Трассирующая Нормализация,
Основанная на Игровой Семантике и Частичных Вычислениях

Даниил Березун [1] Neil D. Jones [2]

[1]Санкт-Петербургский Государственный Университет

[2]DIKU, University of Copenhagen (prof. emeritus)

2017

Игровая Семантика (Game Semantics, GS)

Игровая Семантика (Game Semantics, GS)

- Задача *полной абстракции* для языка PCF (full abstraction) *Plotkin* 1977

Игровая Семантика (Game Semantics, GS)

- Задача *полной абстракции* для языка PCF (full abstraction) *Plotkin* 1977
- GS — первое решение
  - *Abramsky, Jagadeesan, Malacaria* (1994)
  - *Hyland, Ong* (1994)

Игровая Семантика (Game Semantics, GS)

- Задача *полной абстракции* для языка PCF (full abstraction) *Plotkin* 1977
- GS — первое решение
  - *Abramsky, Jagadeesan, Malacaria* (1994)
  - *Hyland, Ong* (1994)
- Денотационная составляющая
- Операционная составляющая

Игровая Семантика (Game Semantics, GS)

- Задача *полной абстракции* для языка PCF (full abstraction) *Plotkin* 1977
- GS — первое решение
  - *Abramsky, Jagadeesan, Malacaria* (1994)
  - *Hyland, Ong* (1994)
- Денотационная составляющая
  - Арены, стратегии, категории
- Операционная составляющая
  - Взаимодействие (interaction)
  - Danos, Regnier. PAM, KAM, IAM (1996, 1999, 2004)

Трассирующая Нормализация

ONP (Oxford Normalization Procedure) *Ong* 2015

- Нестандартный подход к вычислениям

Трассирующая Нормализация

ONP (Oxford Normalization Procedure) *Ong* 2015

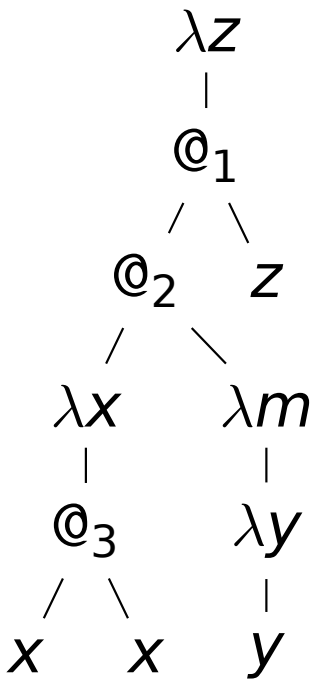
- Нестандартный подход к вычислениям
  - Нормализация путём “обхода” входного терма (traversal-besed normalization)
  - Множество *трасс* определяет η-длинную β-нормальную форму

Трассирующая Нормализация

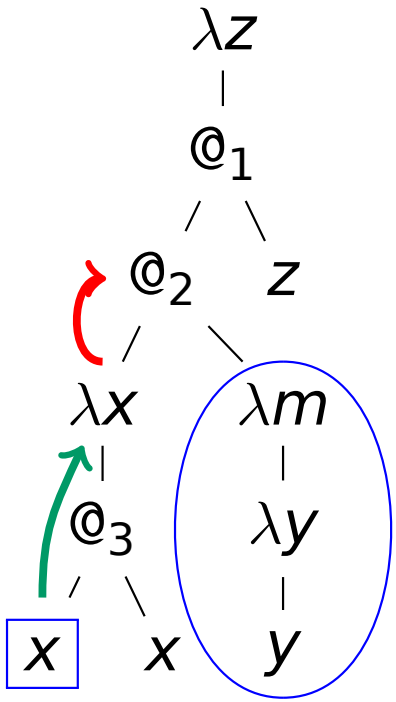
ONP (Oxford Normalization Procedure) *Ong* 2015

- Нестандартный подход к вычислениям
  - Нормализация путём “обхода” входного терма (traversal-besed normalization)
  - Множество *трасс* определяет η-длинную β-нормальную форму
  - Реализует головную линейная редукцию

