

Example of order-1 grammar:

$$L(G) = \left\{ \begin{array}{c} \text{br} \\ \text{br} \quad\quad \text{br} \\ a_1 \quad \ddots \quad a_1 \quad \ddots \\ \quad\quad \text{br} \quad\quad\quad \text{br} \\ \quad\quad a_n \quad a_{n+1} \quad\quad a_n \quad a_{n+1} \end{array} \;\middle|\; \begin{array}{l} a_i \in \{\mathrm{a}, \mathrm{b}\} \\ \\ n \geq 0 \end{array} \right\}$$

Relation to λ-calculus

a higher-order grammar

$=$ a ground closed term of λY-calculus with first-order constants and finite non-determinism

$$S \to F\ \mathrm{a} \qquad F\ x \to \mathrm{br}\ x\ x$$
$$S \to F\ \mathrm{b} \qquad F\ x \to F(\mathrm{br}\ \mathrm{a}\ x)$$
$$F\ x \to F(\mathrm{br}\ \mathrm{b}\ x)$$

---

let rec $S = F\ \mathrm{a} \oplus F\ \mathrm{b}$ in
let rec $F\ x = \mathrm{br}\ x\ x \oplus F(\mathrm{br}\ \mathrm{a}\ x) \oplus F(\mathrm{br}\ \mathrm{b}\ x)$
in S

Word grammar/language

A word grammar G has terminals

- 0-ary: $\boldsymbol{e}$
- 1-ary: a, b, c, …

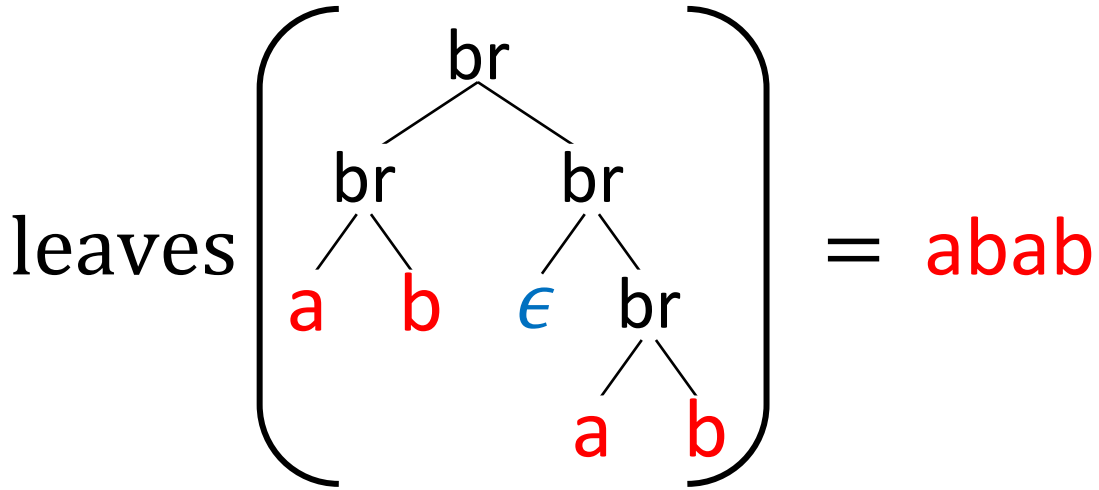
$$L_{\text{word}}(G) = \{\ a_1 a_2 \ldots a_n \mid S \rightarrow^* \begin{array}{|c|}\hline a_1 \\ \hline a_2 \\ \hline \vdots \\ \hline a_n \\ \hline \boldsymbol{e} \\ \hline \end{array}\ \}$$

Frontier grammar/language

A frontier grammar G has terminals

- 2-ary: br
- 0-ary: a, b, c, … , ϵ (ϵ : "empty-word")

$$L_{\text{leaf}}(G) = \{\, \text{leaves}(\pi) \mid \pi \in L(G) \,\}$$



Main theorem

For any $n \in \{0, 1, \ldots\}$,

$\{L_{\text{word}}(G) \mid G\text{: order}-(n+1) \text{ word grammar}\}$

$=$

$\{L_{\text{leaf}}(G) \mid G\text{: order}-n \text{ frontier grammar}\}$

Outline

- Introduction
- Definition and main theorem
- Application
  - General method for reasoning about higher-order languages
- Proof of main theorem
- Related work and conclusion

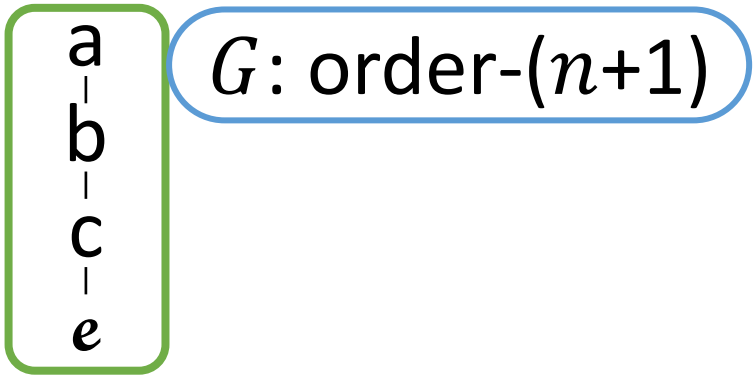
General proof method by theorem: induction on order [Damm 82]

