Лекция 10. Деревья

Неориентированные и ориентированные деревья. Эквивалентность разных определений деревьев. Деревья и формулы (выражения). Обходы деревьев.

Ключевые слова: неориентированное дерево, ориентированное дерево, корень, лист, ветвь, предок вершины, потомок вершины, высота дерева, глубина вершины, поддерево, лес, бинарное (двоичное) дерево, прямой (префиксный) обход дерева, обратный (суффиксный) обход дерева, инфиксный обход бинарного дерева.

Неориентированные и ориентированные деревья

Деревья являются одним из интереснейших классов графов, используемых для представления различного рода иерахических структур.

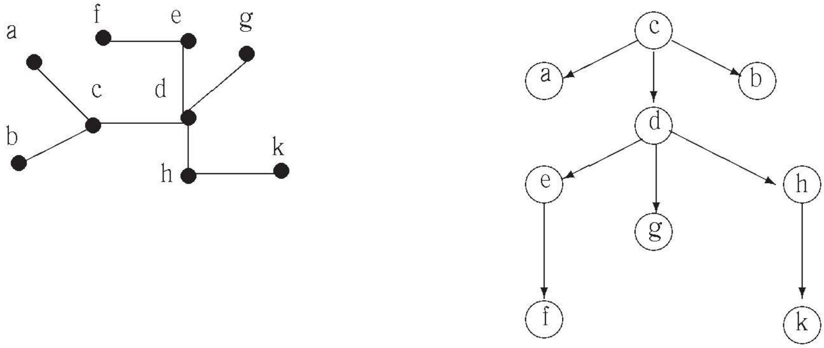
Определение 10.1. *Неориентированный граф называется* ***деревом,*** *если он связный и в нем нет циклов.*

Определение 10.2. *Ориентированный граф $G = (V, E)$ называется* ***(ориентированным) деревом,*** *если*
1) в нем есть одна вершина $r \in E$, в которую не входят ребра; она называется ***корнем*** *дерева;*
2) в каждую из остальных вершин входит ровно по одному ребру;
3) все вершины достижимы из корня.

На рисунке 1 на стр. 125 показаны примеры неориентированного дерева G_1 и ориентированного дерева G_2. Обратите внимание на то, что дерево G_2 получено из G_1 с помощью выбора вершины c в качестве корня и ориентации всех ребер в направлении "от корня".

Это не случайно. Докажите самостоятельно следующее утверждение о связи между неориентированными и ориентированными деревьями.

Лемма 10.1. *Если в любом неориентированном дереве $G = (V, E)$ выбрать произвольную вершину $v \in V$ в качестве корня и сориентировать все ребра в направлении "от корня", т.е. сделать v началом всех инцидентных ей ребер, вершины, смежные с v — началами всех инцидентных им еще не сориентированных ребер и т.д., то полученный в результате ориентированный граф G' будет ориентированным деревом.*



Неориентированное дерево G_1 Ориентированное дерево G_2

Рис. 1. Неориентированное и ориентированное деревья

Неориентированные и ориентированные деревья имеют много эквивалентных характеристик.

Теорема 10.1. *Пусть $G = (V, E)$ — неориентированный граф. Тогда следующие условия эквивалентны.*
(1) G является деревом.
(2) Для любых двух вершин в G имеется единственный соединяющий их путь.
(3) G связен, но при удалении из E любого ребра перестает быть связным.
(4) G связен и $|E| = |V| - 1$.
(5) G ациклический и $|E| = |V| - 1$.
(6) G ациклический, но добавление любого ребра к E порождает цикл.