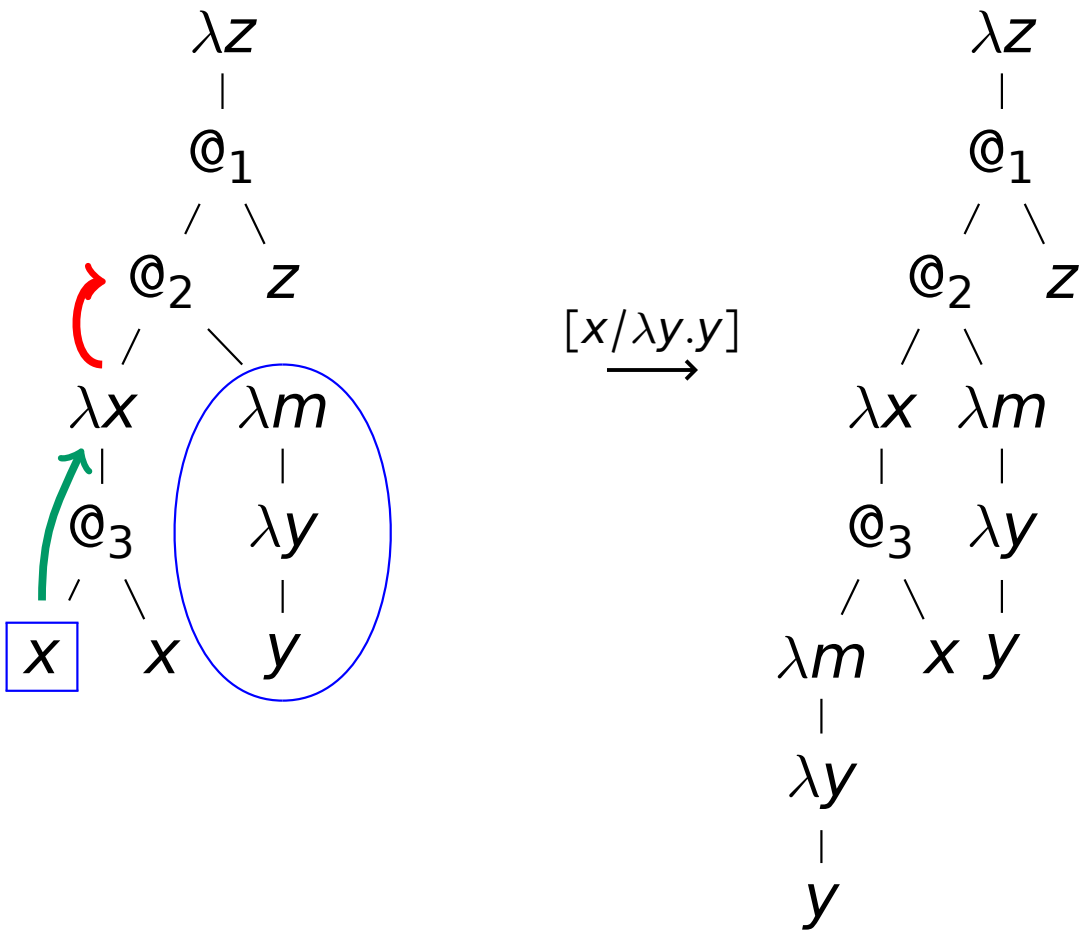
Головная Линейная Редукция



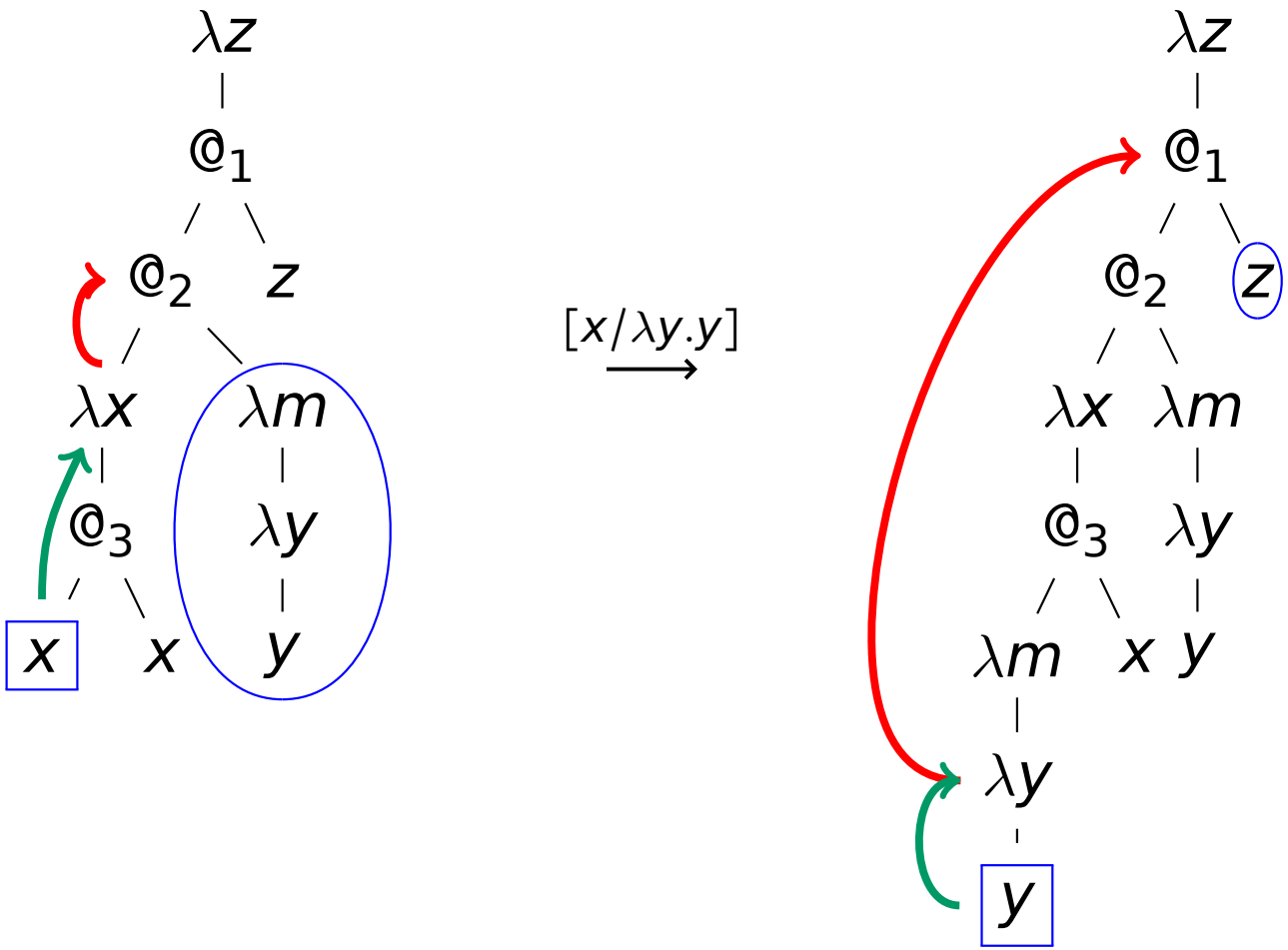
Головная Линейная Редукция



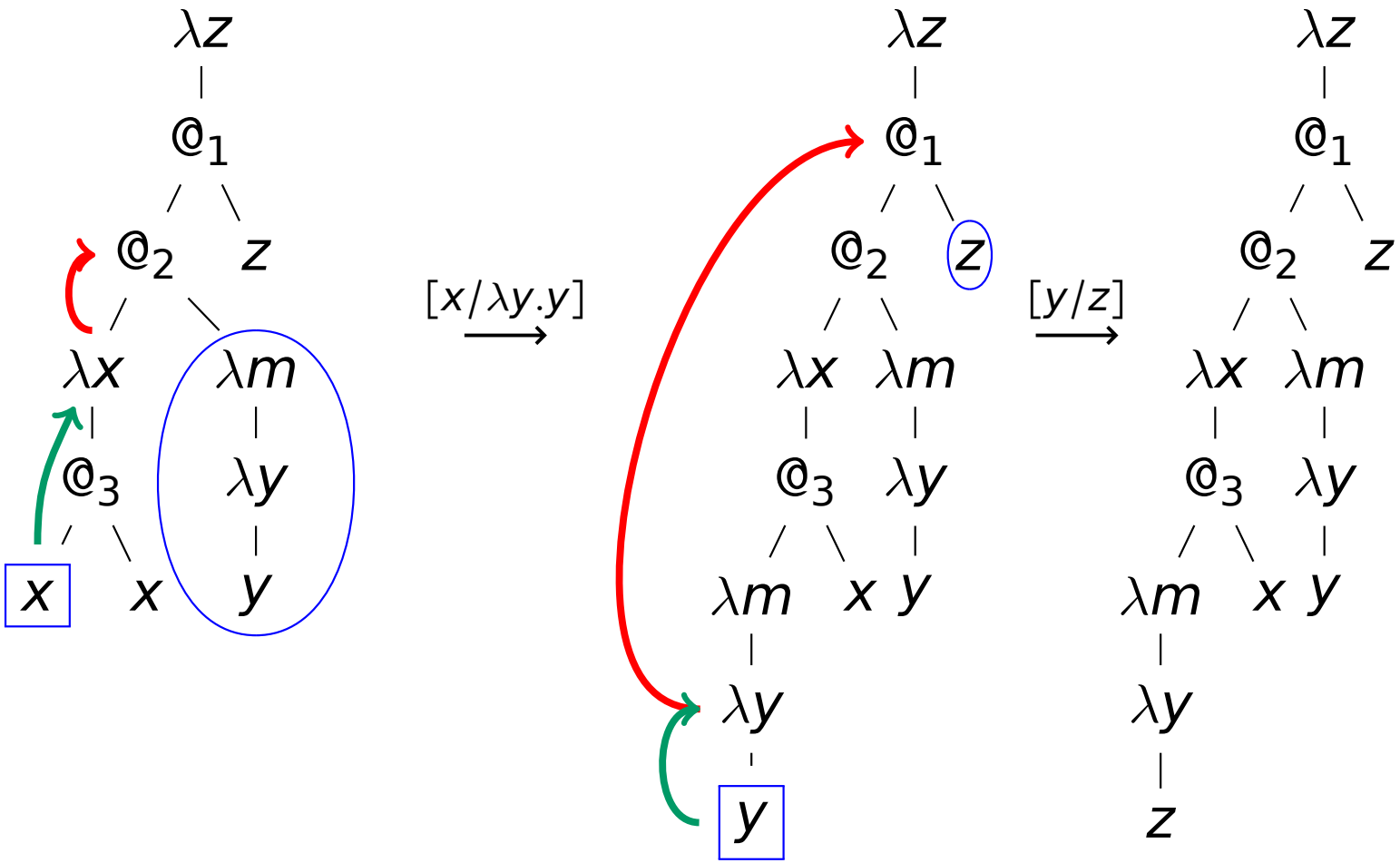
Головная Линейная Редукция



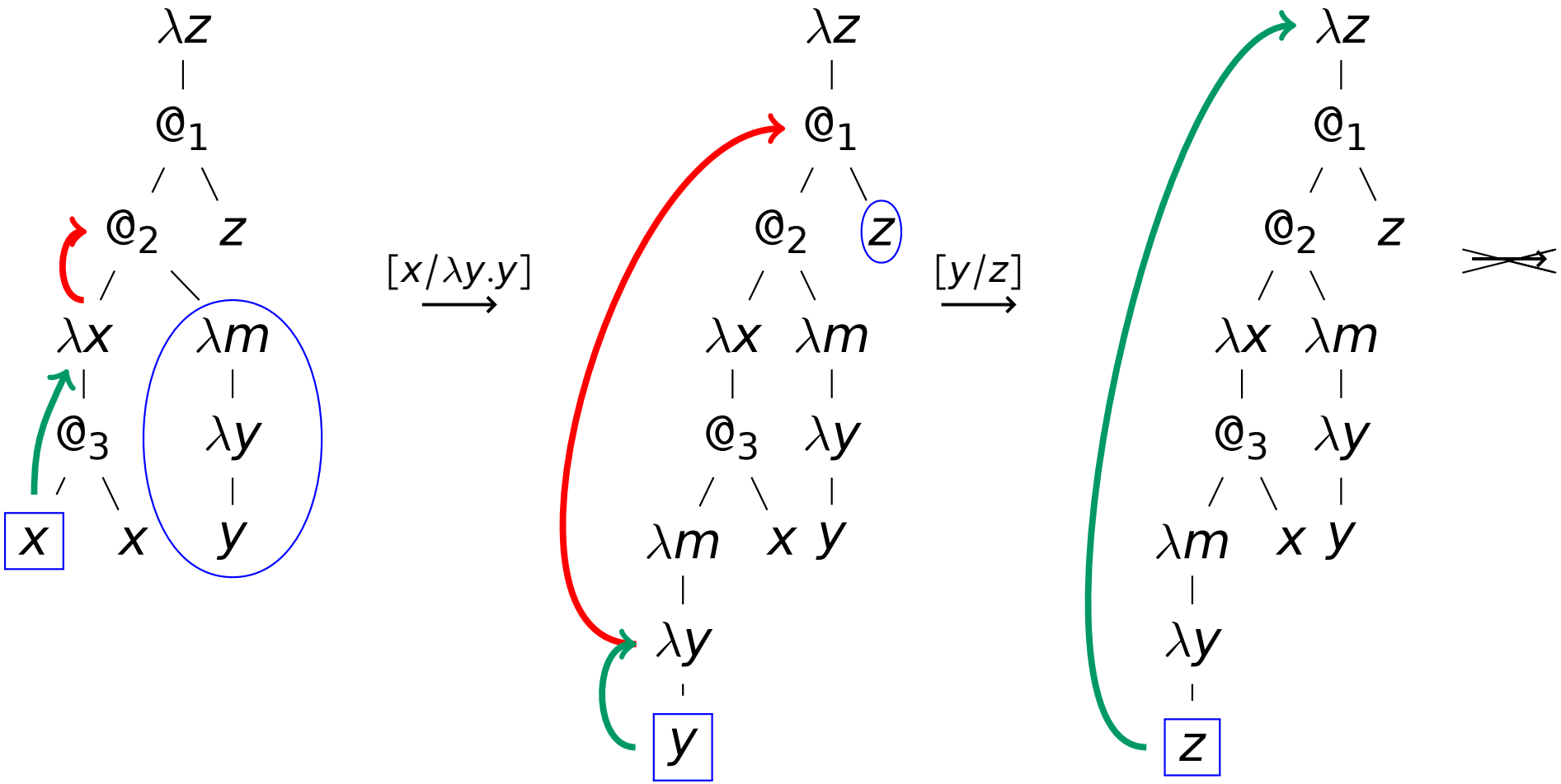
Головная Линейная Редукция



Головная Линейная Редукция



Головная Линейная Редукция



ONP

- Трассы вместо замыканий, окружений, . . .

ONP

- Трассы вместо замыканий, окружений, . . .

- Трасса (traversal)
  - Последовательность *токенов* вершин дерева терма *M*
    - *M* — программа
    - токен — программная точка
  - Каждый токен может быть снабжён *указателем* (backpointer, justifier)

ONP

- Трассы вместо замыканий, окружений, . . .
- Трасса (traversal)
  - Последовательность *токенов* вершин дерева терма M
    - M — программа
    - токен — программная точка
  - Каждый токен может быть снабжён *указателем* (backpointer, justifier)
- Простое типизированное лямбда исчисление
- Вход: термы в η-длинной форме

ONP

- Трассы вместо замыканий, окружений, ...
