Поиск множества паттернов в тексте

Допустим, у нас есть **k** паттернов, максимальная длина - M. Ищем в тексте длины N

За сколько можно это сделать уже известными алгоритмами?

Поиск множества паттернов в тексте

Допустим, у нас есть **k** паттернов, максимальная длина - M. Ищем в тексте длины N

За сколько можно это сделать уже известными алгоритмами?

КМП работает за O(N + M). Ищем паттерны по-очереди, получается суммарное время

$$k \cdot O(N + M) = O(k \cdot N + k \cdot M)$$

Можем ли искать быстрее?

Построим бор

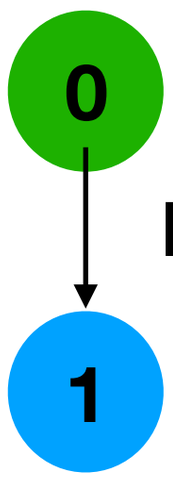
Пусть у нас есть паттерны HE, SHE, HIS, HER

Добавляем 0-ю вершину - корень бора

Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

Добавим HE



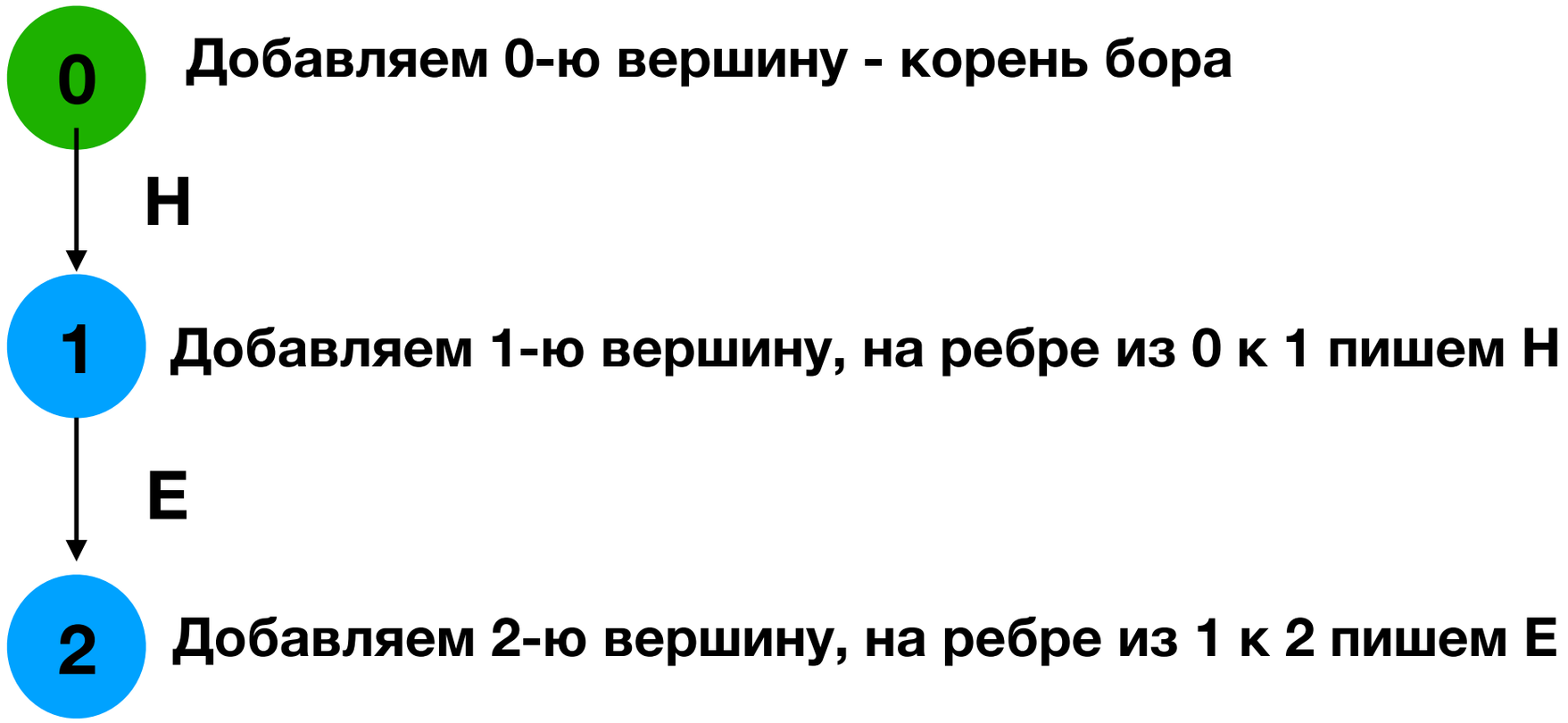
Добавляем 0-ю вершину - корень бора

Добавляем 1-ю вершину, на ребре из 0 к 1 пишем H

Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

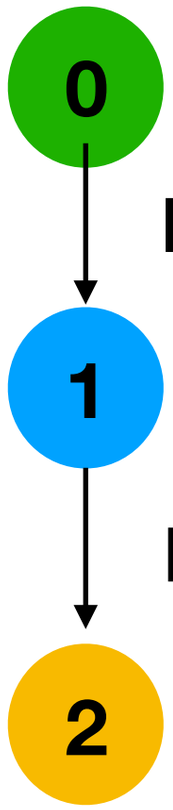
Добавим HE



Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

Добавим HE



Добавляем 0-ю вершину - корень бора

Добавляем 1-ю вершину, на ребре из 0 к 1 пишем H

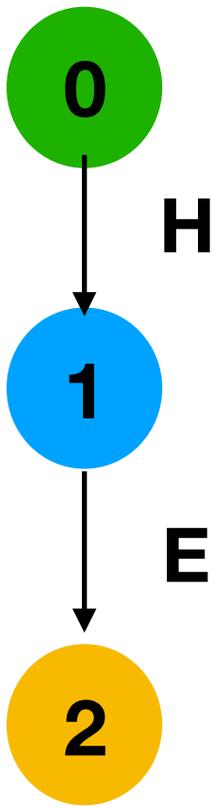
Добавляем 2-ю вершину, на ребре из 1 к 2 пишем E

Так как мы достигли конца паттерна,
помечаем вершину ккак конечную

Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

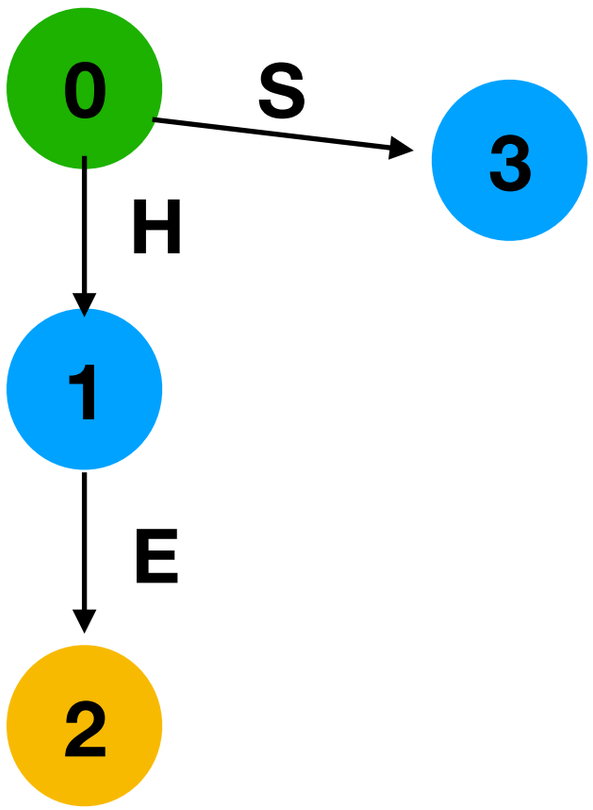
Аналогично добавляем SHE, начиная с 0-й вершины



Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

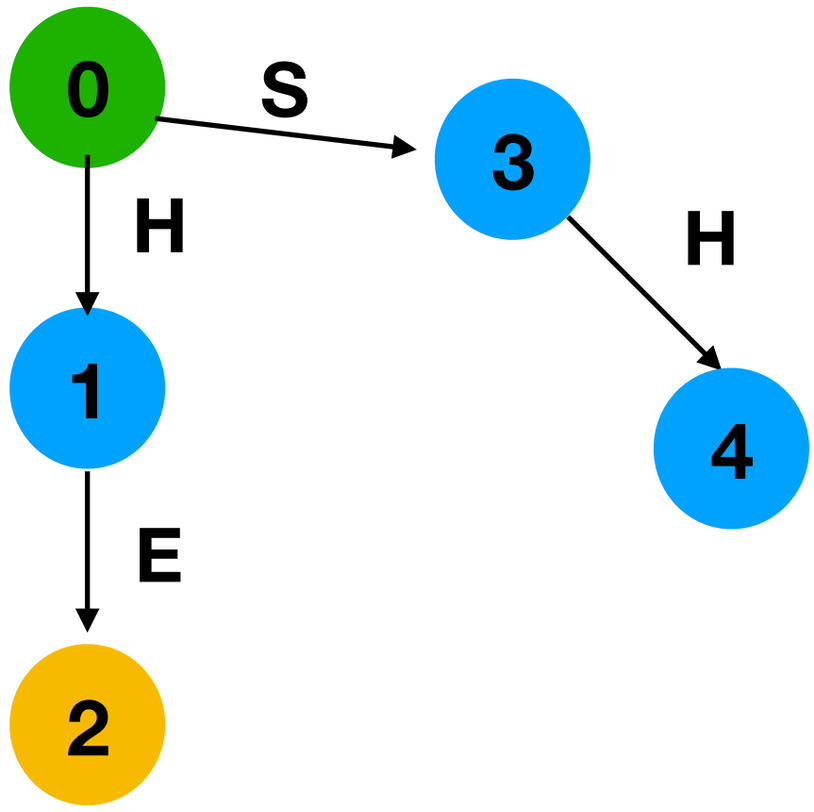
Аналогично добавляем SHE, начиная с 0-й вершины



Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

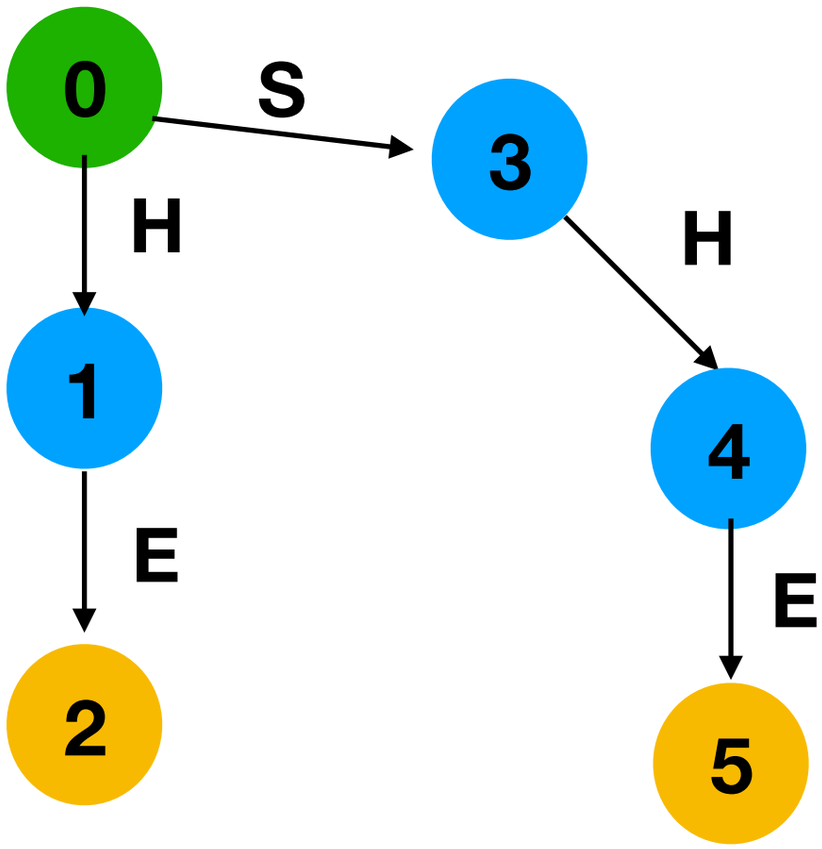
Аналогично добавляем SHE, начиная с 0-й вершины



Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

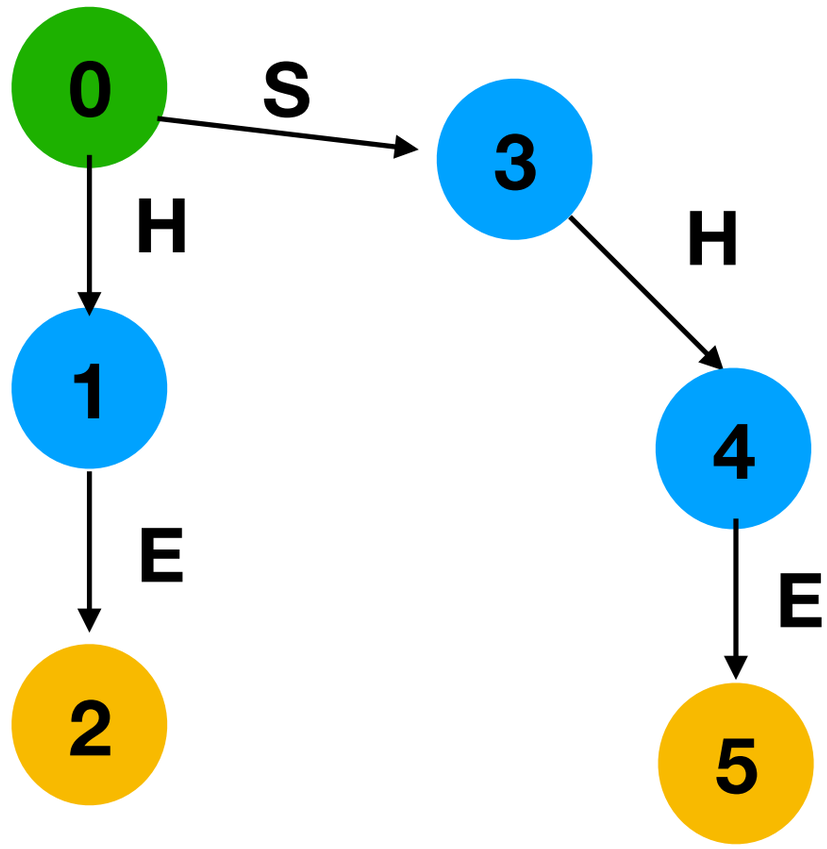
Аналогично добавляем SHE, начиная с 0-й вершины



Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

Начинаем добавлять HIS

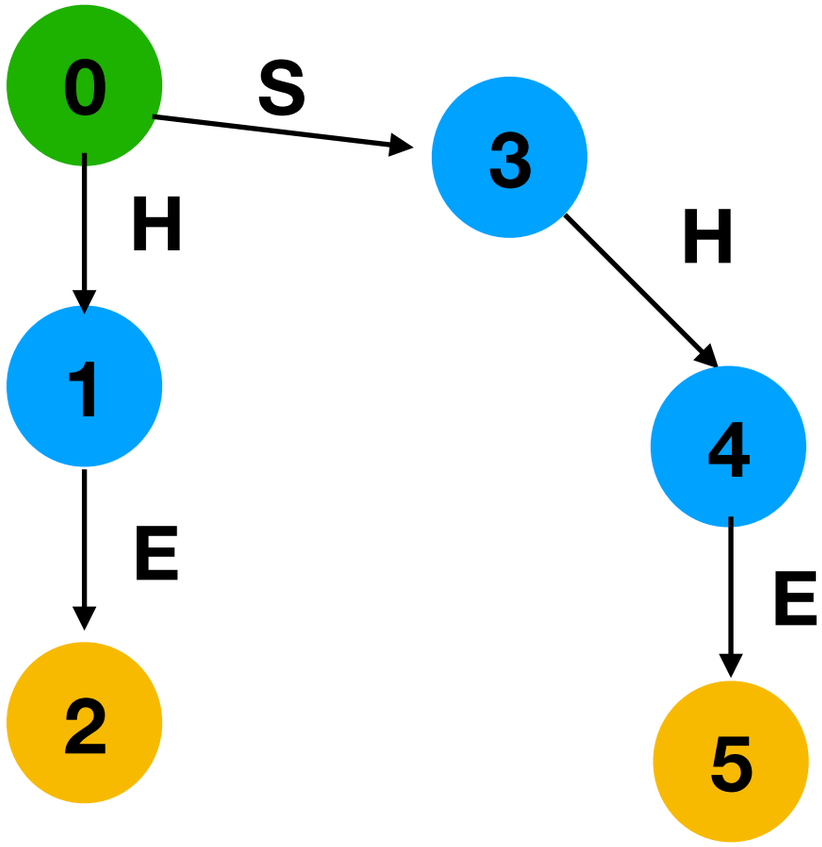


Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

Начинаем добавлять HIS

H уже есть, потому просто переходим в вершину 1



Построим бор

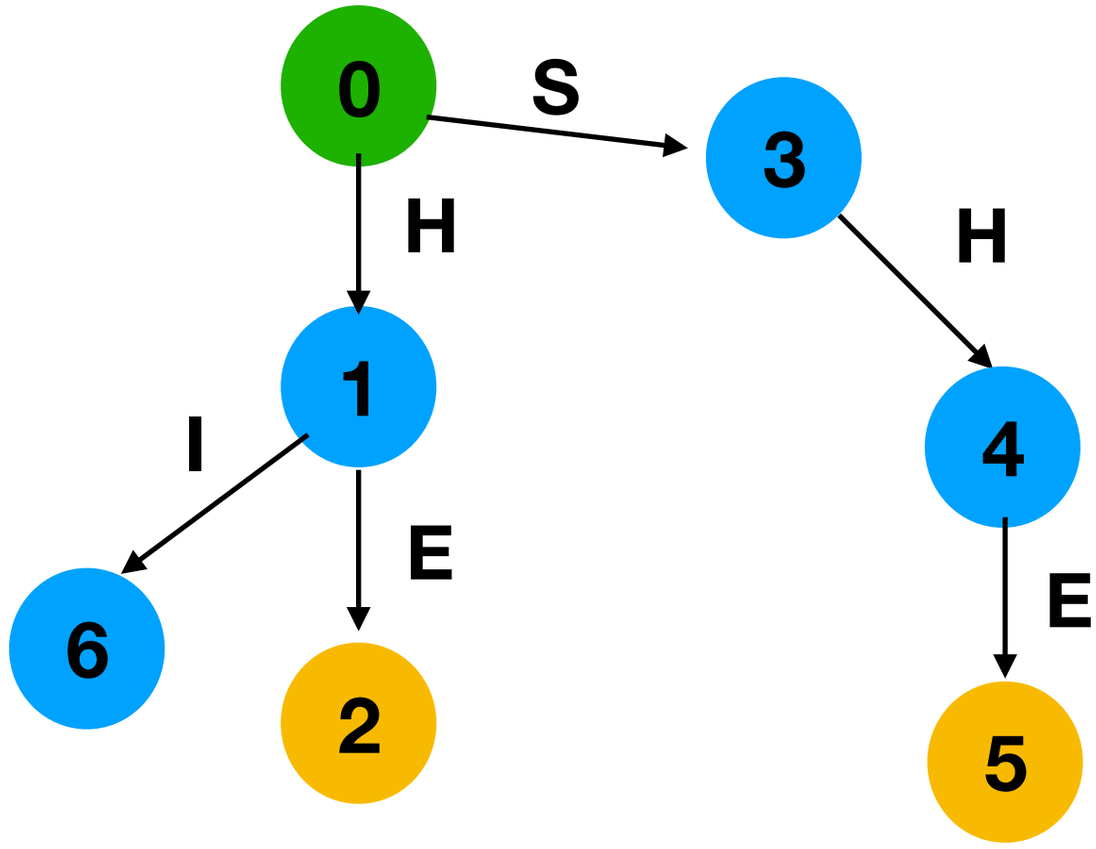
Пусть у нас есть паттерны HE, SHE, HIS, IS, I

Начинаем добавлять HIS

H уже есть, потому просто переходим в вершину 1

Далее добавляем вершину 6 и соединяем ее ребром с 1-й

Далее аналогично



Построим бор

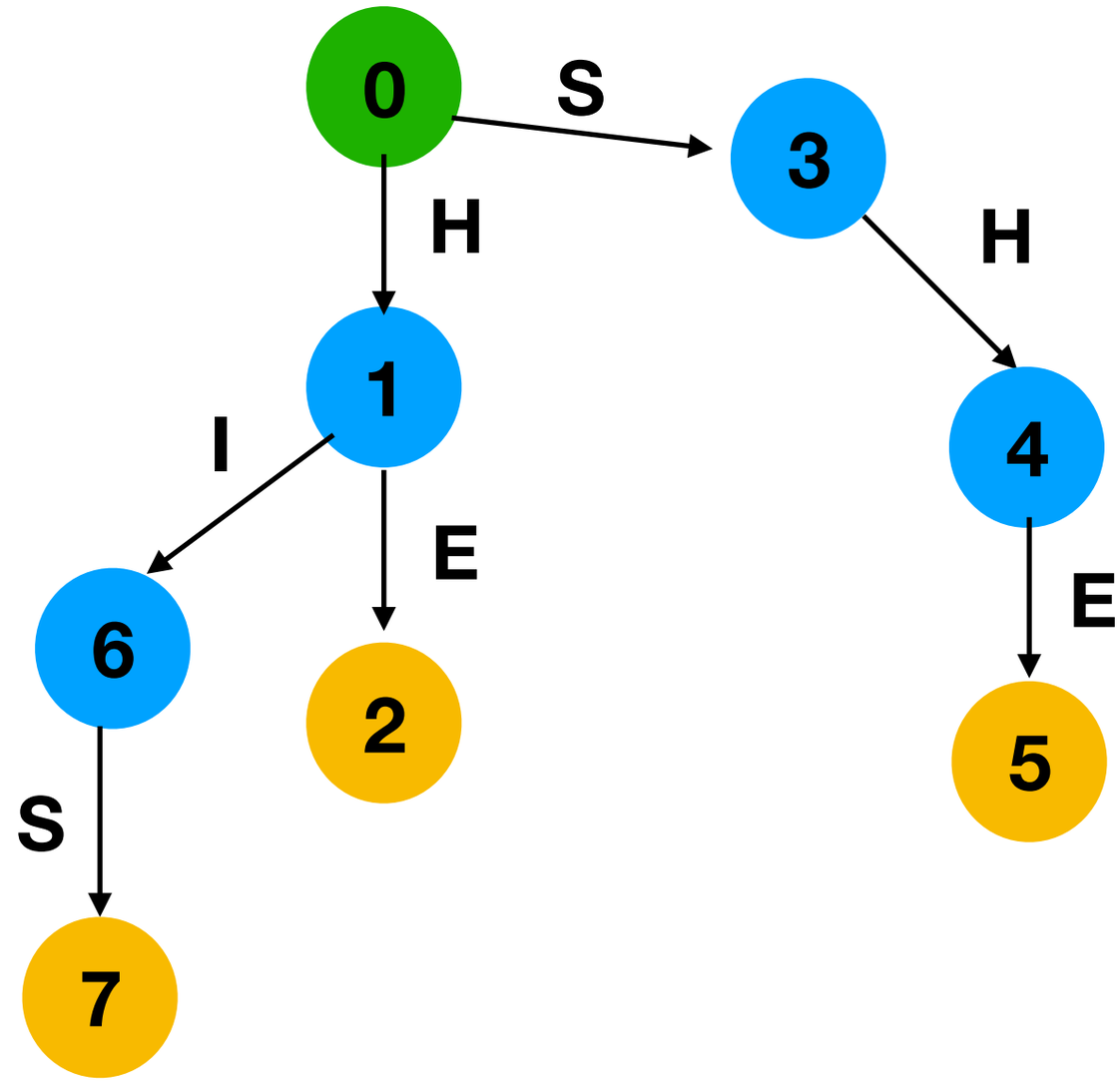
Пусть у нас есть паттерны HE, SHE, HIS, IS, I

Начинаем добавлять HIS

H уже есть, потому просто переходим в вершину 1

Далее добавляем вершину 6 и соединяем ее ребром с 1-й

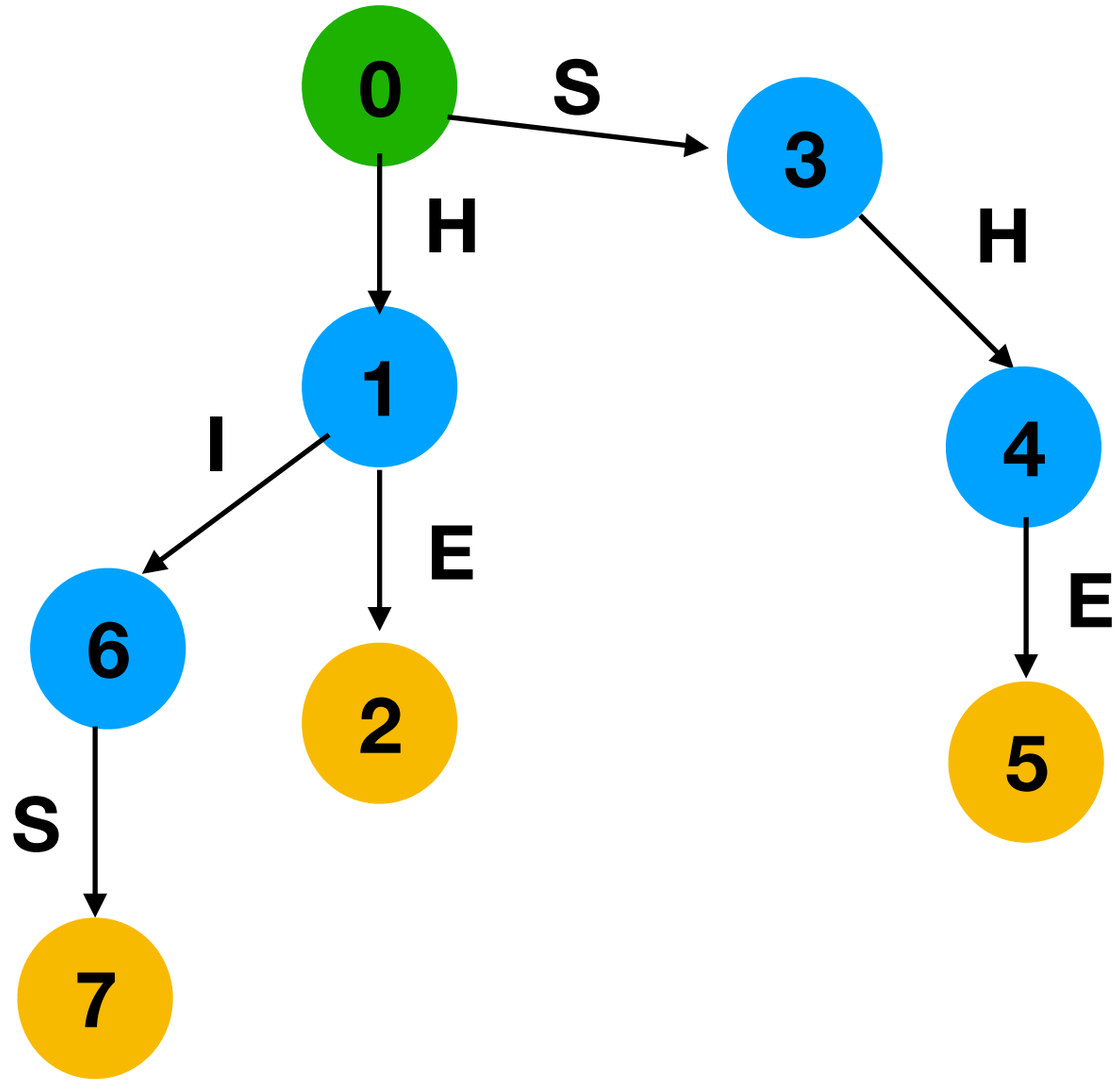
Далее аналогично



Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, HER

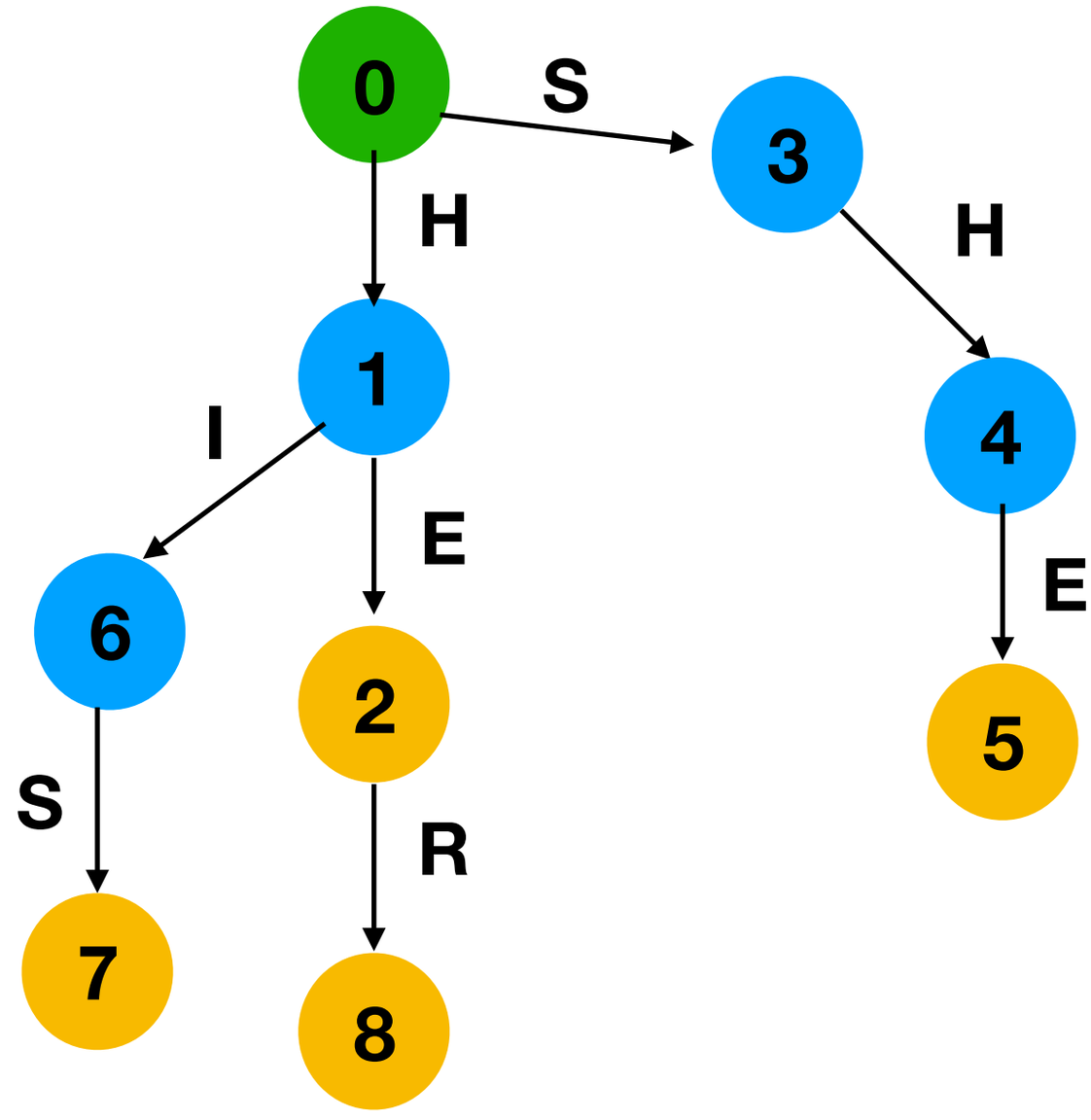
Добавляем HER
Идем по ребру H из 0 в 1.
Далее идем из 1 в 2 по ребру E



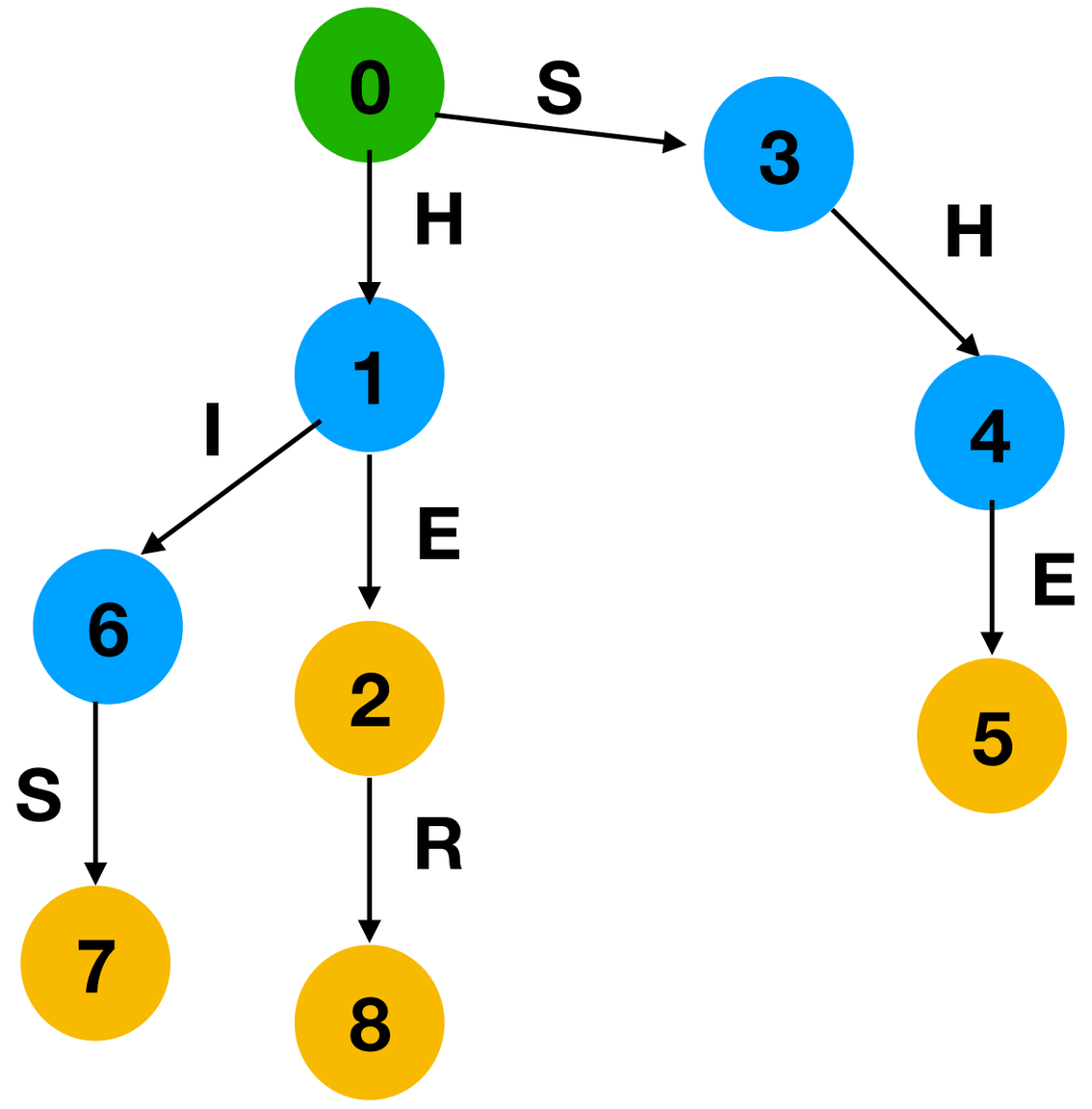
Построим бор

Пусть у нас есть паттерны HE, SHE, HIS, IS, I

Добавляем R



За сколько строим бор?