Доказательство. (1) $\Rightarrow$ (2): Если бы в G некоторые две вершины соединялись двумя путями, то, очевидно, в G имеелся бы цикл. Но это противоречит определению дерева в (1).
(2) $\Rightarrow$ (3): Если G связен, но при удалении некоторого ребра $(u, v) \in E$ не теряет связности, то между u и v имеется путь, не содержащий это ребро. Но тогда в G имеется не менее двух путей, соединяющих u и v, что противоречит условию (2).
(3) $\Rightarrow$ (4): Предоставляется читателю (см. задачу 9.4 на стр. 122).
(4) $\Rightarrow$ (5): Если G содержит цикл и является связным, то при удалении любого ребра из цикла связность не должна нарушиться, но ребер останется $|E| = |V - 2|$, а по задаче 9.4 на стр. 122(a) в связном графе должно быть не менее $|V - 1|$ ребер. Полученное противоречие

показывает, что циклов в G нет и выполнено условие (5).
(5) $\Rightarrow$ (6): Предположим, что добавление ребра (u, v) к E не привело к появлению цикла. Тогда в G вершины u и v находятся в разных компонентах связности. Так как $|E| = |V - 1|$, то в одной из этих компонент, пусть это (V_1, E_1), число ребер и число вершин совпадают: $|E_1| = |V_1|$. Но тогда в ней имеется цикл (см. задачу 9.4 (б)), что противоречит ацикличности G.
(6) $\Rightarrow$ (1): Если бы G не был связным, то нашлись бы две вершины u и v из разных компонент связности. Тогда добавление ребра (u, v) к E не привелобы к появлению цикла, что противоречит (6). Следовательно, G связен и является деревом.

Для ориентированных деревьев часто удобно использовать следующее индуктивное определение.

Определение 10.3. *Определим по индукции класс ориентированных графов $\mathcal{D}$, называемых деревьями. Одновременно для каждого из них определим выделенную вершину — корень.*
1) Граф $T_0 = (V, E)$, с единственной вершиной $V = \{v\}$ и пустым множеством ребер $E = \emptyset$ является деревом (входит в $\mathcal{D}$). Вершина v называется корнем этого дерева.
2) Пусть графы $T_1 = (V_1, E_1), \ldots, T_k = (V_k, E_k)$ с корнями $r_1 \in V_1, \ldots, r_k \in V_k$ принадлежат $\mathcal{D}$, а r_0 — новая вершина, т.е. $r_0 \notin \bigcup_{i=1}^{k} V_i$. Тогда классу $\mathcal{D}$ принадлежит также следующий граф $T = (V, E)$, где $V = \{r_0\} \cup \bigcup_{i=1}^{k} V_i$, $E = \{(r_0, r_i) \mid i = 1, \ldots, k\} \cup \bigcup_{i=1}^{k} E_i$. Корнем этого дерева является вершина r_0.
3) Других графов в классе $\mathcal{D}$ нет.

Рисунок 2 на стр. 127 иллюстрирует это определение.

Теорема 10.2. *Определения ориентированных деревьев 10.2 и 10.3 эквивалентны.*

Доказательство. $\Rightarrow$ Пусть граф $G = (V, E)$ удовлетворяет условиям определения 10.2. Покажем индукцией по числу вершин $|V|$, что $G \in \mathcal{D}$.
Если $|V| = 1$, то единственная вершина $v \in V$ является по свойству (1) корнем дерева, т.е. в этом графе ребер нет: $E = \emptyset$. Тогда $G = T_0 \in \mathcal{D}$.

Предположим, что всякий граф с $\leq n$ вершинами, удовлетворяющий определению 10.2 входит в $\mathcal{D}$. Пусть граф $G = (V, E)$ с $(n+1)$-й вершиной удовлетворяет условиям определения 10.2. По условию (1) в нем имеется вершина-корень r_0. Пусть из r_0 выходит k ребер и они ведут в вершины