- Трасса (traversal)
  - Последовательность *токенов*
    вершин дерева терма *M*
    - *M* — программа
    - токен — программная точка
  - Каждый токен может быть снабжён *указателем* (backpointer, justifier)
- Простое типизированное лямбда исчисление
- Вход: термы в η-длинной форме
- Типы используются для
  - Построения η-длинной формы
  - Доказательства корректности алгоритма

Эта Работа

- UNP — расширение ONP до беспипового λ-исчисления

Эта Работа

- UNP — расширение ONP до беспипового λ-исчисления

- Применение частичных вычислений к UNP позволяет:
  - Получить *низкоуровневое* представление (LLL) λ-терма $[\![spec]\!](\text{UNP}, M) = \text{target code}_{LLL}$
  - Сгенерировать *компилятор* на основе UNP

$$[\![cogen]\!](\text{UNP}) \in \begin{array}{|ccc|} \hline \text{ULC} & \rightarrow & \text{LLL} \\ \hline & \text{L} & \\ \hline \end{array}$$

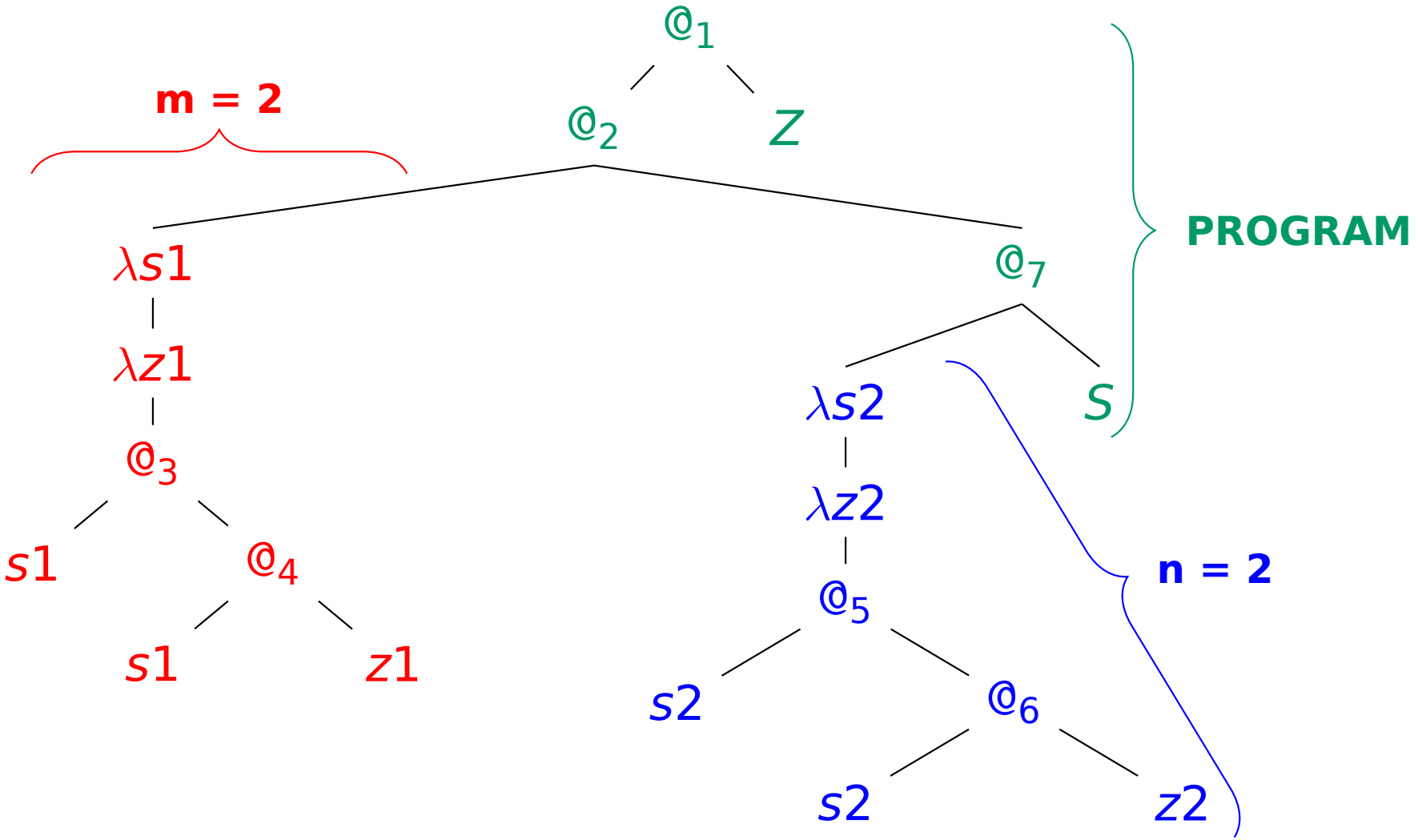
Пример

- Нумералы Чёрча: $\lambda s\ .\ \lambda z\ .\ s(\cdots(sz)\cdots)$
- Умножение нумералов