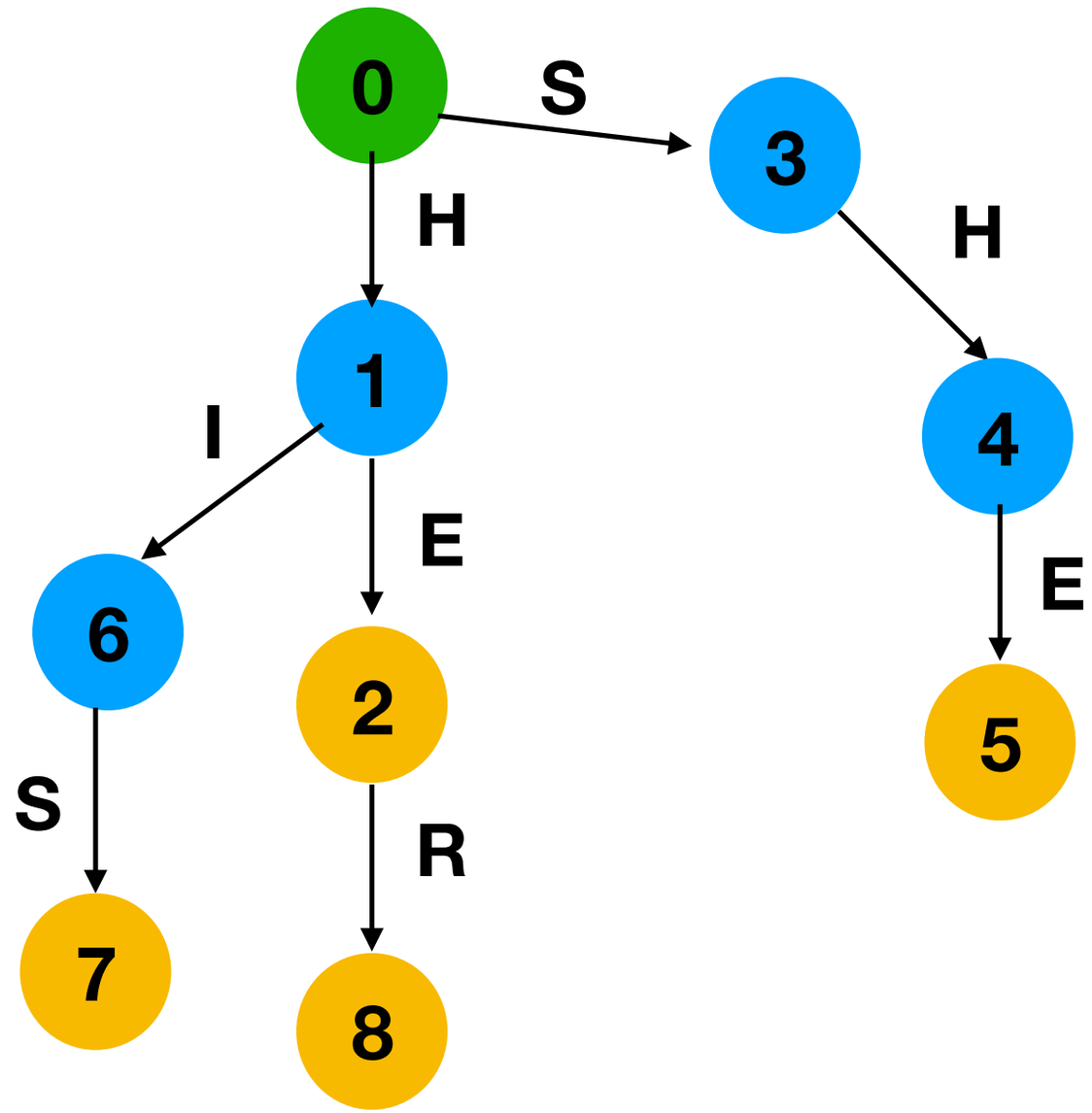


Добавление каждого паттерна - за линейной время (считаем, что находим и/или добавляем переход за O(1) (например, используя хэш-таблицы)), потому

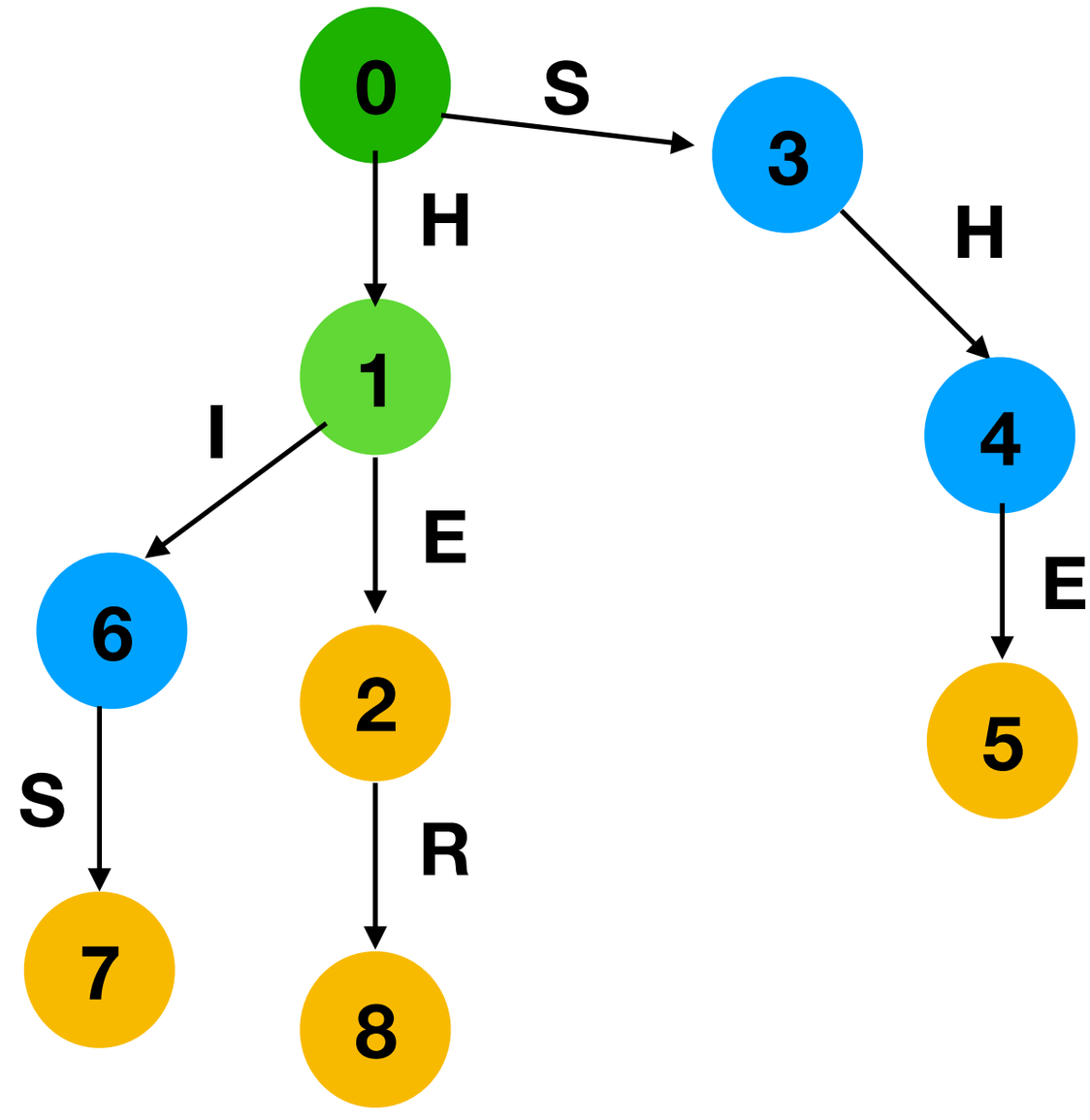
$$k \cdot O(M) = O(k \cdot M)$$

Попробуем поискать в строке

HERHEHIS



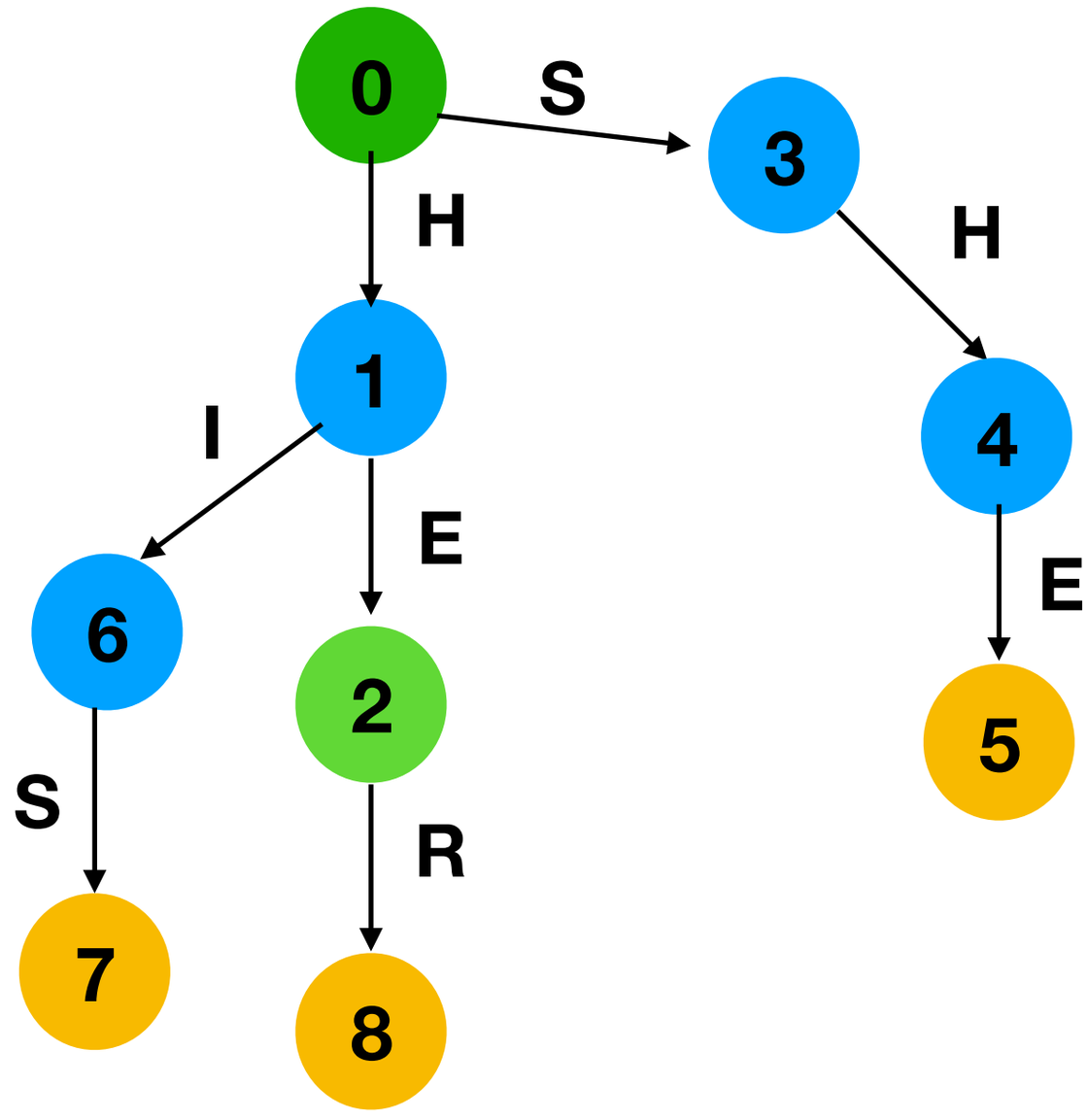
Попробуем поискать в строке



HERHEHIS

Идем по ребру с прочтенным символом, если оно есть

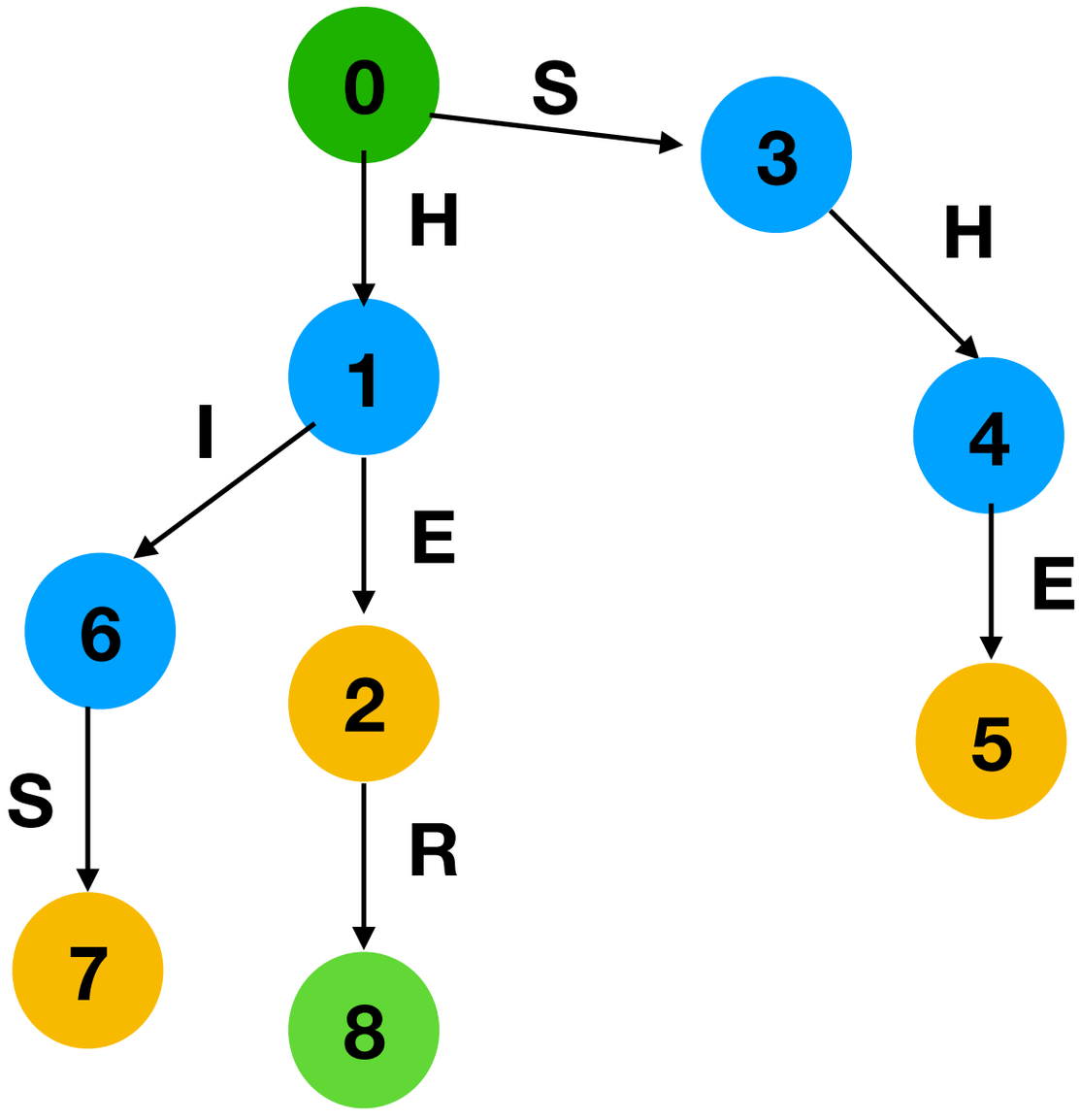
Попробуем поискать в строке



HERHEHIS

Если пришли в отмеченное состояние - то сообщаем о находке. Нашли HE

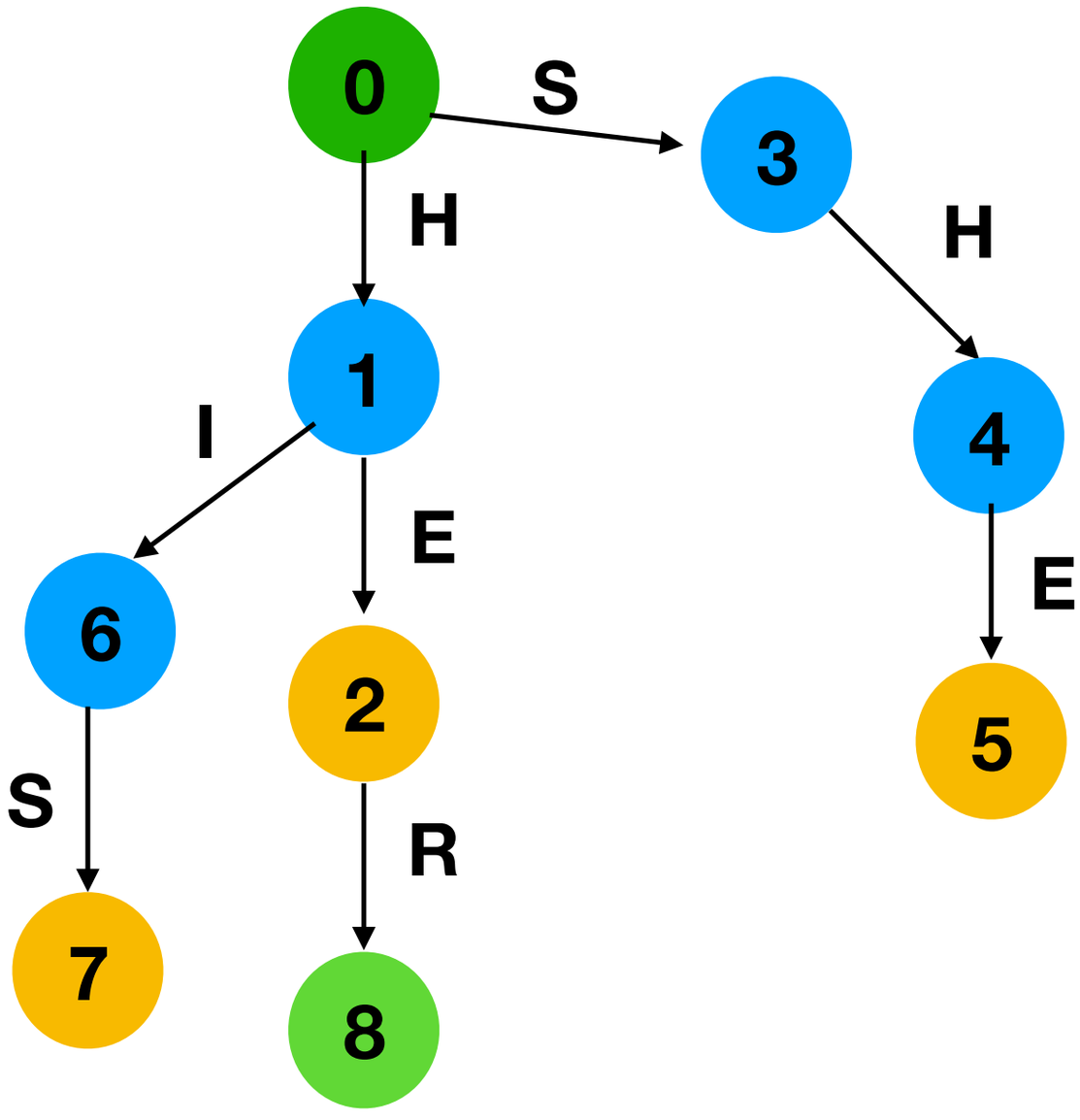
Попробуем поискать в строке



HERHEHIS

Нашли HER

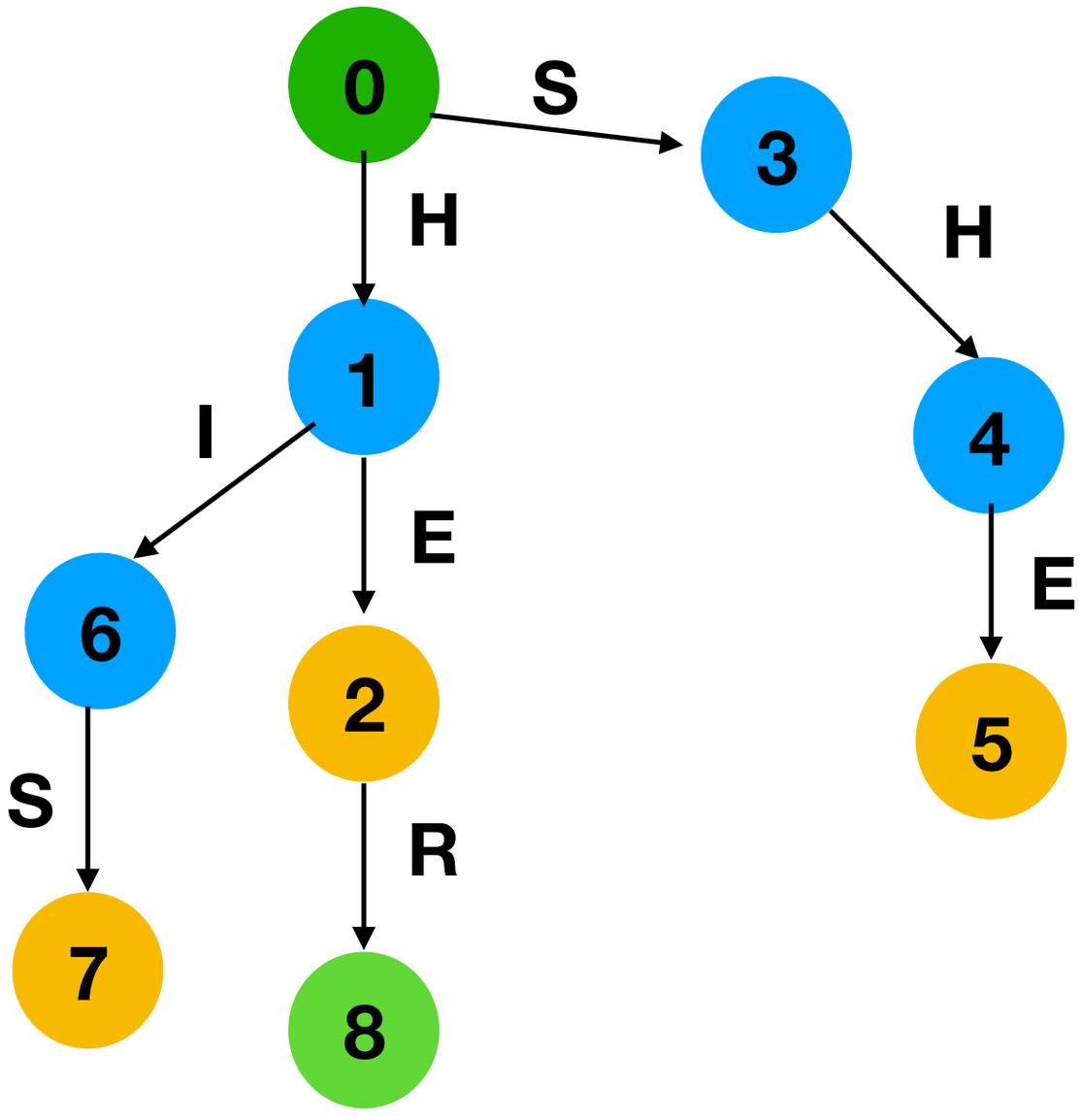
Попробуем поискать в строке



HERHEHIS

Если пришли в конец ветви - переходим в корень

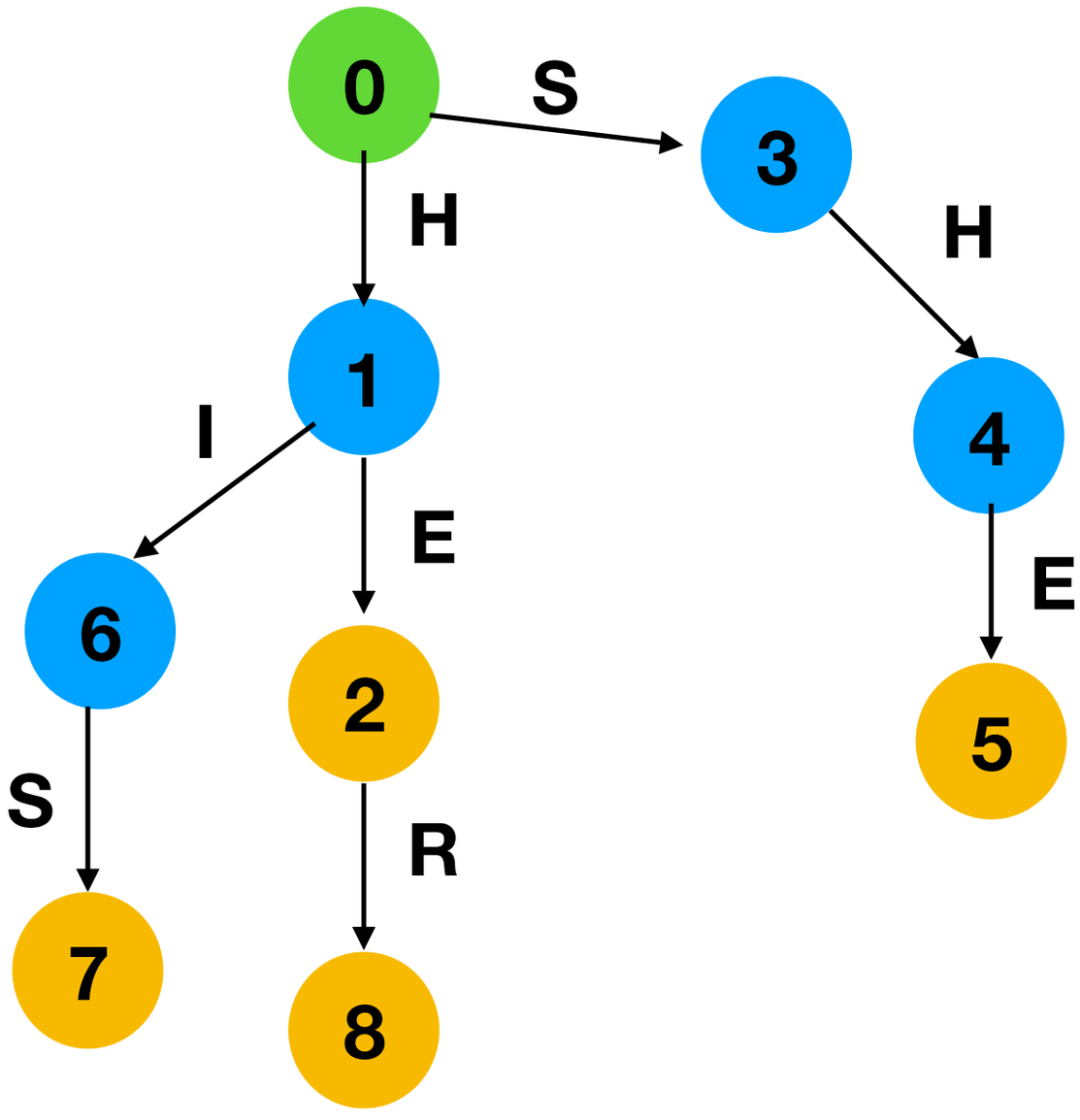
Попробуем поискать в строке



HERHEHIS

Если не можем прочитать символ из текущей вершины - идем в корень

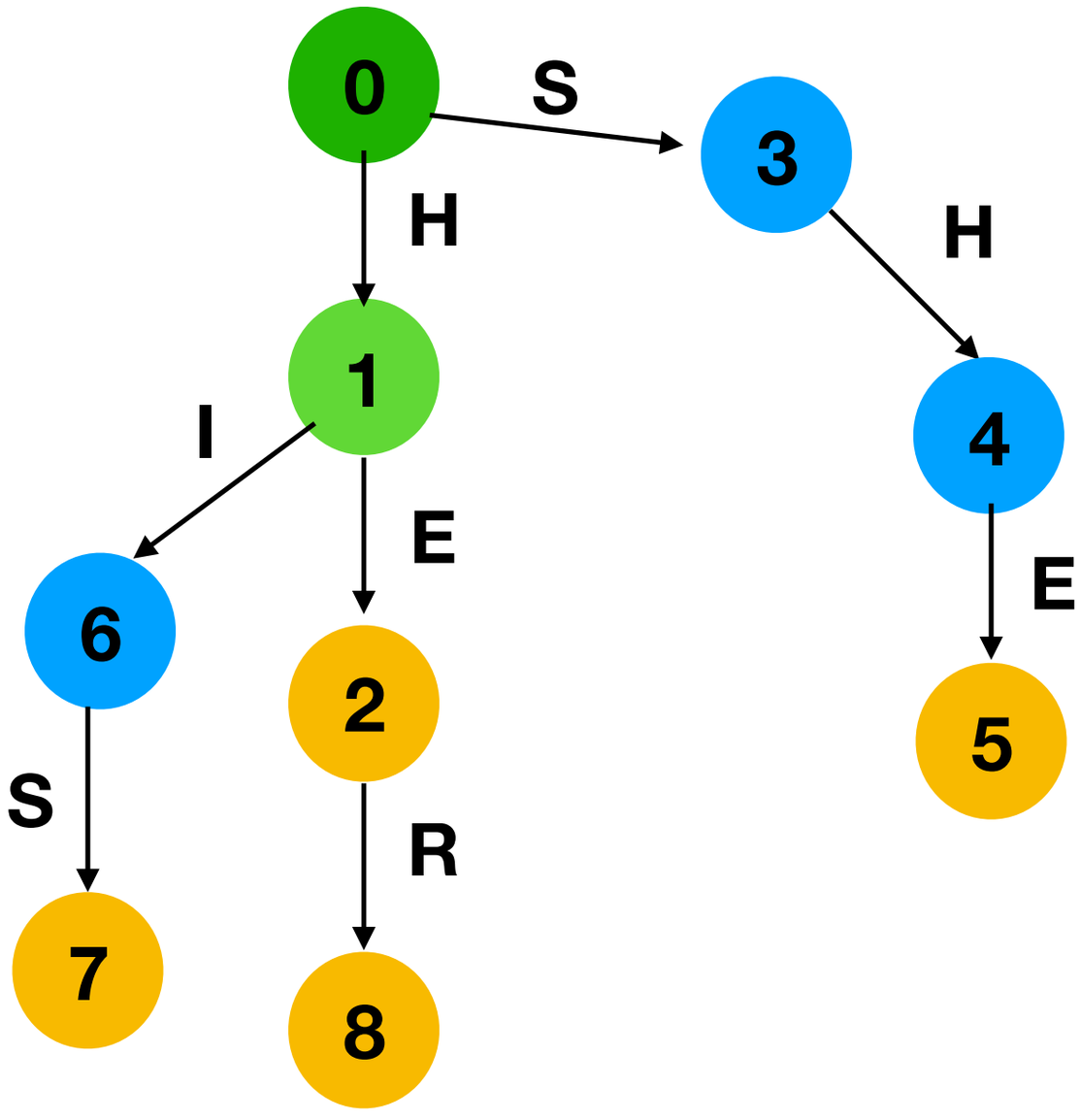
Попробуем поискать в строке



HERHEHIS

Если не можем прочитать символ из текущей вершины - идем в корень

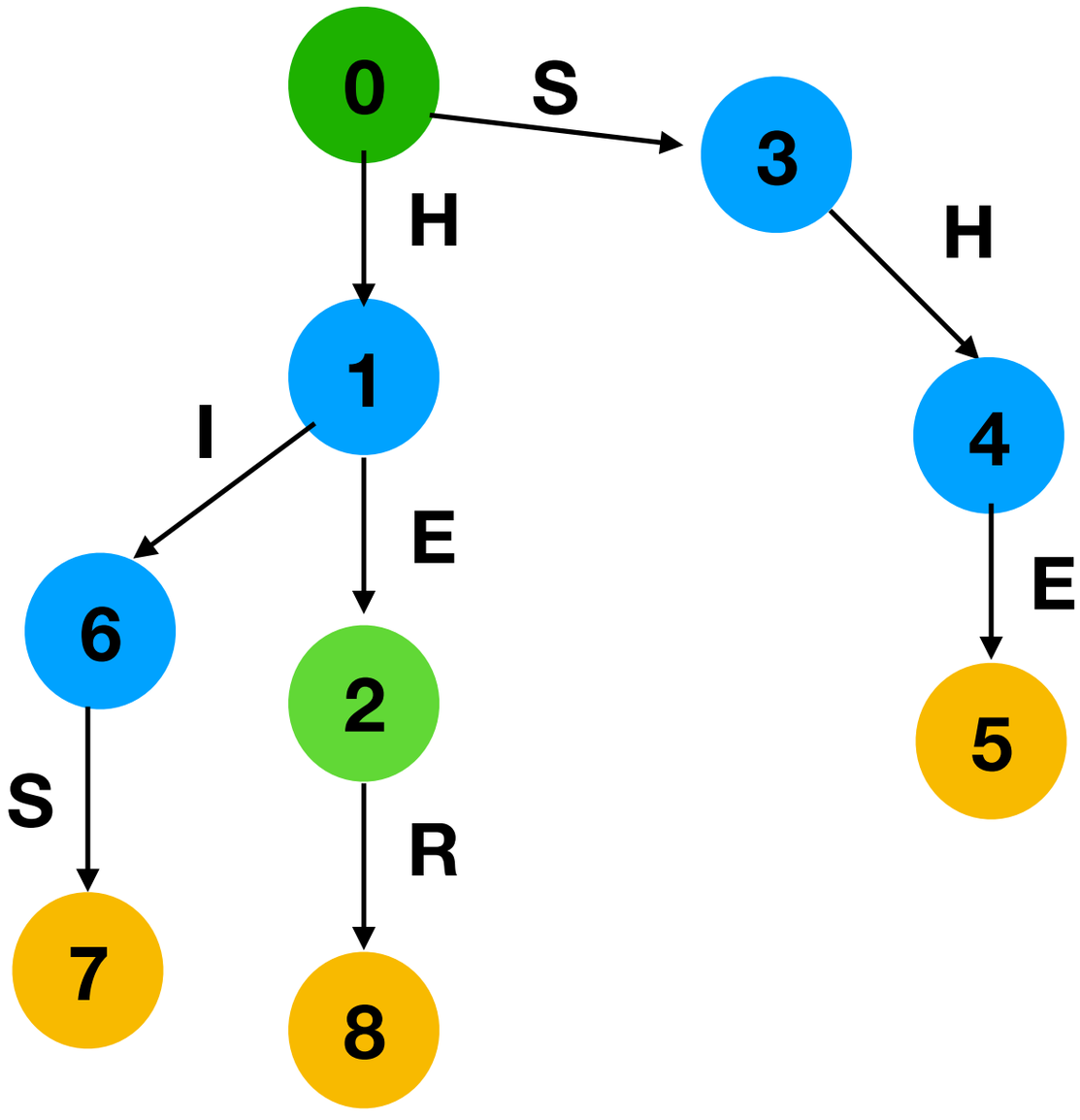
Попробуем поискать в строке



HERHEHIS

Если не можем прочитать символ из текущей вершины - идем в корень

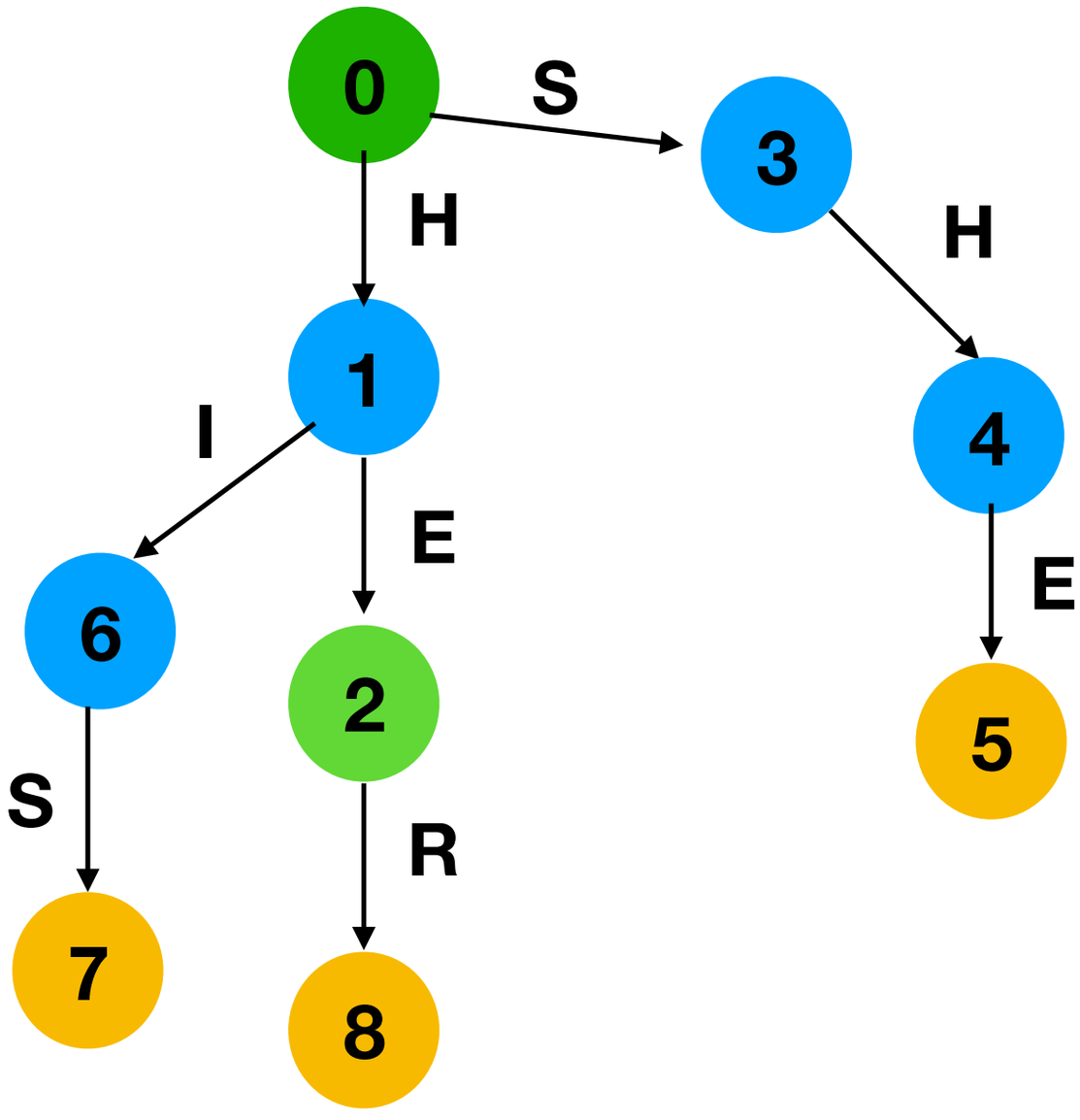
Попробуем поискать в строке



HERHEHIS

Нашли HE

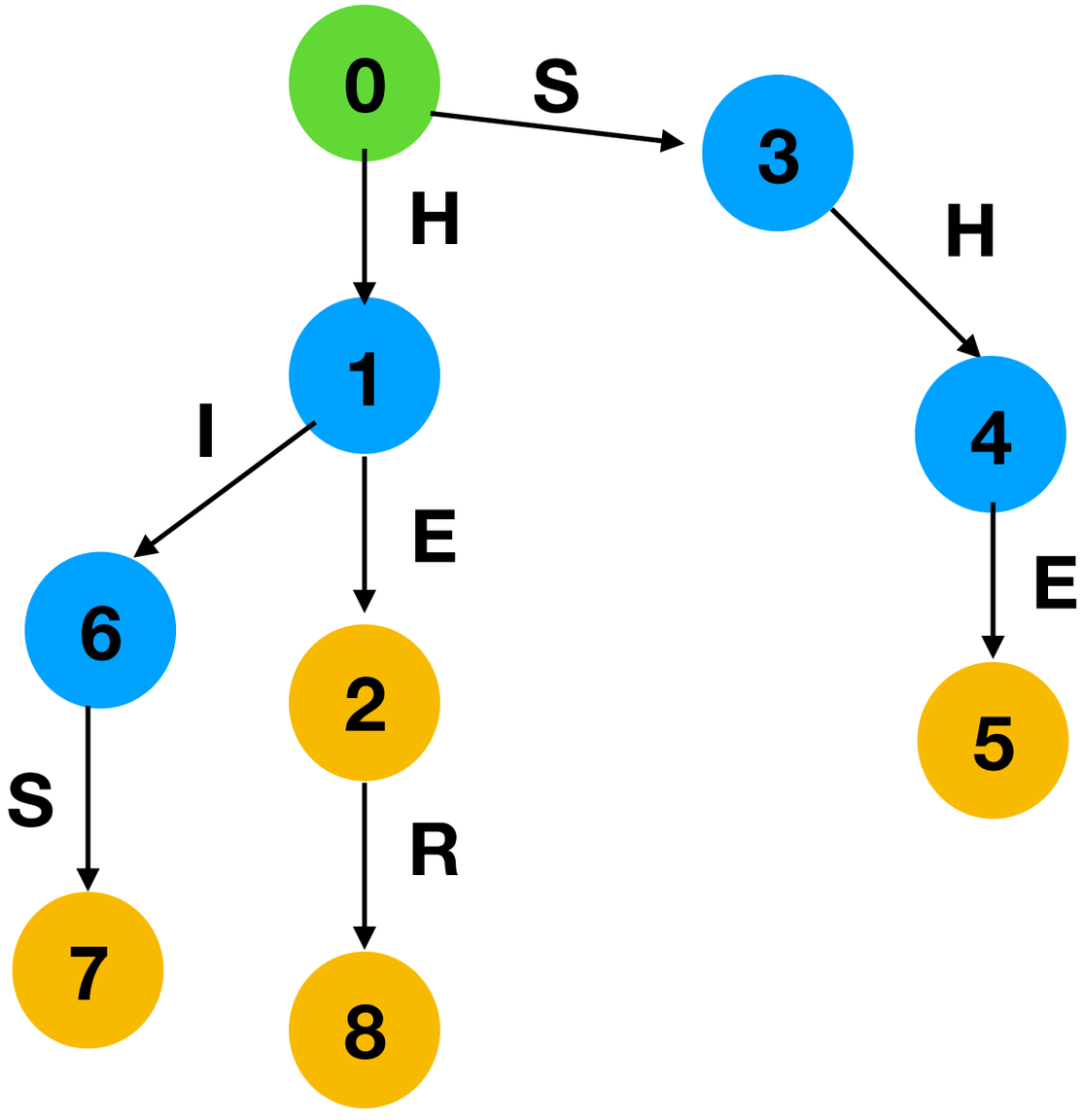
Попробуем поискать в строке



HERHEHIS

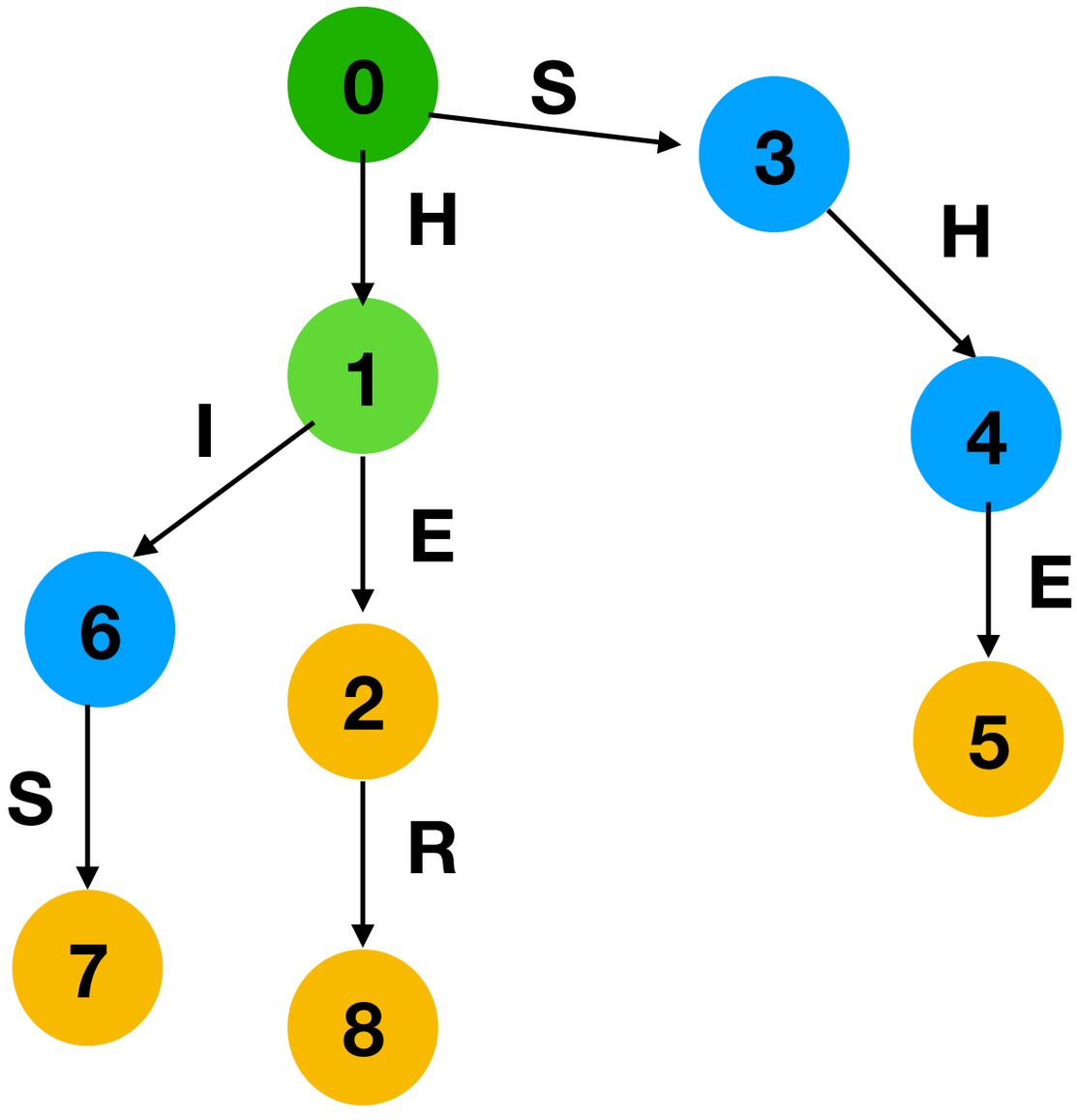
Если не можем прочитать символ из текущей вершины - идем в корень