General proof method by theorem: induction on order [Damm 82]

P: property on word languages

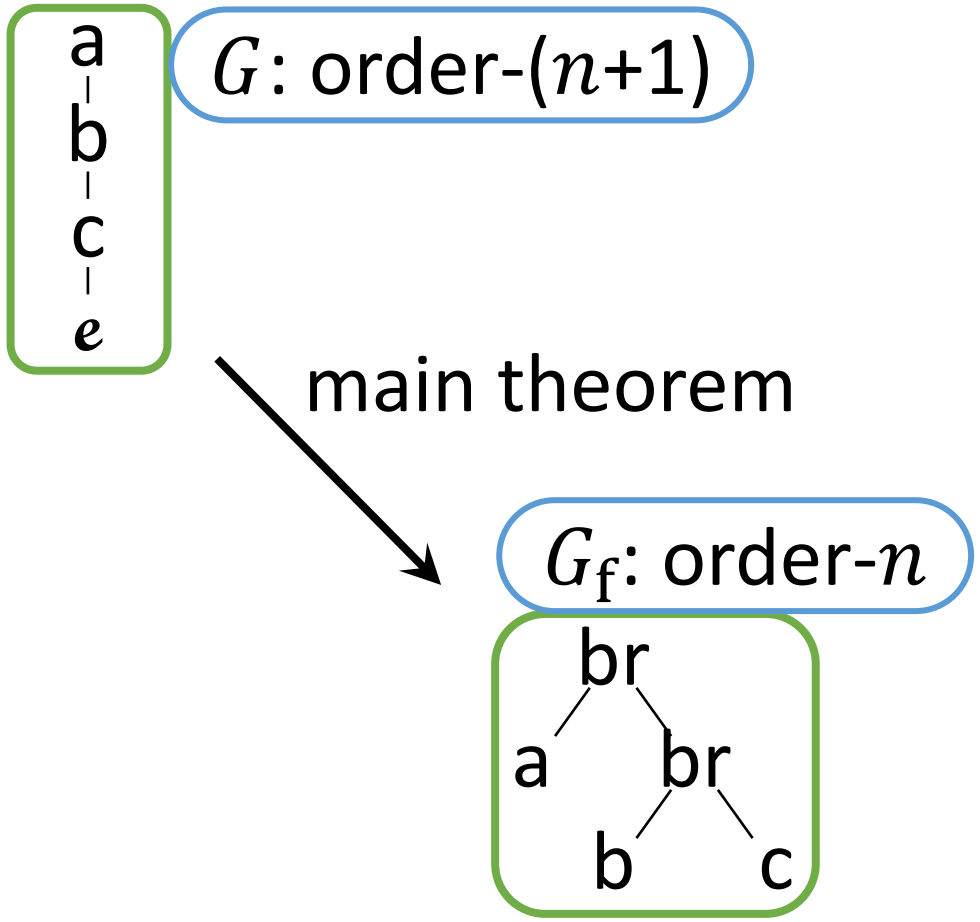
General proof method by theorem: induction on order [Damm 82]

P: property on word languages



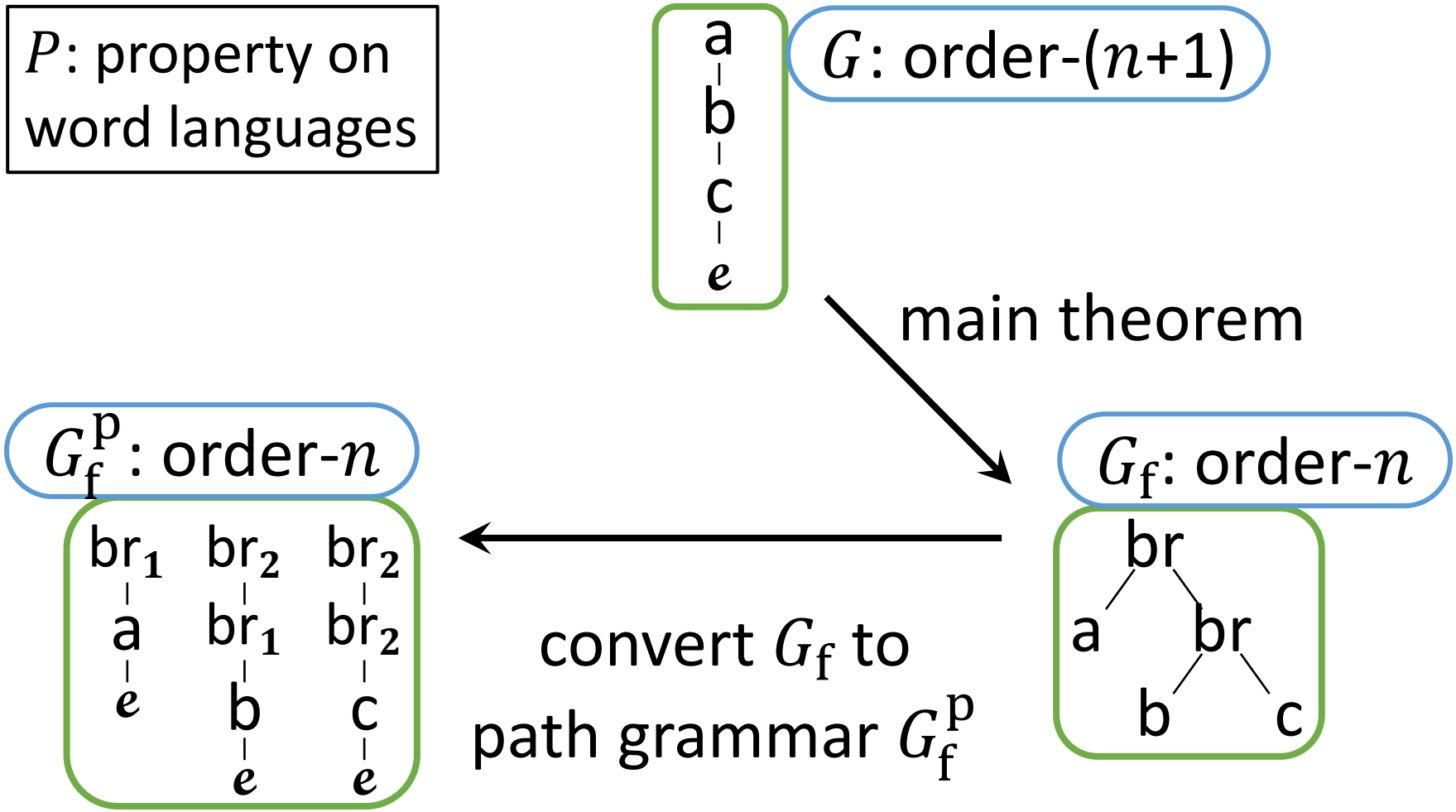
General proof method by theorem: induction on order [Damm 82]