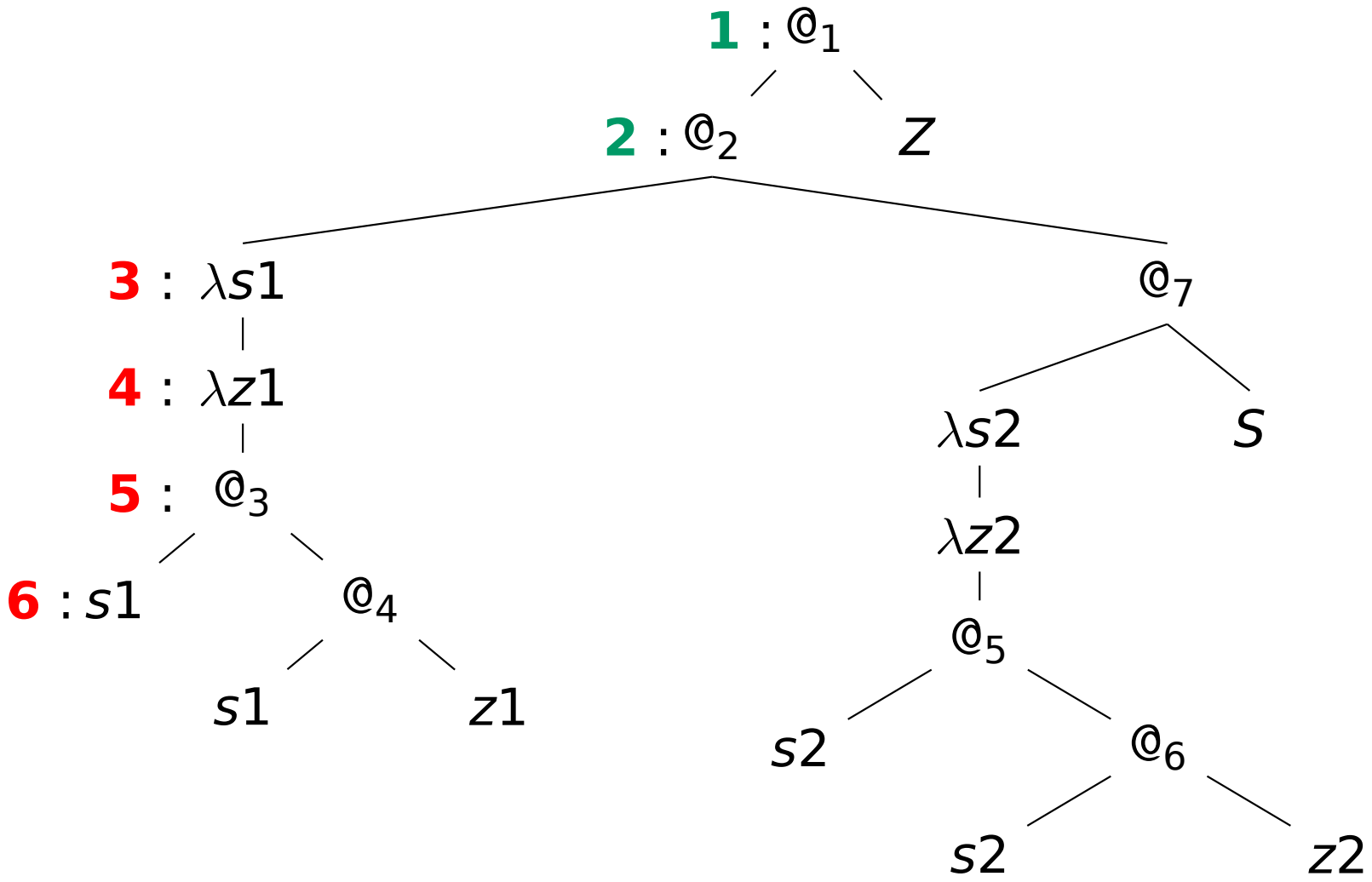
$$mul = \lambda mnsz\ .\ m(ns)z$$

- Нормальная форма $2 \times 2 : S@(S@(S@(S@Z)))$

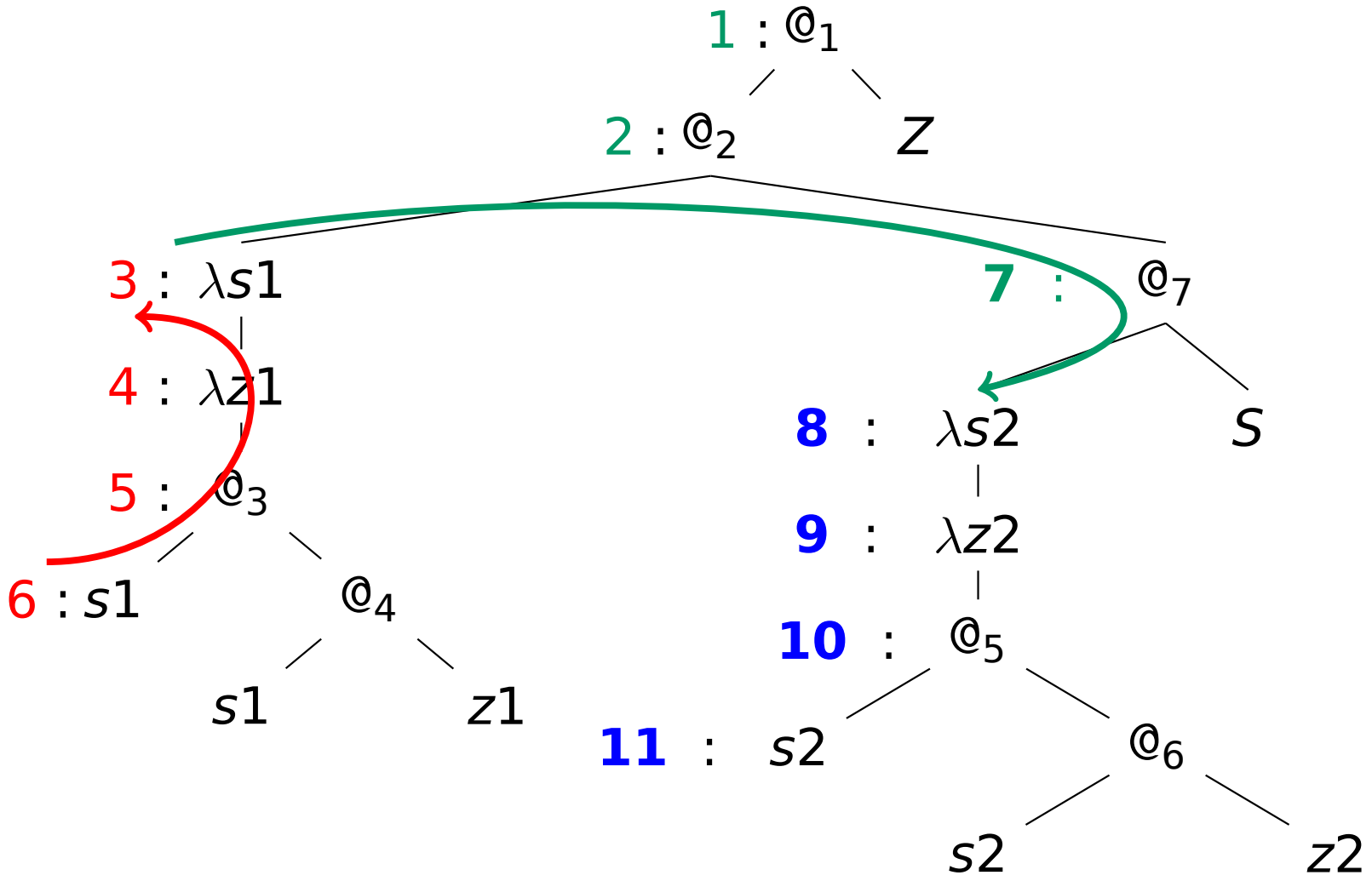