Попробуем поискать в строке



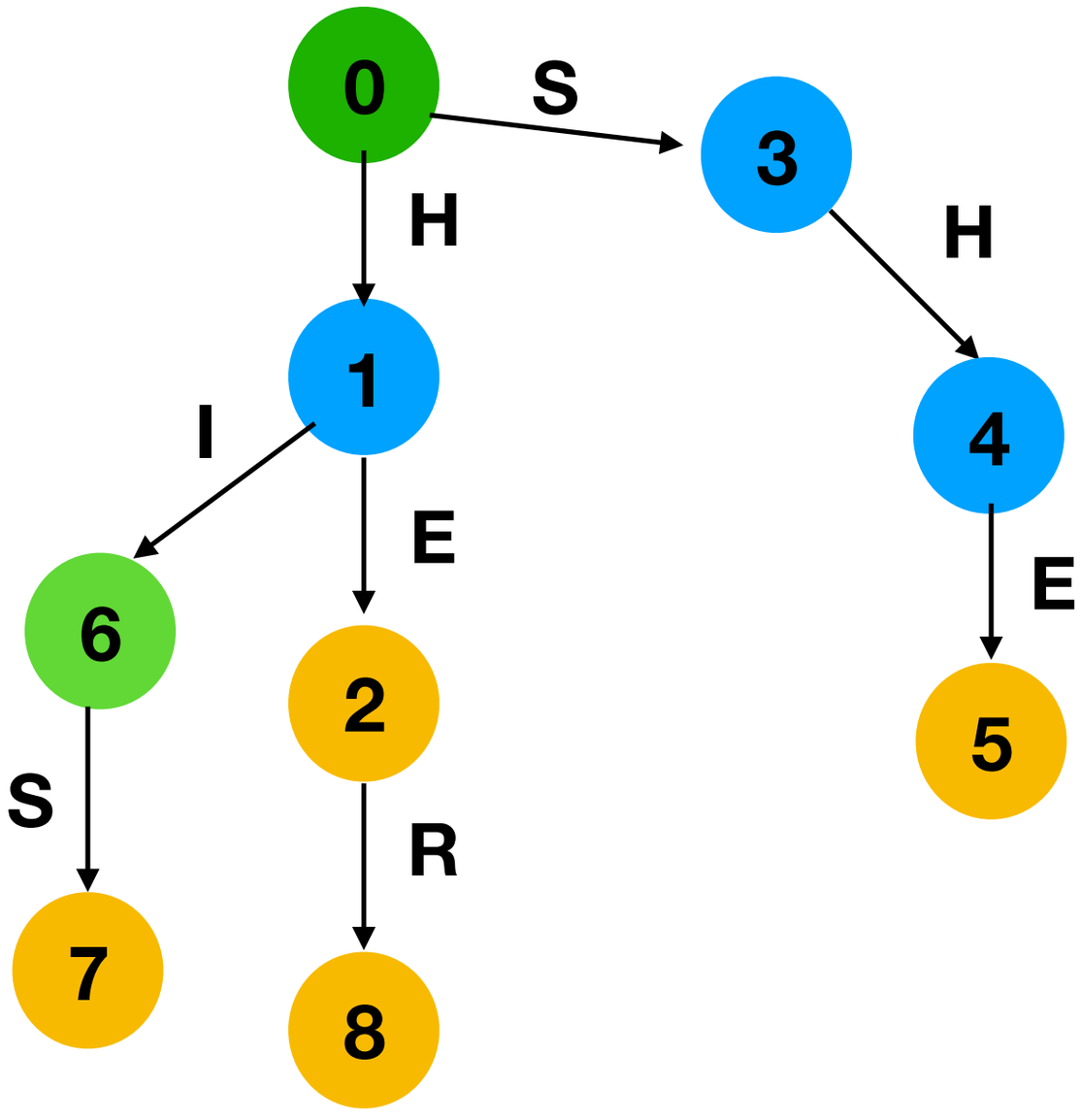
HERHEHIS

Попробуем поискать в строке



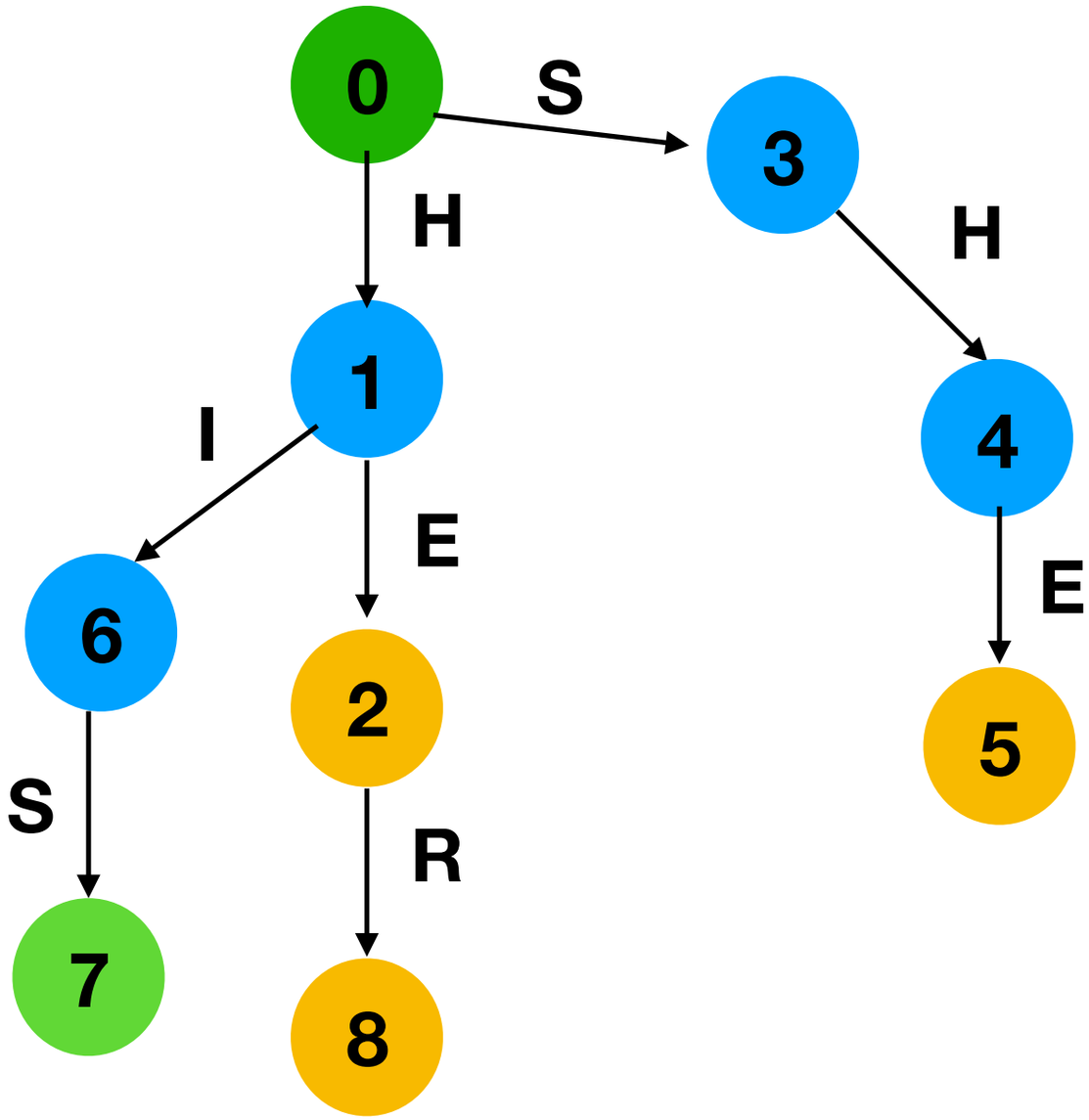
HERHEHIS

Попробуем поискать в строке



HERHEHIS

Попробуем поискать в строке

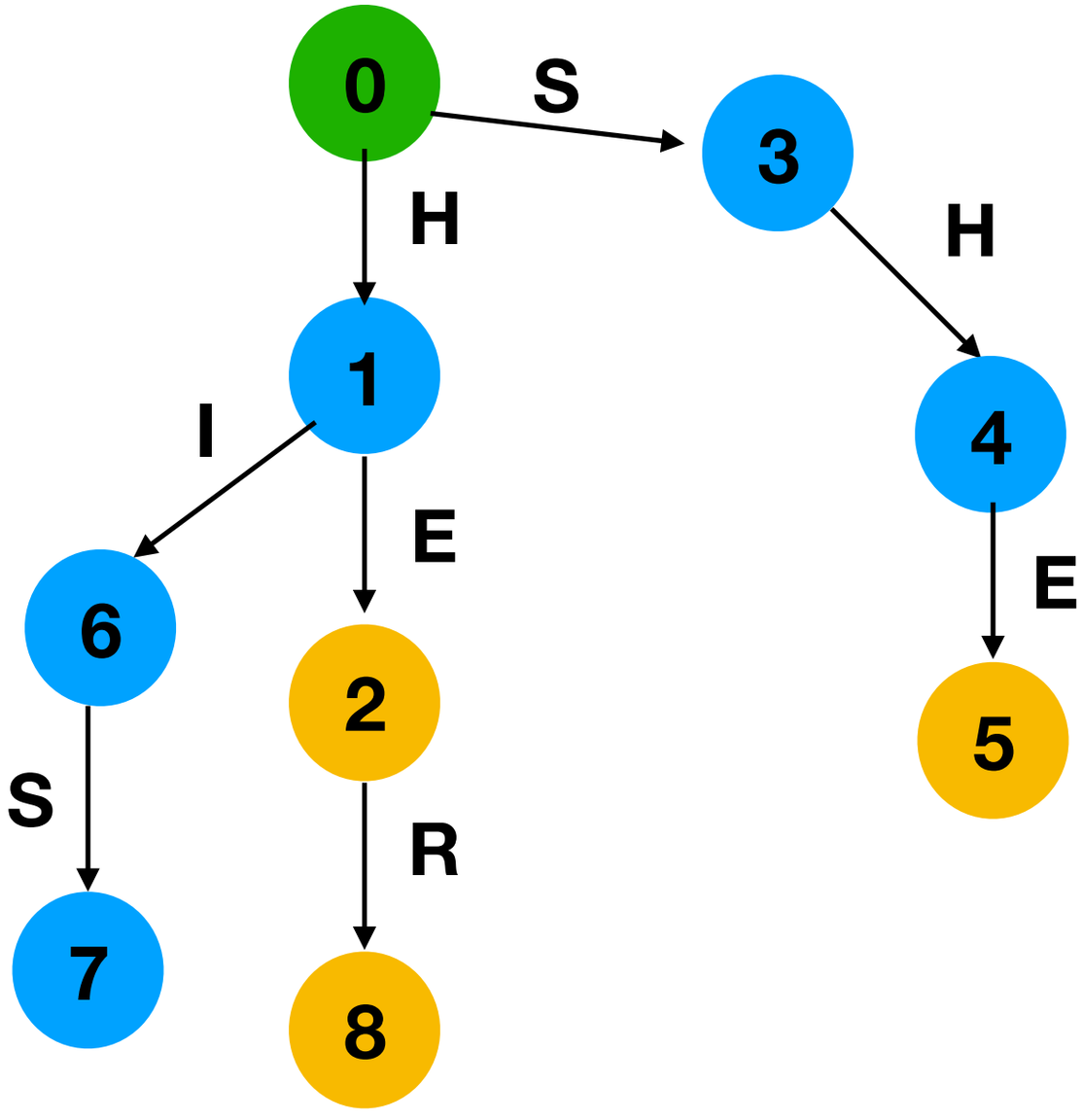


HERHEHIS

Нашли HIS

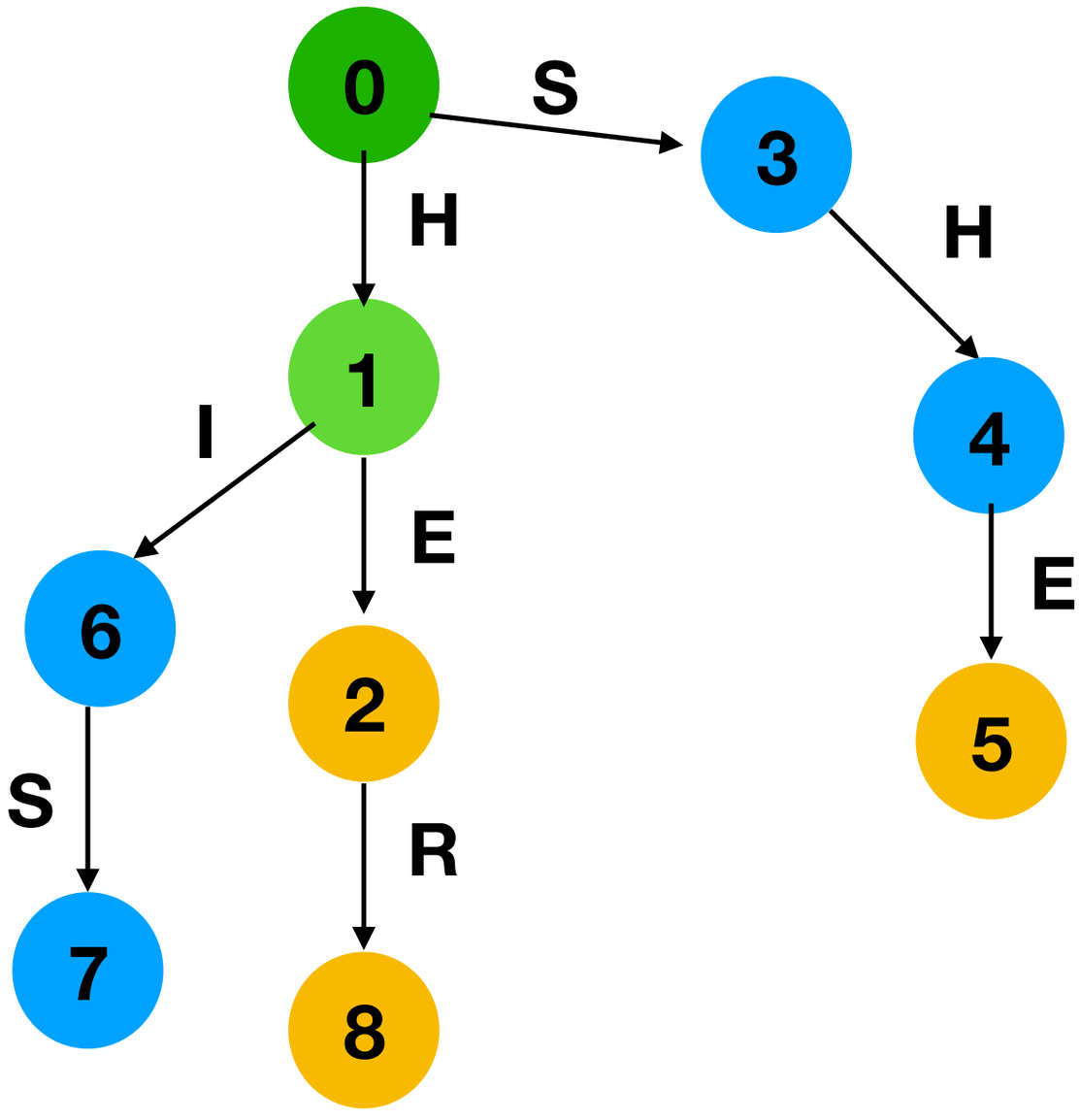
Нашли все, что было

Попробуем поискать в строке



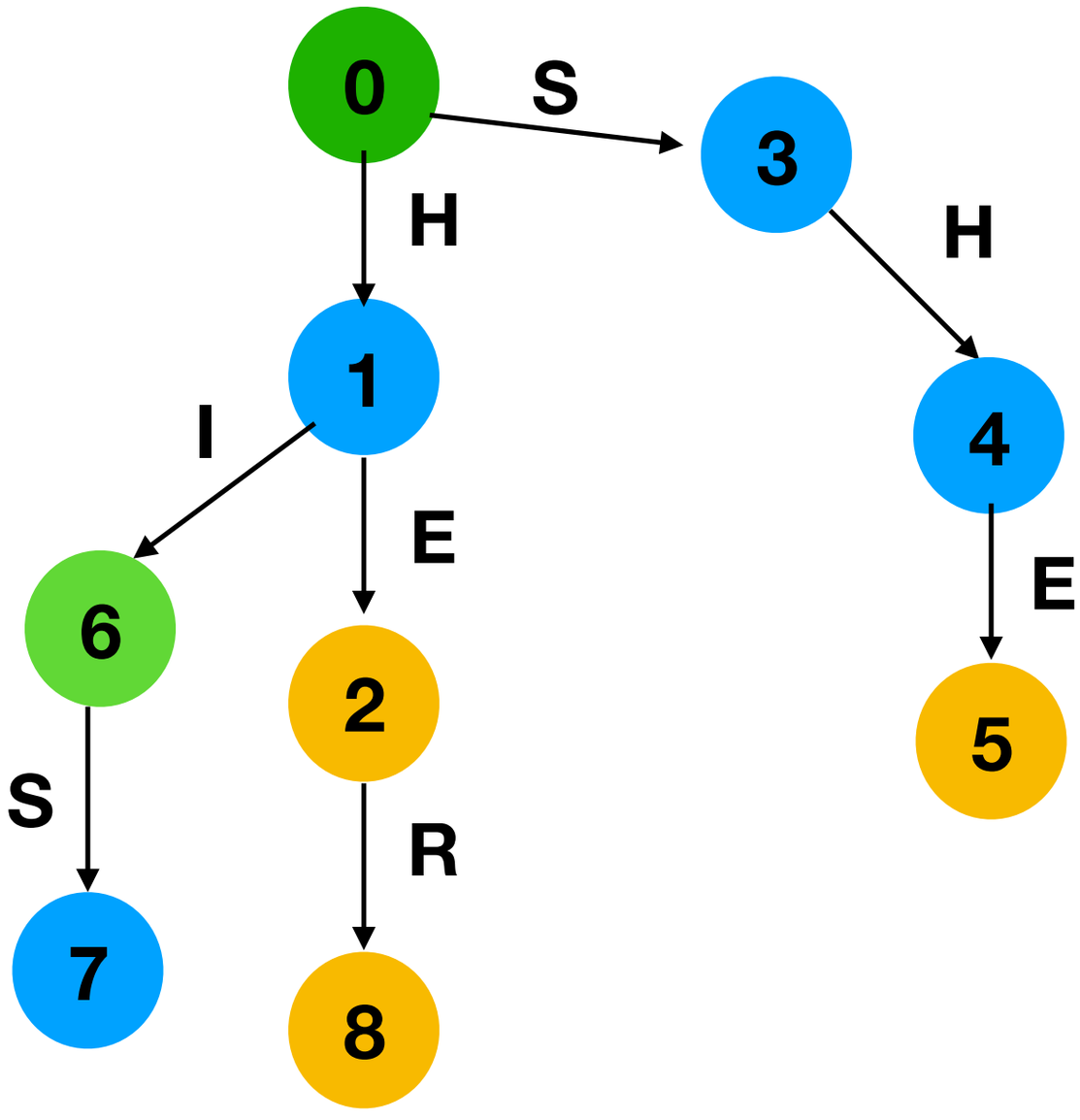
HISHER

Попробуем поискать в строке



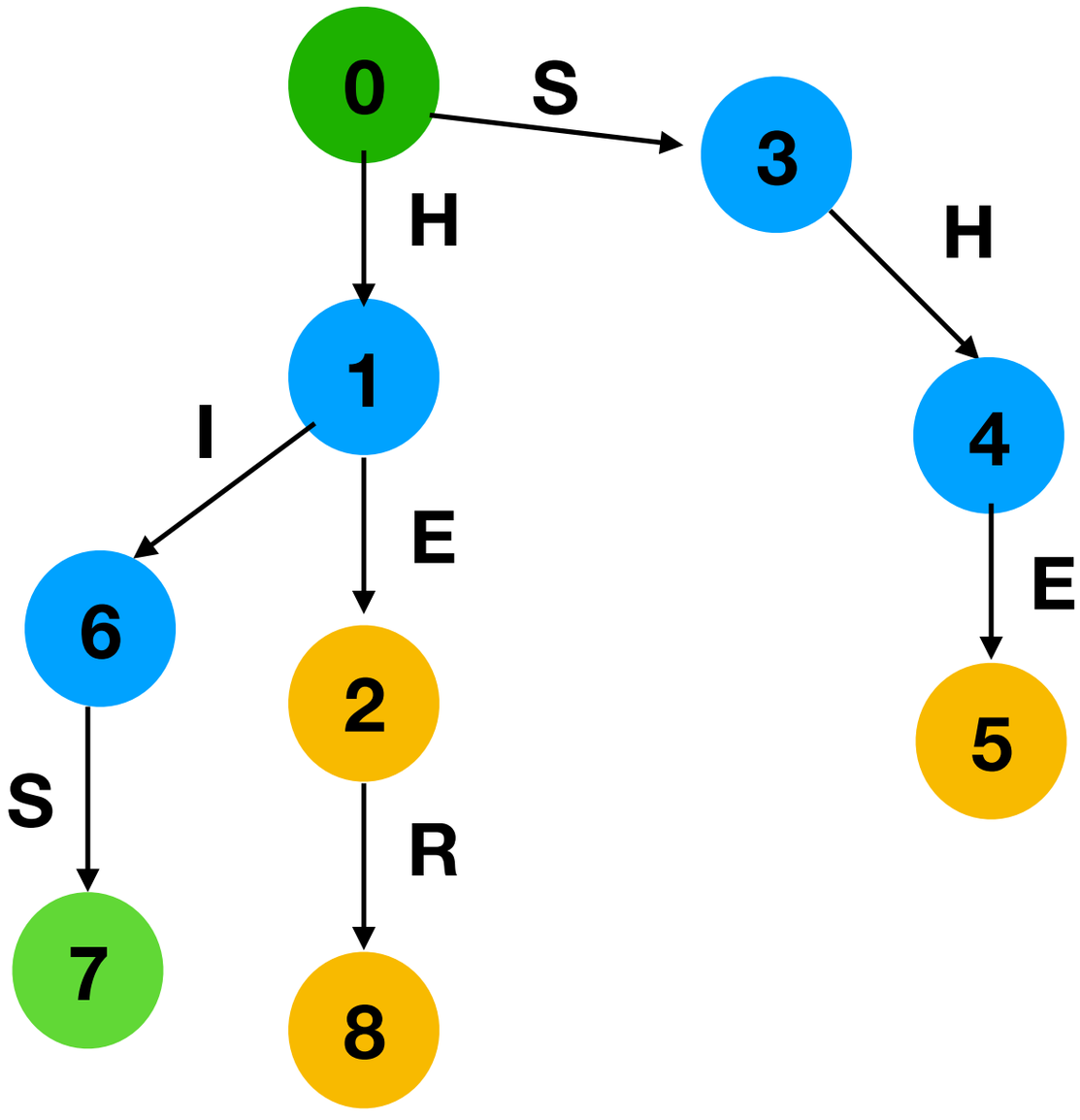
HISHER

Попробуем поискать в строке



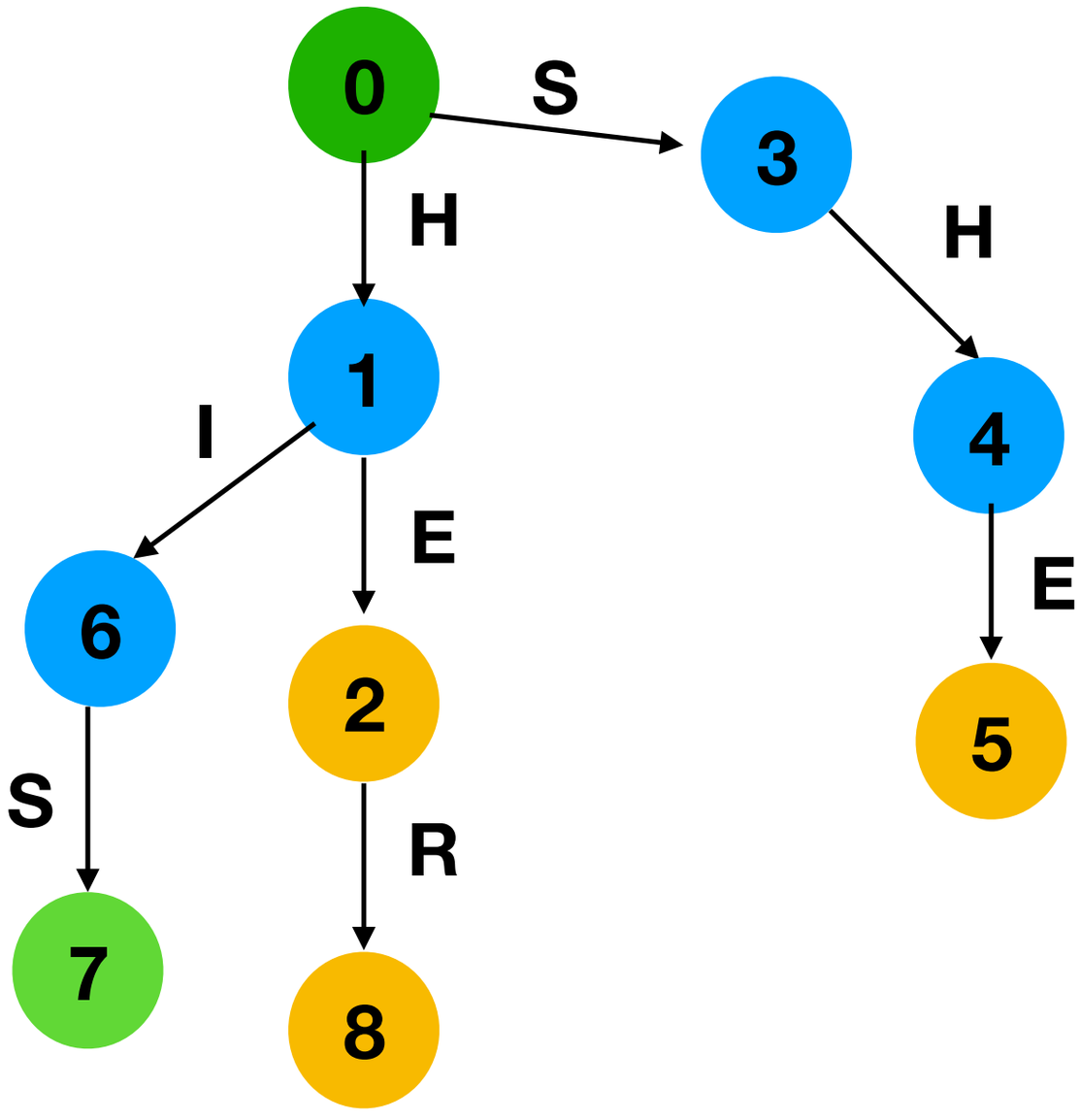
HISHER

Попробуем поискать в строке



HISHER

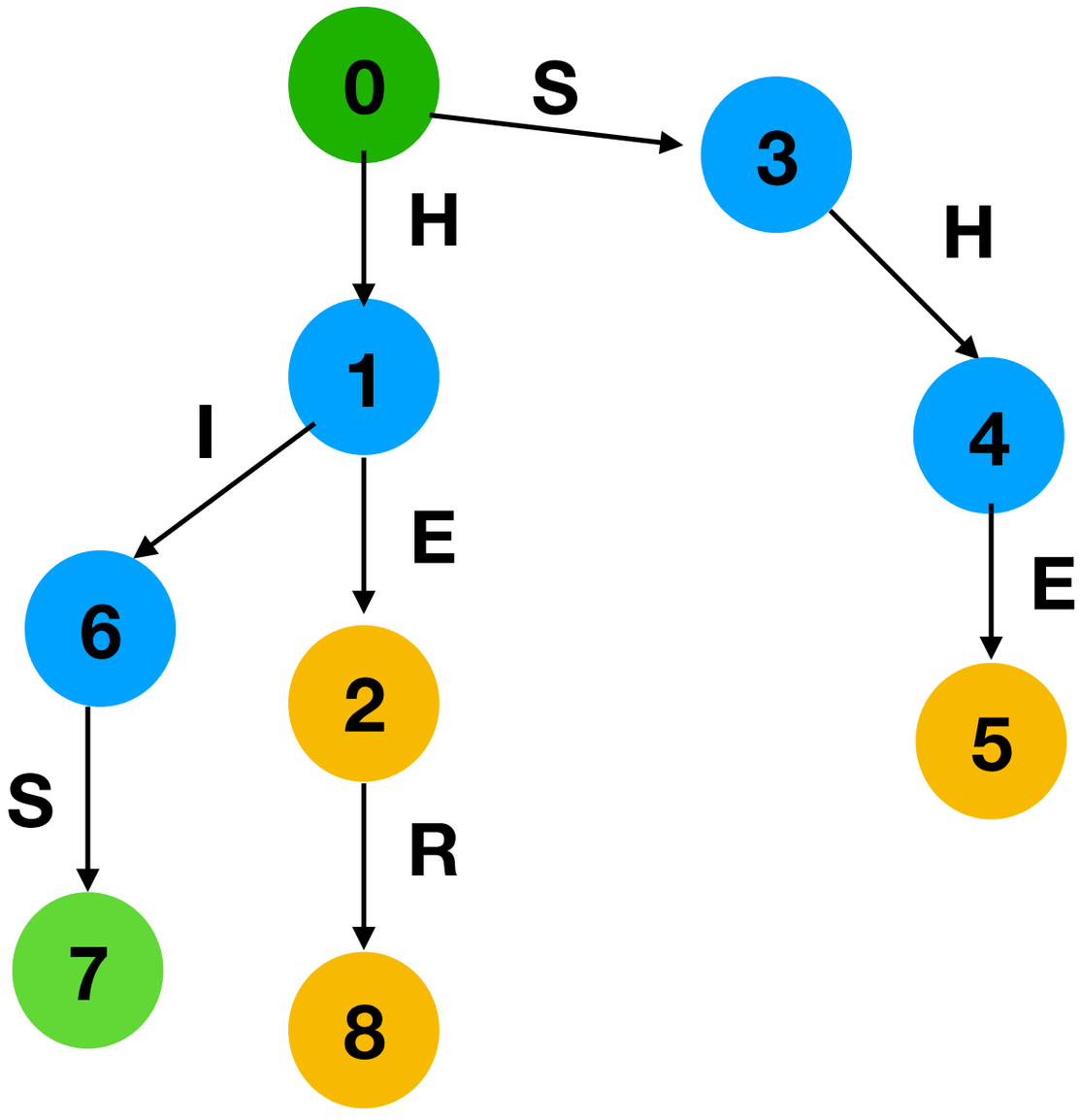
Попробуем поискать в строке



HISHER

Нашли HIS

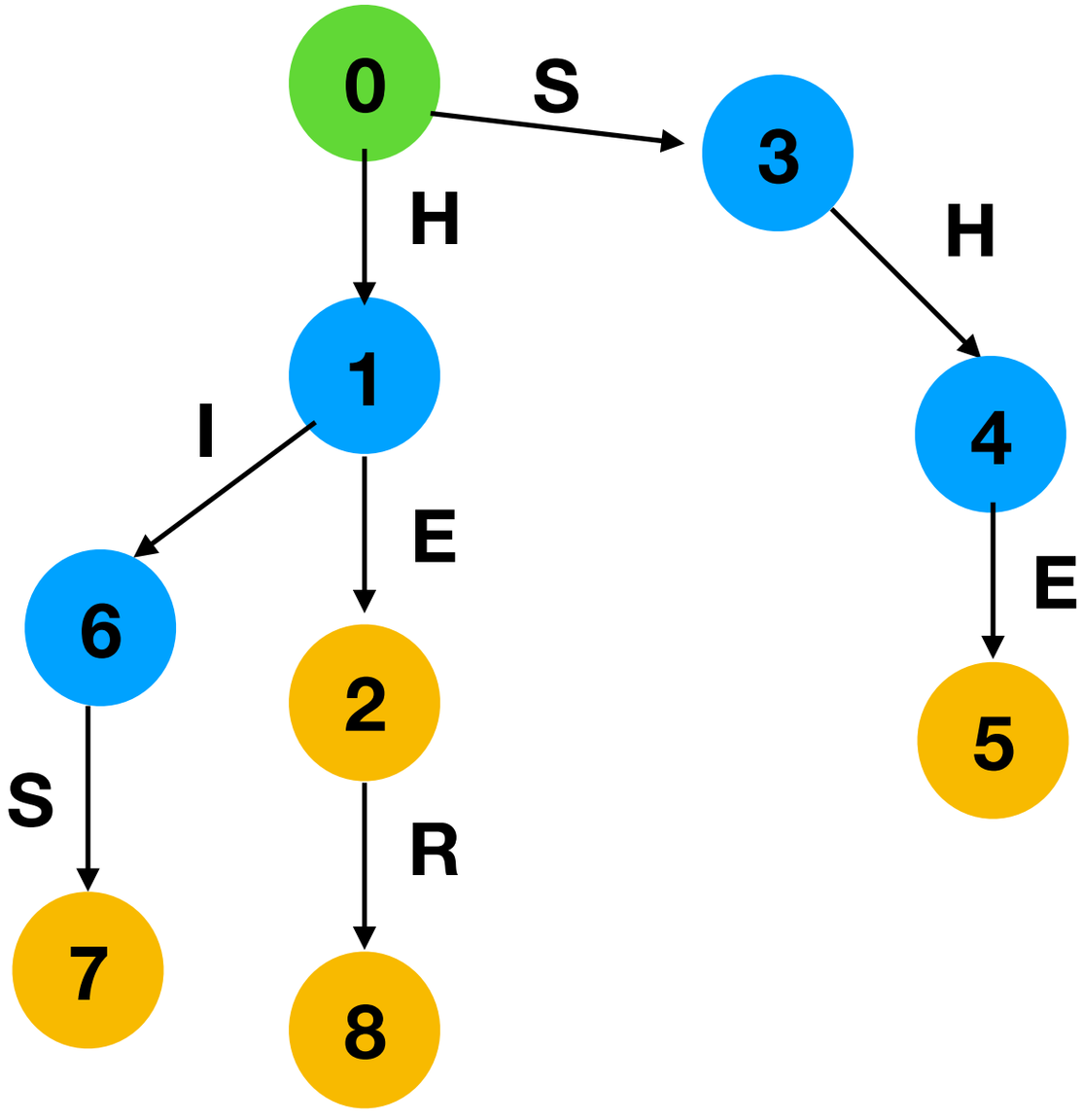
Попробуем поискать в строке



HISHER

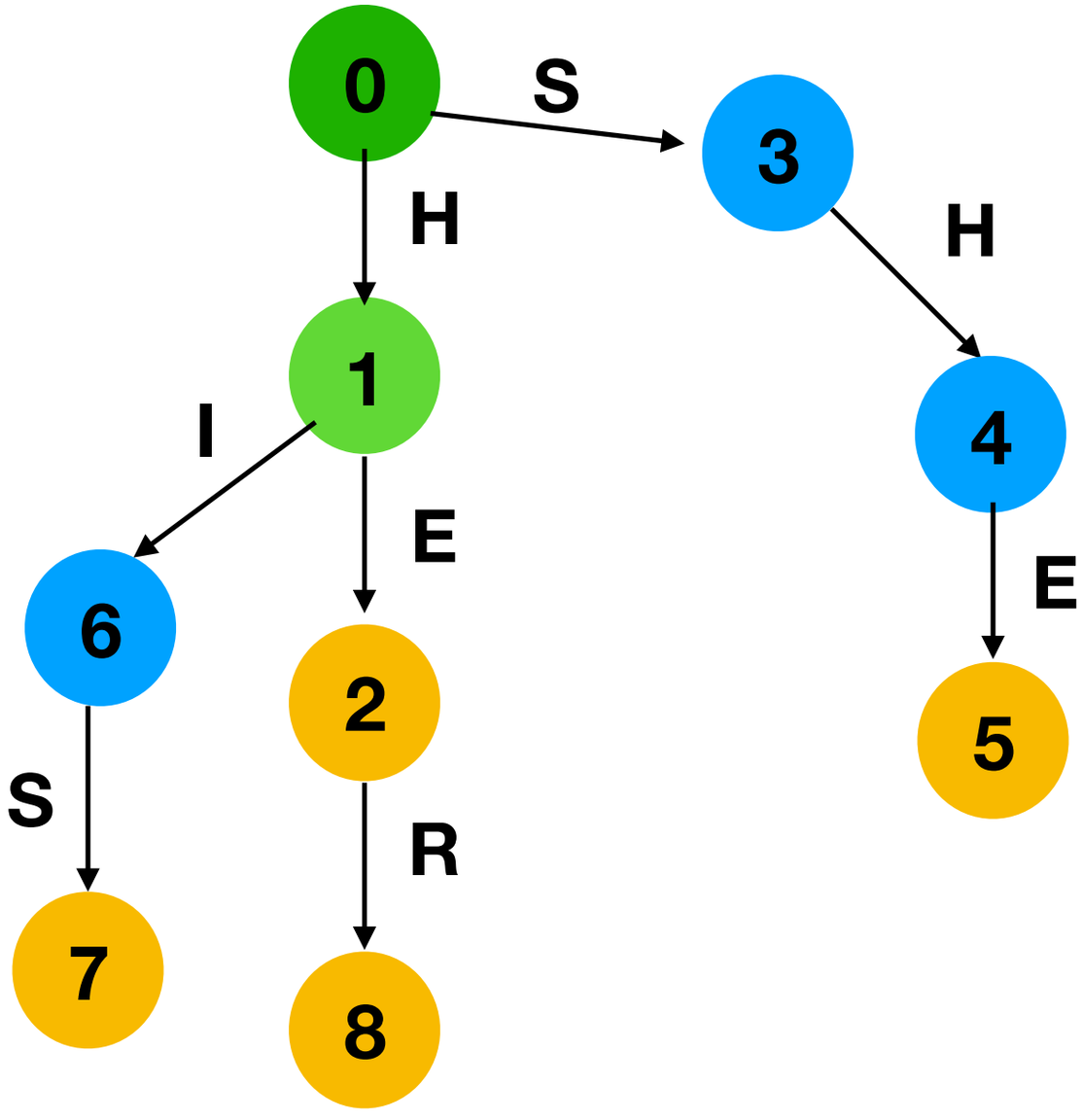
Прочитать H из вершины не можем - идем в корень