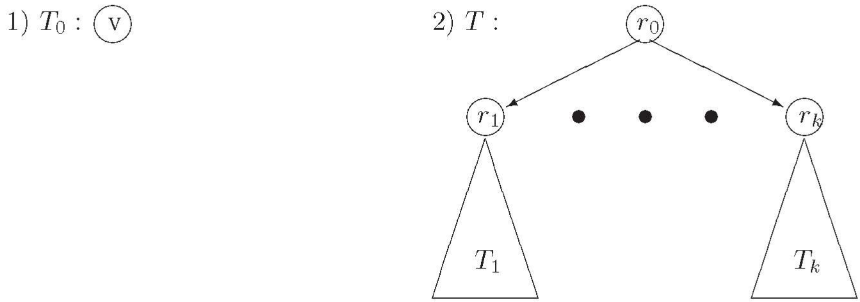


Рис. 2. Индуктивное определение ориентированных деревьев

$r_1, \ldots, r_k$ $(k \geq 1)$. Обозначим через G_i, $(i = 1, \ldots, k)$ граф, включающий вершины $V_i = \{v \in V \mid v$ *достижима из* $r_i\}$ и соединяющие их ребра $E_i \subseteq E$. Легко понять, что G_i удовлетворяет условиям условиям определения 10.2. Действительно, в r_i не входят ребра, т.е. эта вершина — корень G_i. В каждую из остальных вершин из V_i входит по одному ребру как и в G. Если $v \in V_i$, то она достижима из корня r_i по определению графа G_i. Так как $|V_i| \leq n$, то по индуктивному предположению $G_i \in \mathcal{D}$. Тогда граф G получен по индуктивному правилу (2) определения 10.3 из деревьев $G_1, \ldots, G_k$ и поэтому принадлежит классу $\mathcal{D}$.

$\Leftarrow$ Если некоторый граф $G = (V, E)$ входит в класс $\mathcal{D}$, то выполнение условий (1)-(3) определения 10.2 для него легко установить индукцией по определению 10.2. Предоставляем это читателю в качестве упражнения.

С ориентированными деревьями связана богатая терминология, пришедшая из двух источников: ботаники и области семейных отношений.

Корень — это единственная вершина, в которую не входят ребра, ***листья*** — это вершины, из которых не выходят ребра. Путь из корня в лист называется ***ветвью*** дерева. ***Высота дерева*** — это максимальная из длин его ветвей. ***Глубина вершины*** — это длина пути из корня в эту вершину. Для вершины $v \in V$, подграф дерева $T = (V, E)$, включающий все достижимые из v вершины и соединяющие их ребра из E, образует ***поддерево*** T_v дерева T с корнем v (см. задачу 10.3 на стр. 132). ***Высота вершины*** v — это высота дерева T_v. Граф, являющийся объединением нескольких непересекающихся деревьев, называется ***лесом.***

Если из вершины v ведет ребро в вершину w, то v называется ***отцом*** w, а w – ***сыном*** v (в последнее время в англоязычной литературе употребляется асексуальная пара терминов: родитель — ребенок). Из определения дерева непосредственно следует, что у каждой вершины кроме корня имеется единственный отец. Если из вершины v ведет путь в вершину w, то v называется ***предком*** w, а w – ***потомком*** v. Вершины, у которых общий отец, называются ***братьями*** или ***сестрами.***

Выделим еще один класс графов, обобщающий ориентированные деревья, — ***ориентированные ациклические графы.*** Два вида таких размеченных графов будут использованы далее для представления булевых функций. У этих графов может быть несколько корней — вершин, в которые не входят ребра, и в каждую вершину может входить несколько ребер, а не одно, как у деревьев.

Деревья и формулы (выражения)

Напомним, что в главе 2 было введено общее понятие ***формулы*** над системой функций $\mathcal{B}$ (определение 3.2 на стр. 39), которое применимо для произвольных функций, а не только булевых. В главе 4 аналогичные синтаксические объекты для логики предикатов названы ***термами*** (определение 7.1 на стр. 85), а в языках программирования такие конструкции часто называются ***выражениями.***

Итак, пусть формула над множеством функций $\mathbf{F}$, множеством констант $\mathbf{C}$ и множеством переменных $\mathbf{Var}$ определяется индуктивно по следующим правилам.

- Переменная из $\mathbf{Var}$ есть формула.
- Константа из $\mathbf{C}$ есть формула.
- Если $g_1, \ldots, g_k$ — формулы, а $f^{(k)}$ — k-местная функция из $\mathbf{F}$, то $f(g_1, \ldots, g_k)$ — это формула.

Обозначим множество всех таких формул через $\mathcal{F}(\mathbf{F}, \mathbf{C}, \mathbf{Var})$.

Рассмотрим класс упорядоченных размеченных деревьев $\mathcal{T}(\mathbf{F}, \mathbf{C}, \mathbf{Var})$, листья которых помечены элементами из $(\mathbf{C} \cup \mathbf{Var})$, а внутренние вершины — функциями из $\mathbf{F}$, причем, если вершина помечена символом k-местной функции из $\mathbf{F}$, то у нее имеется k сыновей.