Пример



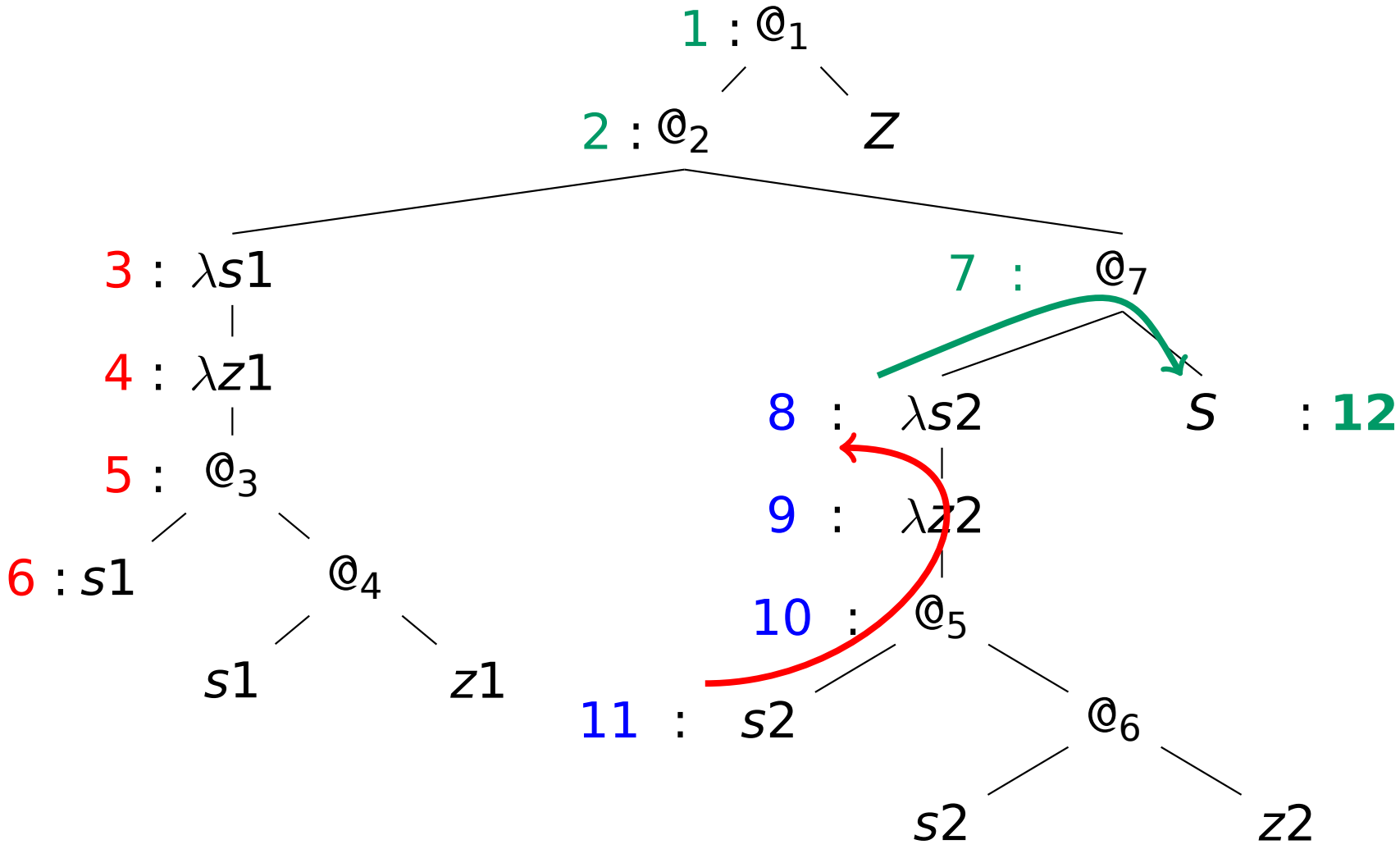
Пример



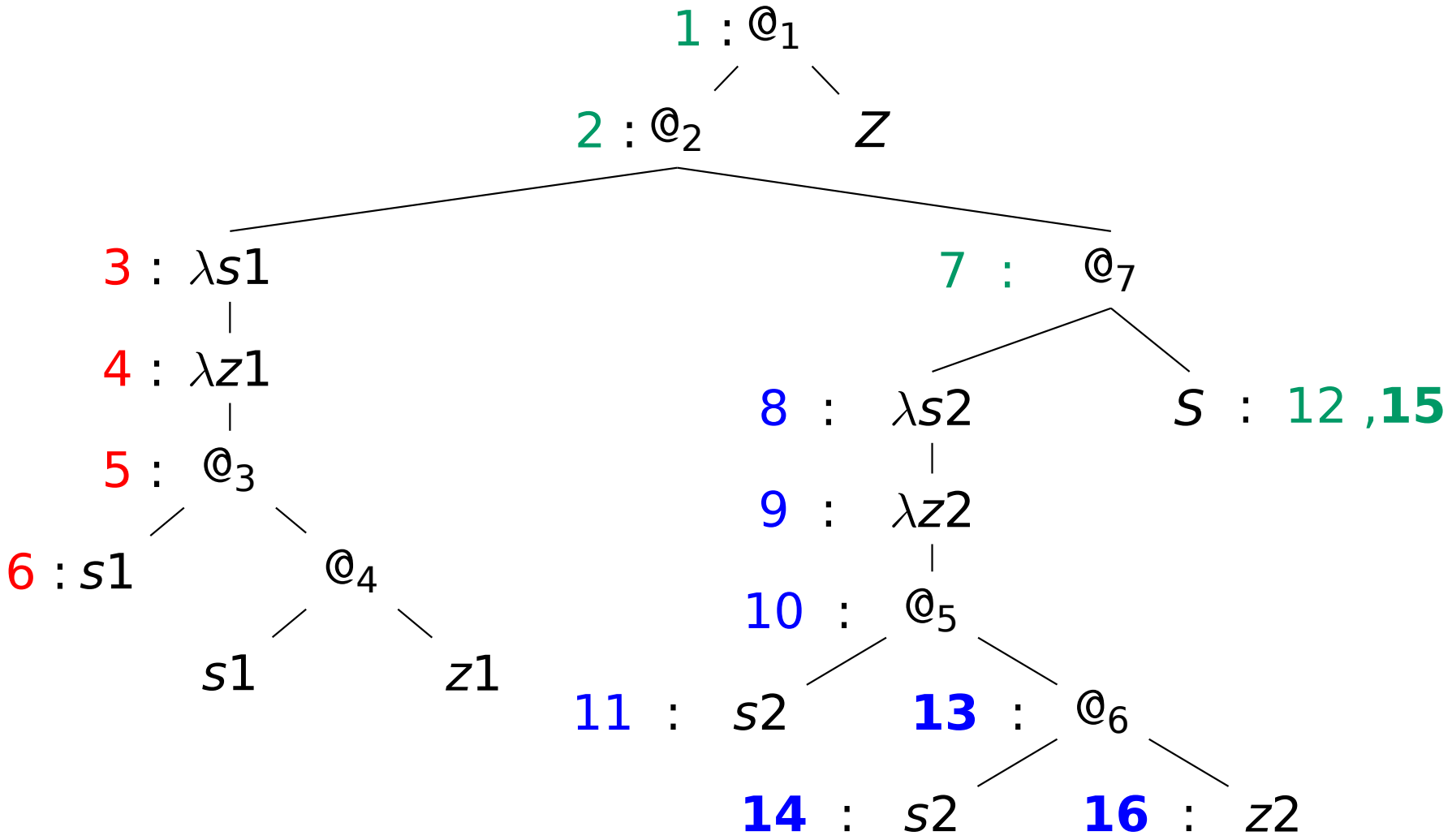
Пример



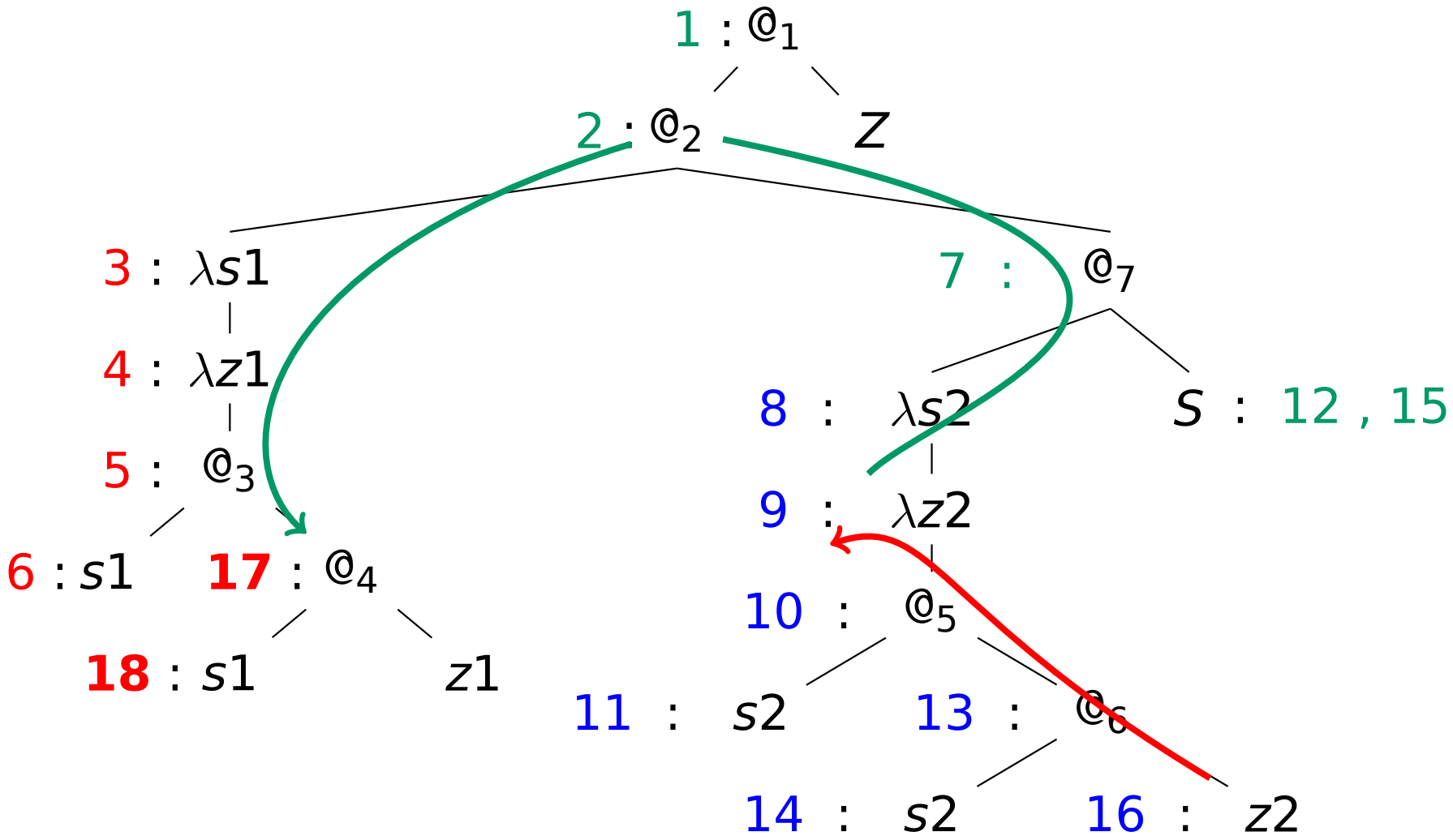
Пример