P: property on word languages

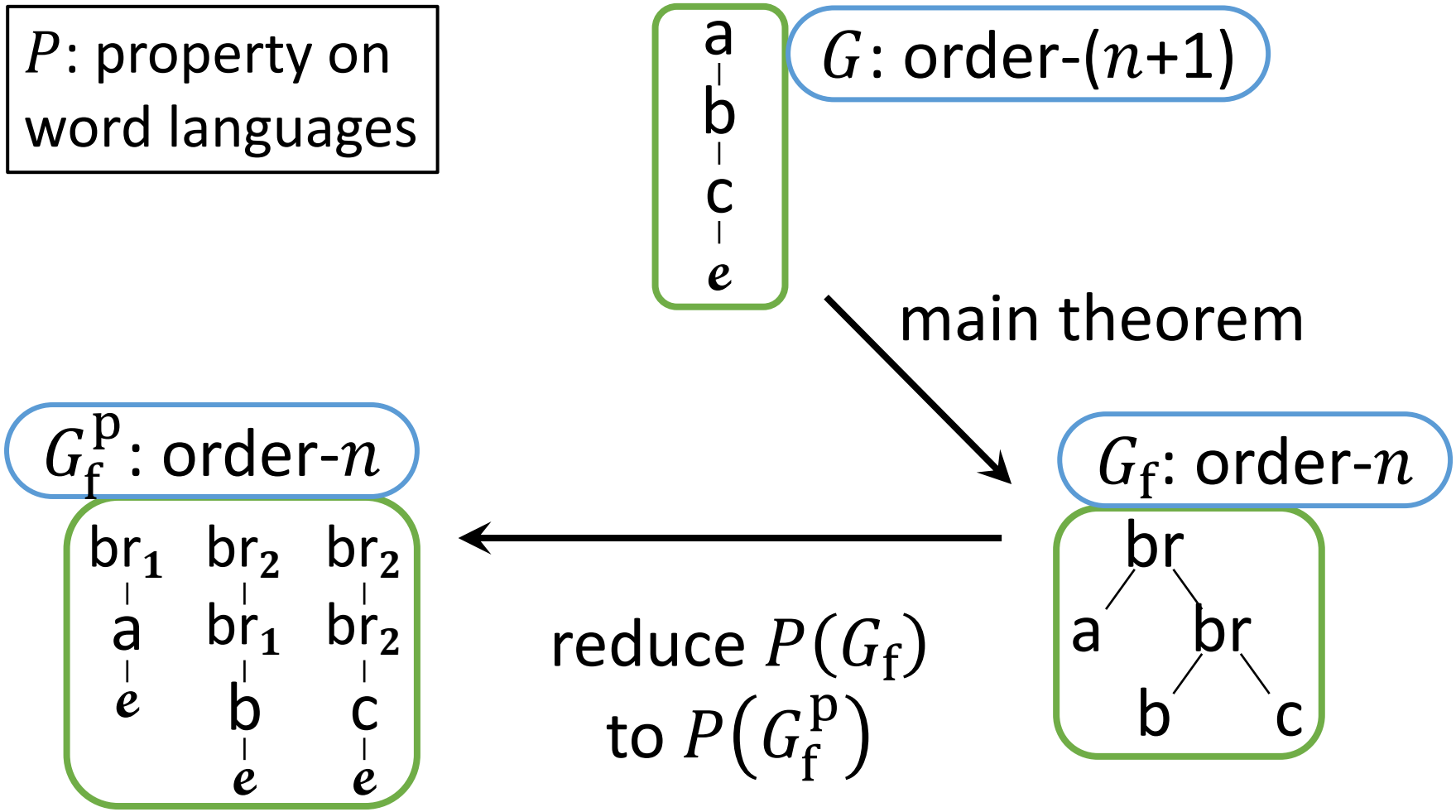


General proof method by theorem: induction on order [Damm 82]