Предложение 10.1. *Между множеством формул* $\mathcal{F}(\mathbf{F}, \mathbf{C}, \mathbf{Var})$ *и множеством деревьев* $\mathcal{T}(\mathbf{F}, \mathbf{C}, \mathbf{Var})$ *имеется взаимно однозначное соответствие.*

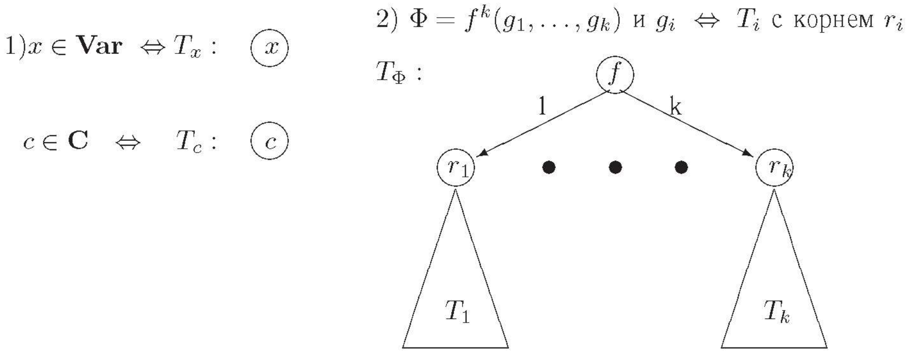


Рис. 3. Индуктивное определение связи между формулами и деревьями

Доказательство. Это соответствие легко устанавливается индукцией по определениям формул и деревьев. Оно показано выше на рис. 3.

Пример 10.1. *Рассмотрим, например, класс обычных арифметических формул над множеством функций* $\mathbf{F} = \{+, -, *, :\}$, *целочисленных констант* $\mathbf{C} = \{0, 1, 2, \dots\}$ *и переменных* $\mathbf{Var} = \{x, y, z, \dots\}$. *Пусть формула* $\Phi = +(*(5, +(x, 7)), (: (y, +(x, 7)))$ *(ее обычное представление* $\Phi = 5 * (x + 7) + y : (x + 7)$ *)*

Тогда в соответствии с предложением 10.1 эта формула представляется деревом T_Φ, изображенном на рис. 4.

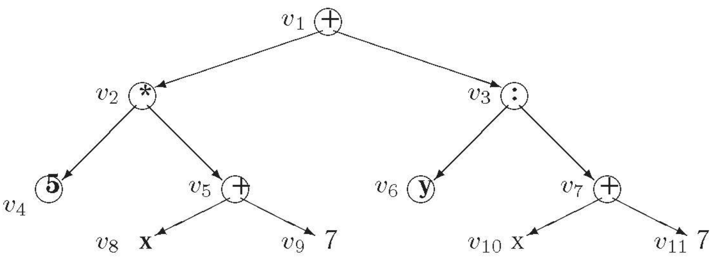


Рис. 4. Дерево T_Φ

На этом рисунке не указаны явно номера ребер, выходящих из внутренних вершин дерева, которые идентифицируют порядок аргументов операций. Предполагается, что для коммутативных операций +, * это несущественно, а для некоммутативных, таких, как :, первый аргумент расположен левее второго.

Заметим, что у деревьев, представляющих арифметические или логические (булевские) формулы, внутренние вершины имеют не более 2-х сыновей. Такие деревья образуют важный подкласс ориентированных деревев, называемых бинарными или двоичными.

Определение 10.4. *Ориентированное дерево называется* ***бинарным*** *или* ***двоичным,*** *если у каждой его внутренней вершины имеется не более двух сыновей, причем ребра, ведущие к ним помечены двумя разными метками (обычно используются метки из пар: "левый" — "правый", 0 — 1, + — – и т.п.)*

Бинарное дерево называется ***полным,*** *если у каждой его внутренней вершины имеется два сына и все его ветви имеют одинаковую длину.*

Ориентированные ациклические графы также используются для представления формул. Они получаются из соответствующих деревьев при склеивании вершин, представляющих одинаковые подформулы. Для формулы $\Phi = 5 * (x + 7) + y : (x + 7)$ такой граф G_Φ получается при склеивании вершин v_5 и v_7 дерева T_Φ, представляющих подформулу $(x + 7)$.