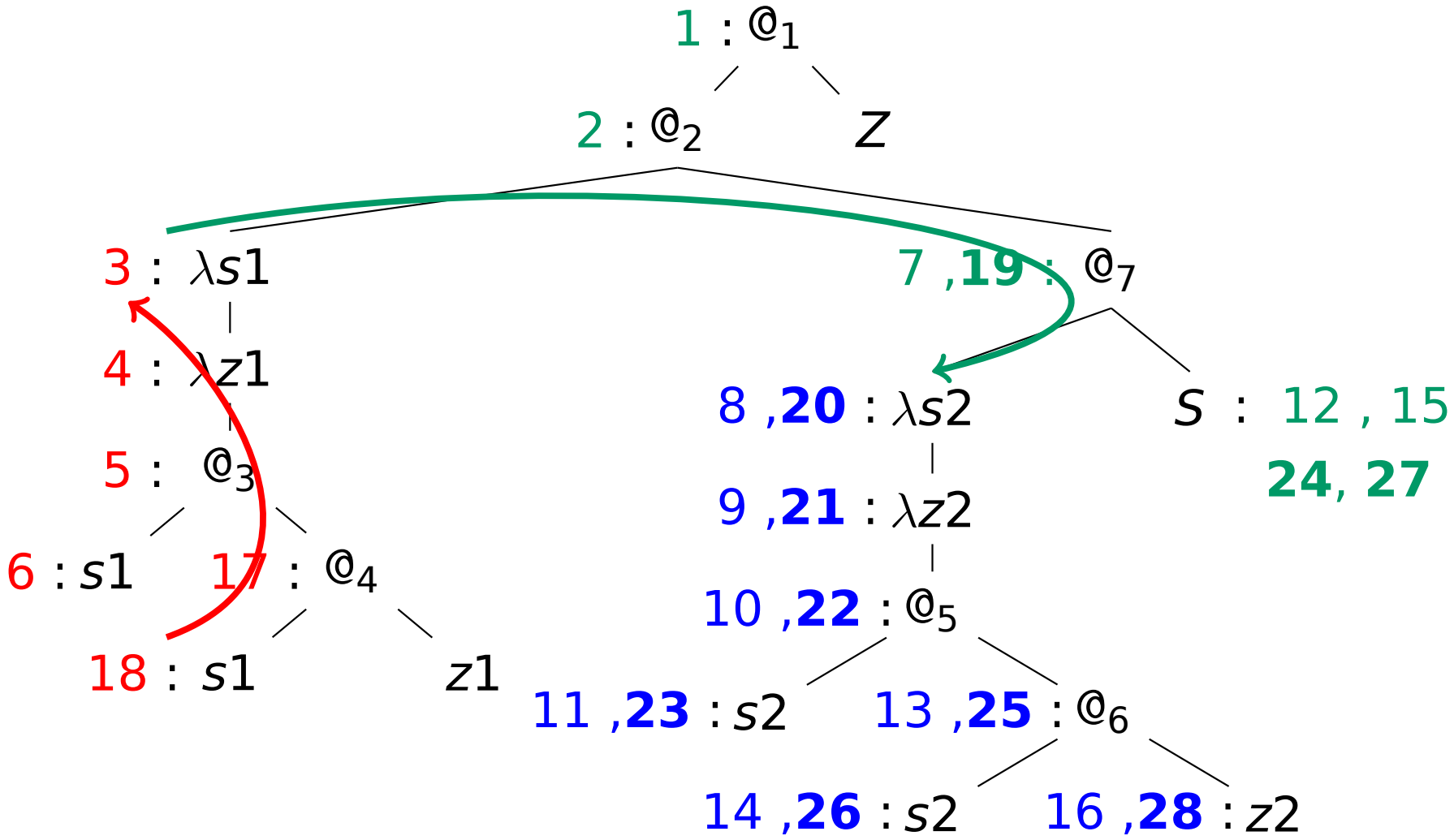
Пример


1 : @1
2 : @2
Z
3 : λs1
7 : @7
4 : λz1
8 : λs2
S : 12 ,15
5 : @3
9 : λz2
6 : s1
@4
10 : @5
s1
z1
11 : s2
13 : @6
14 : s2
16 : z2