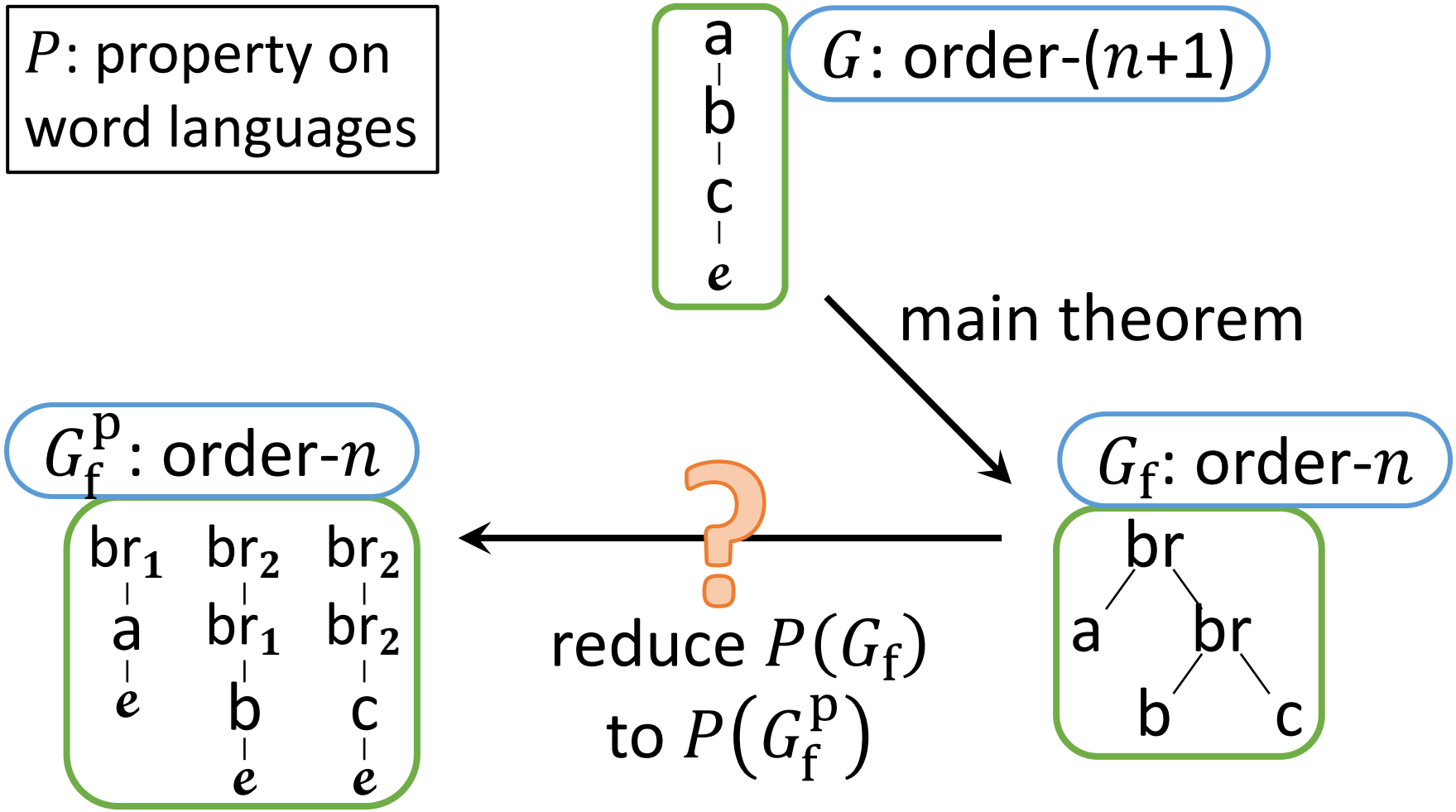
P: property on word languages



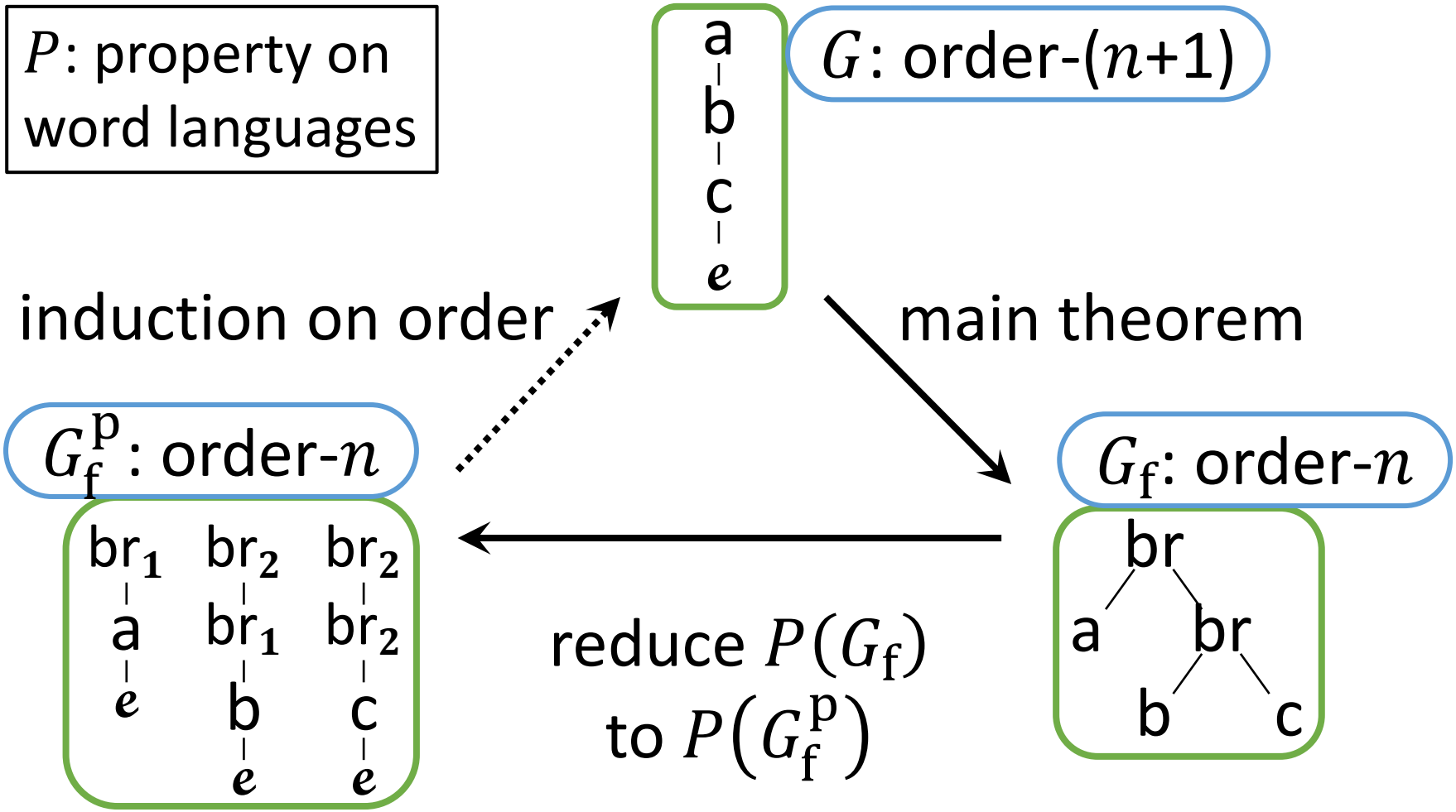
General proof method by theorem: induction on order [Damm 82]



General proof method by theorem: induction on order [Damm 82]



General proof method by theorem: induction on order [Damm 82]



Outline

- Introduction
- Definition and main theorem
- Application
  - General method for reasoning about higher-order languages
- Proof of main theorem
- Related work and conclusion

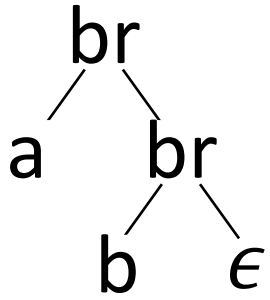
Proof outline

1. Constructing two transformations:
   - order-n frontier $\rightarrow$ order-(n+1) word
     - easy
   - order-(n+1) word $\rightarrow$ order–n frontier
     - difficult
2. Proving that they preserve languages

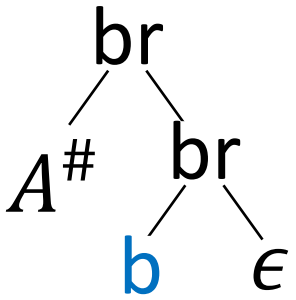