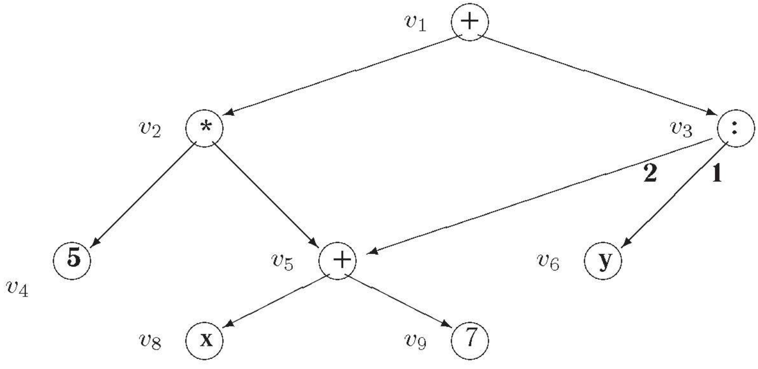


Рис. 5. Ациклический граф G_Φ

На этом рисунке явно указаны номера ребер, выходящих из вершины v_3, которые определяют порядок аргументов приписанной этой вершине операции :. Ясно, что при отсутствии такого указания и использовании порядка " по умолчанию" — первый аргумент слева — граф представлял бы другое выражение.

Обходы деревьев

Часто при обработке представленной в дереве информации требуется обойти некоторым регулярным способом все его вершины. Имеется два естественных стандартных способа обхода деревьев. Каждый из них позволяет линейно упорядочить вершины дерева и тем самым представить его "двумерную структуру" в виде линейной последовательности вершин.

Прямой (префиксный) обход дерева основан на принципе: "сначала родитель, затем дети". Определим индукцией по построению дерева T в определении 10.3 его прямое представление $\text{ПР}(T)$ следующим образом.

1) Если $T_0 = (\{v\}, \emptyset)$, то $\text{ПР}(T_0) = v$.

2) Если T получено из деревьев $T_1, \ldots, T_k$ и нового корня r_0 по пункту (2) определения 10.3 то $\text{ПР}(T) = r_0\,\text{ПР}(T_1) \ldots \text{ПР}(T_k)$.

Обратный (суффиксный) обход дерева основан на противоположном принципе: "сначала дети, затем родитель". Вот его индуктивное определение.

1) Если $T_0 = (\{v\}, \emptyset)$, то $\text{ОБР}(T_0) = v$.

2) Если T получено из деревьев $T_1, \ldots, T_k$ и нового корня r_0 по пункту (2) определения 10.3, то $\text{ОБР}(T) = \text{ОБР}(T_1) \ldots \text{ОБР}(T_k)\, r_0$.

Для бинарных деревьев, внутренние вершины которых имеют не более 2-х сыновей, помеченных как "левый" и "правый", можно естественно определить еще один способ обхода — ***инфиксный (внутренний) обход,*** основанный на принципе: "сначала левый сын, затем родитель, а затем правый сын". Он определяется следующим образом.

1) Если $T_0 = (\{v\}, \emptyset)$, то $\text{ИНФ}(T_0) = v$.

2) Если T получено из деревьев T_1, T_2 и нового корня r_0 по пункту (2) определения 10.3 на стр. 126, то $\text{ИНФ}(T) = \text{ИНФ}(T_1) r_0 \text{ИНФ}(T_2)$. (Если одно из деревьев T_1, T_2 пусто, то соответствующее ему инфиксное представление тоже пусто).