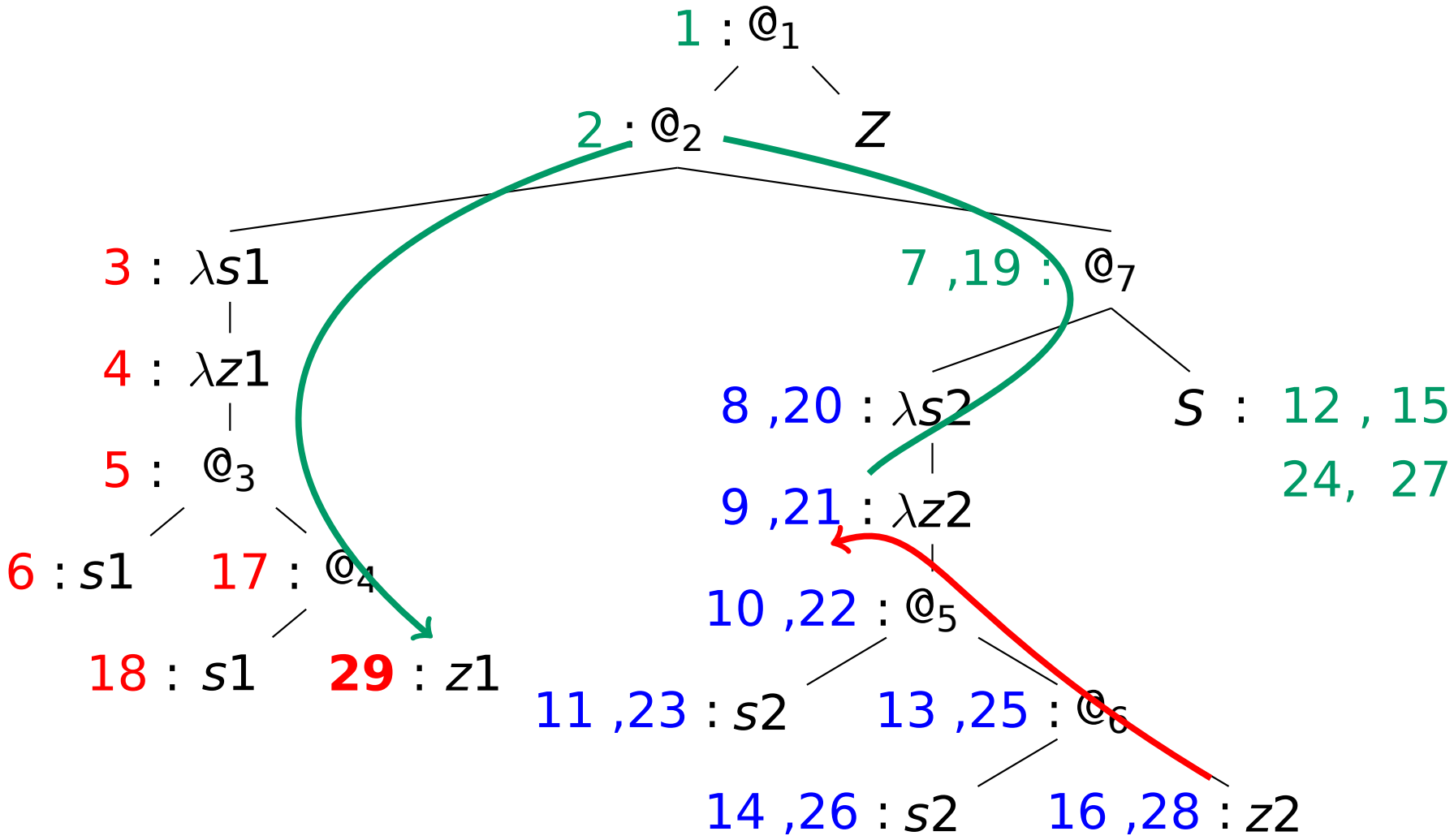
Пример


1 : @1
2 : @2
Z
3 : λs1
7 : @7
4 : λz1
8 : λs2
S : 12 , 15
5 : @3
9 : λz2
6 : s1
17 : @4
10 : @5
18 : s1
z1
11 : s2
13 : @6
14 : s2
16 : z2

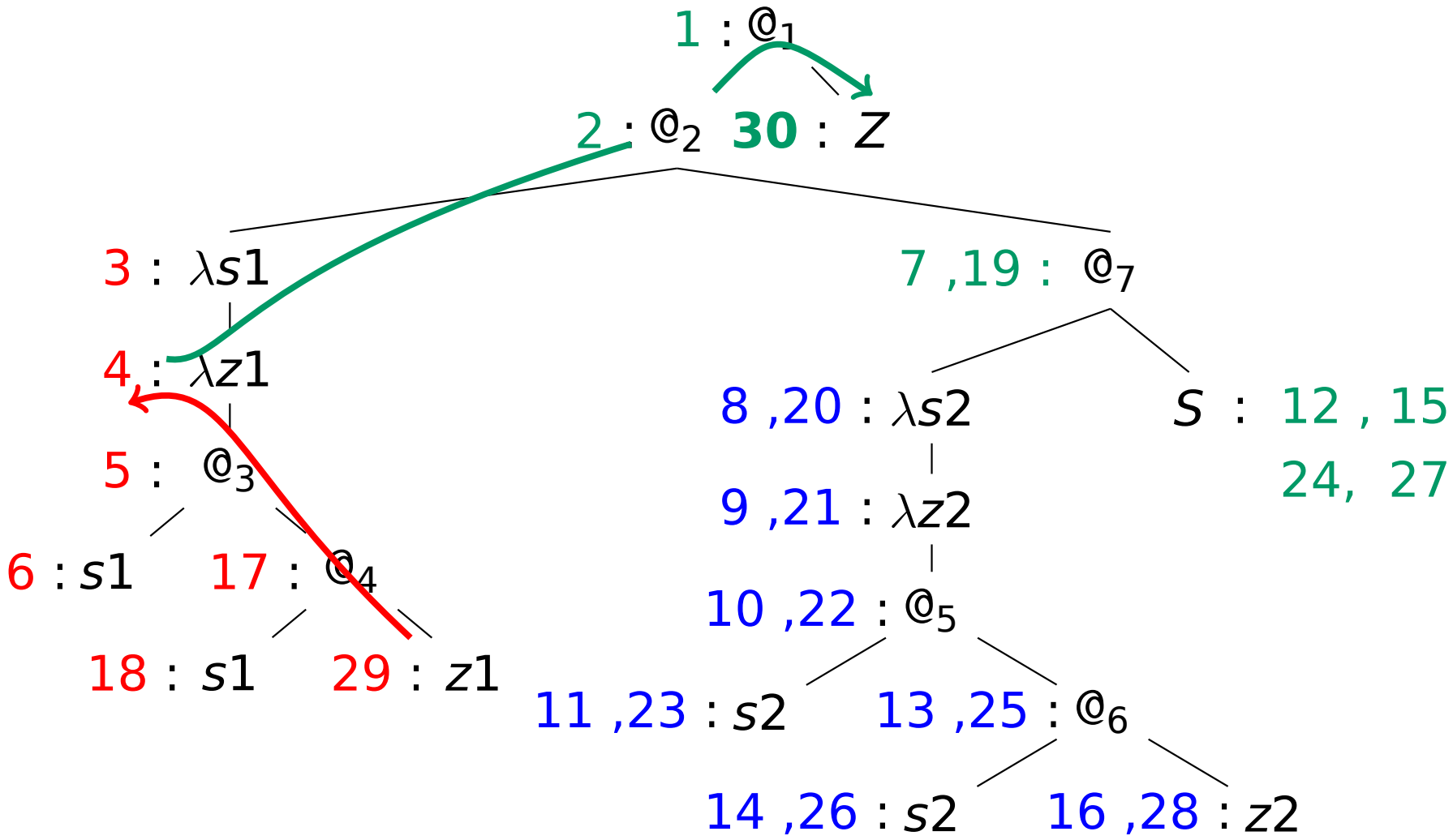
Пример


1 : @1
2 : @2
Z
3 : λs1
7 , 19 : @7
4 : λz1
8 , 20 : λs2
S : 12 , 15
24, 27
5 : @3
9 , 21 : λz2
6 : s1
17 : @4
10 , 22 : @5
18 : s1
z1
11 , 23 : s2
13 , 25 : @6
14 , 26 : s2
16 , 28 : z2