Order-(n+1) word ⇒ order-n frontier (basic idea)

"$t_1\ t_2 \Rightarrow \mathrm{br}\ t_1^{\#}\ t_2^{\#}$" $(t_1 : o \to o)$

- Example:
  - a (b $\boldsymbol{e}$) ⇒



- Problem:
  - consider $A\ x \to a\ \boldsymbol{e}$
  - NG: A (b $\boldsymbol{e}$) ⇒



  - OK: A (b $\boldsymbol{e}$) ⇒ $A^{\#}$

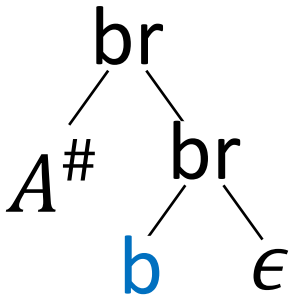
Order-(n+1) word ⇒ order-n frontier (basic idea)

"$t_1\ t_2 \ \Rightarrow\ \mathrm{br}\ t_1^{\#}\ t_2^{\#}$" $\quad (t_1 : o \to o)$

- Exar
  - a

We need useless analysis of arguments

- Problem:
  - consider $A\ x \to a\ e$
  - NG: $A\ (\mathrm{b}\ e) \ \Rightarrow$



  - OK: $A\ (\mathrm{b}\ e) \Rightarrow A^{\#}$

Useless analysis of arguments by intersection types

$\delta ::= o \mid \delta_1 \wedge \cdots \wedge \delta_k \to \delta$

(we write $\top$ for empty intersection)