Попробуем поискать в строке



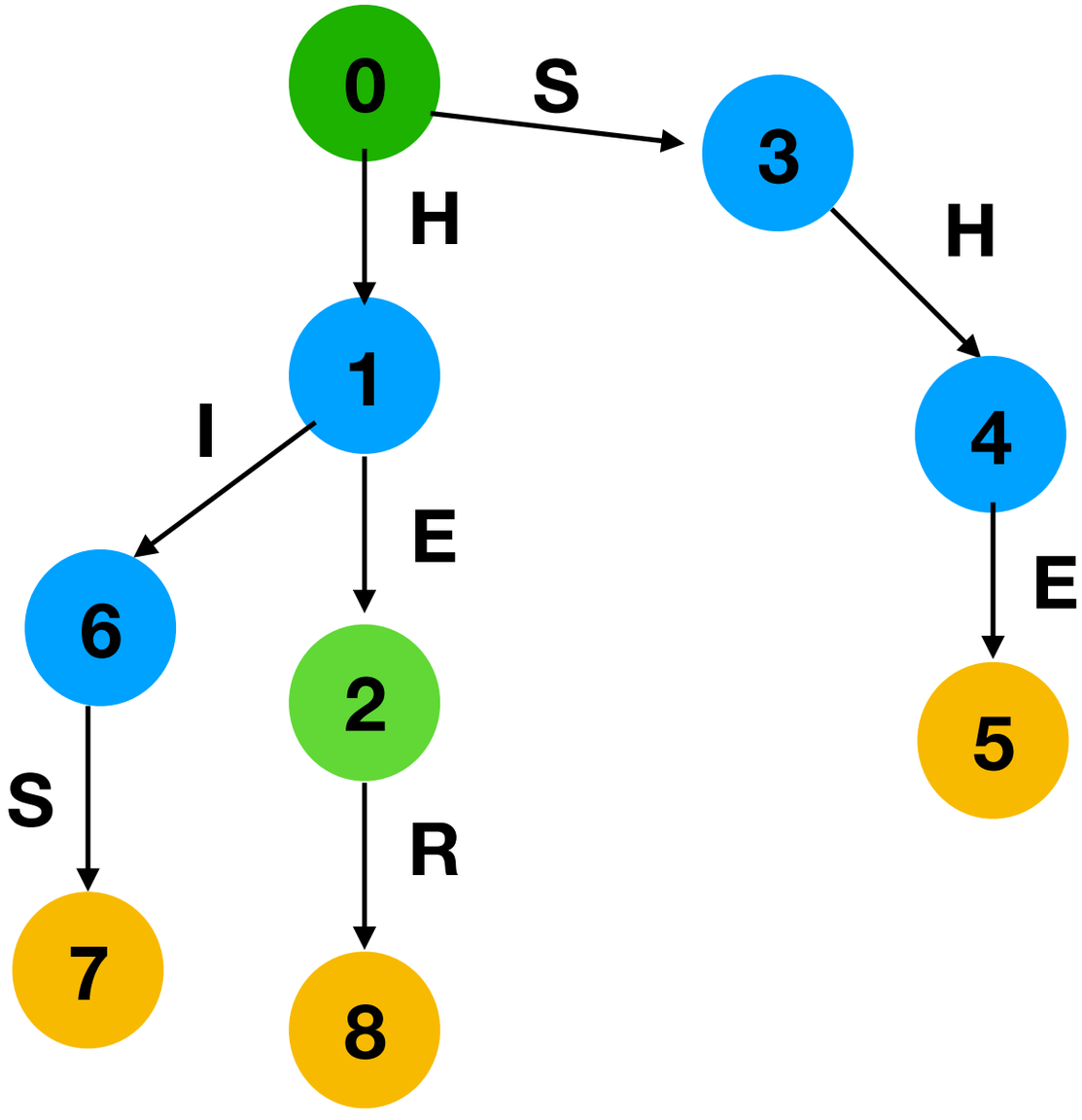
HISHER

Попробуем поискать в строке



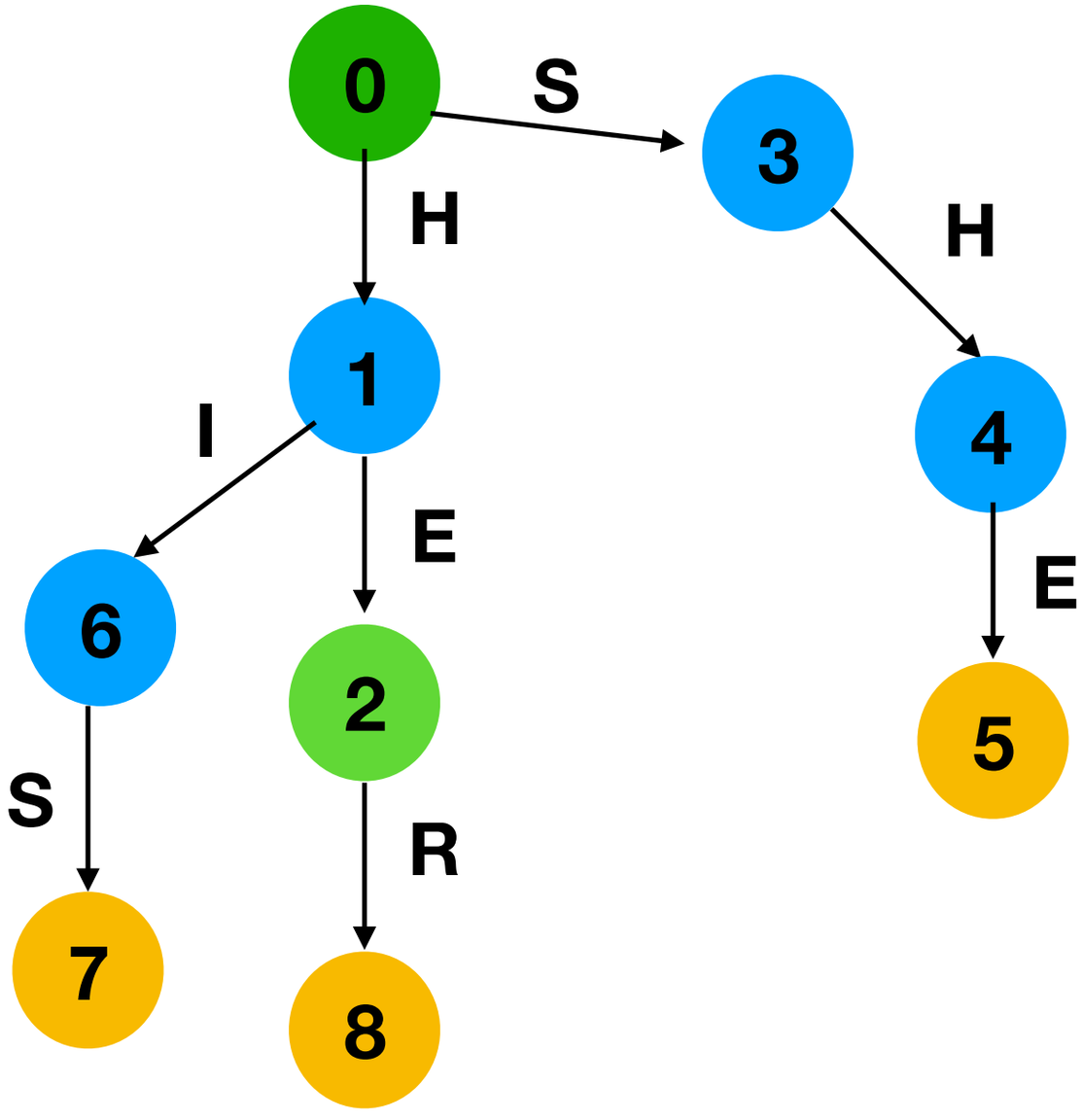
HISHER

Попробуем поискать в строке



HISHER

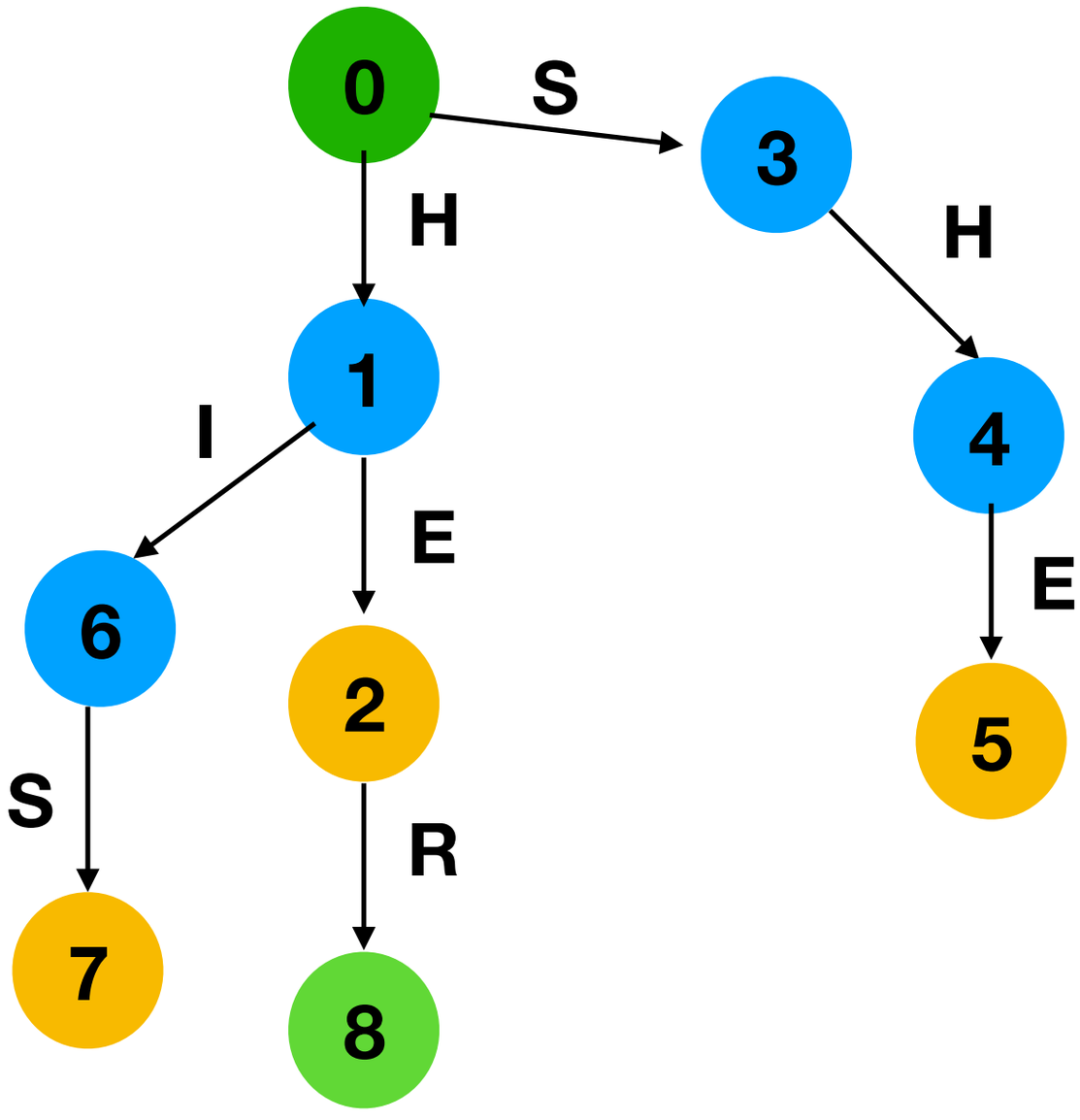
Попробуем поискать в строке



HISHER

Нашли HE

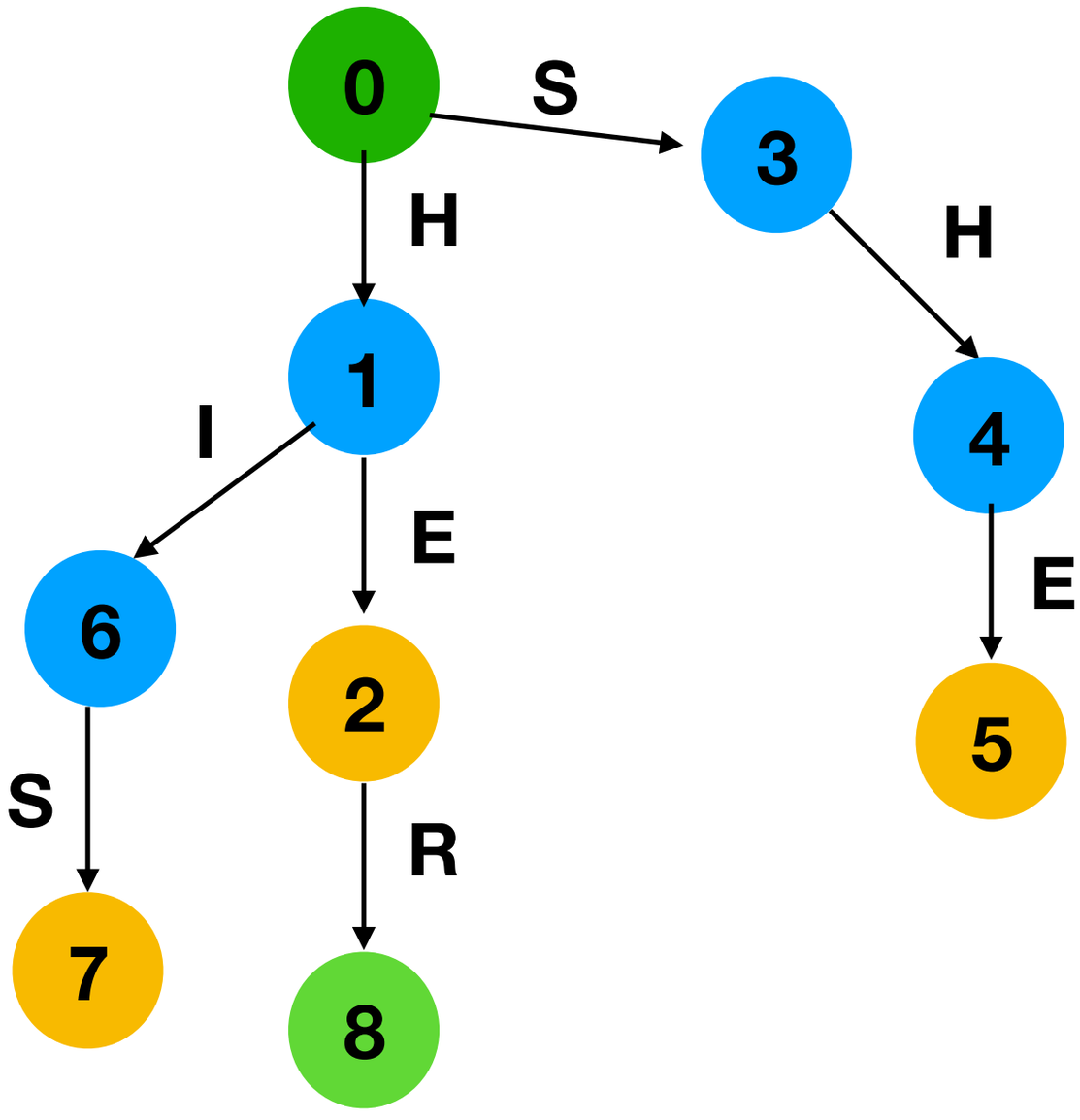
Попробуем поискать в строке



HISHER

Нашли HER

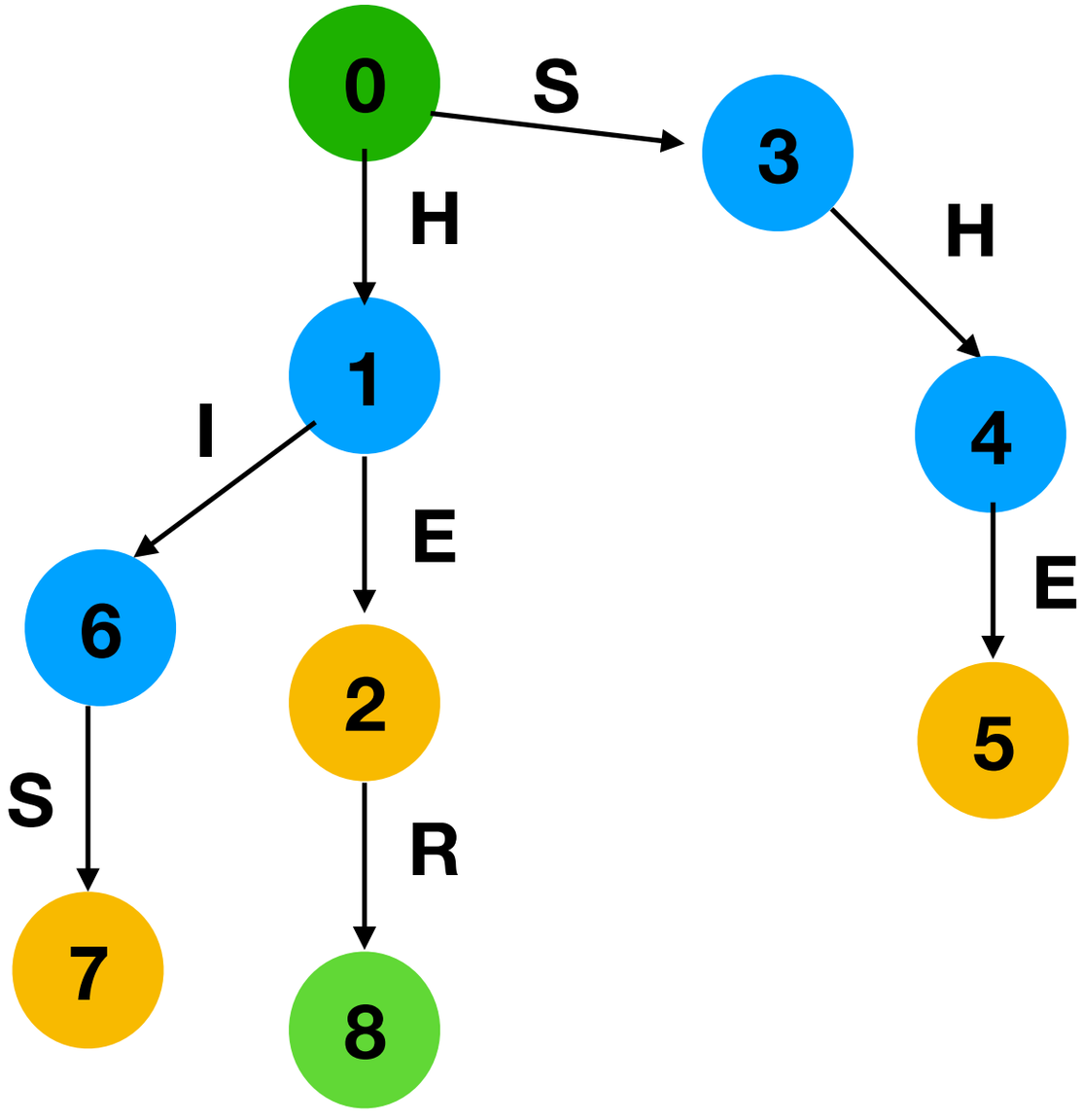
Попробуем поискать в строке



HISHER

Мы не нашли SHE, а оно было

Попробуем поискать в строке



HISHER

Мы не нашли SHE, а оно было

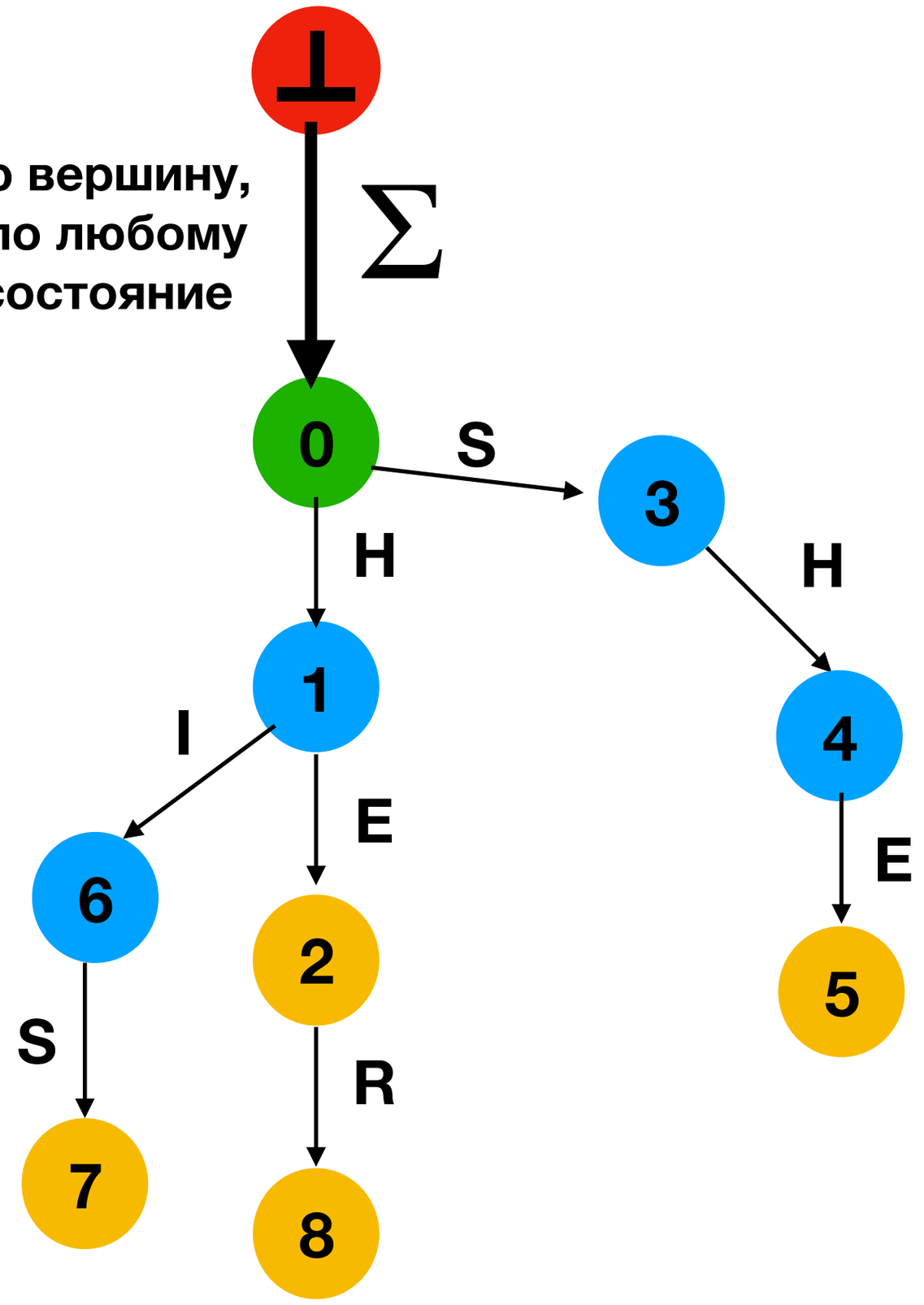
Надо переходить не в корень

Суффикс-ссылка

Суффикс-ссылка - длина наибольшего собственного суффикса строки, которому соответствует некоторая вершина в боре (этот суффикс можно прочитать, идя от корня)

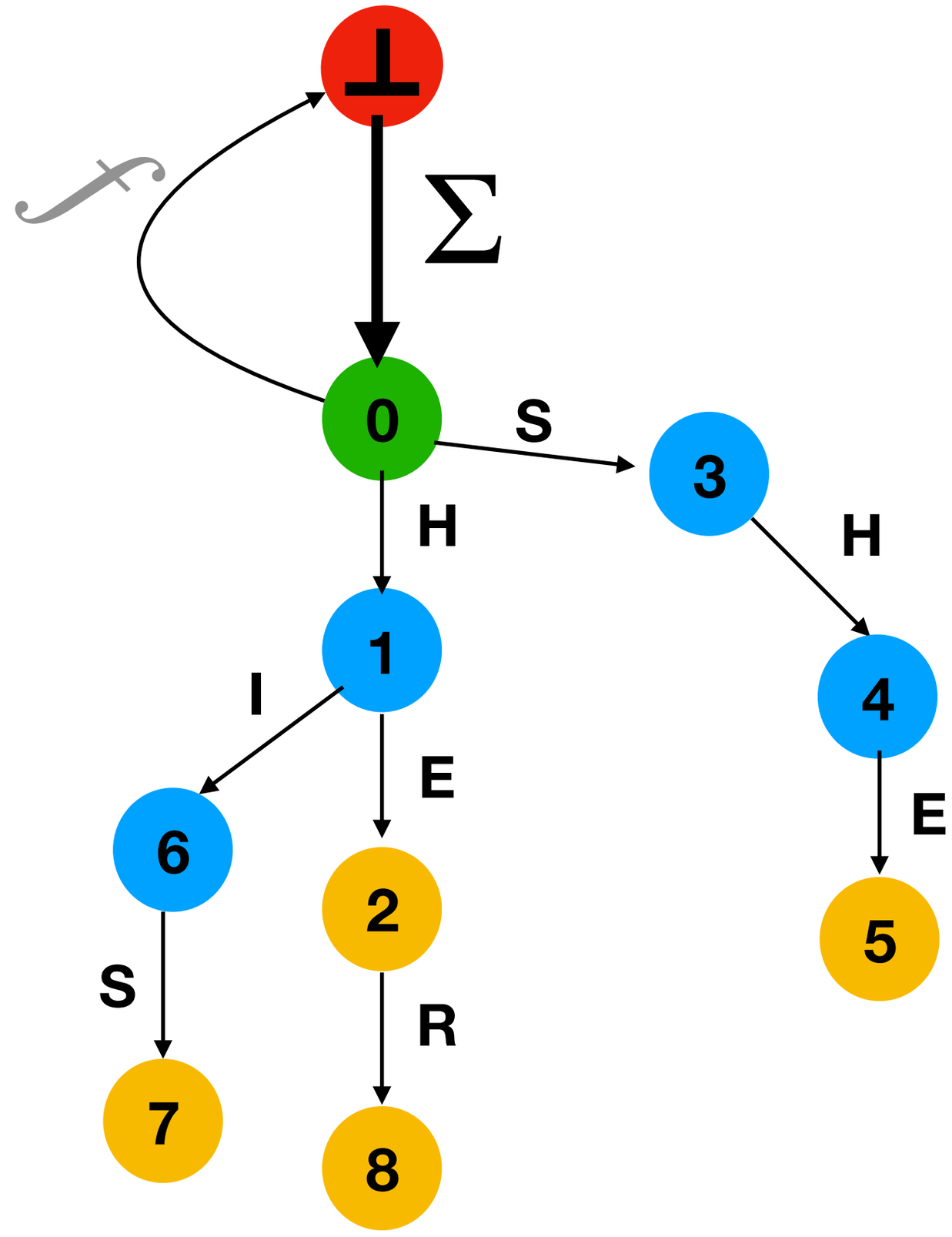
Суффикс-ссылка

Добавим в бор фиктивную вершину, из которой есть переход по любому символу алфавита в 0-е состояние



Суффикс-ссылка

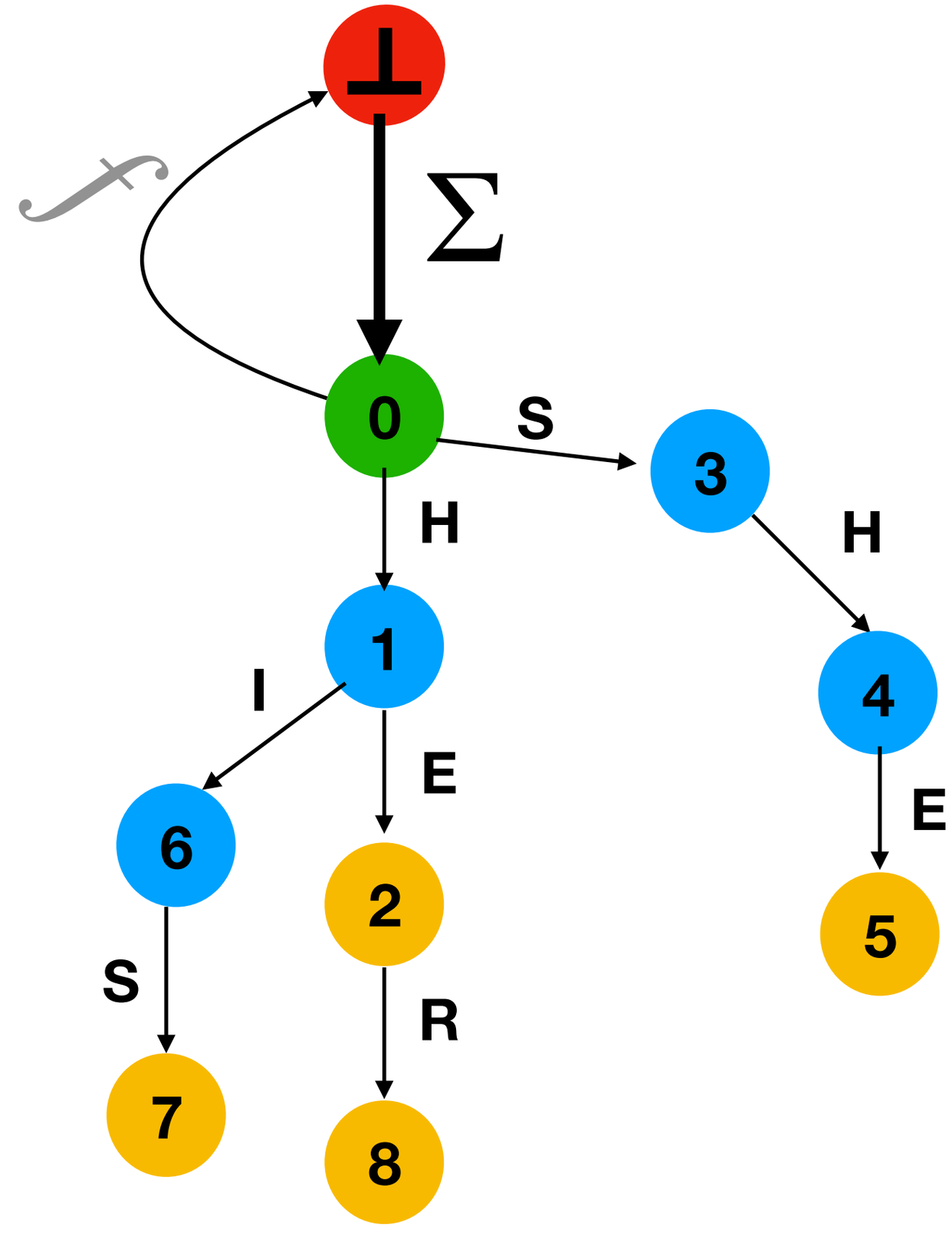
Пусть суффикс-ссылка корня указываем на нее



Суффикс-ссылка

Далее суффикс ссылка вершины v высчитывается следующим образом:

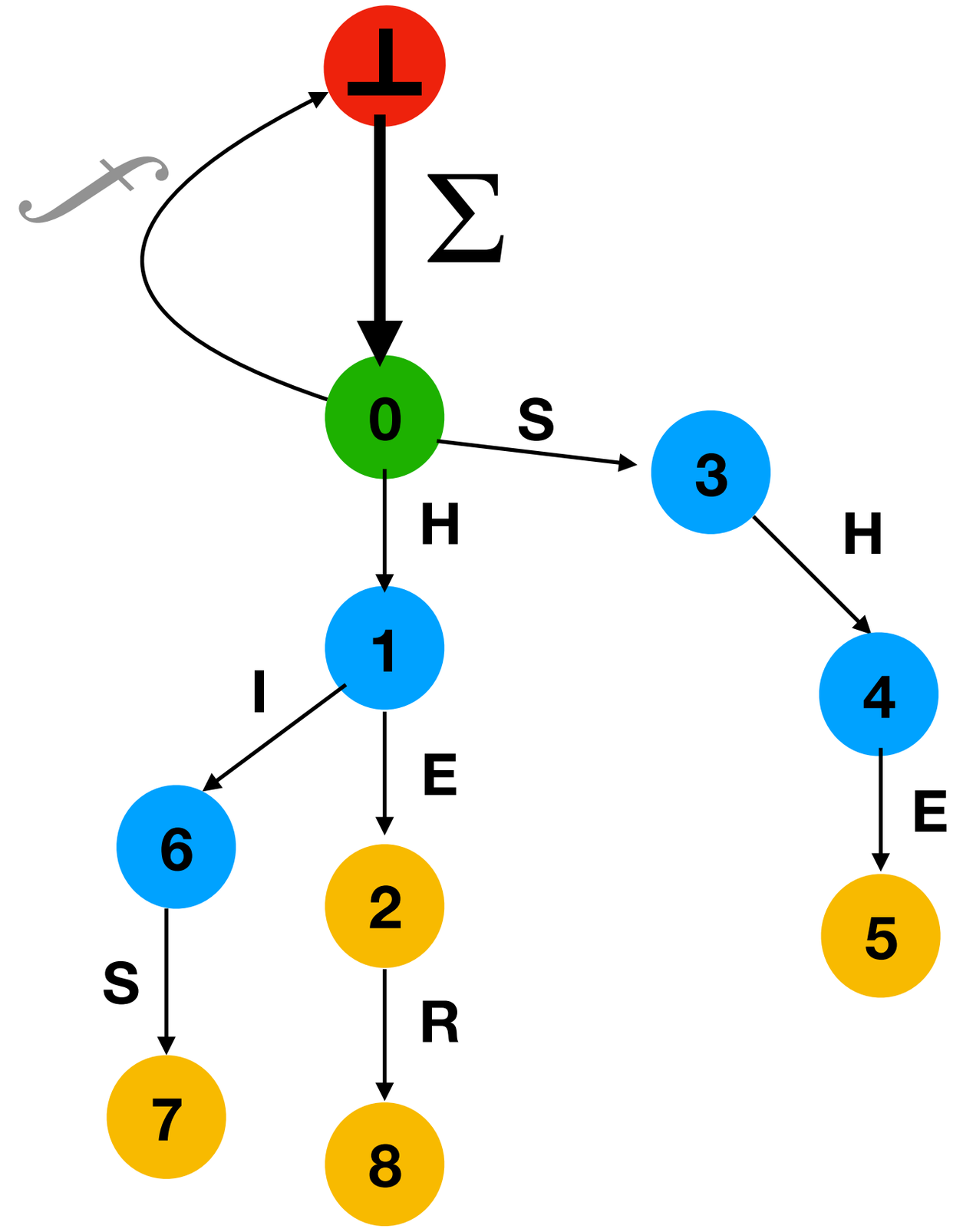
1) **перейти в предка u вершины v**
2) **Перейти по суффиксной ссылке предка в вершину h**
3) **Попытаться пройти из этой вершины h по символу, который вел из предка u в вершину v**
4) **Если получилось - то поставить суффиксную ссылку из v в вершину, которую мы смогли перешли**
5) **Иначе: перейти по суффиксной ссылке вершины h**



Суффикс-ссылка

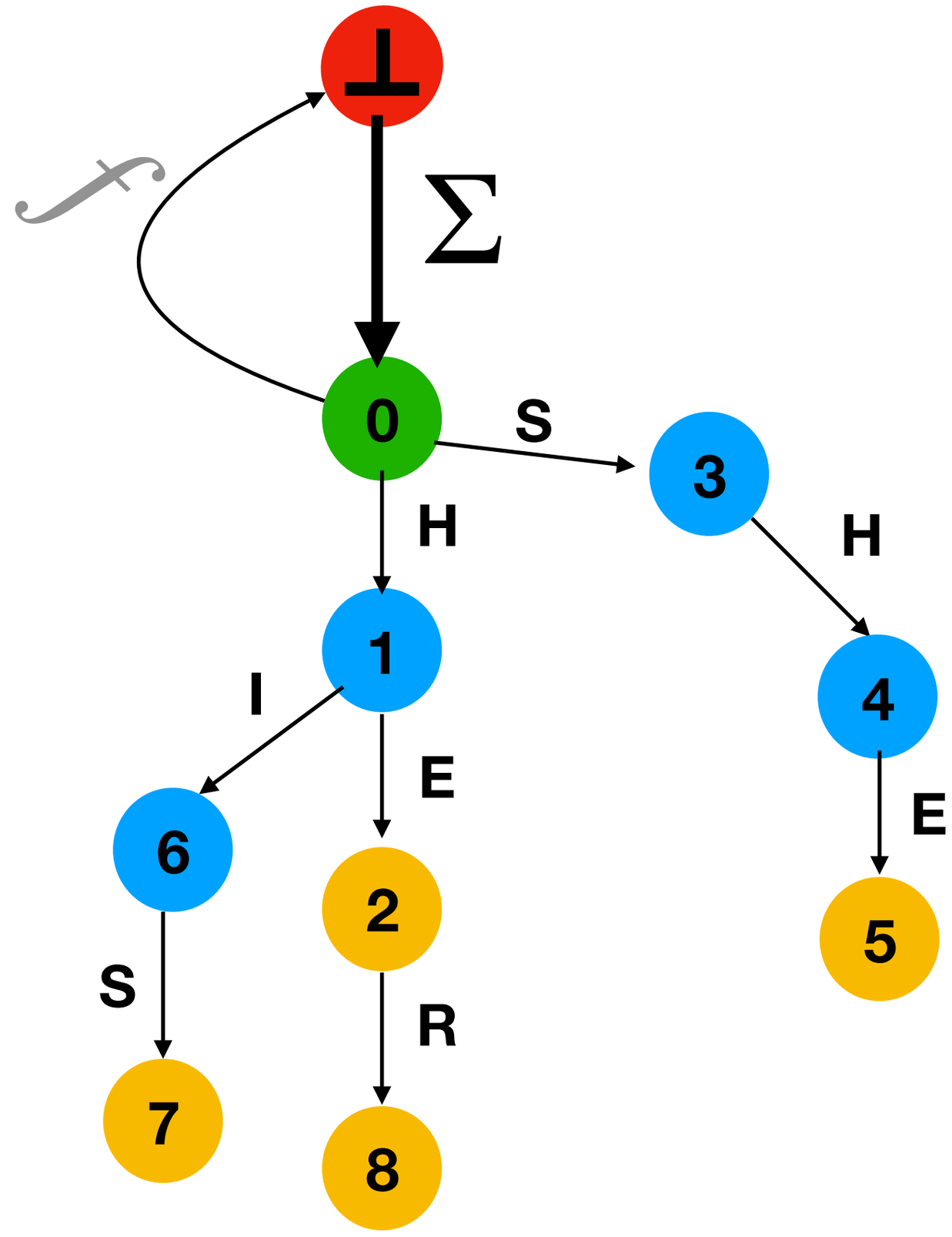
Далее суффикс ссылка вершины v высчитывается следующим образом:

1) **перейти в предка u вершины v**
2) **Перейти по суффиксной ссылке предка в вершину h**
3) **Попытаться пройти из этой вершины h по символу, который вел из предка u в вершину v**
4) **Если получилось - то поставить суффиксную ссылку из v в вершину, которую мы смогли перешли**
5) **Иначе: перейти по суффиксной ссылке вершины h**
6) **Суффиксы ссылки считаем сначала у вершин, более близких к корню**



Суффикс-ссылка

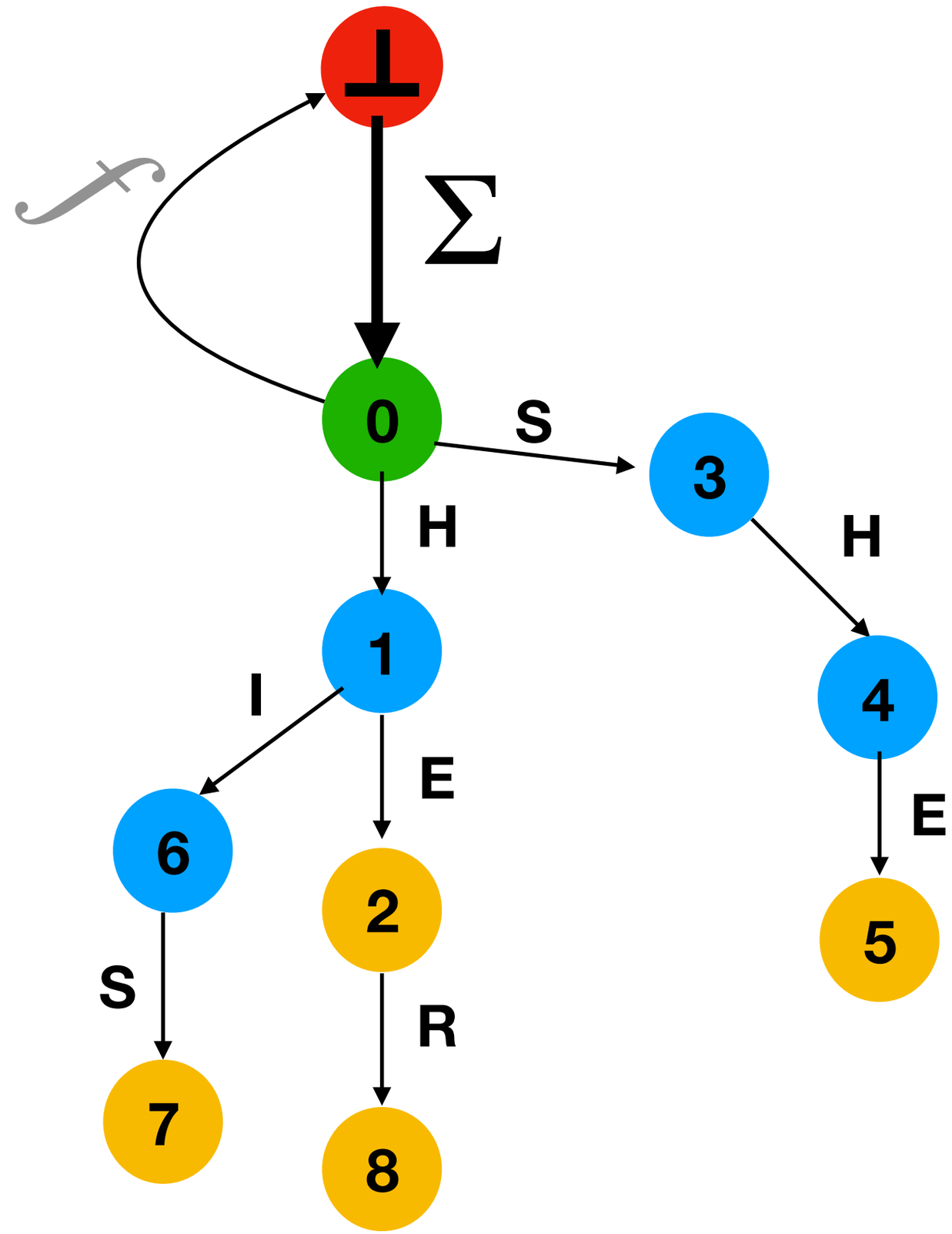
Посчитаем суффиксную ссылку вершины 3



Суффикс-ссылка

Посчитаем суффиксную ссылку вершины 3

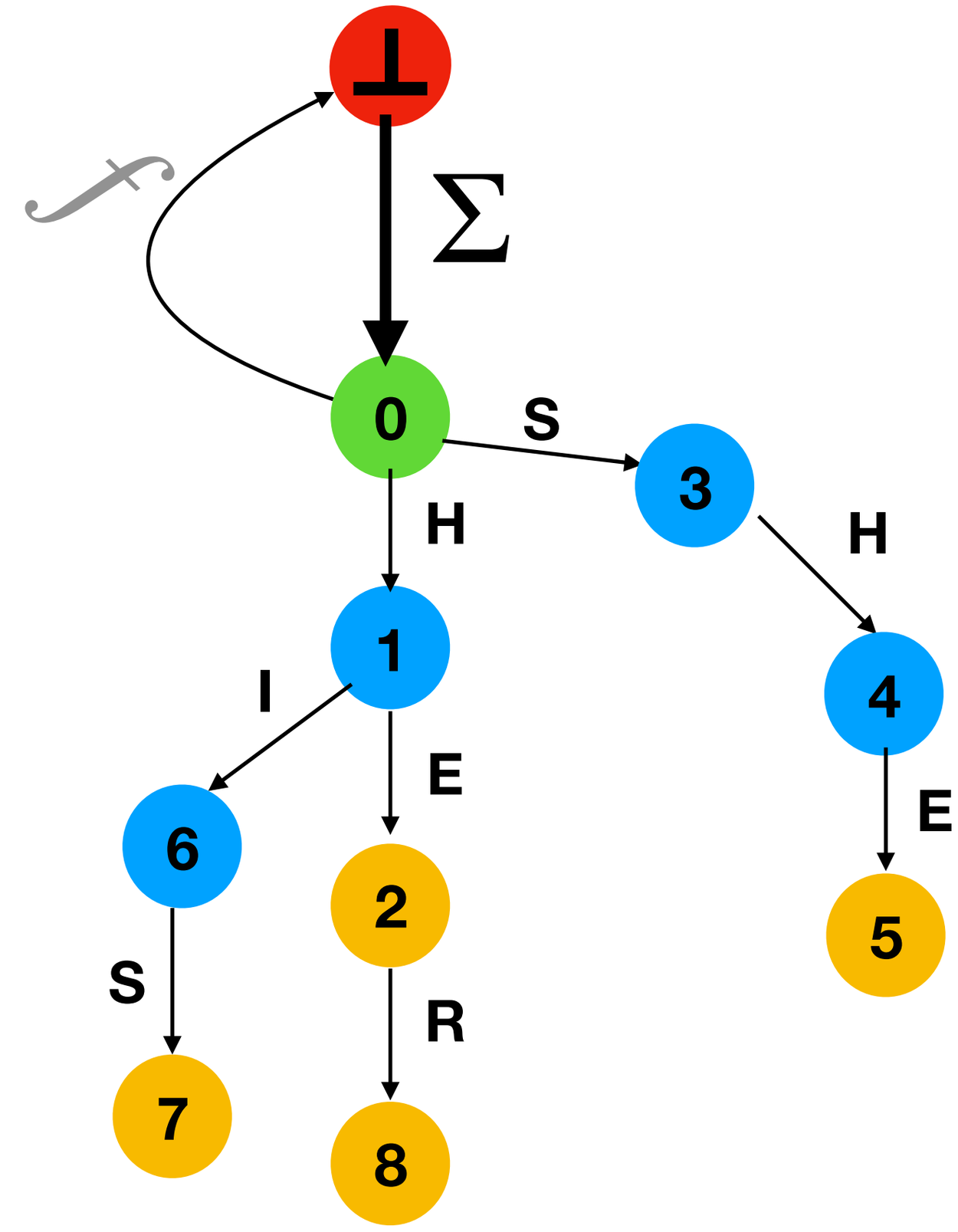
1) **Предок вершины 3 - корень.**
2) **Проходим по его суффиксной ссылке в фиктивную вершину**
3) **Из нее мы можем пройти по любому символу в 0 (корень)**
4) **Значит, суффиксная ссылка 3 ведет в корень**