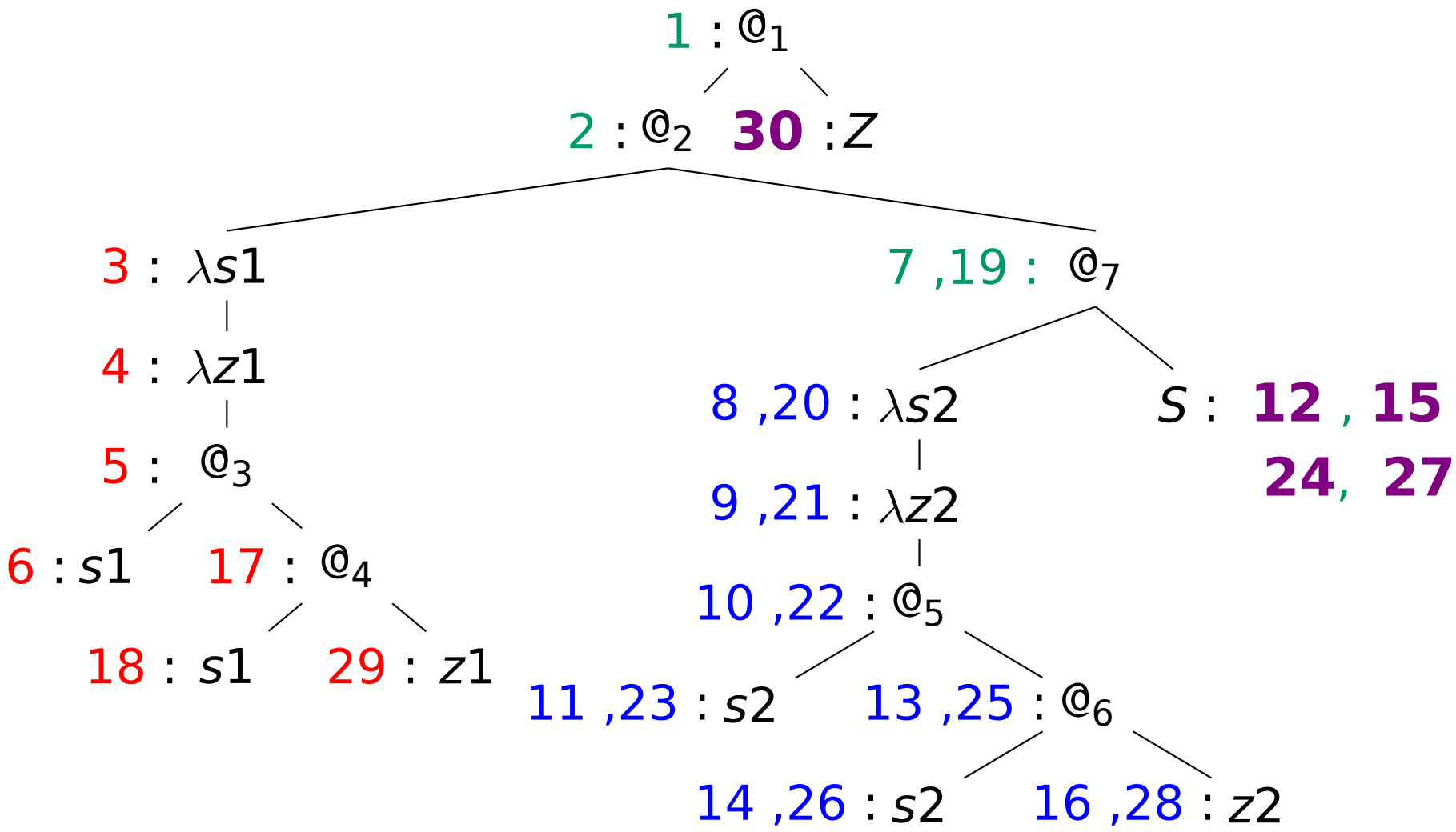
Пример


1 : @1
2 : @2
Z
3 : λs1
7 ,19 : @7
4 : λz1
8 ,20 : λs2
S : 12 , 15
24, 27
5 : @3
9 ,21 : λz2
6 : s1
17 : @4
10 ,22 : @5
18 : s1
29 : z1
11 ,23 : s2
13 ,25 : @6
14 ,26 : s2
16 ,28 : z2

Пример


1 : @1
2 : @2
30 : Z
3 : λs1
7 ,19 : @7
4 : λz1
8 ,20 : λs2
S : 12 , 15
24, 27
5 : @3
9 ,21 : λz2
6 : s1
17 : @4
10 ,22 : @5
18 : s1
29 : z1
11 ,23 : s2
13 ,25 : @6
14 ,26 : s2
16 ,28 : z2

Пример



Result: **S S S S Z**

Кононический Обход

Как выбрать правильный порядок обхода?

Кононический Обход

Как выбрать правильный порядок обхода?

Мы представили 5 семантик[1]

- Семантика 1 — классическая β-редукция
- Семантика 5 — трассирующая (à la Ong)

  Leftmost head linear reduction
- Инвариант: порядок обхода

[1] D.Berezun, N. D. Jones. PEPM 2017

Пошаговое Представление UNP

Семантика 1: классическая β**-редукция**
(substitution-based)

Пошаговое Представление UNP

Семантика 1: классическая β**-редукция**
(substitution-based)

Семантика 2: **окружения** (environment-based):

$$\rho \in \mathit{Env} = \mathit{Variable} \rightharpoonup \mathit{Exp} \times \mathit{Env}$$

Пошаговое Представление UNP

Семантика 1: классическая β**-редукция**
(substitution-based)

Семантика 2: **окружения** (environment-based):

$$\rho \in \mathit{Env} = \mathit{Variable} \rightharpoonup \mathit{Exp} \times \mathit{Env}$$

$$\begin{array}{lcll} [\![x]\!]\rho & = & \text{let } (e_0, \rho_0) = \rho(x) & \text{in } [\![e_0]\!]\rho_0 \\ [\![e_1 @ e_2]\!]\rho & = & \text{let } (\lambda x.e_0, \rho_0) = [\![e_1]\!]\rho_0 & \text{in } [\![e_0]\!]\rho_0[x \mapsto (e_2, \rho)] \end{array}$$

Пошаговое Представление UNP

Семантика 1: классическая β**-редукция** (substitution-based)

Семантика 2: **окружения** (environment-based):

$$\rho \in Env = Variable \rightharpoonup Exp \times Env$$

$$\begin{array}{lcll} [\![x]\!]\rho & = & \text{let } (e_0, \rho_0) = \rho(x) & \text{in } [\![e_0]\!]\rho_0 \\ [\![e_1 @ e_2]\!]\rho & = & \text{let } (\lambda x.e_0, \rho_0) = [\![e_1]\!]\rho_0 & \text{in } [\![e_0]\!]\rho_0[x \mapsto (e_2, \rho)] \end{array}$$

- Не композициональна, но
- Полу-композициональна:
  - при каждом вызове $[\![e]\!]\rho$, e — подвыражение M

важно для частичных вычислений

Пошаговое Представление UNP

Семантика 3: окружения и **продолжения** (continuations)

Пошаговое Представление UNP

Семантика 3: окружения и **продолжения** (continuations)

- Продолжения: линеаризация потока управления
- Дефункционализация: данные как замена вызова продолжений

Пошаговое Представление UNP

Семантика 3: окружения и **продолжения** (continuations)

- Продолжения: линеаризация потока управления
- Дефункционализация: данные как замена вызова продолжений

- Пример:
  $[\![e_1 @ e_2]\!]\ \rho = \text{ let } (\lambda x.e_0, \rho_0) = [\![e_1]\!]\rho \text{ in } [\![e_0]\!]\rho_0[x \mapsto (e_2, \rho)]$
  $\Rightarrow$
  $[\![e_1 @ e_2]\!]\ \rho\ \kappa = [\![e_1]\!]\rho\ \langle Kapp\ e_2\ \rho\ \kappa\rangle$
  $applycont\ \langle Kapp\ e_2\ \rho\ \kappa\rangle\ e_0\ \rho_0 = [\![e_0]\!]\ \rho_0[x \mapsto (e_2, \rho)]\ \kappa$

Пошаговое Представление UNP

Семантика 4: окружения и **история**

Пошаговое Представление UNP

Семантика 4: окружения и **история**

- Вместо продолжения κ — *история h*
- История — трасса с указателями

$$h \in H = (Exp \times Env \times H)^\star$$

Пошаговое Представление UNP

Семантика 5: **только** история (UNP)

- Вместо ρ — *указатель* в историю

Пошаговое Представление UNP

Семантика 5: **только** история (UNP)

- Вместо ρ — *указатель* в историю
- Эффект: все аргументы нормализатора являются данными первого порядка (first-order data)

Текущая работа

- Доказательство корректности UNP

Текущая работа

- Доказательство корректности UNP
  - Формальное описание в виде систем переходов
    - HLR
    - CHLR
    - BUNP $\subset$ UNP
    - UNP

Текущая работа

- Доказательство корректности UNP
  - Формальное описание в виде систем переходов
    - HLR
    - CHLR
    - BUNP ⊂ UNP
    - UNP
  - HLR завершается совместно с HR

Текущая работа

- Доказательство корректности UNP
  - Формальное описание в виде систем переходов
    - HLR
    - CHLR
    - BUNP ⊂ UNP
    - UNP
  - HLR завершается совместно с HR
  - CHLR нормализует терм iff терм нормализуем

Текущая работа

- Доказательство корректности UNP
  - Формальное описание в виде систем переходов
    - HLR
    - CHLR
    - BUNP $\subset$ UNP
    - UNP
  - HLR завершается совместно с HR
  - CHLR нормализует терм iff терм нормализуем
  - Бисимуляция
    - HLR ~ BUNP
    - CHLR ~ UNP

Что дальше?

- “Оптимизировать” LLL

Что дальше?

- “Оптимизировать” LLL
- Определить *HO*-игры над LLL

Что дальше?

- “Оптимизировать” LLL
- Определить *HO*-игры над LLL
- Доказательство корректности через игровую семантику

Что дальше?

- “Оптимизировать” LLL
- Определить *HO*-игры над LLL
- Доказательство корректности через игровую семантику
- Сложность (complexity), анализ потока данных
- η-расширение “на лету” (on-the-fly)