Пример 10.2. *Построим в соответствии с этими определениями три разных обхода бинарного дерева T_Φ, изображенного на рис. 4 на стр. 129 (в скобках после вершины указана ее метка).*

$\text{ПР}(T_\Phi) = v_1(+)v_2(*)v_4(5)v_5(+)v_8(x)v_9(7)v_3(:)v_6(y)v_7(+)v_{10}(x)v_{11}(7)$.
$\text{ОБР}(T_\Phi) = v_4(5)v_8(x)v_9(7)v_5(+)v_2(*)v_6(y)v_{10}(x)v_{11}(7)v_7(+)v_3(:)v_1(+)$.
$\text{ИНФ}(T_\Phi) = v_4(5)v_2(*)v_8(x)v_5(+)v_9(7)v_1(+)v_6(y)v_3(:)v_{10}(x)v_7(+)v_{11}(7)$.

Для упорядоченного размеченного дерева T из класса $\mathcal{T}(\mathbf{F}, \mathbf{C}, \mathbf{Var})$ по любому из указанных обходов $\text{ПР}(T), \text{ОБР}(T)$ и, если дерево бинарное, – $\text{ИНФ}(T)$ можно однозначно восстановить само дерево T (см. задачу 10.6).

Замечание. Для вычислительных приложений особенно интересен обратный обход, иногда называемый обратной польской записью. По нему компилятор легко строит программу вычисления соответствующего выражения.

Задачи

Задача 10.1. *Докажите, что если в связном неориентированном графе число вершин равно числу ребер, то можно выбросить одно из ребер так, что после этого граф станет деревом.*

Задача 10.2. *Пусть $G = (V, E)$ — неориентированное дерево и $v \in V$ — произвольная вершина. Докажите, что если для каждого ребра $(u, w) \in E$ выбрать ориентацию от u к w, если им заканчивается путь из v в w, и ориентацию от w к u, если им заканчивается путь из v в u, то полученный ориентированный граф будет ориентированным деревом с корнем v. Используйте это утверждение для доказательства следующего факта: если в неориентированном дереве $G = (V, E)$ имеется вершина степени $d > 1$, то в нем имеется по крайней мере d вершин степени 1.*

Задача 10.3. *Пусть $T = (V, E)$ — это ориентированное дерево с корнем $v_0 \in V$. определим для каждой вершины $v \in V$ подграф $T_v = (V_v, E_v)$ следующим образом: V_v — это множество вершин, достижимых из v в T, а E_v — это множество ребер из E, оба конца которых входят в V_v. Доказать, что*
а) T_v является деревом с корнем v;
б) если две разные вершины v и u имеют одинаковую глубину, то деревья T_v и T_u не пересекаются.

Задача 10.4. *Пусть $G = (V, E)$ – ориентированный граф с $n > 1$ вершинами. Докажите, что G является (ориентированным) деревом тогда и только тогда, когда в G нет циклов, имеется одна вершина r, в которую не входят ребра, а в каждую из остальных вершин $v \in V \setminus \{r\}$ входит ровно одно ребро.*

Задача 10.5. *Пусть корень ориентированного дерева T имеет 5 сыновей, а каждая из остальных внутренних вершин имеет три или четыре сына, при этом число вершин с 3-я сыновьями вдвое больше числа вершин с 4-я. Сколько всего вершин и ребер в T, если известно, что число его листьев равно 27?*

Задача 10.6. *Для каждого из обходов деревьев ПР(T), ОБР(T) и ИНФ(T) предложите процедуру, восстановления соответствующего дерева* $T \in \mathcal{T}(\mathbf{F}, \mathbf{C}, \mathbf{Var})$.

Задача 10.7. *Докажите по индукции, что в любом бинарном дереве число вершин степени 2 на единицу меньше числа листьев.*

Задача 10.8. *Определите число листьев и число вершин в полном бинарном дереве высоты* h.

Задача 10.9. *Постройте дерево, представляющее следующую логическую формулу*
$\Psi = ((X \vee \neg Y) \wedge \neg(Z \to (X \wedge Y))) \vee (\neg Z + Y)$.
Для полученного дерева определите прямой, обратный и инфиксный обходы.

Задача 10.10. *Постройте дерево и ациклический ориентированный граф, представляющие следующую арифметическую формулу*
$\Phi = (a + b)/(c + a * d) + ((c + a * d) - (a + b) * (c - d))$.

Сколько вершин удалось сократить?