Суффикс-ссылка

Посчитаем суффиксную ссылку вершины 3

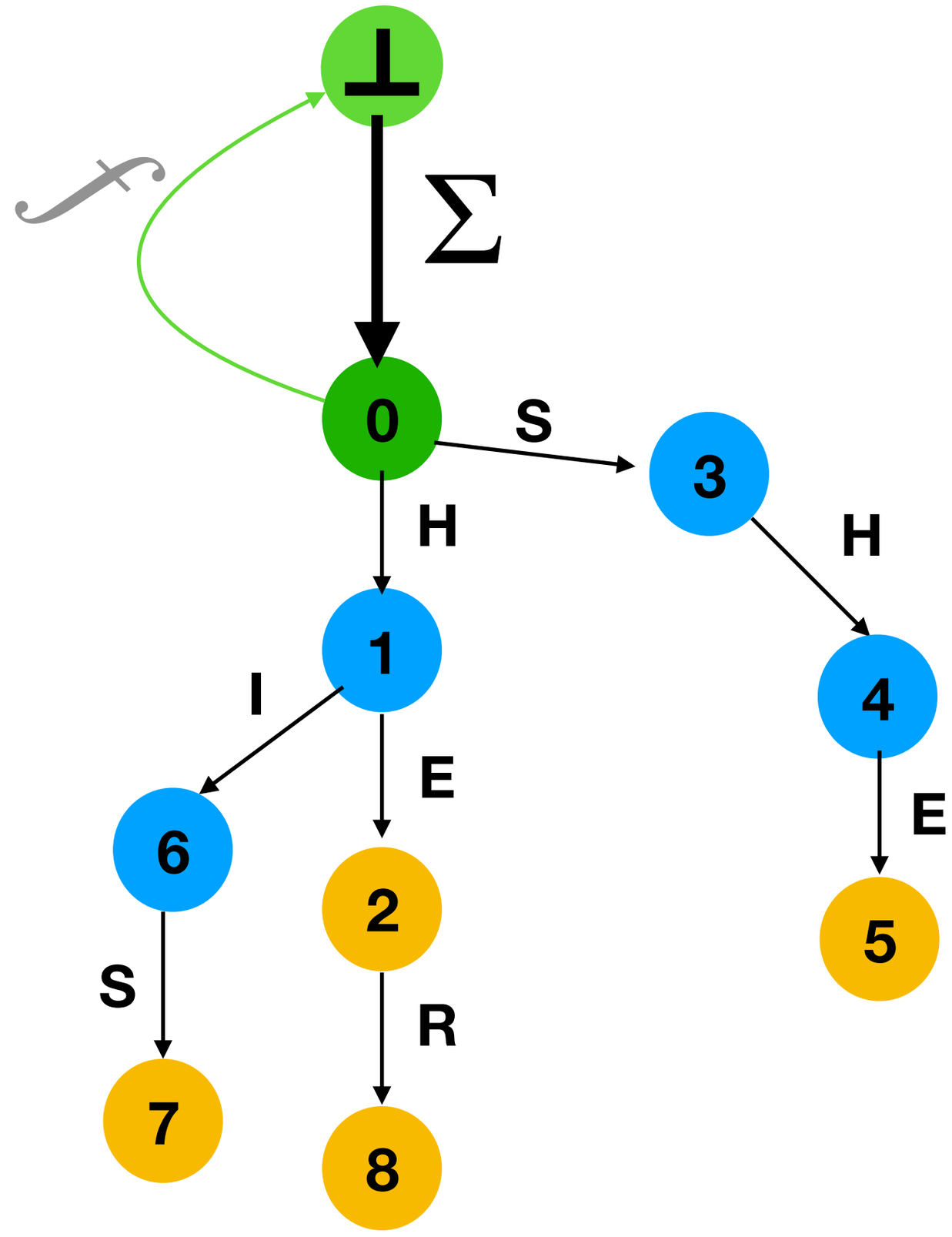
1) **Предок вершины 3 - корень.**



Суффикс-ссылка

Посчитаем суффиксную ссылку вершины 3

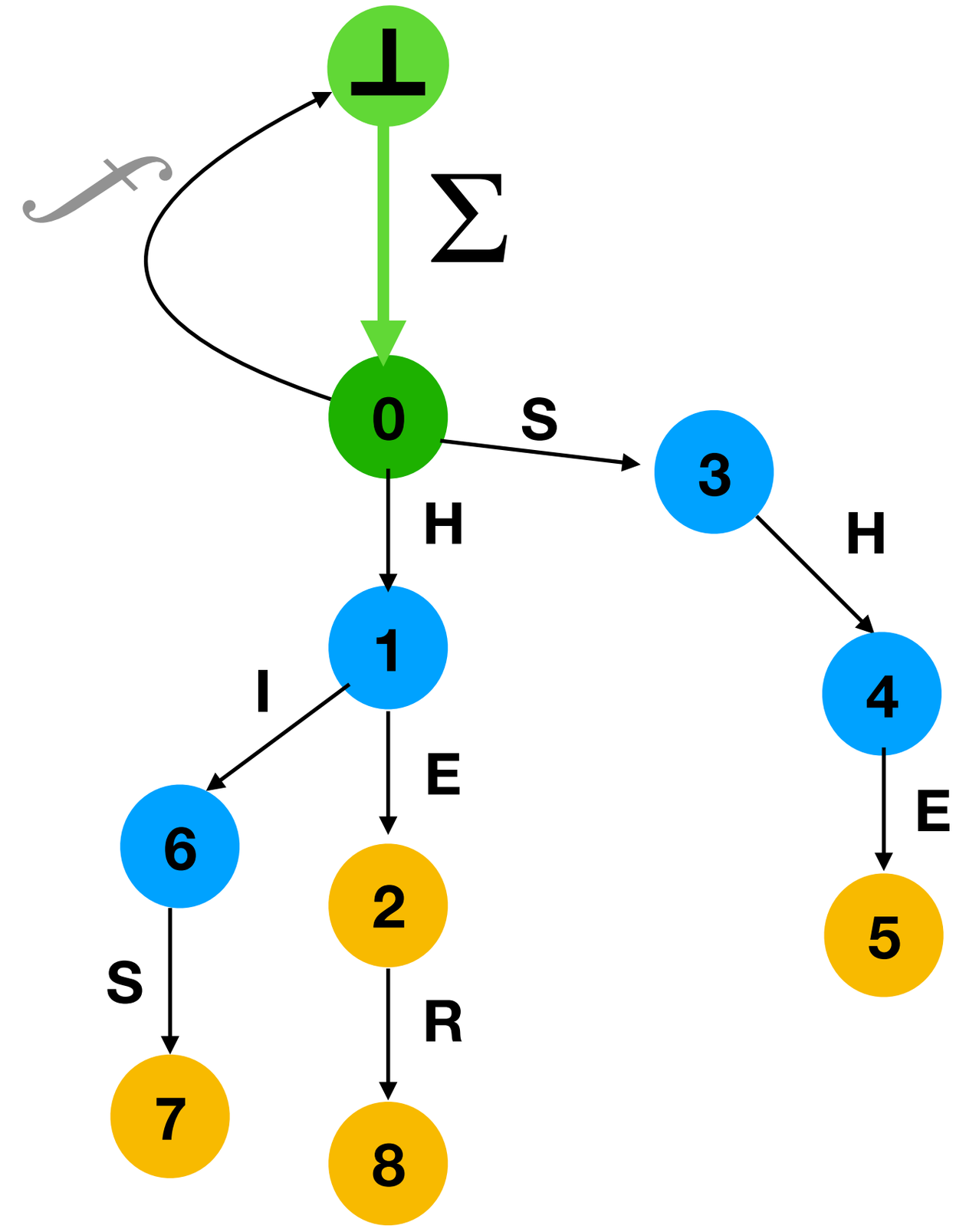
1) **Предок вершины 3 - корень.**
2) **Проходим по его суффиксной ссылке в фиктивную вершину**



Суффикс-ссылка

Посчитаем суффиксную ссылку вершины 3

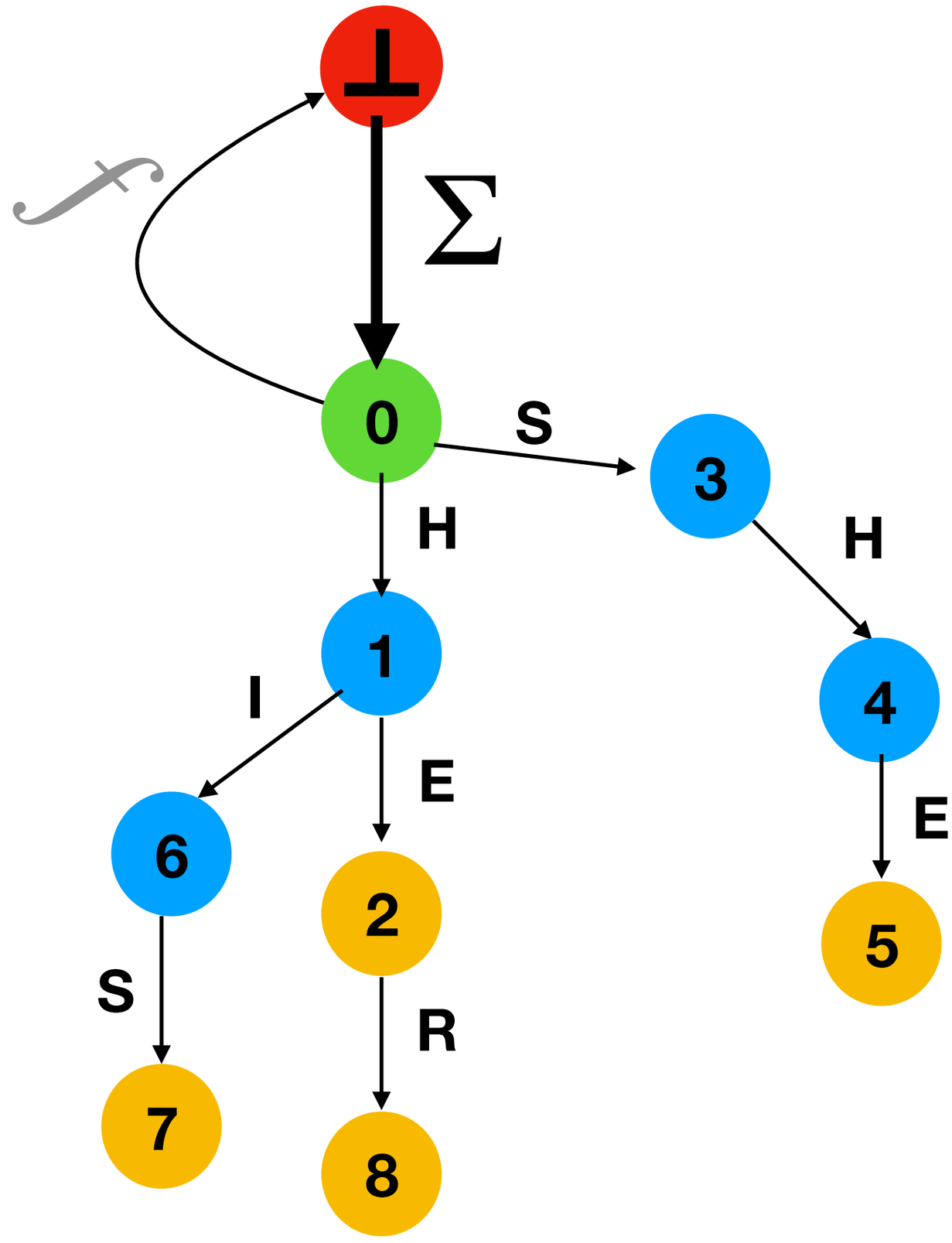
1) **Предок вершины 3 - корень.**
2) **Проходим по его суффиксной ссылке в фиктивную вершину**
3) **Из нее мы можем пройти по любому символу в 0 (корень), в том числе по S**



Суффикс-ссылка

Посчитаем суффиксную ссылку вершины 3

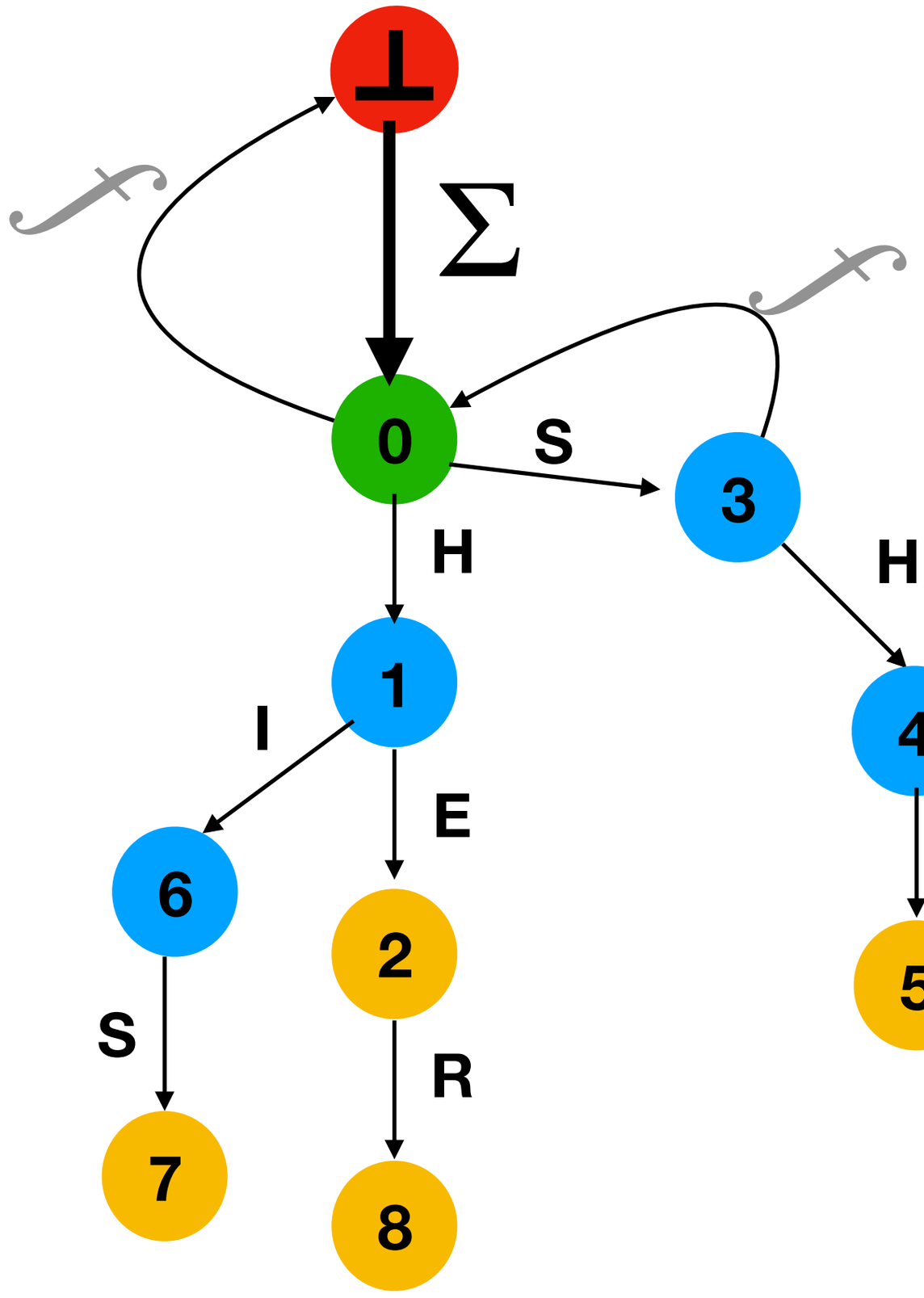
1) **Предок вершины 3 - корень.**
2) **Проходим по его суффиксной ссылке в фиктивную вершину**
3) **Из нее мы можем пройти по любому символу в 0 (корень), в том числе по S**
4) **Значит, суффиксная ссылка 3 ведет в корень**



Суффикс-ссылка

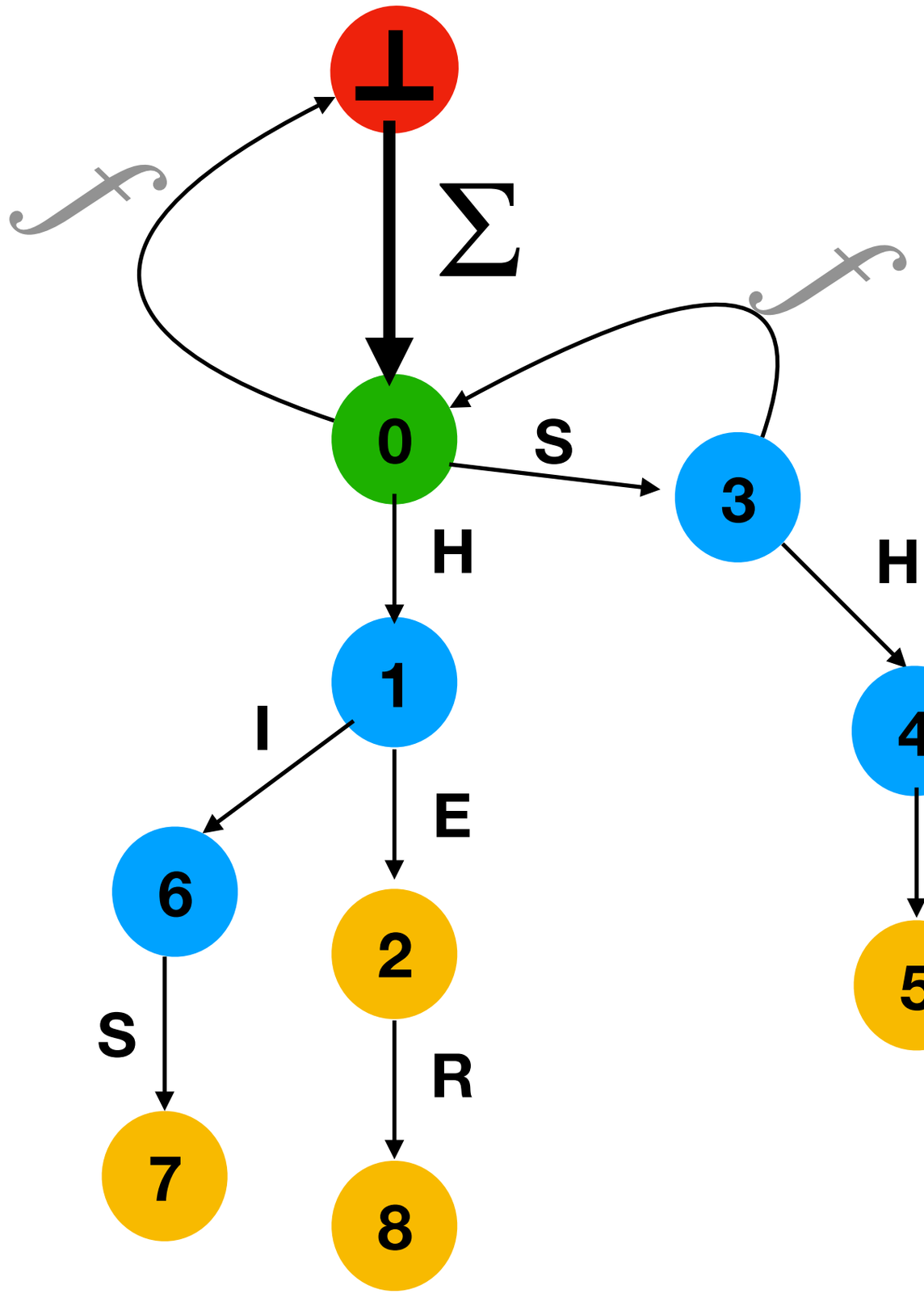
Посчитаем суффиксную ссылку вершины 3

1) **Предок вершины 3 - корень.**
2) **Проходим по его суффиксной ссылке в фиктивную вершину**
3) **Из нее мы можем пройти по любому символу в 0 (корень), в том числе по S**
4) **Значит, суффиксная ссылка из 3 ведет в корень**



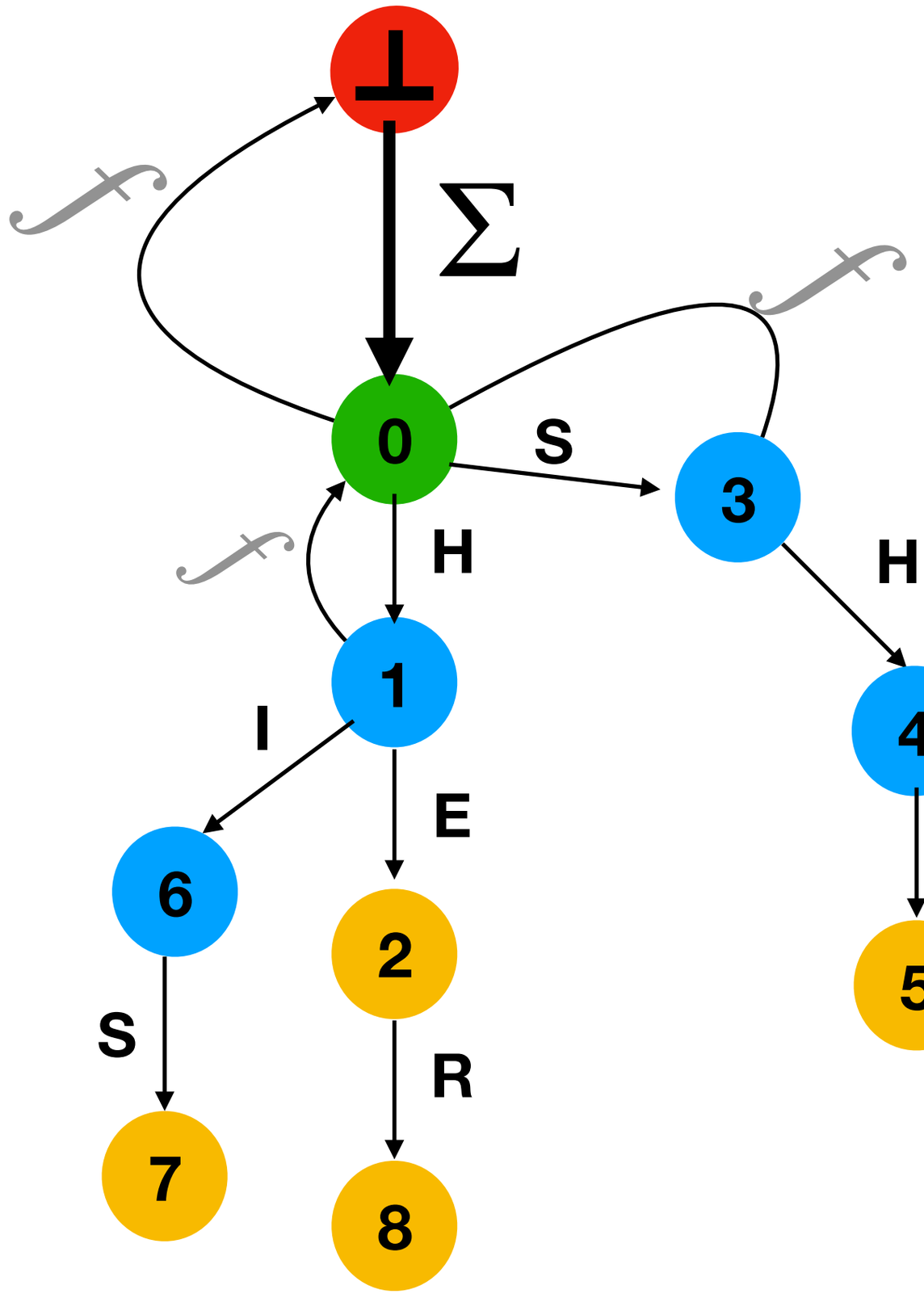
Суффикс-ссылка

Аналогично вычисляем суффиксу ссылку вершины 1



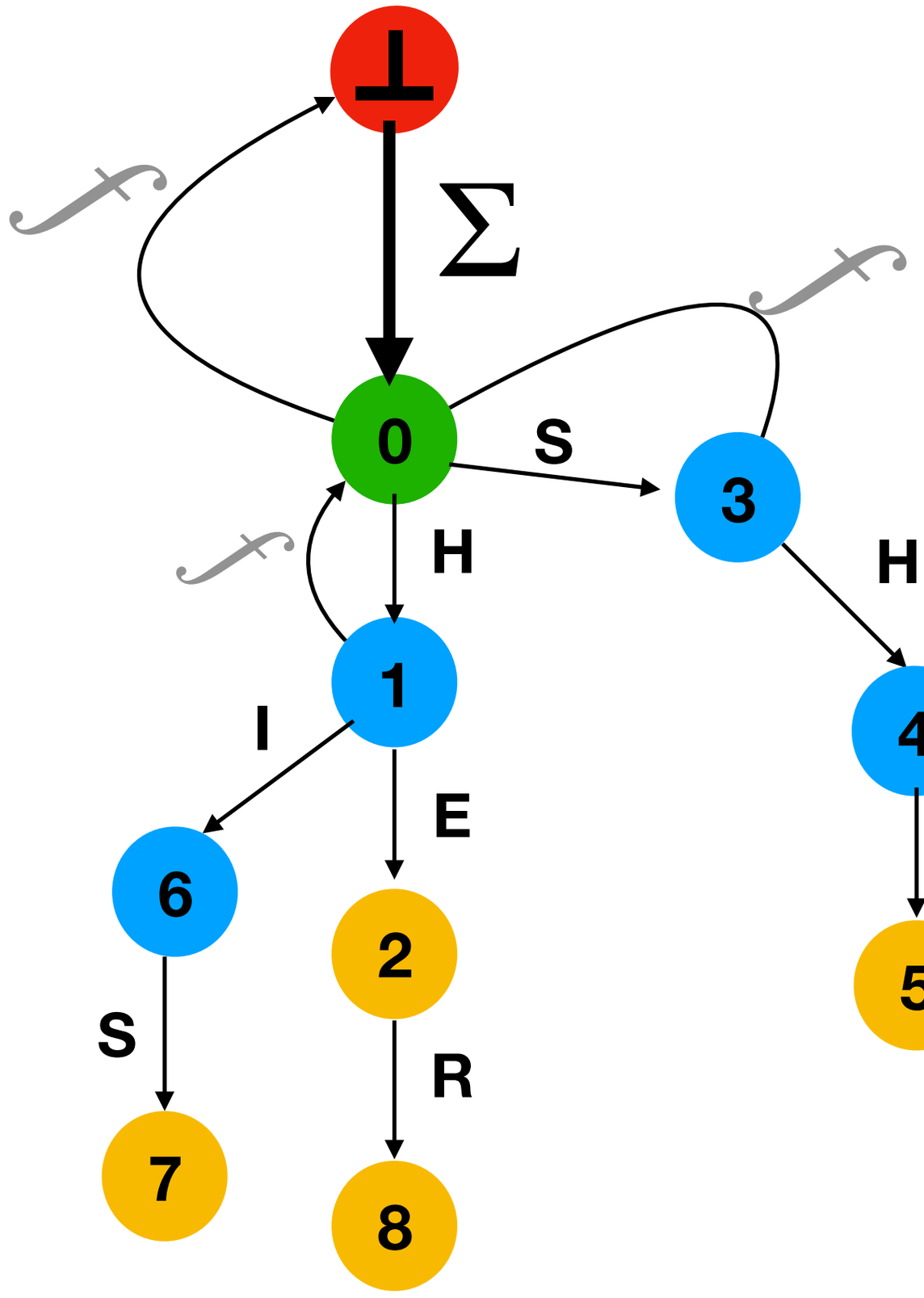
Суффикс-ссылка

Аналогично вычисляем суффиксу ссылку вершины 1



Суффикс-ссылка

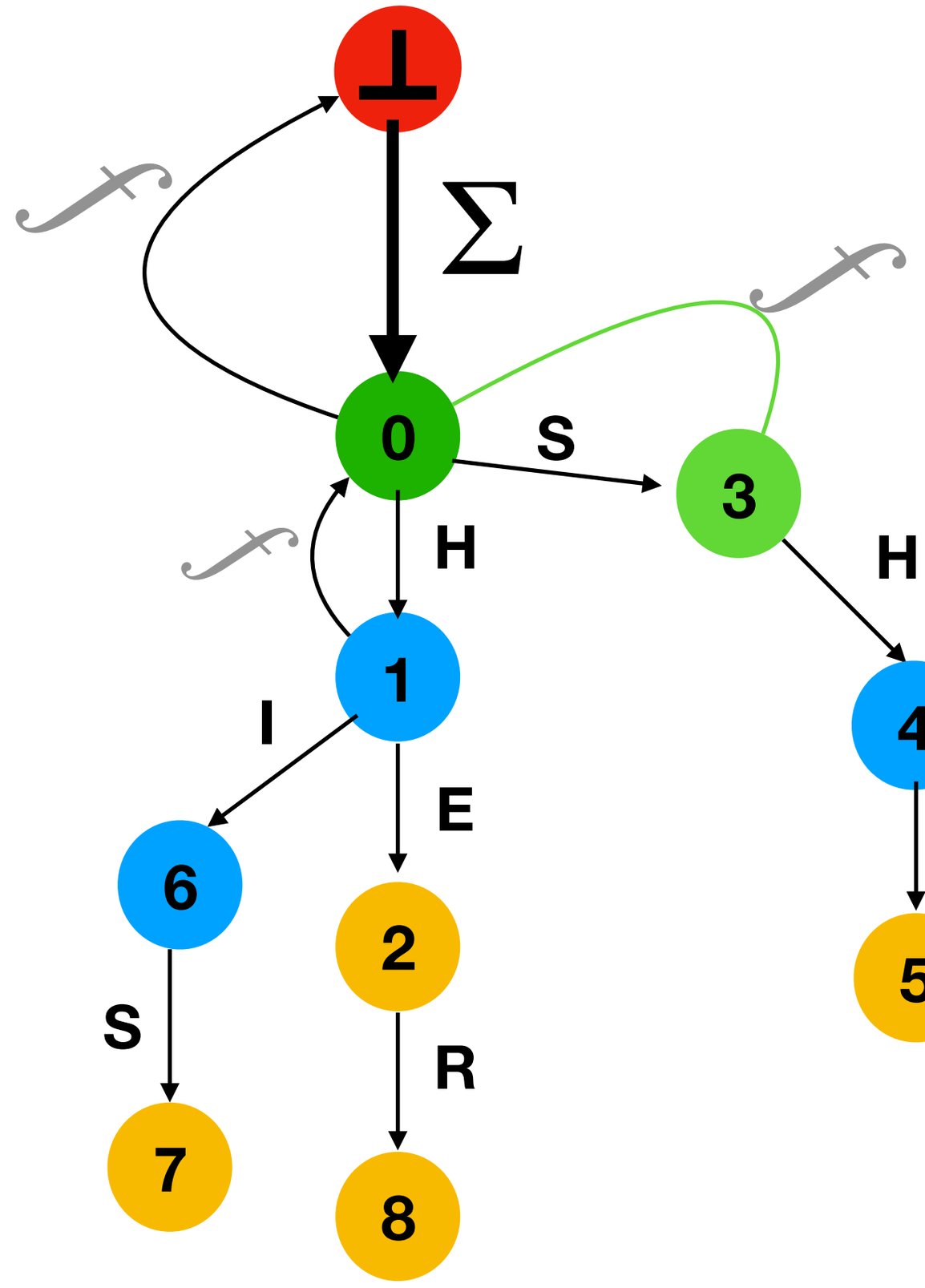
Вычислим суффиксную ссылку вершины 4



Суффикс-ссылка

Вычислим суффиксную ссылку вершины 4

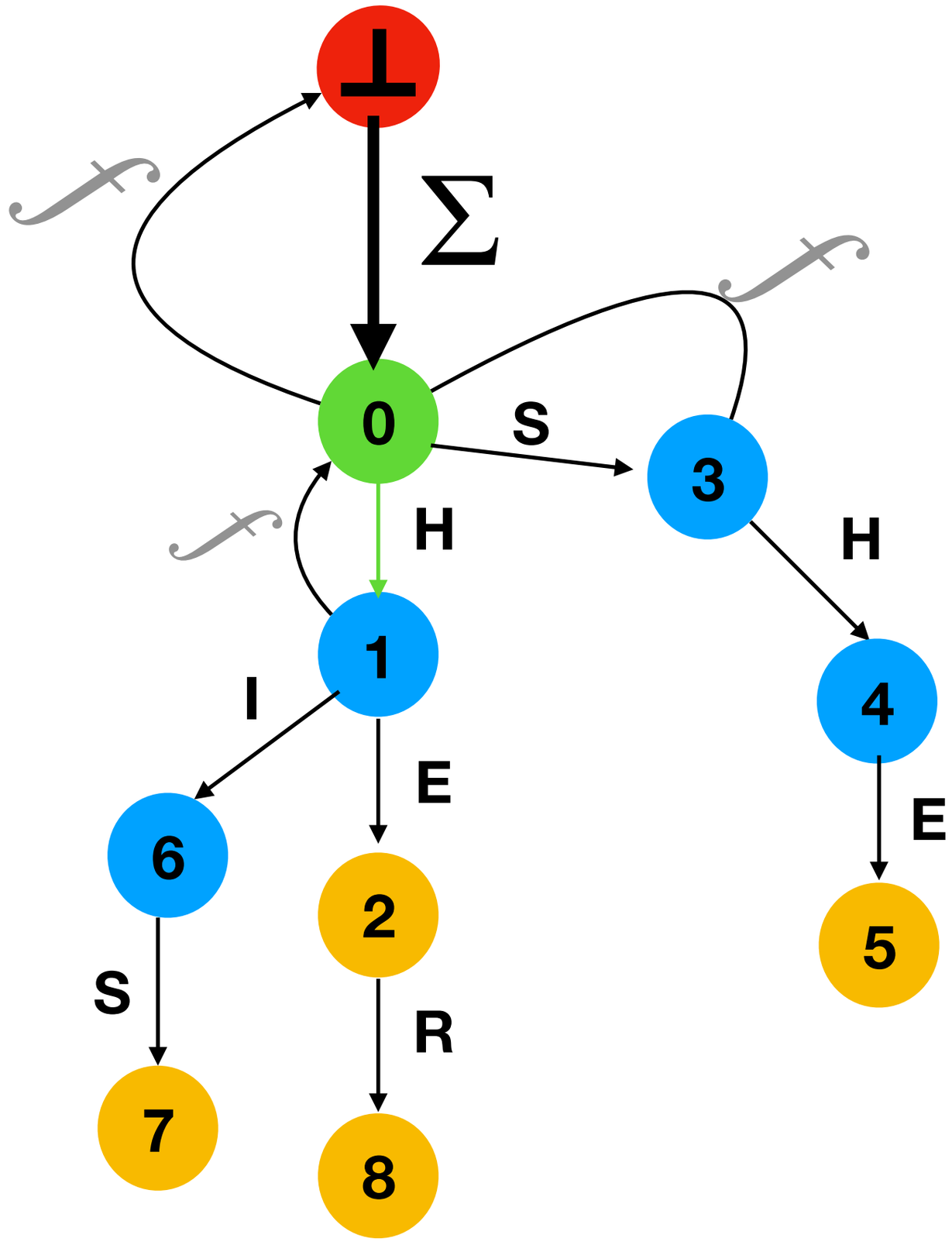
1) **переходим в ее предка - вершину 3**
2) **Переходим по его суффиксной ссылку в корень**



Суффикс-ссылка

Вычислим суффиксную ссылку вершины 4

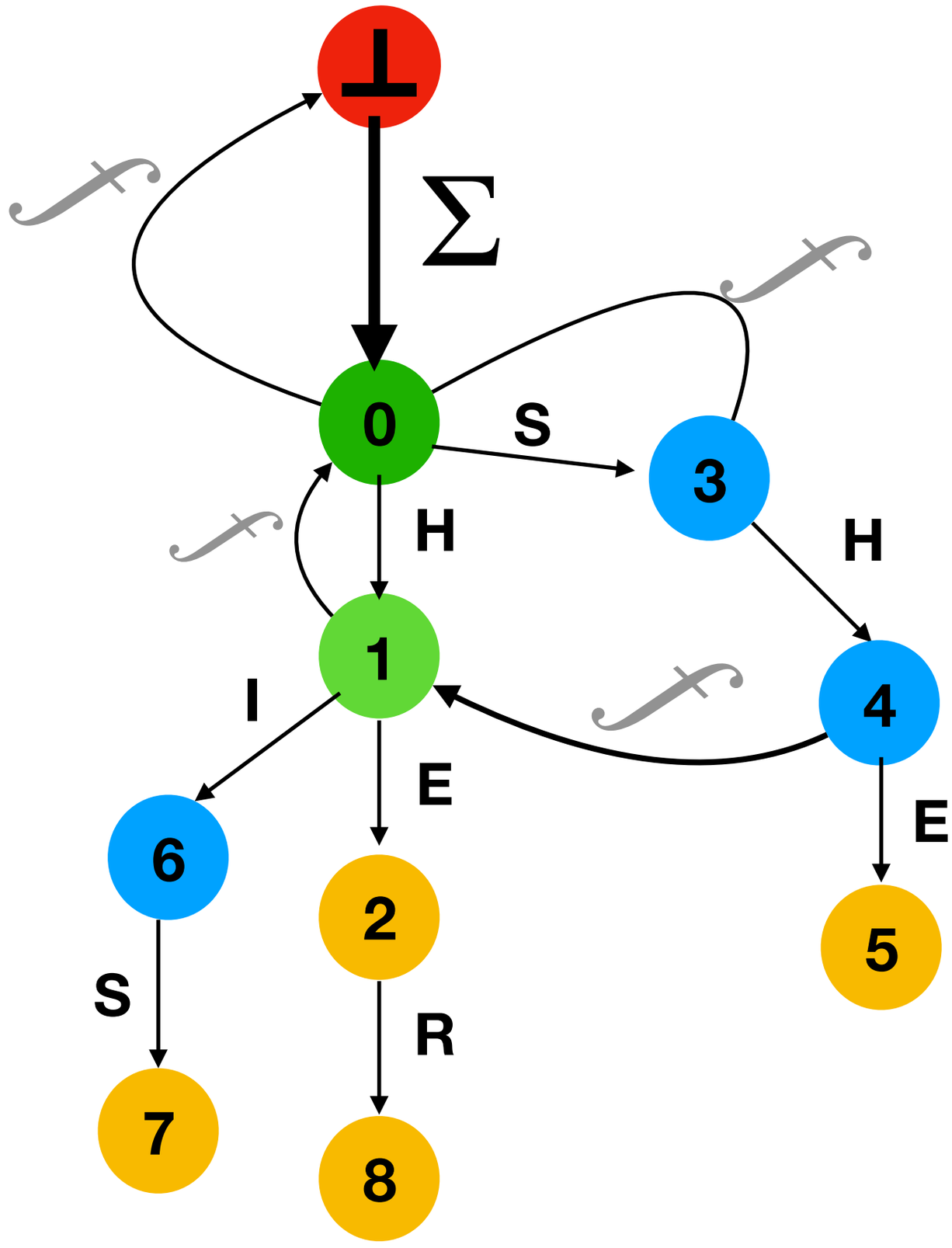
1) **переходим в ее предка - вершину 3**
2) **Переходим по его суффиксной ссылку в корень**
3) **Из корня можно прочитать символ H и попасть в состояние 1**