- e.g., $t : \top \to o \to \top \to o$ uses the second argument, but not the first nor third ones
  - $\top$: argument is not used
  - o: argument is used

Type-directed transformation

- we define transformation of the form

$$\Gamma \vdash t : \delta \Rightarrow u$$

  where Γ is intersection type environment

Type-directed transformation

$$\frac{\Gamma_0 \vdash s : \delta_1 \wedge \cdots \wedge \delta_k \to \delta \Rightarrow v \qquad \Gamma_i \vdash t : \delta_i \Rightarrow U_i \text{ and } \delta_i \neq \mathsf{o} \text{ (for each } i \in \{1,\ldots,k\})}{\Gamma_0 \cup \Gamma_1 \cup \cdots \cup \Gamma_k \vdash st : \delta \Rightarrow vU_1 \cdots U_k}$$

$$\frac{\mathbf{bal}(\Gamma)}{\Gamma, x : \delta \vdash x : \delta \Rightarrow x_\delta}$$

$$\frac{\Gamma_0 \vdash s : \mathsf{o} \to \delta \Rightarrow V \qquad \Gamma_1 \vdash t : \mathsf{o} \Rightarrow U}{\Gamma_0 \cup \Gamma_1 \vdash st : \delta \Rightarrow \mathtt{br}\, V\, U}$$

$$\frac{\delta :: \mathcal{N}(A) \qquad \mathbf{bal}(\Gamma)}{\Gamma \vdash A : \delta \Rightarrow A_\delta}$$

$$\frac{\Gamma \vdash t : \delta \Rightarrow u_i \text{ (for each } i \in \{1,\ldots,k\}) \qquad k \geq 1}{\Gamma \vdash t : \delta \Rightarrow \{u_1, \ldots, u_k\}}$$

$$\frac{\mathbf{bal}(\Gamma)}{\Gamma \vdash \mathtt{e} : \mathsf{o} \Rightarrow \mathtt{e}}$$

$$\frac{\Gamma, x : \delta_1, \ldots, x : \delta_k \vdash t : \delta \Rightarrow u \qquad x \notin dom(\Gamma) \qquad \delta_i \neq \mathsf{o} \text{ for each } i \in \{1,\ldots,k\}}{\Gamma \vdash \lambda x.t : \delta_1 \wedge \cdots \wedge \delta_k \to \delta \Rightarrow \lambda x_{\delta_1} \cdots \lambda x_{\delta_k}.u}$$

$$\frac{\Sigma(a) = 1 \qquad \mathbf{bal}(\Gamma)}{\Gamma \vdash a : \mathsf{o} \to \mathsf{o} \Rightarrow a}$$

$$\frac{\Gamma, x : \mathsf{o} \vdash t : \delta \Rightarrow u}{\Gamma \vdash \lambda x.t : \mathsf{o} \to \delta \Rightarrow [\mathtt{e}/x_\mathsf{o}]u}$$

Application rules (first-order)

Corresponding to the examples:

- $\mathrm{a}\,(\mathrm{b}\,\boldsymbol{e}) \;\Rightarrow\; \mathrm{br}\,\mathrm{a}\,(\mathrm{br}\,\mathrm{b}\,\epsilon)$
- $A(\mathrm{b}\,\boldsymbol{e}) \;\Rightarrow\; A \quad (Ax \to \mathrm{a}\,\boldsymbol{e})$

we have two rules (δ : order-1):

$$\frac{\Gamma \vdash s : o \to \delta \Rightarrow v \qquad \Gamma' \vdash t : o \Rightarrow u}{\Gamma \cup \Gamma' \vdash s\,t : \delta \Rightarrow \mathrm{br}\,v\,u}$$

$$\frac{\Gamma \vdash s : \mathsf{T} \to \delta \Rightarrow v}{\Gamma \vdash s\,t : \delta \Rightarrow v}$$

Application rule (higher-order)

$$\frac{\Gamma \vdash s : \delta_1 \wedge \cdots \wedge \delta_k \to \delta \Rightarrow v \qquad \Gamma_i \vdash t : \delta_i \Rightarrow u_i \quad (\forall i \in \{1, \ldots, k\})}{\Gamma \cup \Gamma_1 \cup \cdots \cup \Gamma_k \vdash s\ t : \delta \Rightarrow v\ u_1 \cdots u_k}$$

- t has different roles $(\delta_1, \ldots, \delta_k)$, due to non-determinism
- s really uses these t's with different roles

$$\frac{\Gamma \vdash s : \delta_1 \wedge \cdots \wedge \delta_k \to \delta \Rightarrow v \qquad \Gamma_i \vdash t : \delta_i \Rightarrow u_i \quad (\forall i \in \{1, \ldots, k\})}{\Gamma \cup \Gamma_1 \cup \cdots \cup \Gamma_k \vdash s\ t : \delta \Rightarrow v\ u_1 \cdots u_k}$$