Суффикс-ссылка

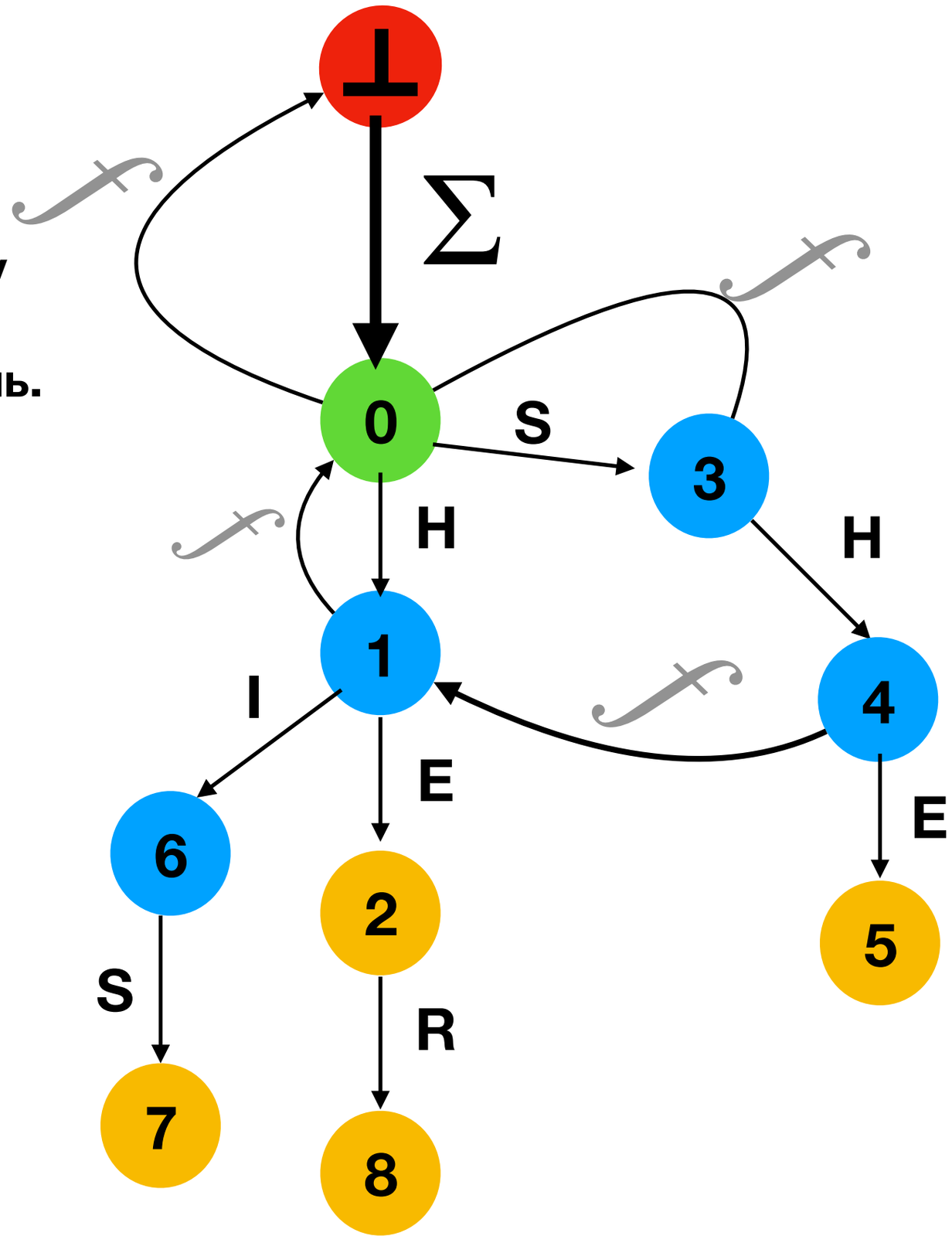
Вычислим суффиксную ссылку вершины 4

1) **переходим в ее предка - вершину 3**
2) **Переходим по его суффиксной ссылку в корень**
3) **Из корня можно прочитать символ H и попасть в состояние 1**
4) **Значит, суффиксная ссылка ведет из 4 в 1**



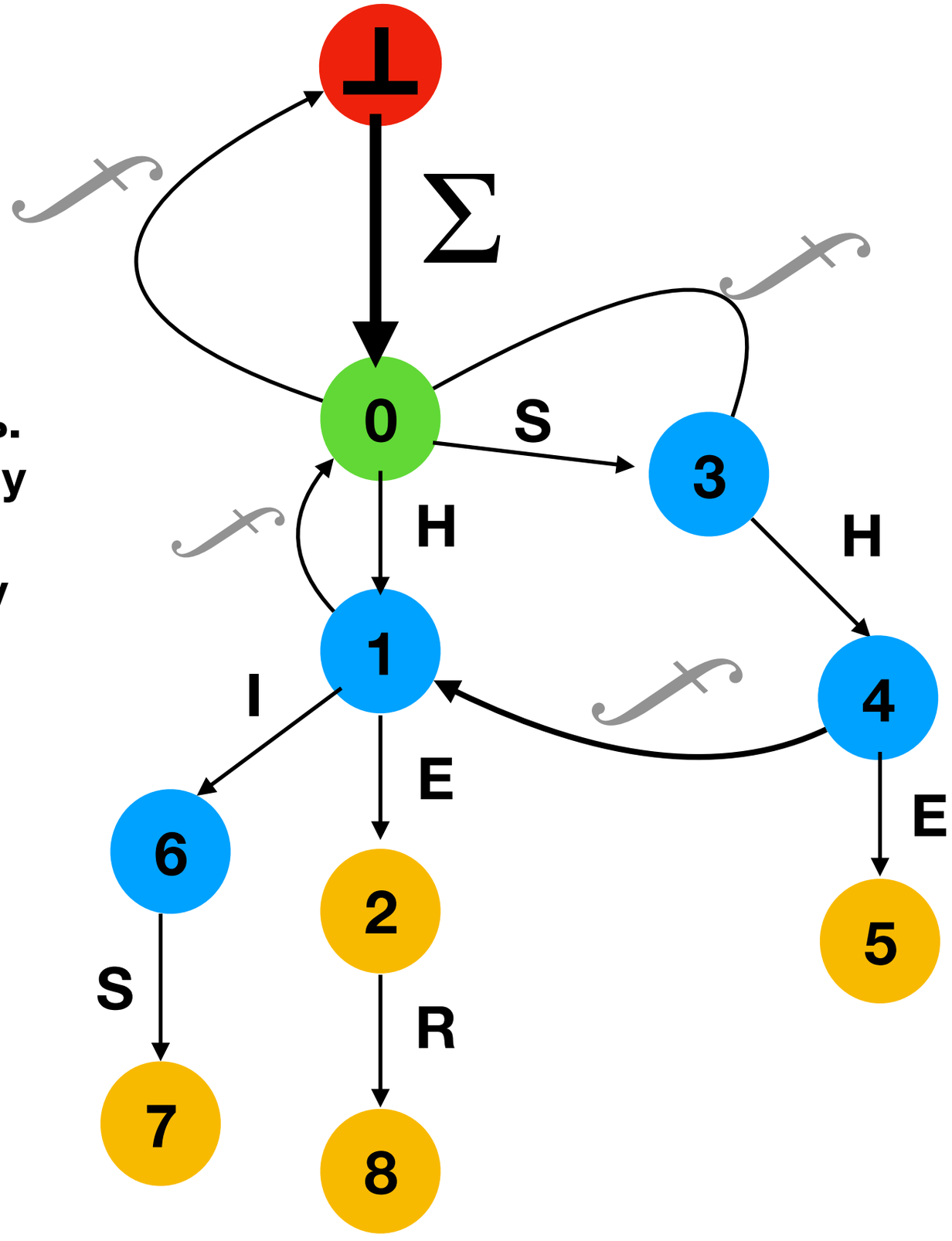
Суффикс-ссылка

Вычислим суффиксную ссылку вершины 6
Здесь мы опять попадем в корень.



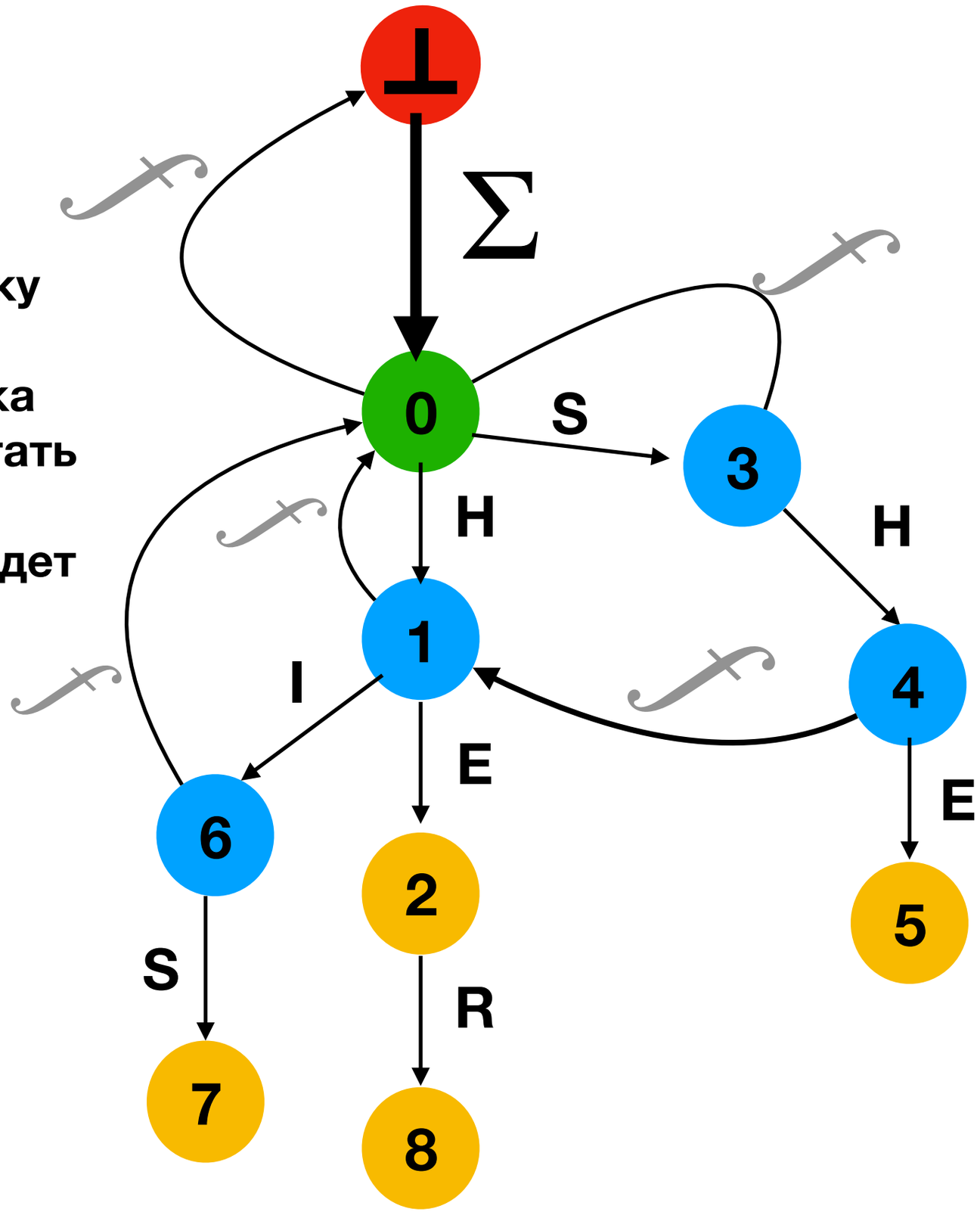
Суффикс-ссылка

Вычислим суффиксную ссылку вершины 6
Здесь мы опять попадем в корень.
Из него нельзя прочитать I, потому по суффиксной ссылке корня переходим в фиктивную вершину
И оттуда переходим в корень.
Значит, суффиксная ссылка вершины 6 ведет в корень



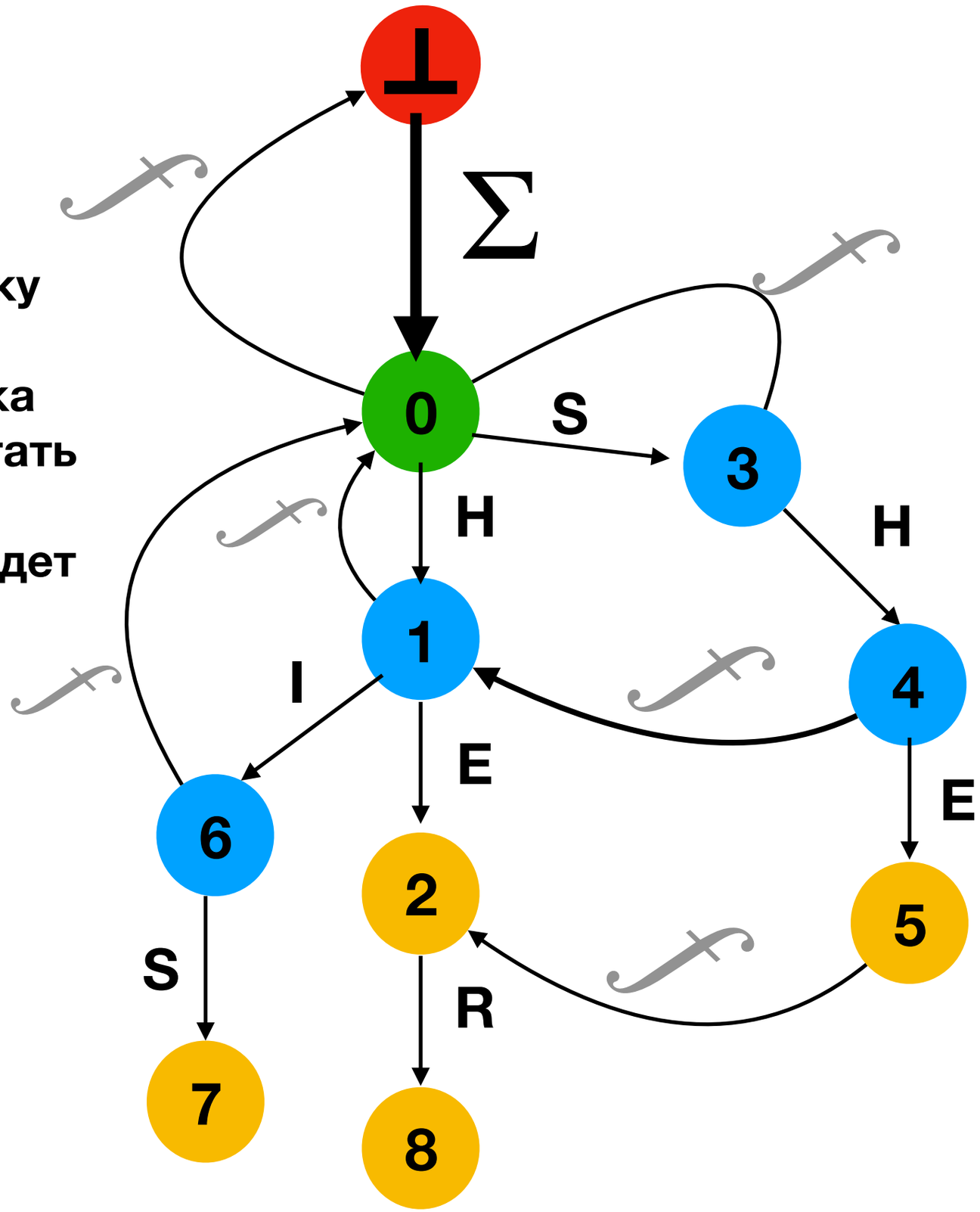
Суффикс-ссылка

Вычислим суффиксную ссылку вершины 5
Суффиксная ссылка ее предка ведет в 1, откуда можно прочитать E и перейти в состояние 2.
Значит, суффиксная ссылка ведет в 2



Суффикс-ссылка

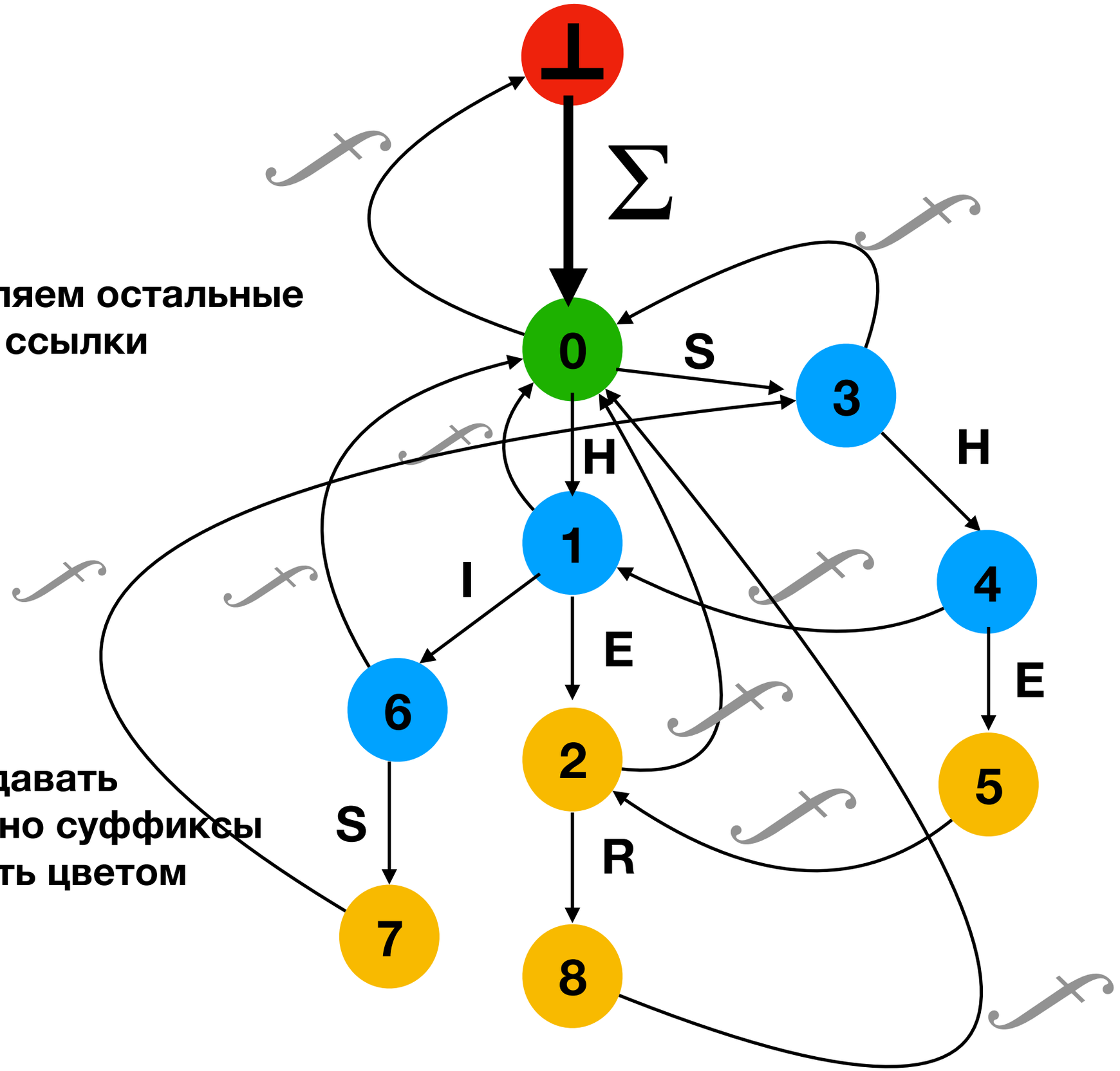
Вычислим суффиксную ссылку вершины 5
Суффиксная ссылка ее предка ведет в 1, откуда можно прочитать E и перейти в состояние 2.
Значит, суффиксная ссылка ведет в 2



Суффикс-ссылка

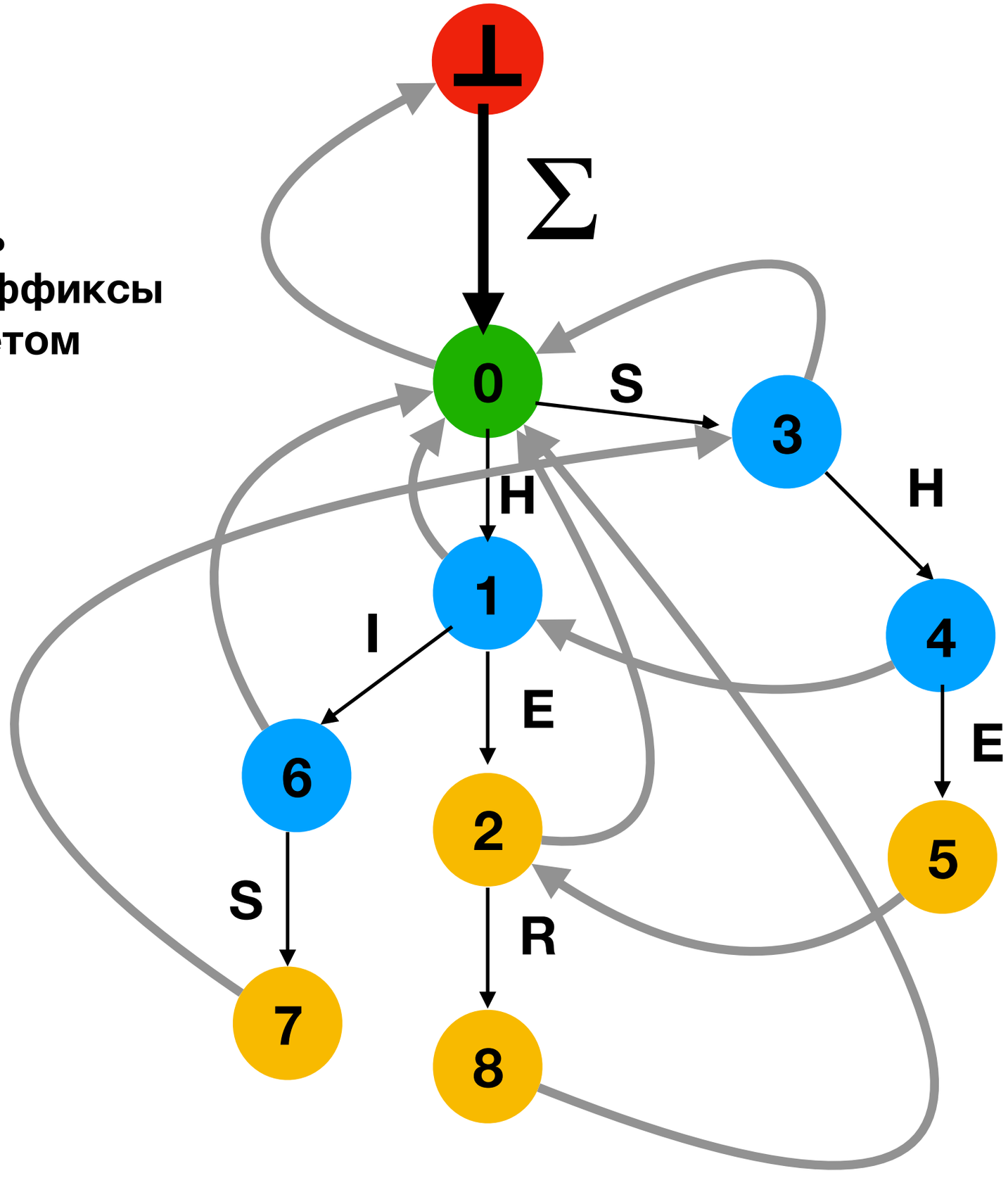
Аналогично вычисляем остальные суффиксы ссылки

Чтобы не создавать нагромождения, можно суффиксы ссылки обозначать цветом



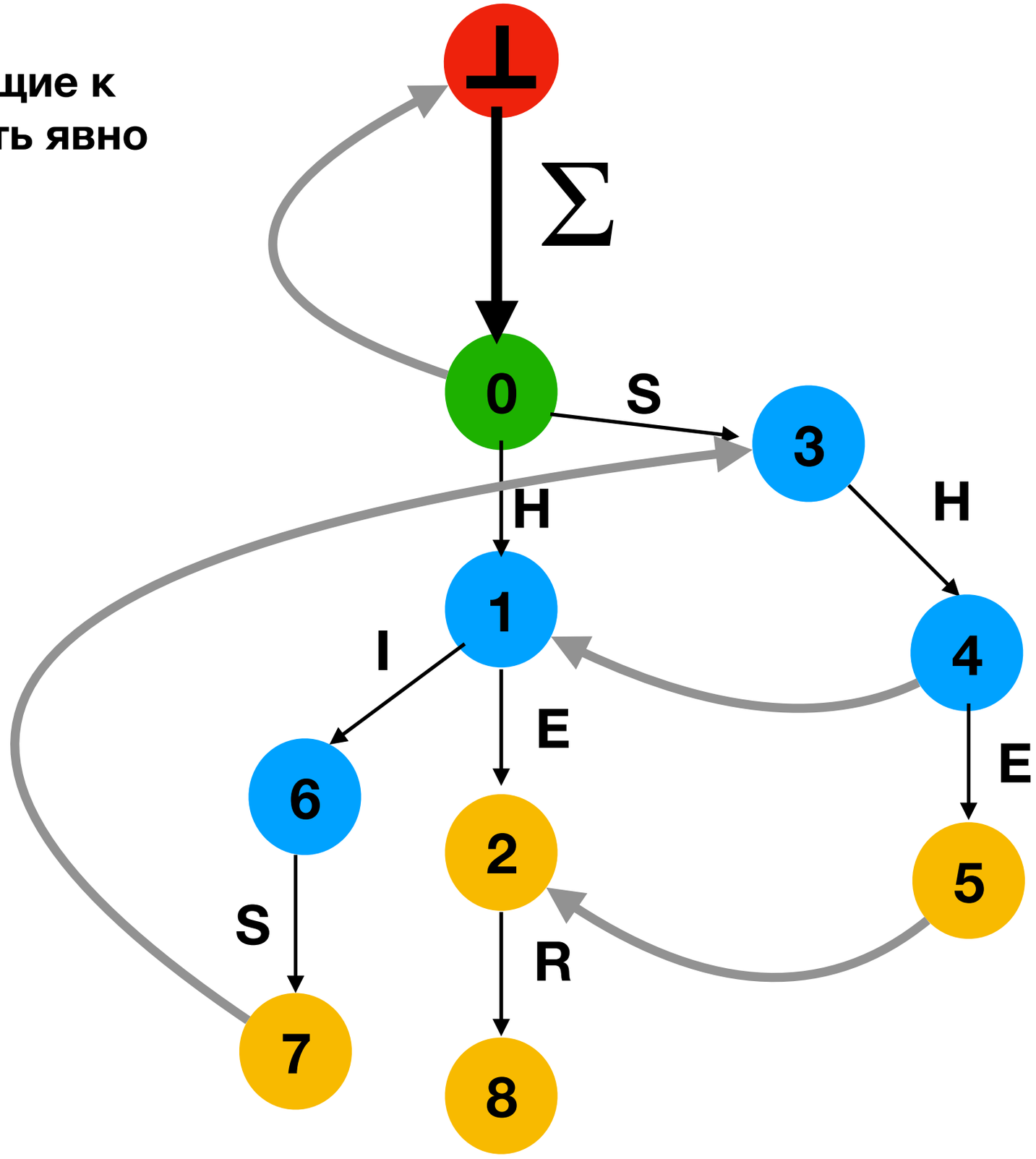
Суффикс-ссылка

Чтобы не создавать нагромождения, можно суффиксы ссылки обозначать цветом

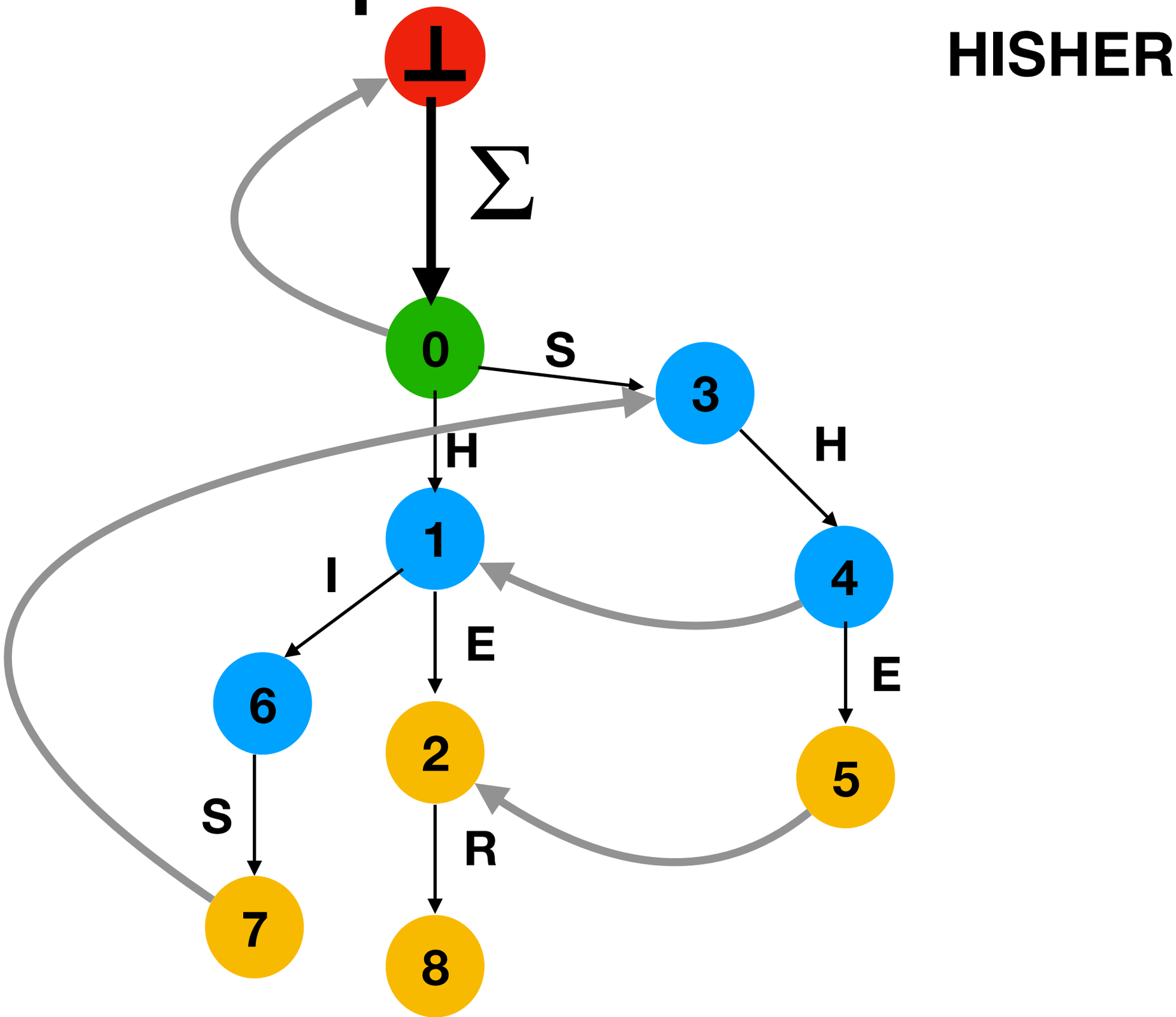


Суффикс-ссылка

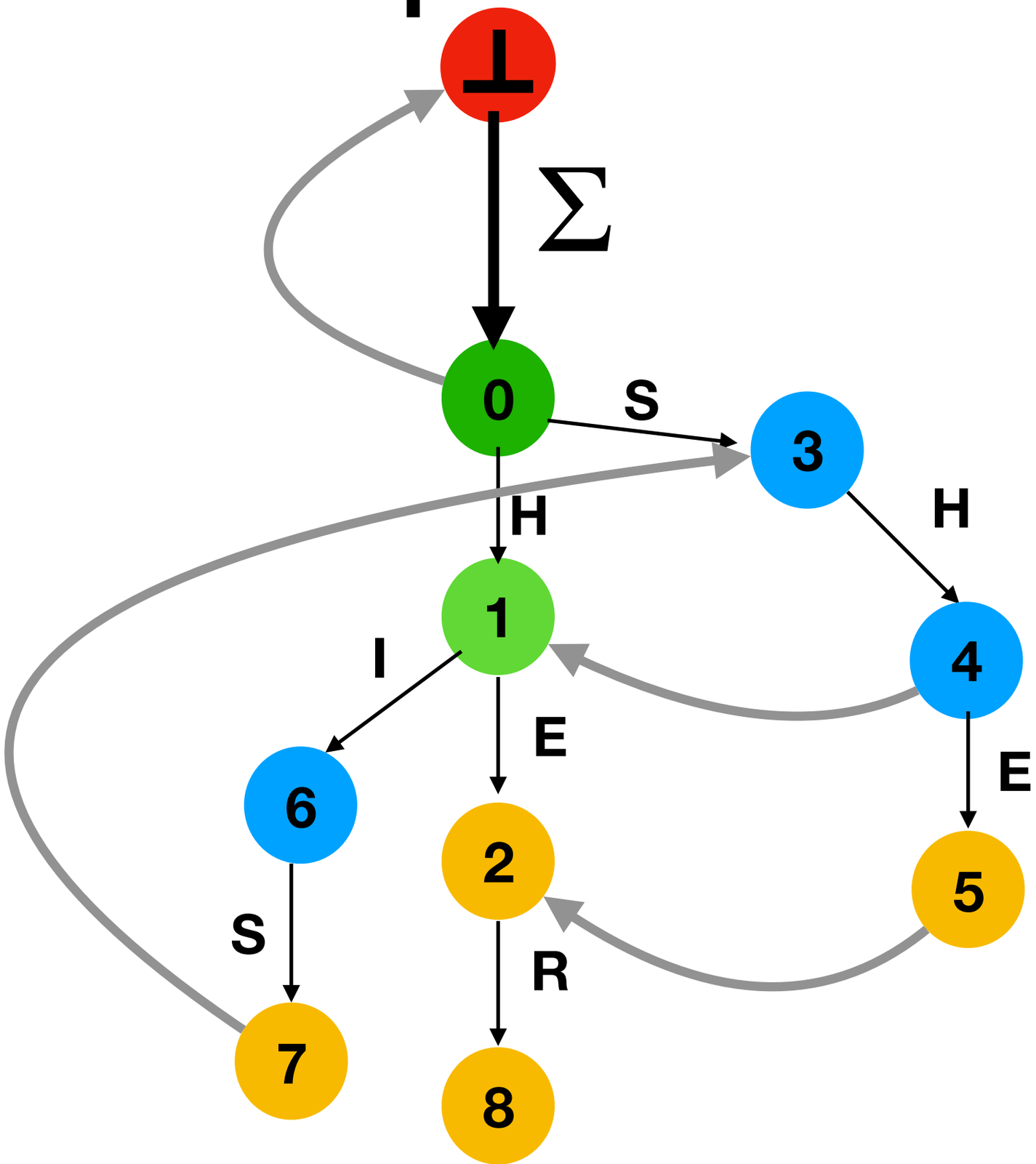
Кроме того, ссылки, ведущие к корню можно не отображать явно