- t has different roles $(\delta_1, \ldots, \delta_k)$, due to non-determinism
- s really uses these t's with different roles

$$\frac{\Gamma \vdash s : \delta_1 \wedge \cdots \wedge \delta_k \to \delta \Rightarrow v \qquad \Gamma_i \vdash t : \delta_i \Rightarrow u_i \quad (\forall i \in \{1, \ldots, k\})}{\Gamma \cup \Gamma_1 \cup \cdots \cup \Gamma \vdash s\ t : \delta \Rightarrow v\ u_1 \cdots u_k}$$

we distinguish "t's with different roles" by labeling intersection types

- t has different roles $(\delta_1, \ldots, \delta_k)$, due to non-determinism
- s really uses these t's with different roles

$$\frac{\Gamma \vdash s : \delta_1 \wedge \cdots \wedge \delta_k \to \delta \Rightarrow v \qquad \Gamma_i \vdash t : \delta_i \Rightarrow u_i \quad (\forall i \in \{1, \ldots, k\})}{\Gamma \cup \Gamma_1 \cup \cdots \cup \Gamma_k \vdash s\ t : \delta \Rightarrow v\ u_1 \cdots u_k}$$

$$\frac{}{x : \delta \vdash x : \delta \Rightarrow x_\delta}$$

$$\frac{A : \kappa \qquad \delta :: \kappa}{\vdash A : \delta \Rightarrow A_\delta}$$

Example of transformation

- Consider $A\ f \to f\,(f\,(f\,(\mathrm{a}\ \boldsymbol{e})))$

  $B\ x \to \mathrm{b}\ x \qquad (\lambda x.\,\mathrm{b}\ x\ :\ o \to o)$

  $B\ x \to\ \boldsymbol{e} \qquad (\lambda x.\ \boldsymbol{e}\ :\ \top \to o)$

- For $A\ B \to B\,(B\,(B\,(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}\,(\mathrm{b}\,(B\,(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}\,(\mathrm{b}\ \boldsymbol{e})$

  we have $f : o \to o,\ f : \top \to o \vdash f\,(f\,(f\,(\mathrm{a}\ \boldsymbol{e}))) : o$

  $\Rightarrow \mathrm{br}\ f_{o\to o}\ (\mathrm{br}\ f_{o\to o}\ f_{\top\to o})$

- This produces a rule:

  $A_{(o\to o)\wedge(\top\to o)\to o}\ f_{o\to o}\ f_{\top\to o} \to \mathrm{br}\ f_{o\to o}\ (\mathrm{br}\ f_{o\to o}\ f_{\top\to o})$

  also: $B_{o\to o} \to \mathrm{b} \qquad B_{\top\to o} \to \epsilon$

- We have: $\vdash A\ B : o \Rightarrow A_{(o\to o)\wedge(\top\to o)\to o}\ B_{o\to o}\ B_{\top\to o}$

Example of transformation

- Consider $A\ f \to f(f(f(\mathrm{a}\ \boldsymbol{e})))$

  $B\ x \to \mathrm{b}\ x \qquad (\lambda x.\mathrm{b}\ x\ :\ o \to o)$

  $B\ x \to \boldsymbol{e} \qquad (\lambda x.\ \boldsymbol{e}\ :\ \top \to o)$

- For $A\ B \to B(B(B(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}(\mathrm{b}(B(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}(\mathrm{b}\ \boldsymbol{e})$

  we have $f : o \to o, f : \top \to o \vdash f(f(f(\mathrm{a}\ \boldsymbol{e}))) : o$

$$\frac{\Gamma \vdash s : \delta_1 \wedge \cdots \wedge \delta_k \to \delta \Rightarrow v \qquad \Gamma_i \vdash t : \delta_i \Rightarrow u_i \quad (\forall i \in \{1, \ldots, k\})}{\Gamma \cup \Gamma_1 \cup \cdots \cup \Gamma_k \vdash s\ t : \delta \Rightarrow v\ u_1 \cdots u_k}$$

- We have: $\vdash A\ B : o \Rightarrow A_{(o\to o)\wedge(\top\to o)\to o}\ B_{o\to o}\ B_{\top\to o}$

Example of transformation

- Consider $A\ f \to f\,(f\,(f\,(\mathrm{a}\ \boldsymbol{e})))$

  $B\ x \to \mathrm{b}\ x \qquad (\lambda x.\,\mathrm{b}\ x\ :\ o \to o)$

  $B\ x \to\ \boldsymbol{e} \qquad (\lambda x.\ \boldsymbol{e}\ :\ \top \to o)$

- For $A\ B \to B\,(B\,(B\,(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}\,(\mathrm{b}\,(B\,(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}\,(\mathrm{b}\ \boldsymbol{e})$

  we have $f : o \to o,\ f : \top \to o \vdash f\,(f\,(f\,(\mathrm{a}\ \boldsymbol{e}))) : o$

  $\Rightarrow \mathrm{br}\ f_{o\to o}\ (\mathrm{br}\ f_{o\to o}\ f_{\top\to o})$

- This produces a rule:

  $A_{(o\to o)\wedge(\top\to o)\to o}\ f_{o\to o}\ f_{\top\to o} \to \mathrm{br}\ f_{o\to o}\ (\mathrm{br}\ f_{o\to o}\ f_{\top\to o})$

  also: $B_{o\to o} \to \mathrm{b} \qquad B_{\top\to o} \to \epsilon$

- We have: $\vdash A\ B : o \Rightarrow A_{(o\to o)\wedge(\top\to o)\to o}\ B_{o\to o}\ B_{\top\to o}$