Попробуем поискать в строке

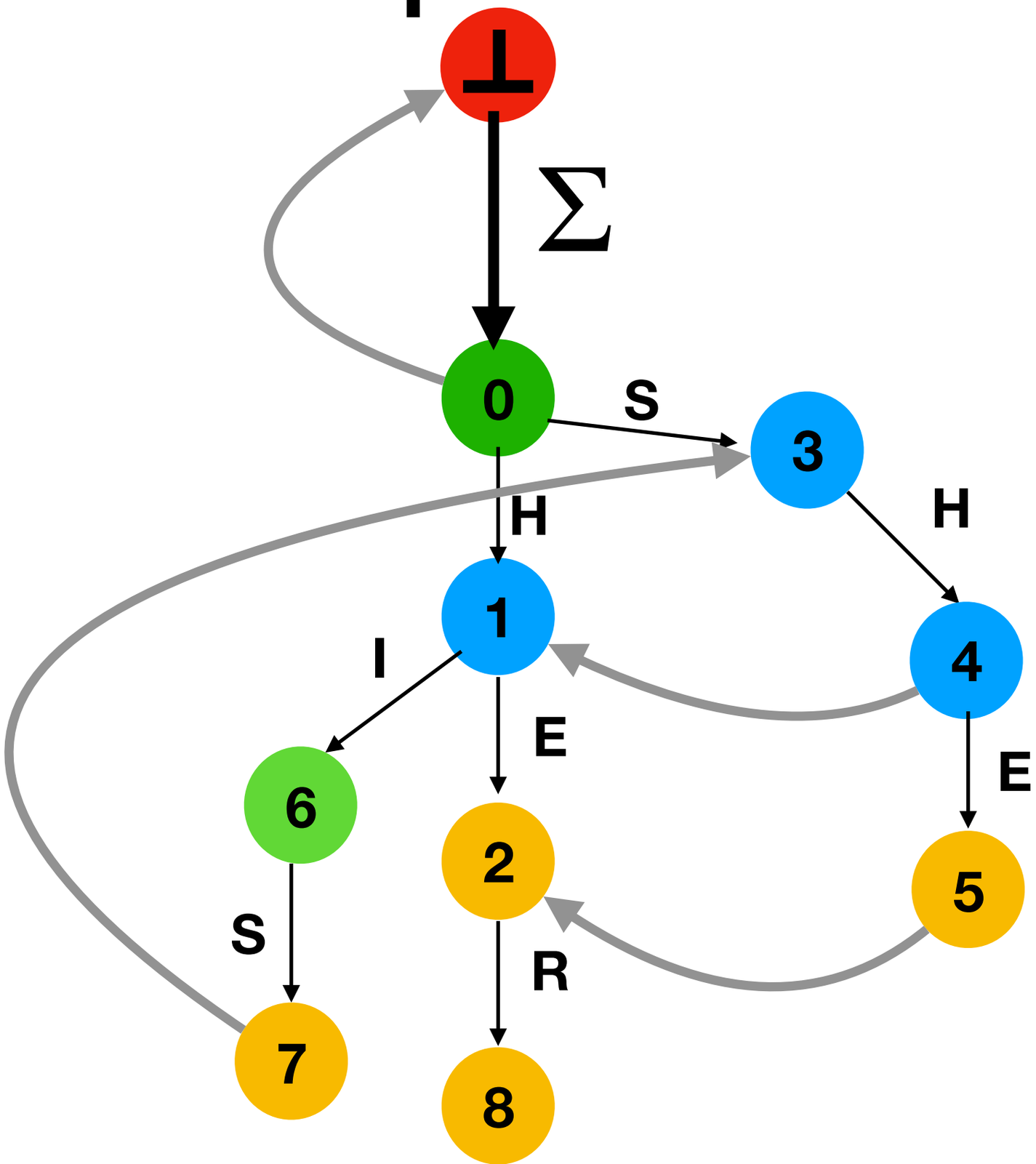


Попробуем поискать в строке



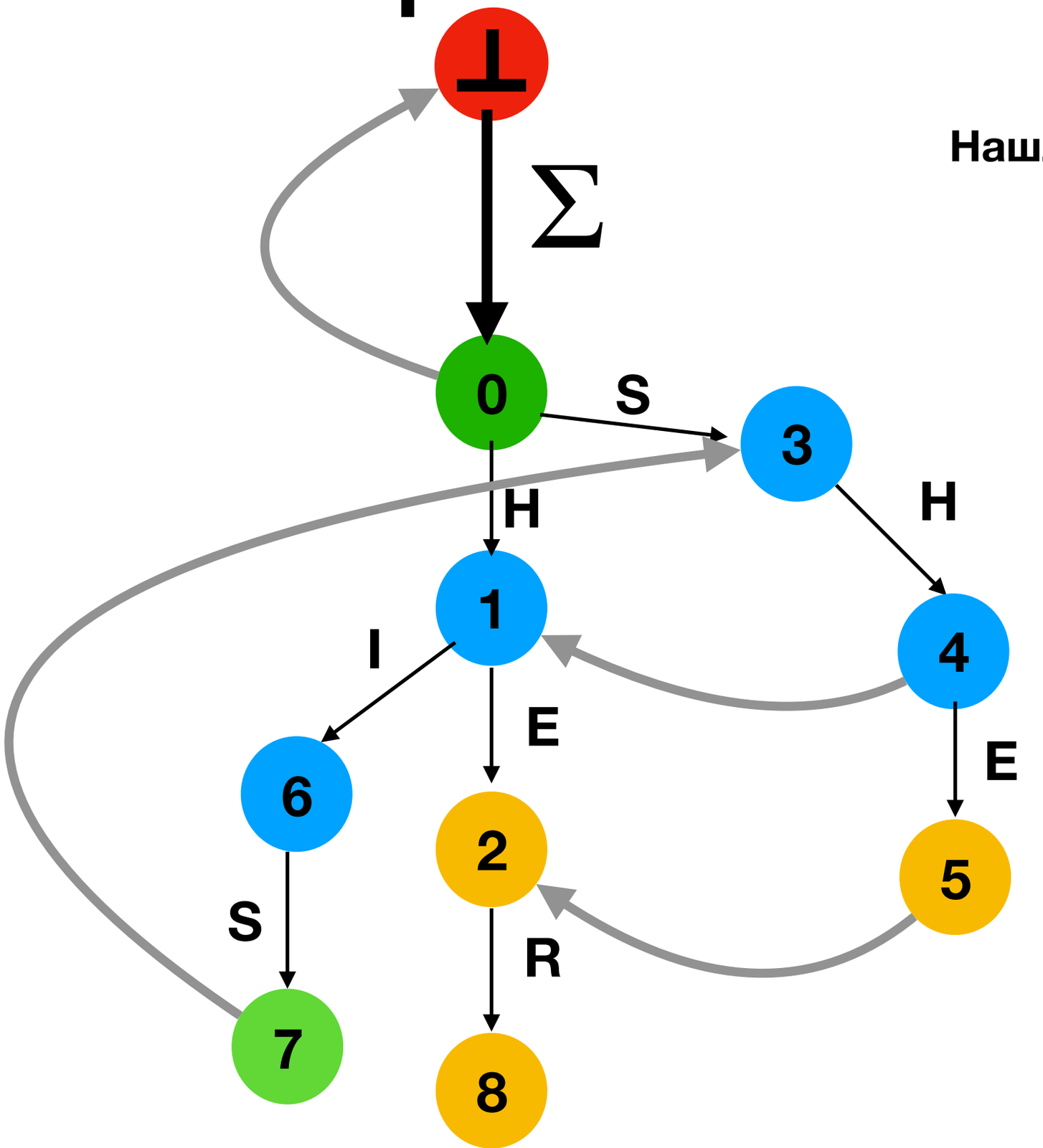
HISHER

Попробуем поискать в строке



HISHER

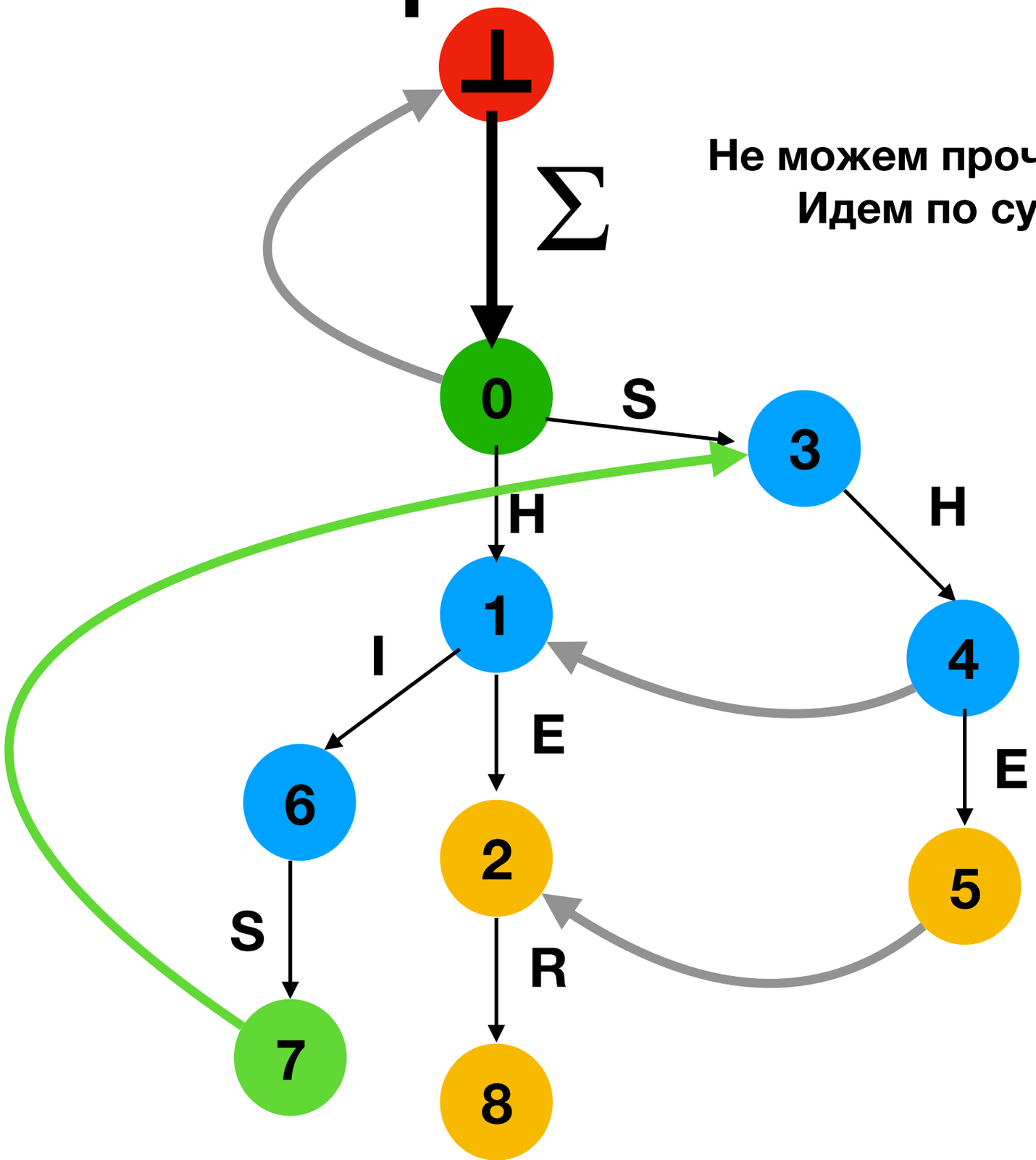
Попробуем поискать в строке



HISHER

Нашли HIS

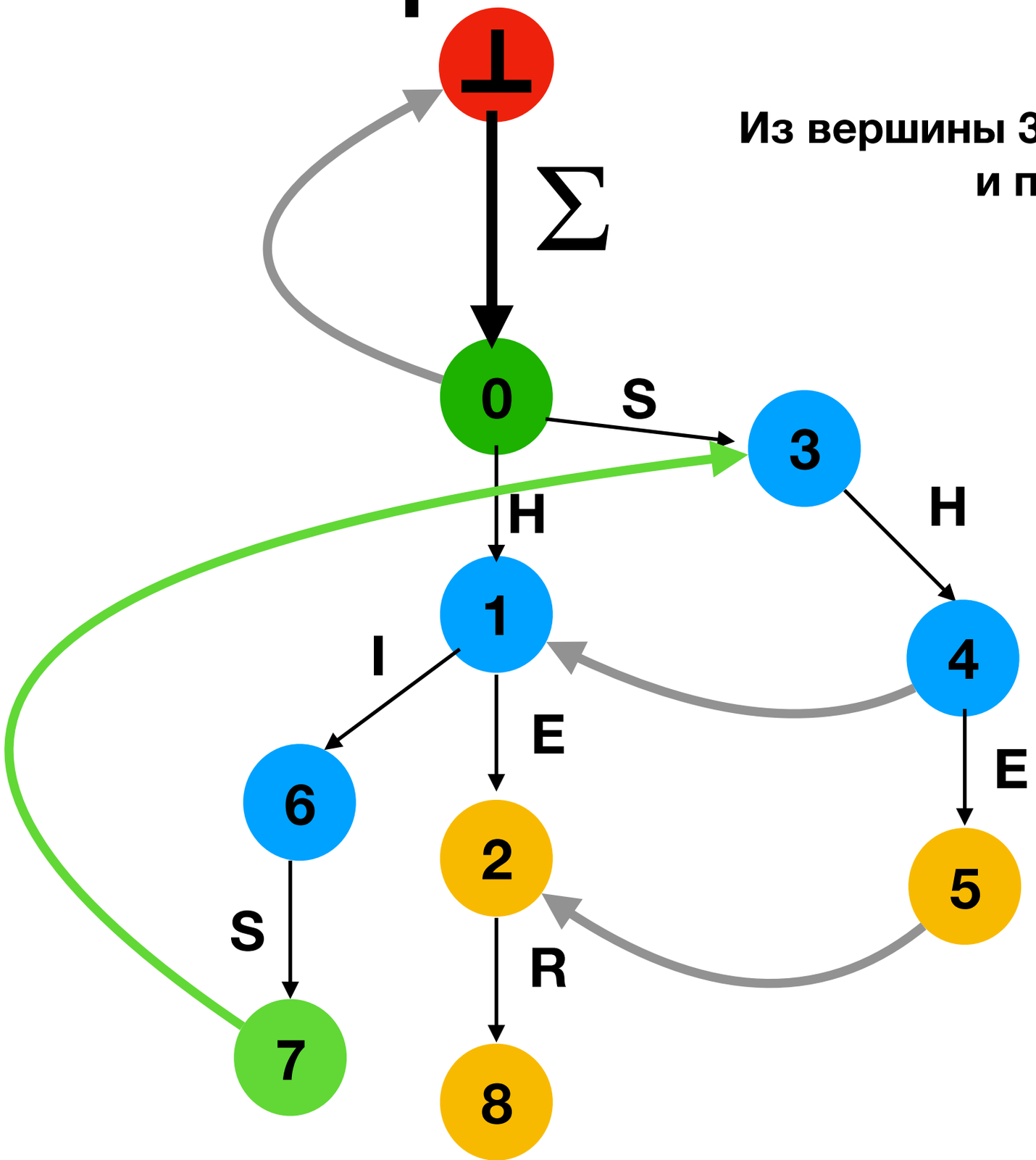
Попробуем поискать в строке



HISHER

Не можем прочитать H из вершины 7.
Идем по суффиксной ссылке

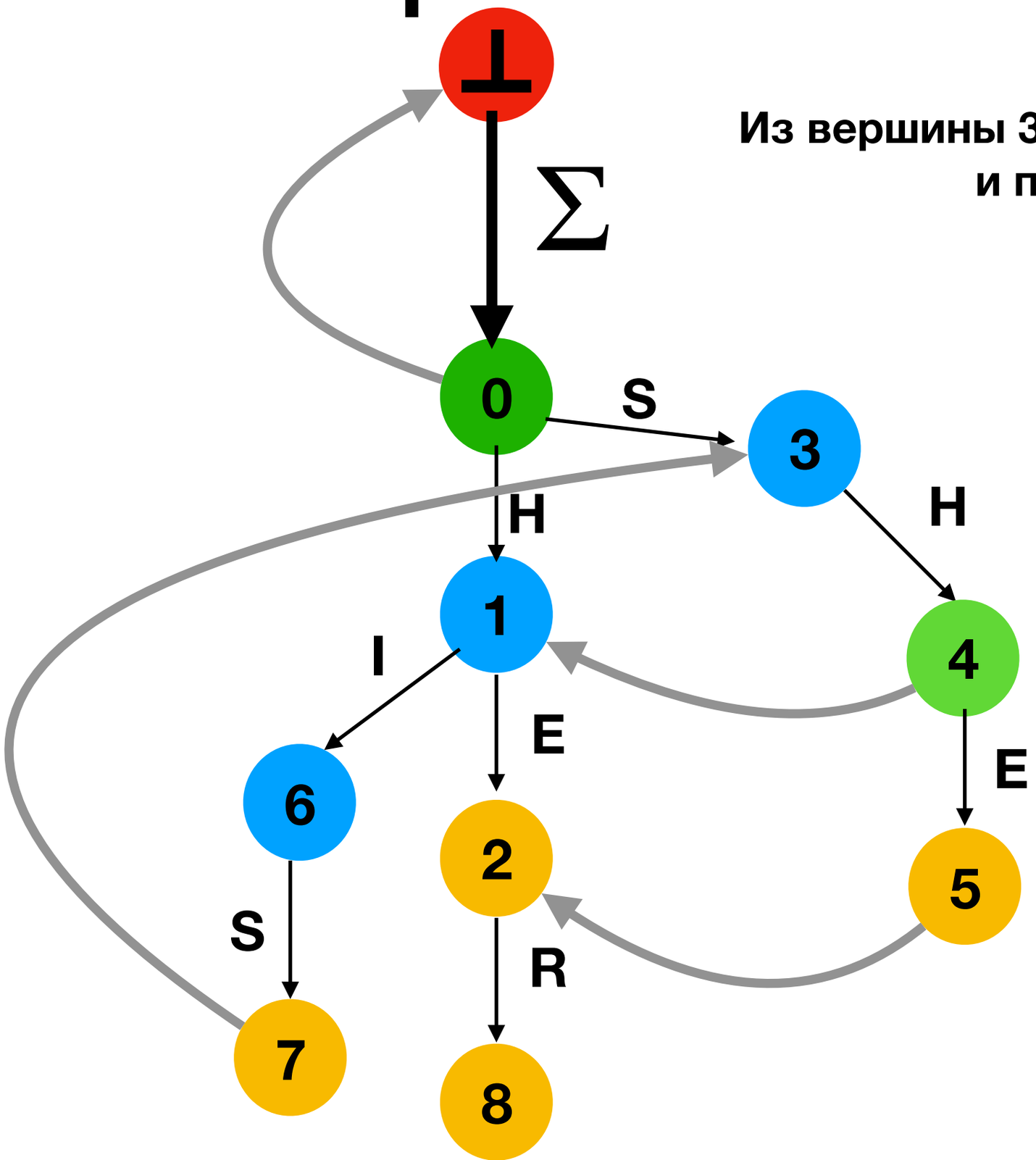
Попробуем поискать в строке



HISHER

Из вершины 3 можем прочитать H и попасть в 4

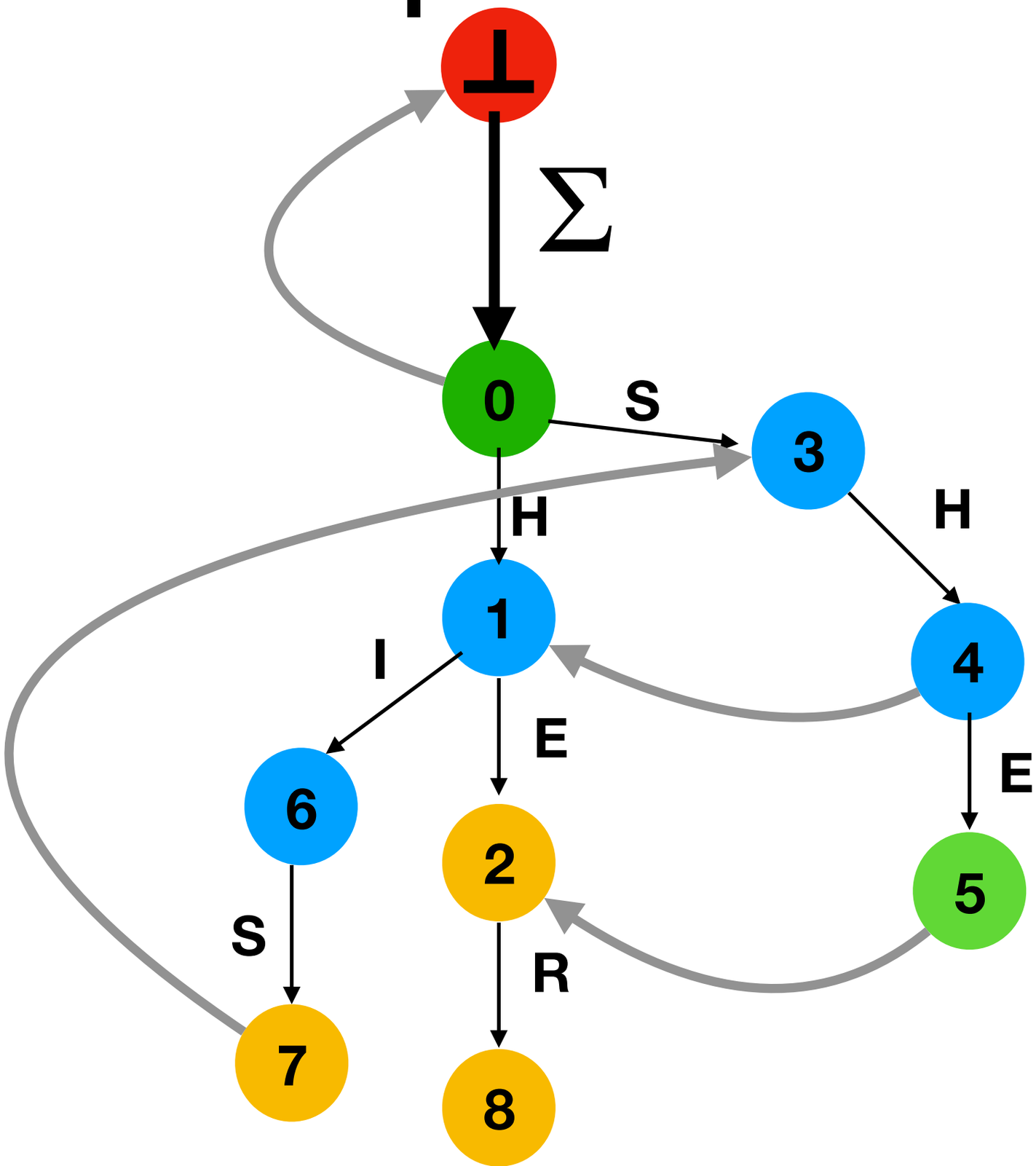
Попробуем поискать в строке



HISHER

Из вершины 3 можем прочитать H и попасть в 4

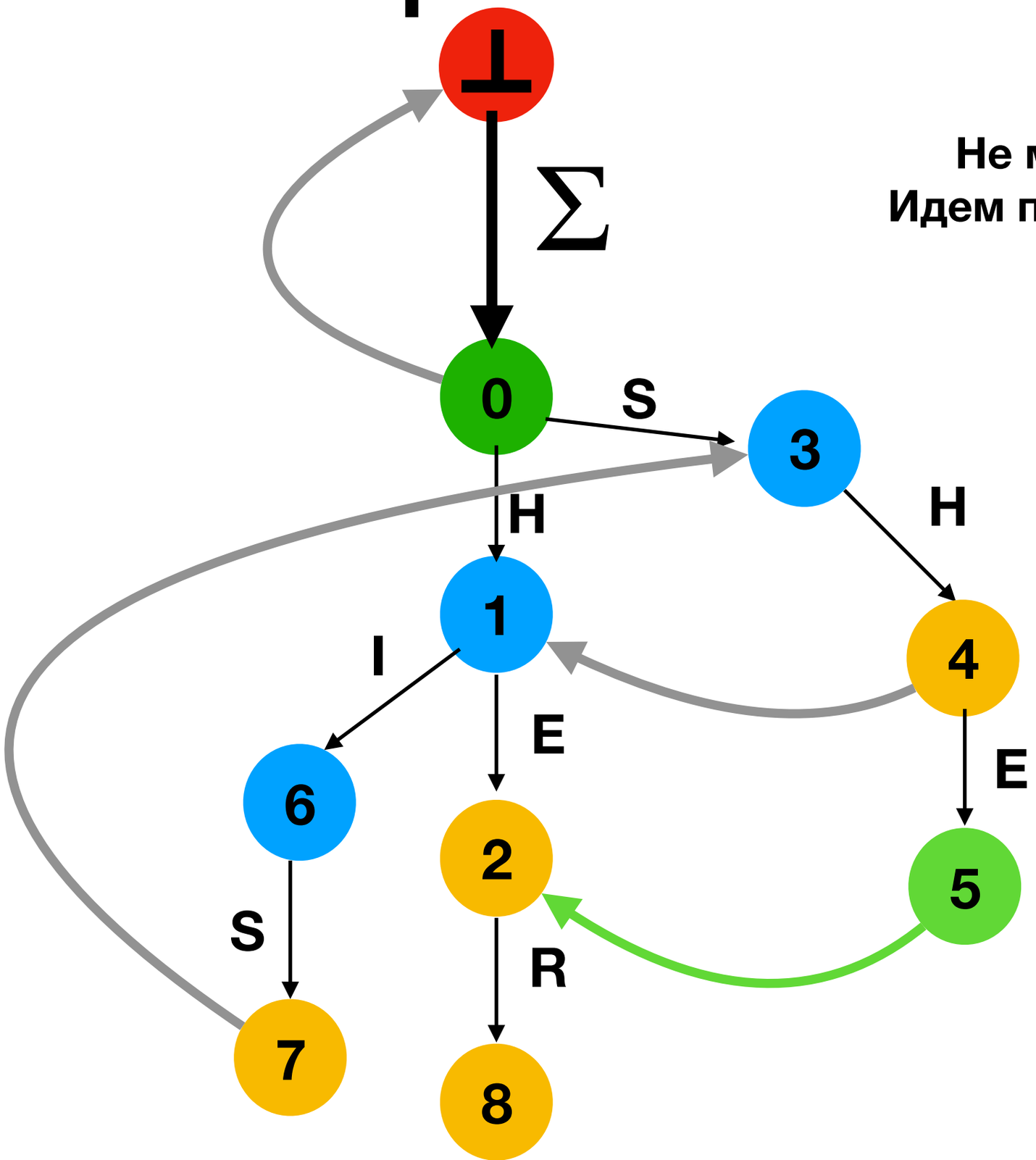
Попробуем поискать в строке



HISHER

Нашли SHE

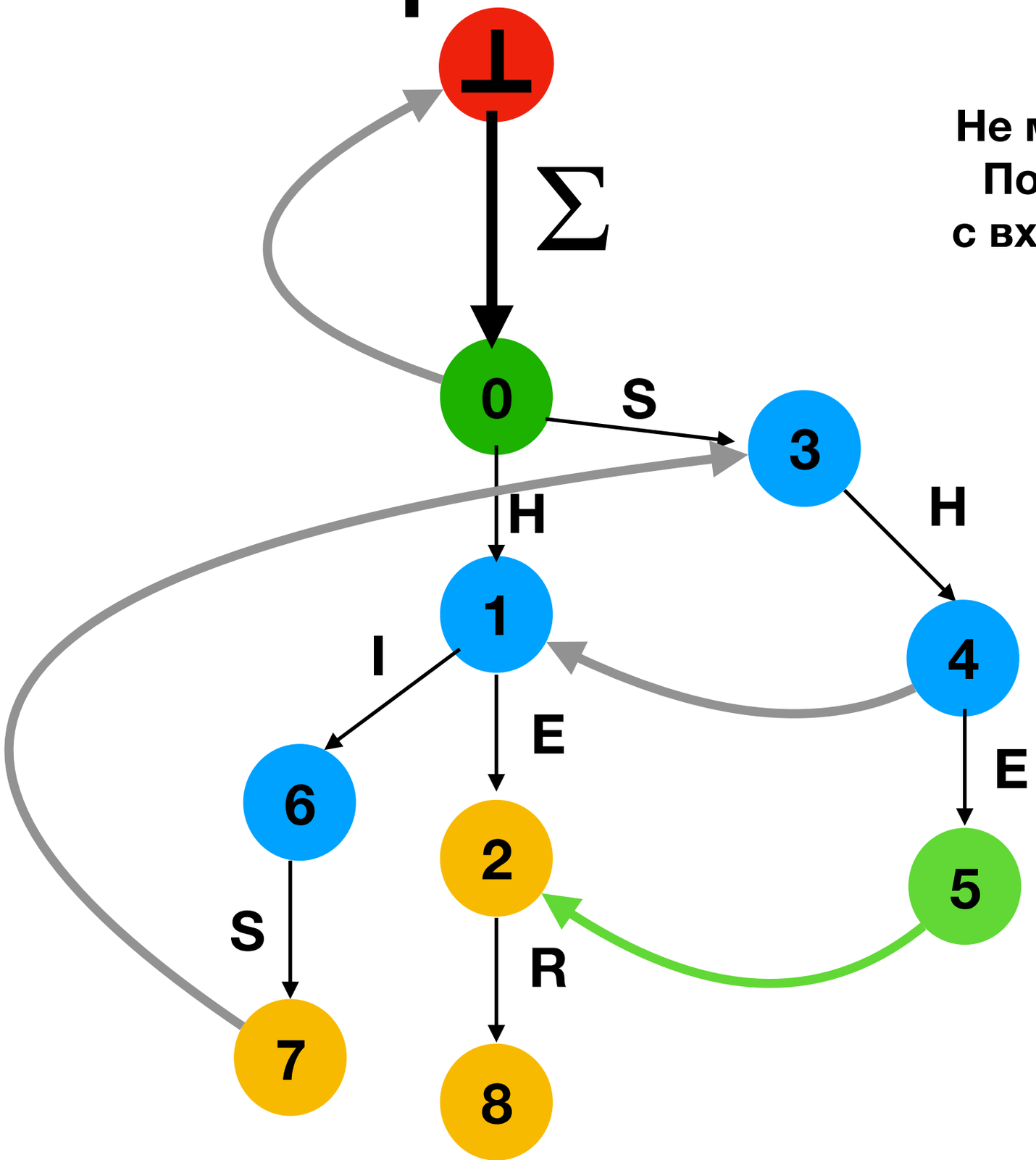
Попробуем поискать в строке



HISHER

Не можем прочитать R.
Идем по суффиксной ссылке

Попробуем поискать в строке

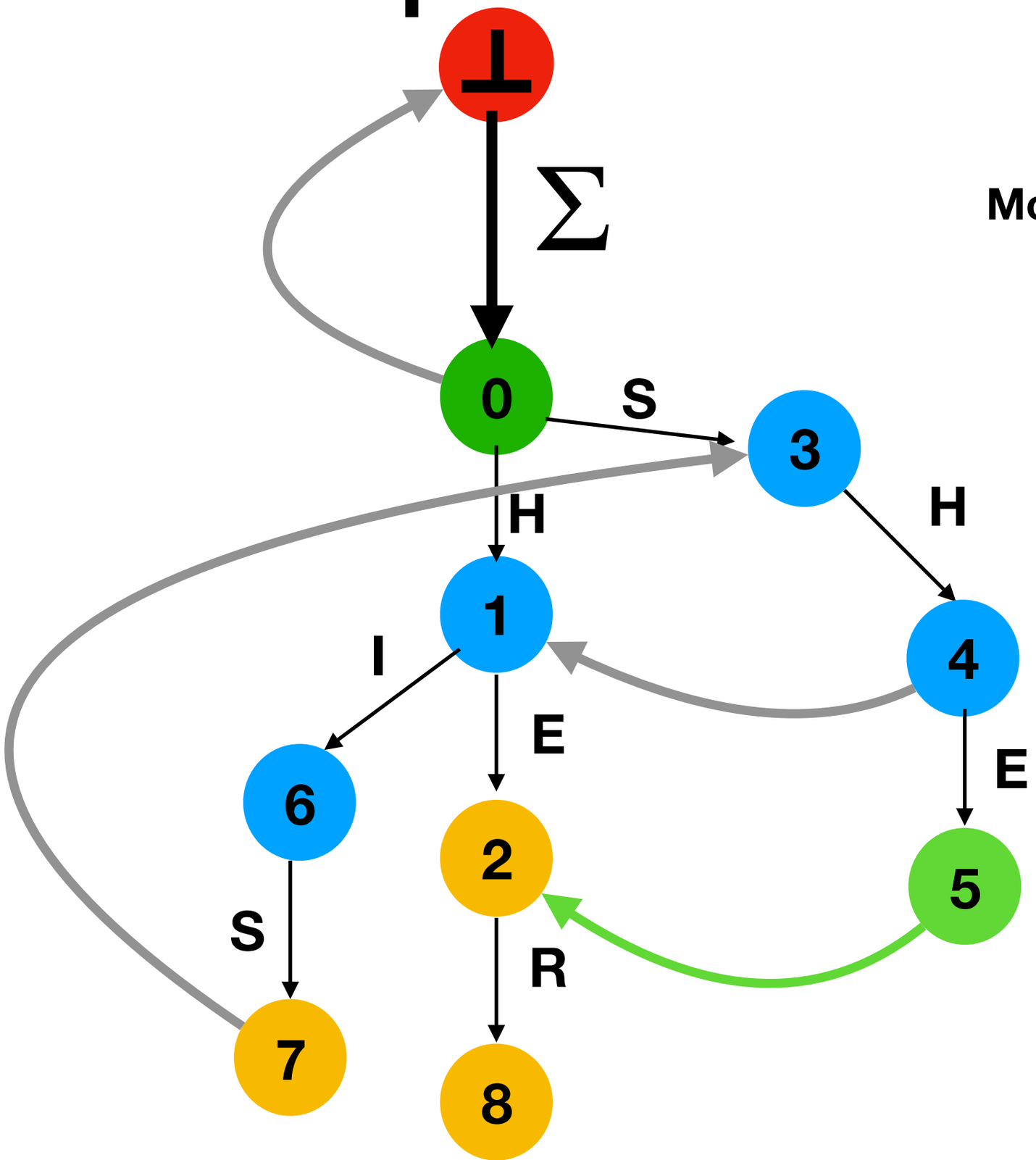


HISHER

Не можем прочитать R.
Попадаем в вершину
с вхождением паттерна

Нашли HE

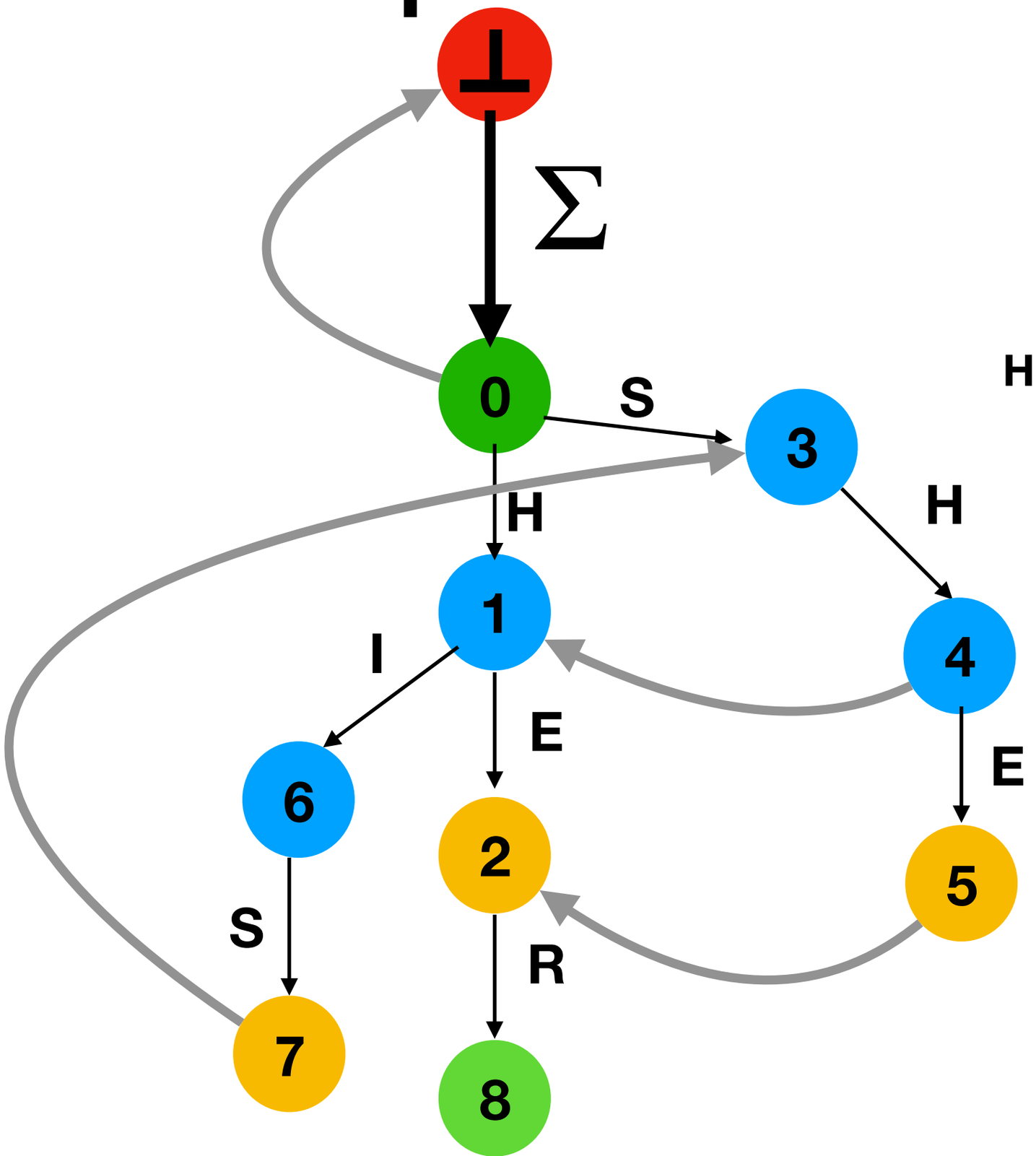
Попробуем поискать в строке



HISHER

Можем прочитать R.

Попробуем поискать в строке



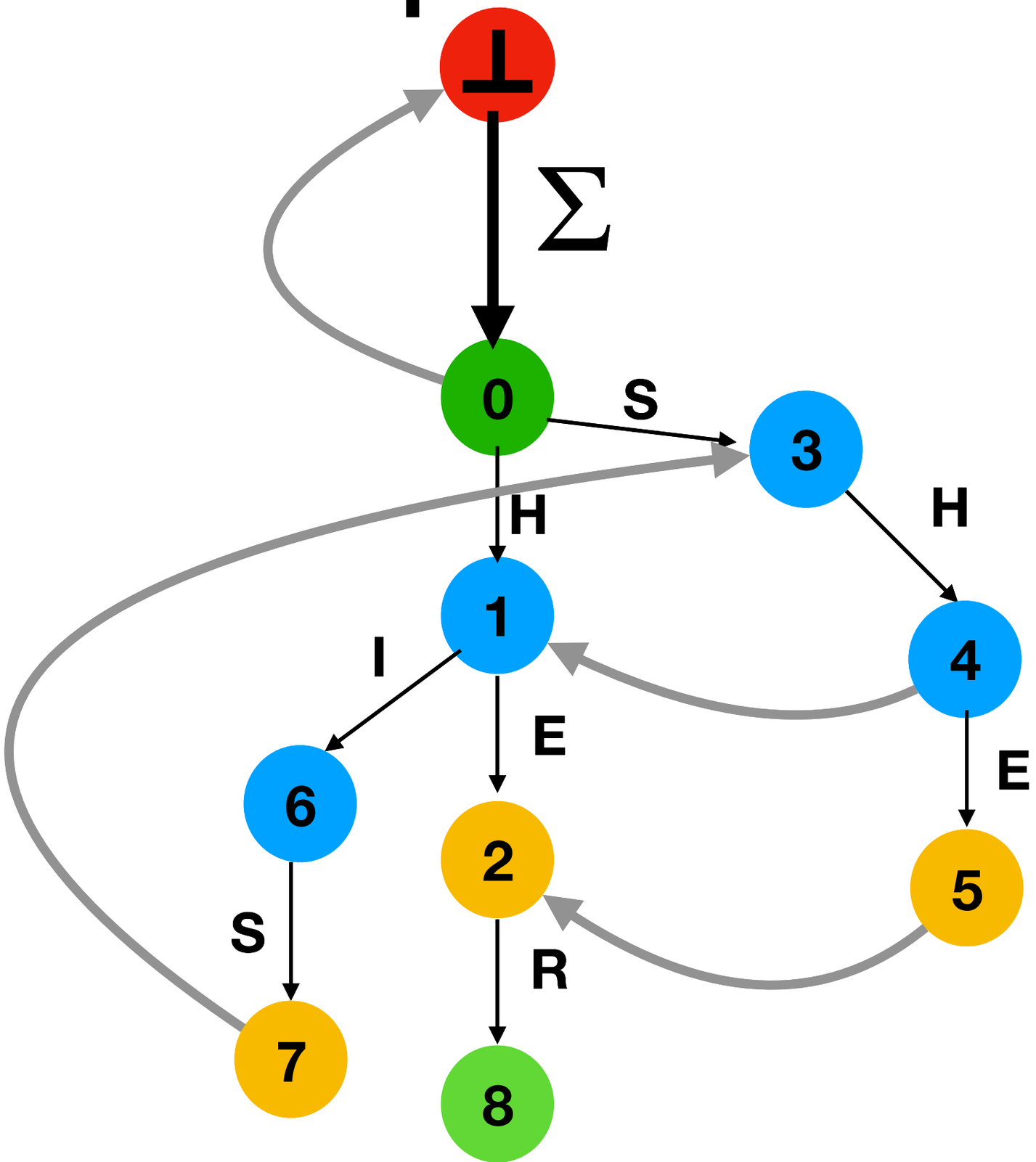
HISHER

Нашли HER

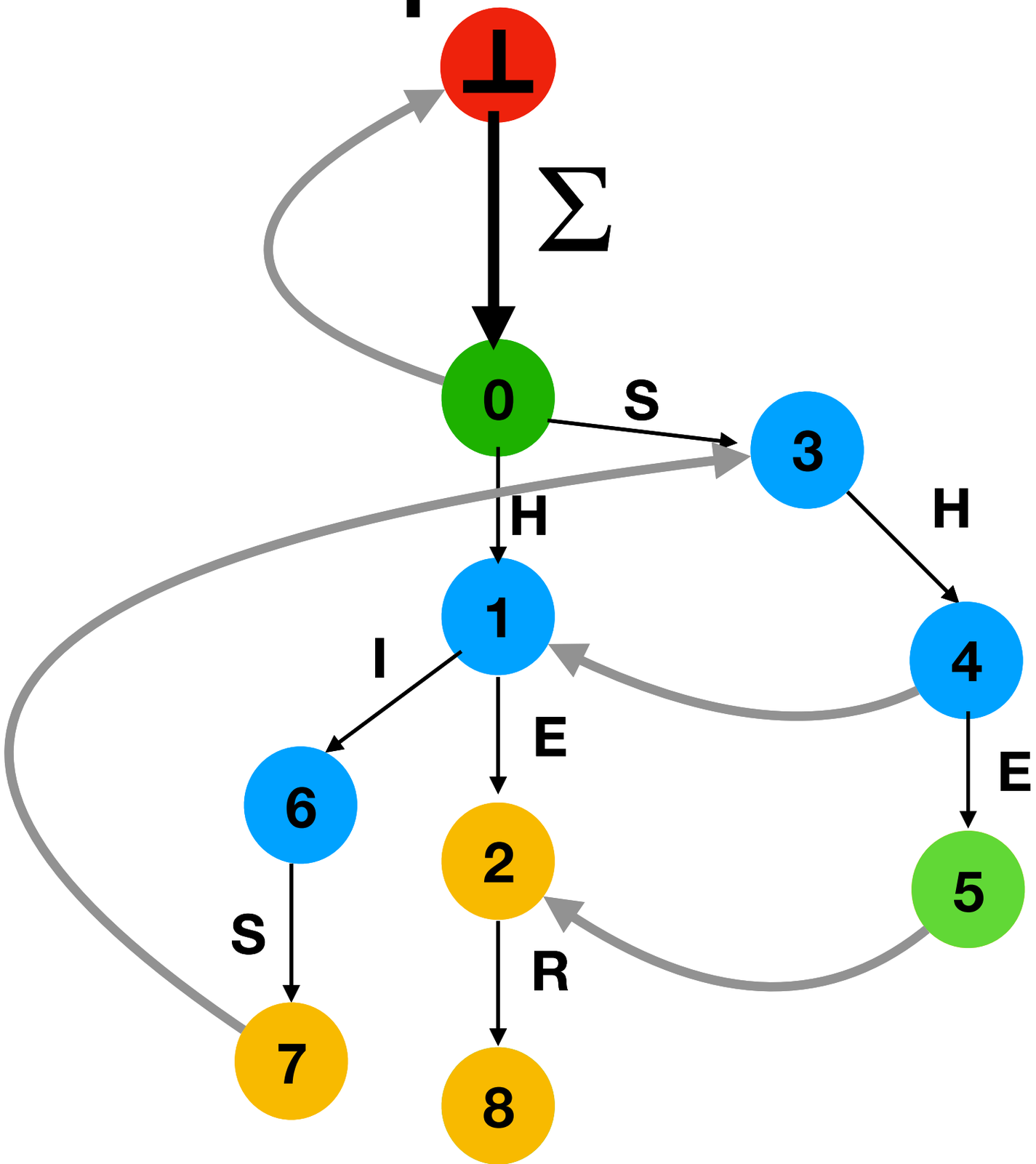
Нашли все вхождения

Попробуем поискать в строке

SHEI



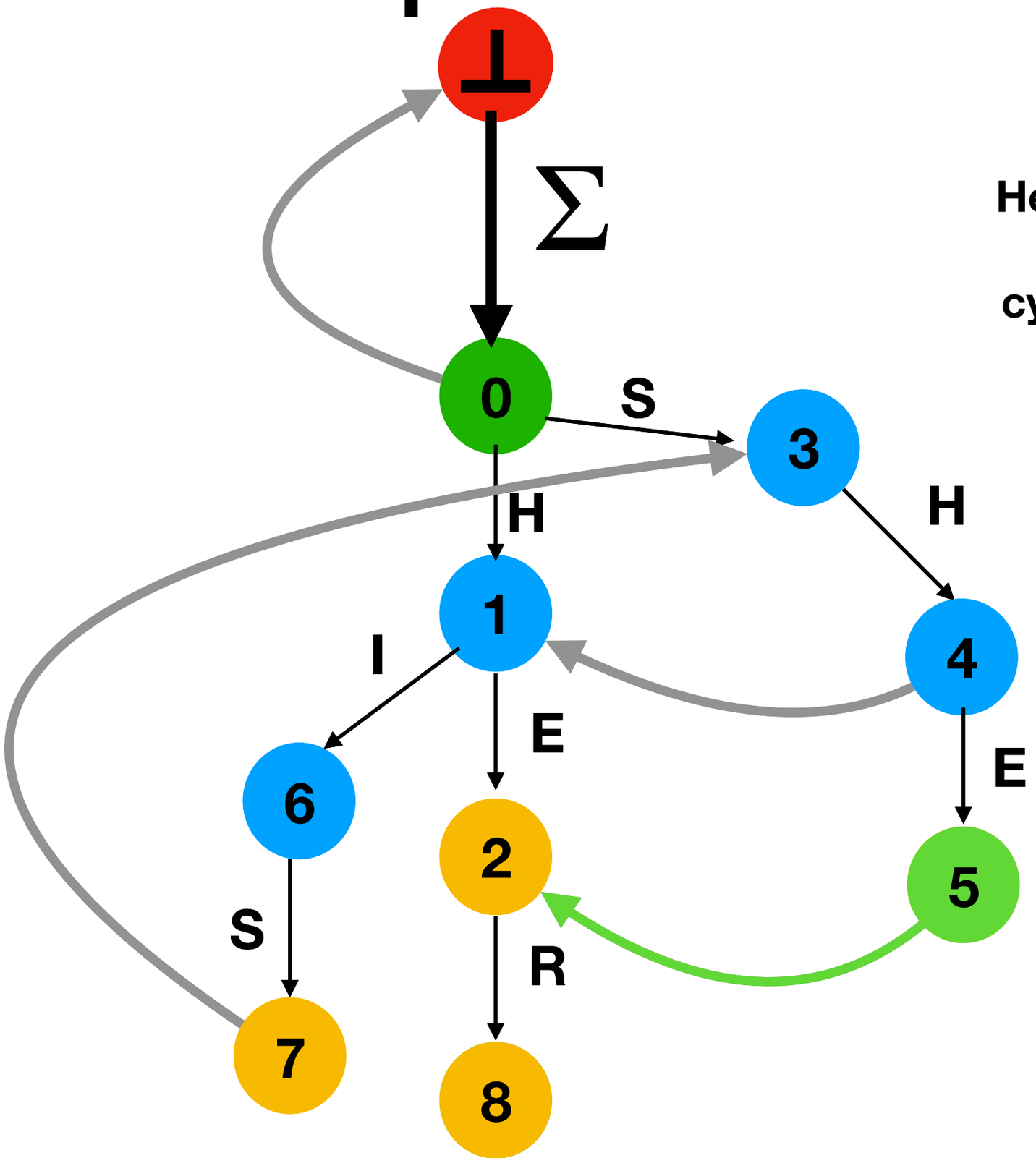
Попробуем поискать в строке



SHEI

Нашли SHE

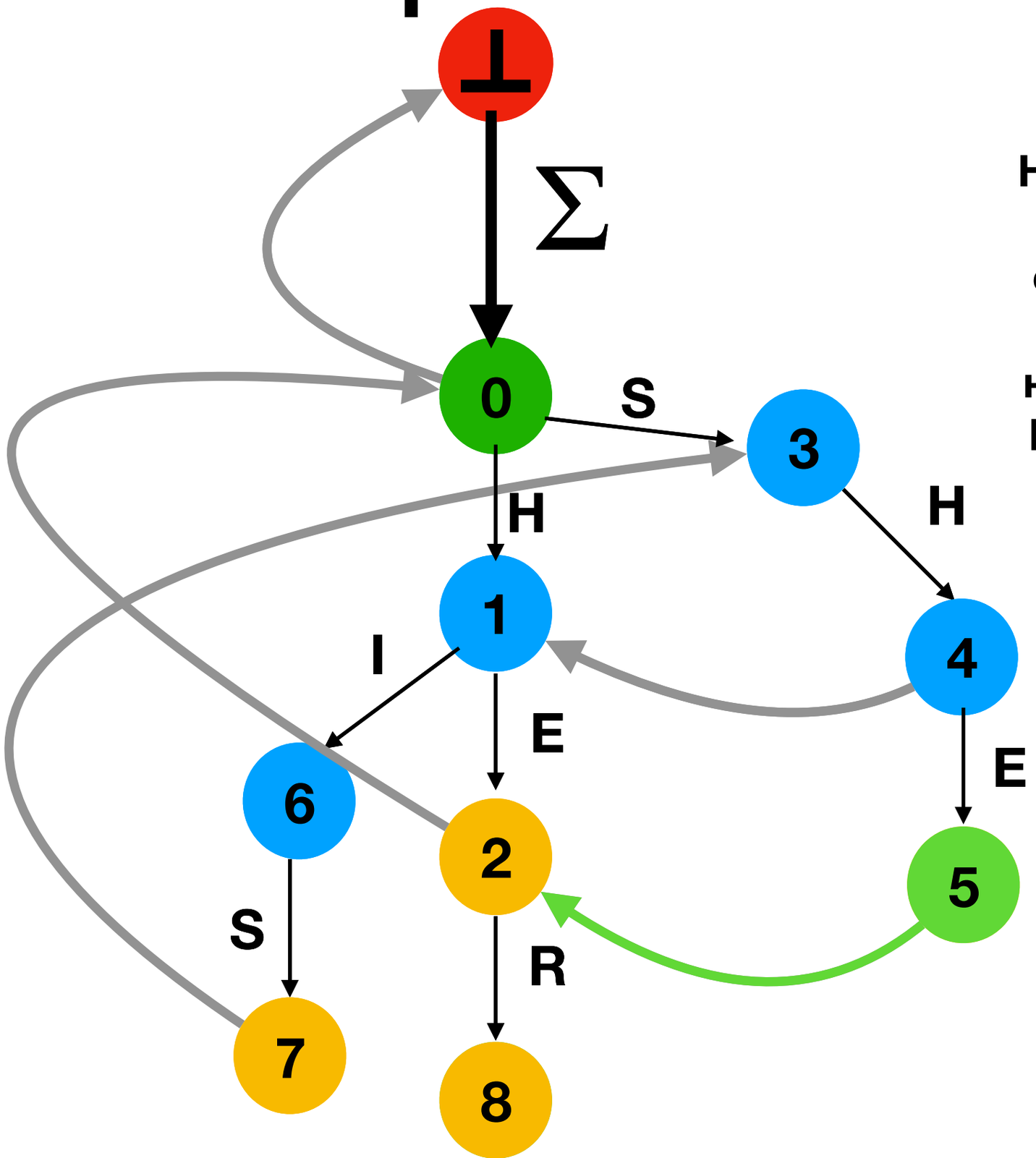
Попробуем поискать в строке



SHEI

Не можем прочитать I. Переходим по суффиксной ссылке - нашли паттерн HE

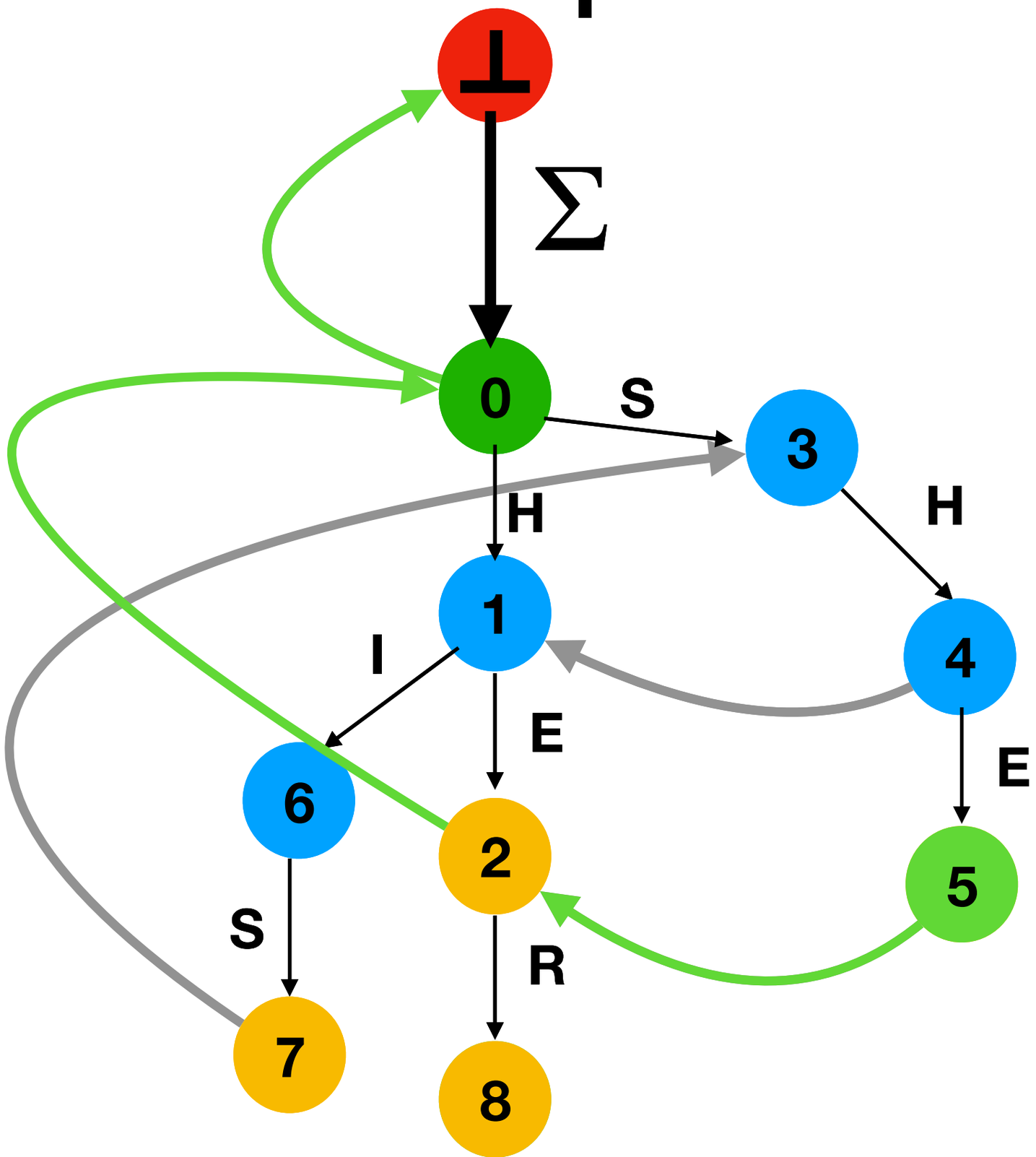
Попробуем поискать в строке



SHEI

Не можем прочитать I.
Переходим по суффиксной ссылке.
Все равно
не можем пройти по I.
Идем по суффиксной ссылке вершины 2.
Она ведет в корень (показали ссылку)

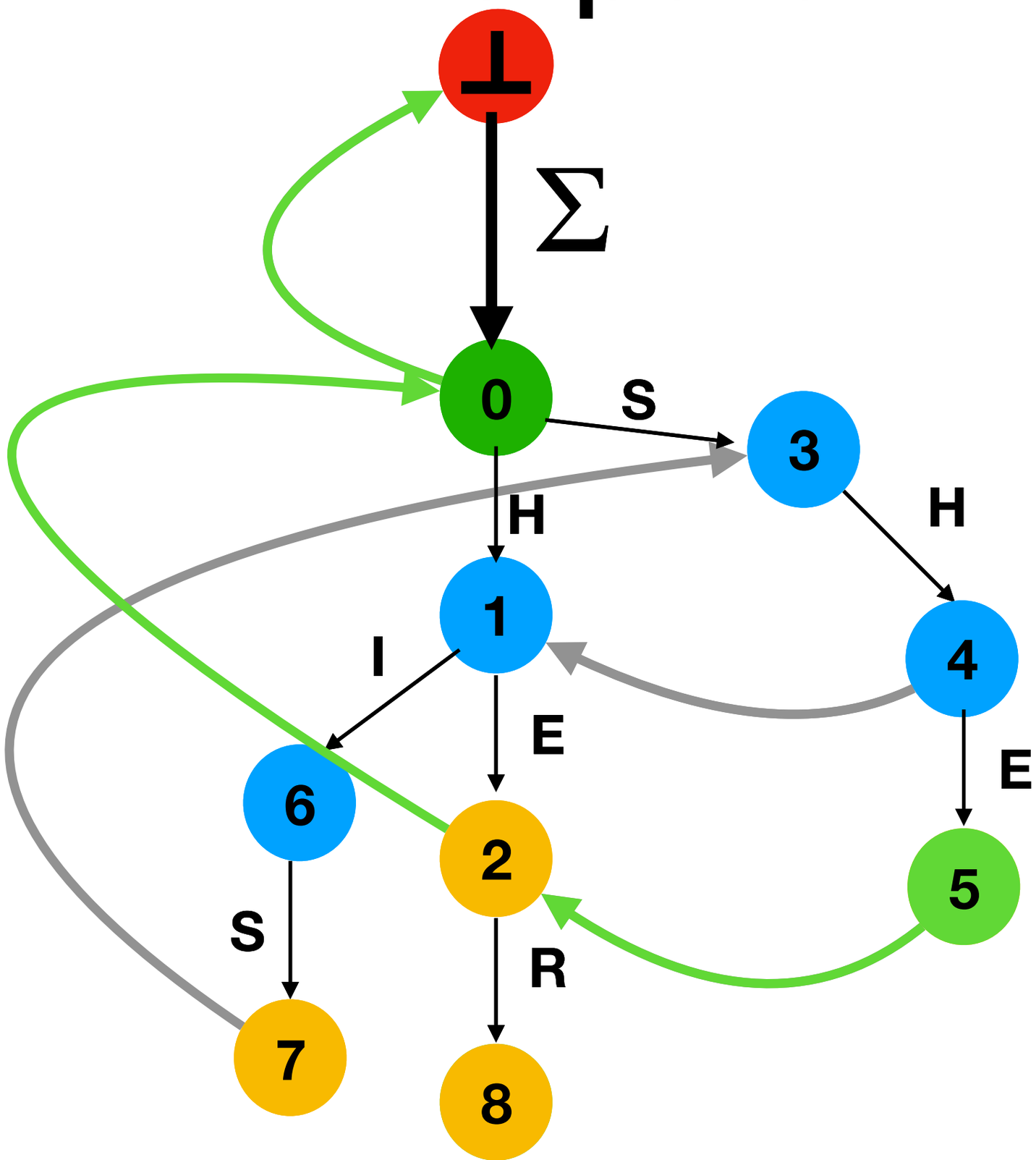
Попробуем поискать в строке



SHEI

Не можем прочитать I.
Переходим по суффиксной ссылке.
Все равно не можем пройти по I.
Идем по суффиксной ссылке вершины 2.
Она ведет в корень (показали ссылку)
Из корня тоже не можем прочитать.
Идем по его суффиксной ссылке в фиктивную вершину

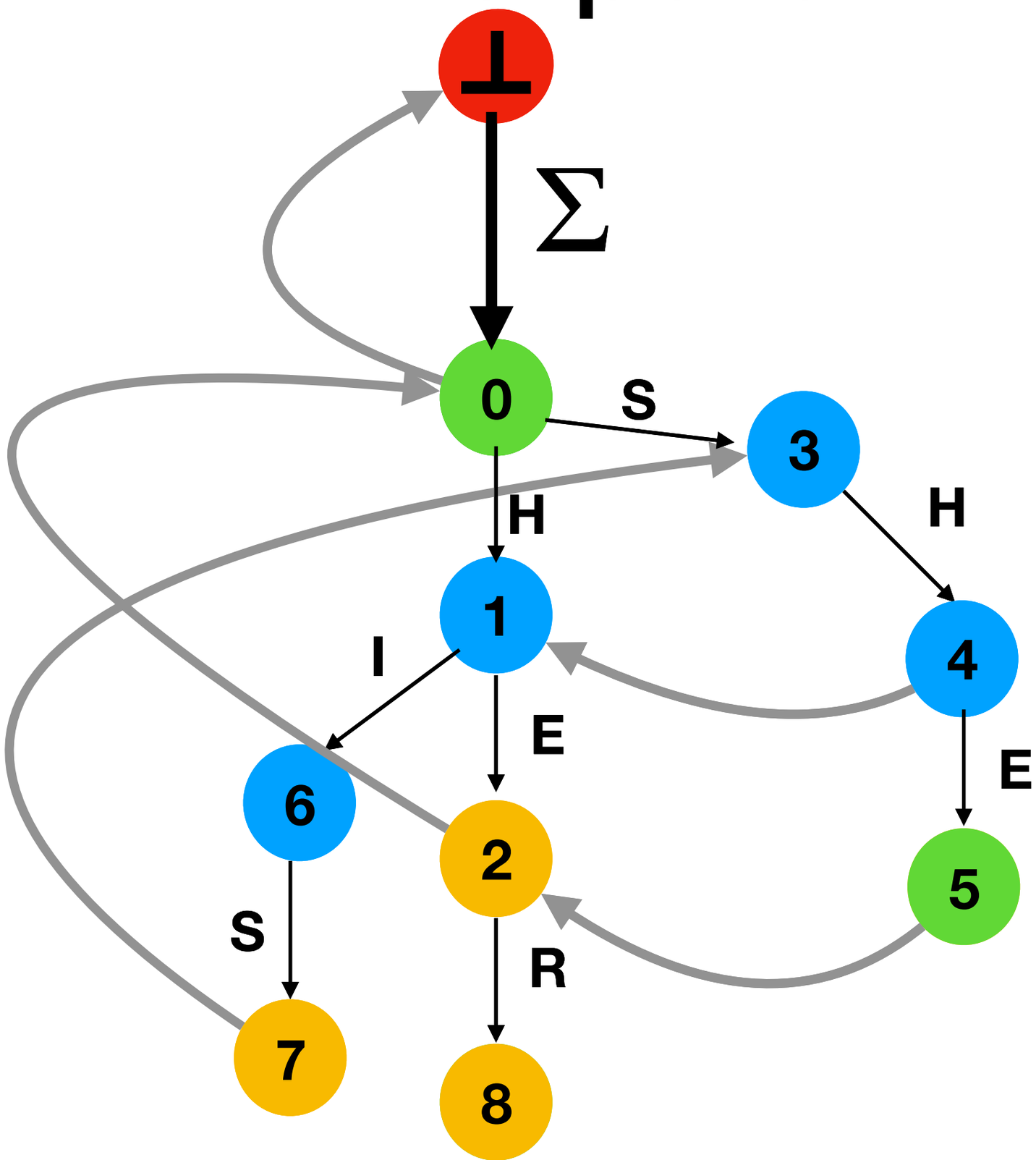
Попробуем поискать в строке



SHEI

Не можем прочитать I.
Переходим по суффиксной ссылке.
Все равно
не можем пройти по I.
Идем по суффиксной ссылке вершины 2.
Она ведет в корень
(показали ссылку)
Из корня
тоже не
можем прочитать.
Идем по его суффиксной ссылке в фиктивную вершину
Из нее можем по символу I
попасть в корень

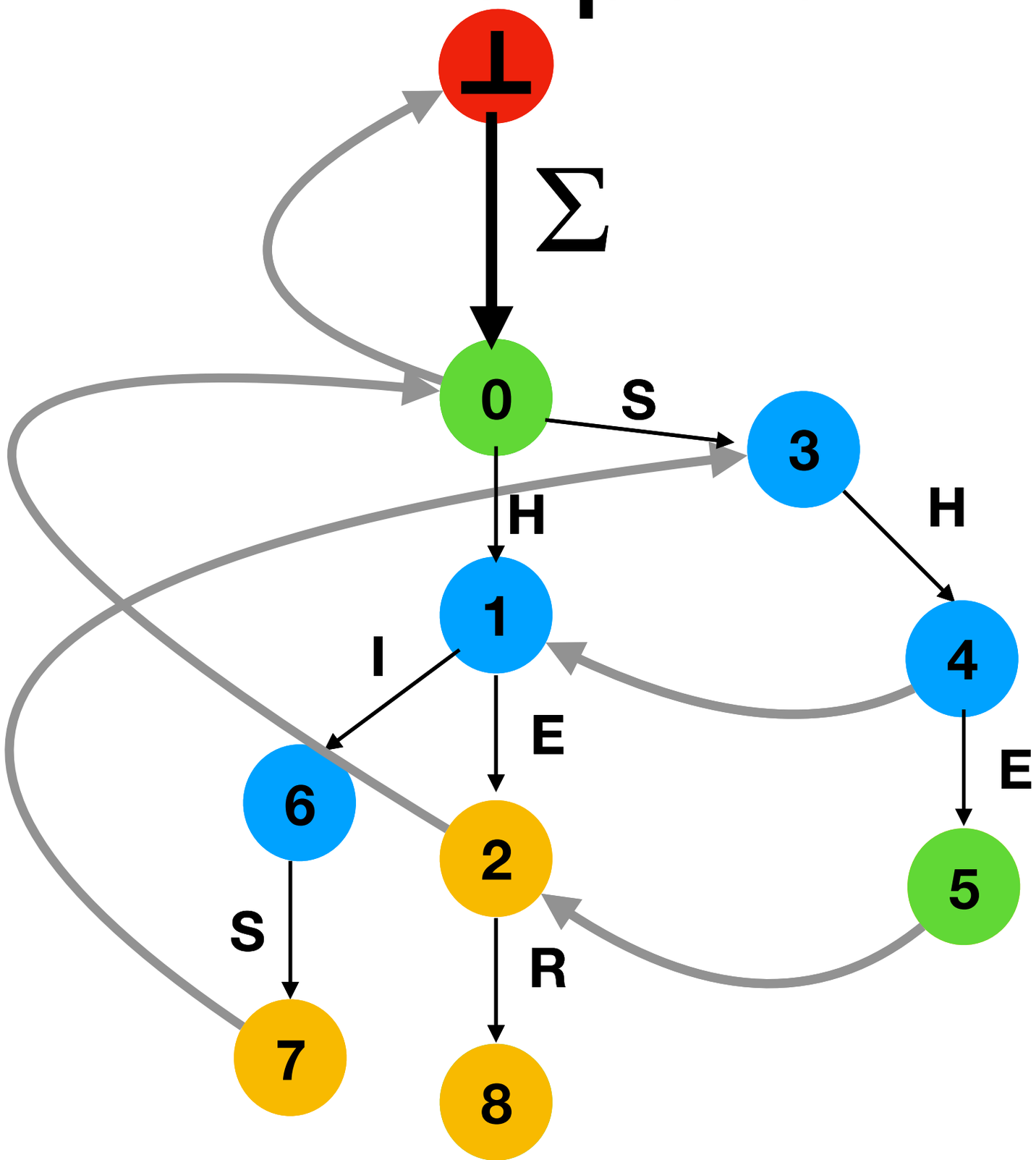
Попробуем поискать в строке



SHEI

Не можем прочитать I.
Переходим по суффиксной ссылке.
Все равно
не можем пройти по I.
Идем по суффиксной ссылке вершины 2.
Она ведет в корень
(показали ссылку)
Из корня
тоже не
можем прочитать.
Идем по его суффиксной ссылке в фиктивную вершину
Из нее можем по символу I
попасть в корень

Попробуем поискать в строке



SHEI

Не можем прочитать I.
Переходим по суффиксной ссылке.
Все равно
не можем пройти по I.
Идем по суффиксной ссылке вершины 2.
Она ведет в корень
(показали ссылку)
Из корня
тоже не
можем прочитать.
Идем по его суффиксной ссылке в фиктивную вершину
Из нее можем по символу I
попасть в корень

Сколько строить суффиксные ссылки?

При вычислении суффиксных ссылок время тратится на переходы в вершины по главным переходам (при переходе к следующему этапу вычисления)
Также время тратится во время переходов по уже посчитанным суффиксным ссылкам в процессе

Сколько строить суффиксные ссылки?

При вычислении суффиксных ссылок время тратится на переходы в вершины по главным переходам (при переходе к следующему этапу вычисления)

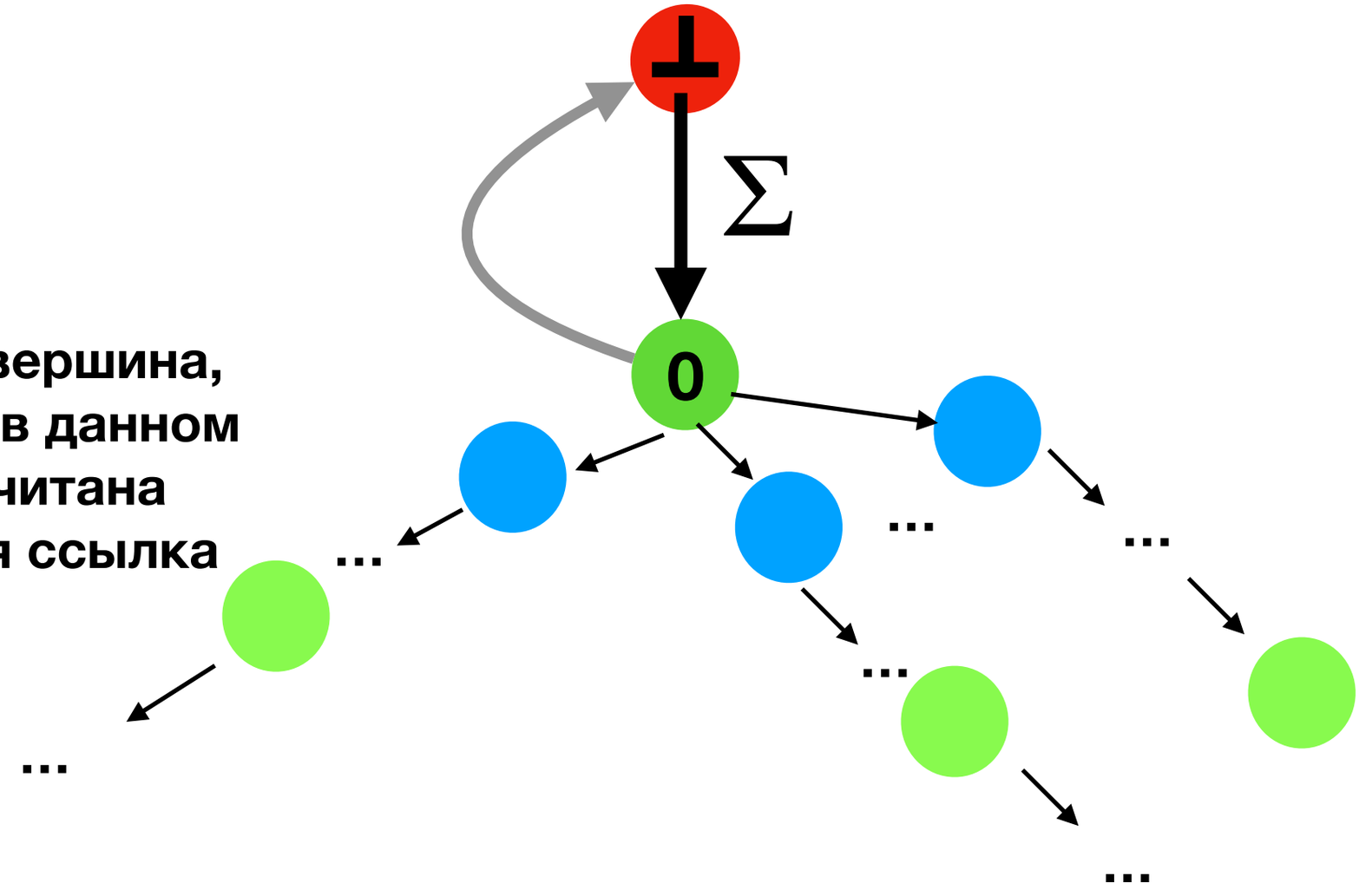
Суффиксная ссылка вычисляется для каждой вершины, и только один раз. Потому посещение вершин займет O(kM)

Сколько строить суффиксные ссылки?

Также время тратится во время переходов по уже посчитанным суффиксным ссылкам в процессе

Пусть высота корня - 1. Высота фиктивной вершины - 0.
Для каждого из k путей, соответствующих данному паттерну, будем хранить высоту суффиксной ссылки, которая ведет из последней вершины пути, для которой на данной момент посчитана суффиксная ссылка.

Последняя вершина, для которой в данном пути подсчитана суффиксная ссылка



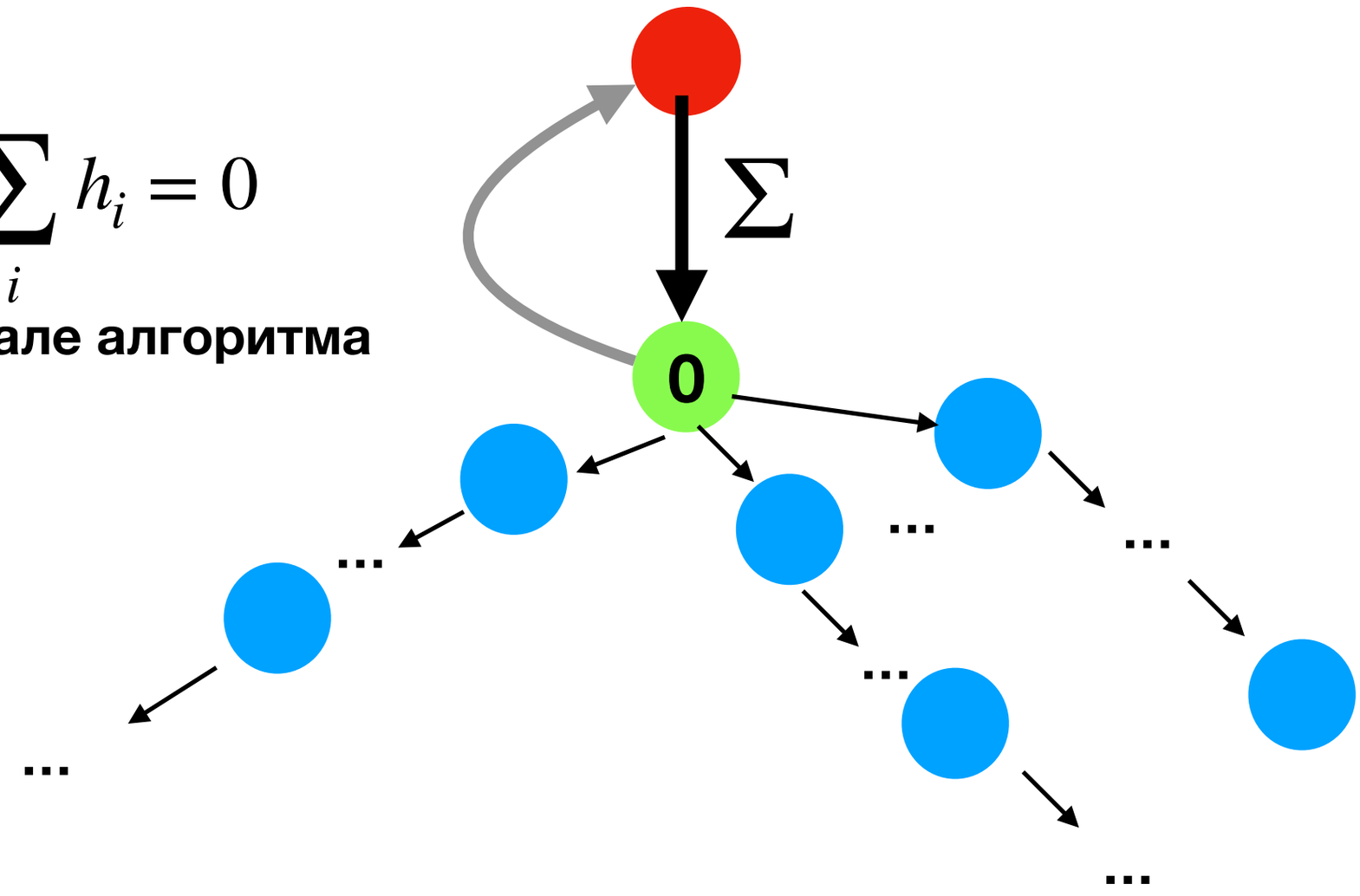
Сколько строить суффиксные ссылки?

Также время тратится во время переходов по уже посчитанным суффиксным ссылкам в процессе

Изначально, так как мы стартуем от корня, эти высоты равны 0 (все пути начинаются с корня, ссылка из корня ведет в фиктивную вершину). Сумма этих высот, тем самым, в начале работы алгоритма равна 0.

$$\sum_i h_i = 0$$

В начале алгоритма



Сколько строить суффиксные ссылки?

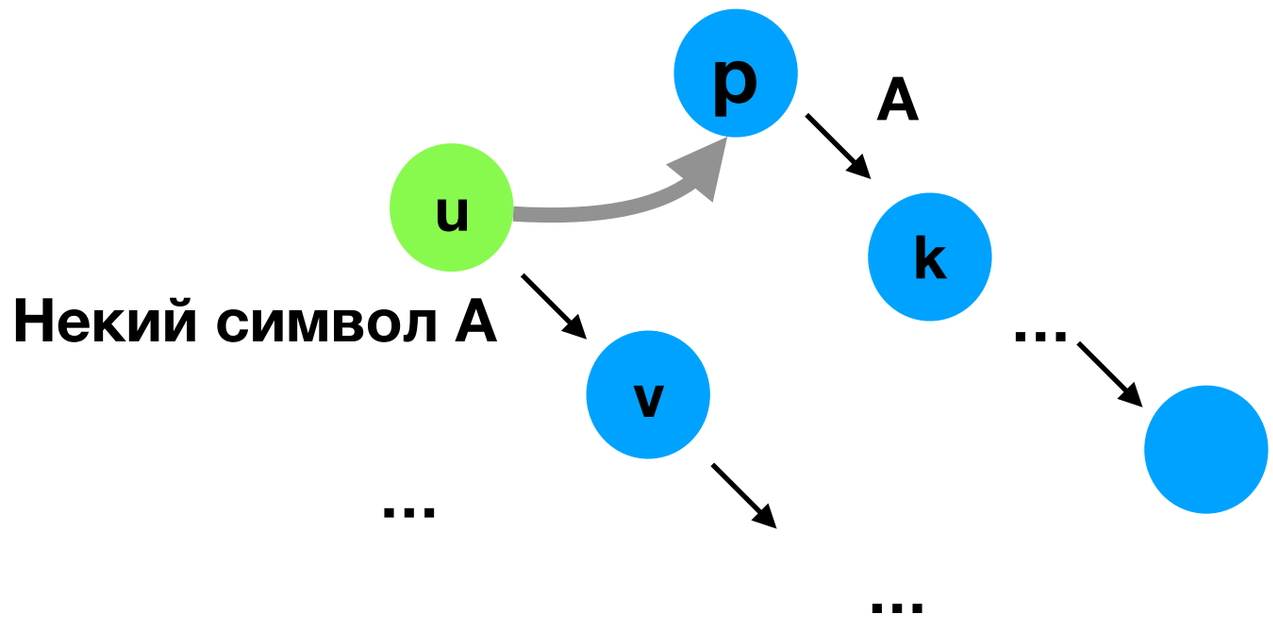
Также, так как у нас нет в боре отрицательных высот, она не может становиться меньше 0

$$\sum_i h_i \geq 0$$

В ходе алгоритма

Воспользуемся методом потенциалов, пусть эта сумма наш потенциал. Когда она увеличивается?

Когда при подсчете суффиксной ссылки вершины ***u*** при переходе по суффиксной ссылке ее предка **v** мы сразу оказываемся в вершине, по которой можно прочитать символ, который вел от вершины **v** к вершине **u.** При этом высота ссылки увеличивается максимально на 1.



Сколько строить суффиксные ссылки?

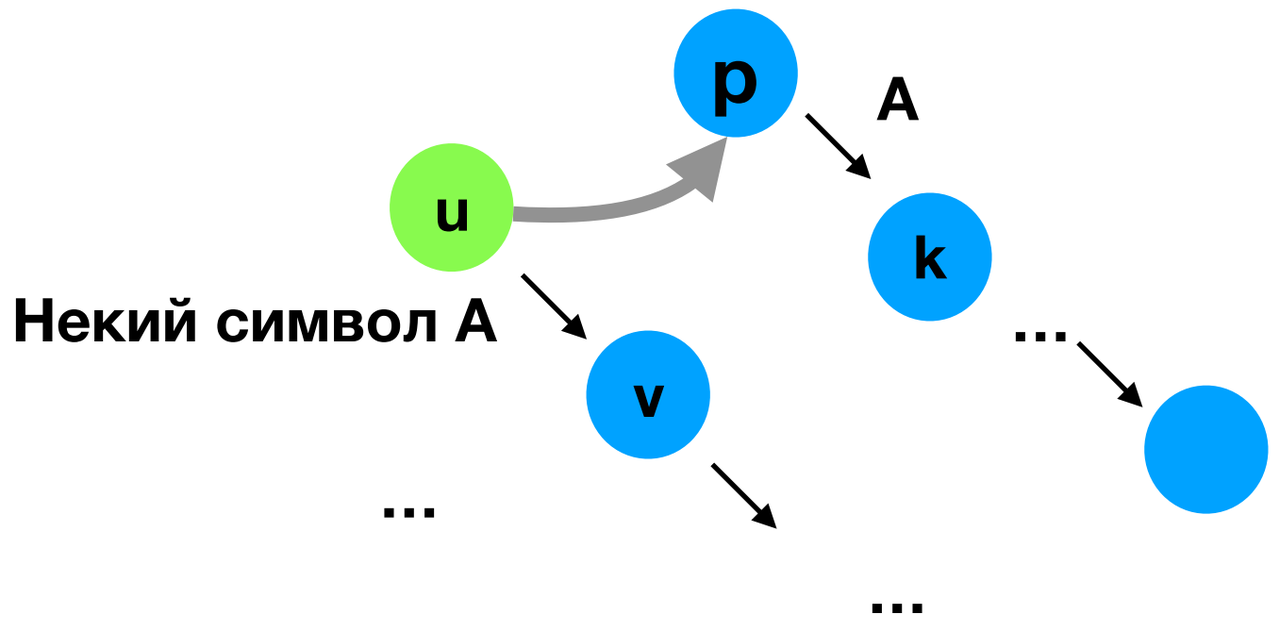
Также, так как у нас нет в боре отрицательных высот, она не может становиться меньше 0

$$\sum_i h_i \geq 0$$

В ходе алгоритма

Воспользуемся методом потенциалов, пусть эта сумма наш потенциал. Когда она увеличивается?

Когда при подсчете суффиксной ссылки вершины ***u*** при переходе по суффиксной ссылке ее предка **v** мы сразу оказываемся в вершине, по которой можно прочитать символ, который вел от вершины **v** к вершине **u.** При этом высота ссылки увеличивается максимально на 1.



Сколько строить суффиксные ссылки?

Также, так как у нас нет в боре отрицательных высот, она не может становиться меньше 0

$$\sum_i h_i \geq 0$$

В ходе алгоритма

Воспользуемся методом потенциалов, пусть эта сумма наш потенциал. Когда она увеличивается?

Когда при подсчете суффиксной ссылки вершины ***u*** при переходе по суффиксной ссылке ее предка **v** мы сразу оказываемся в вершине ***p***, по которой можно прочитать символ, который вел от вершины **v** к вершине ***u***. При этом высота ссылки увеличивается максимально на 1.

Всего вычисляются O(kM) суффиксных ссылок. Таким образом, за все время работы программы, сумма высот не могла стать больше O(kM). Так как каждый “лишний” суффиксный переход уменьшает эту сумму минимум на 1, то мы можем сказать, что было сделано не более O(kM) (по суффиксной ссылке предка) + O(kM) (“лишние переходы”) переходов. Таким образом, всего было сделано O(kM) переходов

Сколько строить суффиксные ссылки?

При вычислении суффиксных ссылок время тратится на переходы в вершины по главным переходам (при переходе к следующему этапу вычисления) - O(kM)
Также время тратится во время переходов по уже посчитанным суффиксным ссылкам в процессе - O(kM). Т.е итогу суффиксные ссылки считаются за O(kM)

$$O(k \cdot M) + O(k \cdot M) = O(k \cdot M)$$

Но еще не конец...

Это автомат не найдет все паттерны(((