Example of transformation

- Consider $A\ f \to f(f(f(\mathrm{a}\ \boldsymbol{e})))$

  $B\ x \to \mathrm{b}\ x \qquad (\lambda x.\mathrm{b}\ x\ :\ o \to o)$

  $B\ x \to \boldsymbol{e} \qquad (\lambda x.\ \boldsymbol{e}\ :\ \top \to o)$

- For $A\ B \to B(B(B(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}(\mathrm{b}(B(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}(\mathrm{b}\ \boldsymbol{e})$

$$A_{(o\to o)\wedge(\top\to o)\to o} B_{o\to o} B_{\top\to o}$$
$$\to \mathrm{br}\ B_{o\to o}\ (\mathrm{br}\ B_{o\to o}\ B_{\top\to o}) \to^{*} \mathrm{br}\ \mathrm{b}\ (\mathrm{br}\ \mathrm{b}\ \epsilon)$$

$A_{(o\to o)\wedge(\top\to o)\to o}\ f_{o\to o}\ f_{\top\to o} \to \mathrm{br}\ f_{o\to o}\ (\mathrm{br}\ f_{o\to o}\ f_{\top\to o})$

also: $B_{o\to o} \to \mathrm{b} \qquad B_{\top\to o} \to \epsilon$

- We have: $\vdash A\ B : o \Rightarrow A_{(o\to o)\wedge(\top\to o)\to o}\ B_{o\to o}\ B_{\top\to o}$

Example of transformation

- Consider $A\ f \to f(f(f(\mathrm{a}\ \boldsymbol{e})))$

  $B\ x \to \mathrm{b}\ x \qquad (\lambda x.\mathrm{b}\ x\ :\ o \to o)$

  $B\ x \to\ \boldsymbol{e} \qquad (\lambda x.\ \boldsymbol{e}\ :\ \top \to o)$

- For $A\ B \to B(B(B(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}(\mathrm{b}(B(\mathrm{a}\ \boldsymbol{e}))) \to \mathrm{b}(\mathrm{b}\ \boldsymbol{e})$

$A_{(o\to o)\wedge(\top\to o)\to o} B_{o\to o} B_{\top\to o}$
$\to \mathrm{br}\ B_{o\to o}\ (\mathrm{br}\ B_{o\to o}\ B_{\top\to o}) \to^{*} \mathrm{br}\ \mathrm{b}\ (\mathrm{br}\ \mathrm{b}\ \epsilon)$

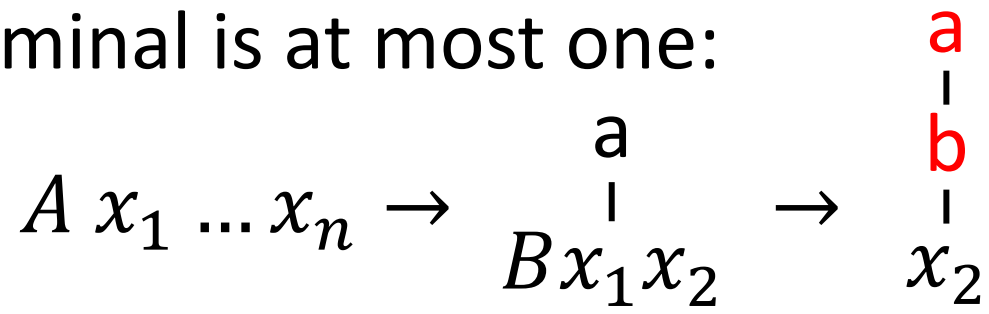
$A_{(o\to o)\wedge(\top\to o)\to o}\ f_{o\to o}\ f_{\top\to o} \to \mathrm{br}\ f_{o\to o}\ (\mathrm{br}\ f_{o\to o}\ f_{\top\to o})$

also: $B_{o\to o} \to \mathrm{b} \qquad B_{\top\to o} \to \epsilon$

- We have: $\vdash A\ B : o \Rightarrow A_{(o\to o)\wedge(\top\to o)\to o}\ B_{o\to o}\ B_{\top\to o}$

Linearity

- argument of type o is used at most once since arity of terminal is at most one:



- linear/non-linear type-based transformation for type-preservation and correctness of the transformation
  - linear: $o, \quad \top \rightarrow o, \quad (o \rightarrow o) \rightarrow o, \quad \ldots$
  - non-linear: $o \rightarrow o, \quad \top \rightarrow o \rightarrow \top \rightarrow o, \quad \ldots$
  - ill-formed: $o \rightarrow o \rightarrow o, \quad \ldots$

Outline

- Introduction
- Definition and main theorem
- Application
  - General method for reasoning about higher-order languages
- Proof of main theorem
- Related work and conclusion

Related work

Study for safe higher-order grammar in ’80

- Safe case of the theorem [Damm, TCS 82]

Intersection-type-based transformation
(for unsafe higher-order grammar)

- higher-order data compression [K.+, HOSC 13]
- context sensitivity of order-2 tree languages [K.+, FoSSaCS 14]
- (general case of) diagonal problem [Clemente+, LICS 16]

Conclusion

Thank you!

the class of order-(n+1) word languages
$=$ the class of order-n frontier languages

- application: induction-method on order
- proof: intersection-type-based transformation with linearity

Future work:

- pumping lemma for higher-order grammar by the induction-method on order (on-going)