Приложение 2

Концепция эквивалентности форм представления бухгалтерской информации и алгоритмов ее преобразования

В настоящем приложении рассматривается фундаментальная проблема эквивалентности разнопредставленных форм и алгоритмов преобразования информации, которая, вообще говоря, не является новой в науке и имеет отношение ко всем без исключения сферам человеческой деятельности, где необходимо решать вопросы сходства или различия; там, где замещение упрощенным эквивалентом реального и потому сложного в своей структуре объекта позволяет решить проблему, которую невозможно было бы решить иным образом. Именно в этом, упомянутым последним качестве, эквивалентирование, т.е. сведение разнопредставленных форм и алгоритмов преобразования информации к подобным им, но более простым для изучения и понимания, имеет непосредственное отношение к моделированию как одному из эффективных методов научных и экспериментальных исследований.

В бухгалтерском учете, если сравнить его с любой другой сферой деятельности, проявления эквивалентности наиболее рельефны и многочисленны, поскольку разнопредставленность одной и той же информации и алгоритмов, приводящих к одному и тому же результату, имманентно присущи именно бухгалтерскому учету и составляют его суть в историческом развитии: «от истоков до наших дней». Достаточно, например, сказать, что сам принцип двойственного отражения операций, на котором базируется весь бухгалтерский учет, — это и есть пример эквивалентного разнопредставления одной и той же информации с помощью двух корреспондирующих между собой логических конструкций — счетов бухгалтерского учета. Можно также привести и множество других примеров: различные, но эквивалентные синтаксические формы записи проводок; различные, но эквивалентные формы регистров бухгалтерского учета; различные, но эквивалентные по результатам — *эквифинальные*[22] формы контроля за достоверностью учетной информации; различные, но эквифинальные методы определения финансовых результатов, и т.д.

В то же время, проблема эквивалентности форм представления бухгалтерской информации и алгоритмов ее преобразования, встречающаяся буквально на каждом шагу в теории и практике бухгалтерского учета, до сих пор даже не обозначена, по крайней мере, в той постановке, которая предлагается в настоящей работе. Но близкие к ней проблемы подобия и сравнимости (сопоставимости) так или иначе рассматриваются в подавляющем большинстве работ по бухгалтерскому учету, аудиту и финансовому анализу, приведенных в списке литературы. Как будет видно из дальнейшего, предлагаемый путь решения поставленной таким образом проблемы — это, по существу, и есть путь к решению центральной проблемы — адэкватности предлагаемой системы моделей бухгалтерского учета ее объектам в существующей системе бухгалтерского учета институционных единиц.

С другой стороны, постановка и рассмотрение обозначенной проблемы, а главное — определение критериев, в соответствии с которыми формы представления информации и алгоритмы ее преобразования могут считаться эквивалентными, имеет непосредственное практическое значение. Связанные с этой проблемой вопросы во многих случаях воспринимаются и решаются неосознанно, т.е. на интуитивном уровне, что приводит к дискуссиям там, где их могло бы и не быть, а иногда выливается и в не столь безобидные последствия в виде нечетких стандартов и инструкций, и, как следствие, — грубых ошибок по их применению в бухгалтерском учете.

Ниже с помощью достаточно простых определений и вытекающих из них утверждений вводятся основные положения предлагаемой в настоящем приложении концепции эквивалент-

22. В дальнейшем наряду с термином «эквивалентный» вместо словосочетания «эвивалентный по результатам» будем использовать термин «эквифинальность», как более краткий. Термин заимствован из менеджмета, где под эквифинальностью понимается существование различных путей для достижения одной и той же цели. Однако, как будет видно из дальнейшего, понятие эквивалентности алгоритмов в общем случае не совпадает с понятием их эквифинальности.

ности форм представления и алгоритмов преобразования информации, которые, как будет видно из дальнейшего, имеют и могут иметь многообразные приложения в бухгалтерском учете.

Вначале введем следующее определение эквивалентности двух любых форм представления информации F_1 и F_2:

Определение 1. *Две формы представления информации* F_1 и F_2 *эквивалентны, если существует прямой алгоритм* A_{12}, *преобразующий* F_1 в F_2 *и обратный к нему алгоритм* A_{21}, *преобразующий* F_2 в F_1.

Если ввести обозначение эквивалентности «$\leftrightarrow$» и преобразования записывать в виде $F1 \xrightarrow{A_{12}} F2$ и, $F2 \xrightarrow{A_{21}} F1$ то определение можно записать сжато:

Определение 1′. $F1 \leftrightarrow F2$, если $\exists$ (существует) A_{12} и A_{21}: $F1 \xrightarrow{A_{12}} F2$ и $F2 \xrightarrow{A_{21}} F1$.

Введенное определение *эквивалентности* — это отношение изоморфизма для которого характерно существование как *прямого*, так и *обратного* преобразования (алгоритма).

Преобразования только в одну сторону с точки зрения введенного определения *неэквивалентны*; они, в отличие от эквивалентных — *изоморфных,* называются *гомоморфными* преобразованиями.

Введенное определение эквивалентности, если оно корректно, должно обладать тремя присущими ему свойствами:

Рефлексивность: «Любая форма эквивалентна самой себе: $F \leftrightarrow F$».

Действительно, если копирование формы, т.е. некий нейтральный алгоритм, подобный сложению с нулем или умножением на 1, можно рассматривать как одно из возможных ее преобразований, то, очевидно, что переобозначив $F1 = F$ и $F2 = F$, из исходного определения получим требуемое: $F \leftrightarrow F$.

Симметричность: «Если $F1 \leftrightarrow F2$, то $F2 \leftrightarrow F1$».

Из определения следует, что $F1 \leftrightarrow F2$, если $F1 \xrightarrow{A_{12}} F2$ и

$F2 \xrightarrow{A_{21}} F1$, а также $F2 \leftrightarrow F1$, если $F2 \xrightarrow{A_{21}} F1$ и $F1 \xrightarrow{A_{12}} F2$. Но это и означает симметричность введенного отношения эквивалентности двух форм представления информации.

Транзитивность: «Если $F1 \leftrightarrow F2$ и $F2 \leftrightarrow F3$, то $F1 \leftrightarrow F3$».

В соответствии с определением: $F1 \leftrightarrow F2$, если $F1 \xrightarrow{A_{12}} F2$ и $F2 \xrightarrow{A_{21}} F1$. В свою очередь: $F1 \leftrightarrow F3$, если $F2 \xrightarrow{A_{23}} F3$ и $F3 \xrightarrow{A_{32}} F2$. Таким образом, даже, если не существует непосредственного преобразования: $F1 \xrightarrow{A_{13}} F3$ и $F3 \xrightarrow{A_{31}} F1$, то алгоритмы A_{13} и A_{31} можно определить как составные, т.е. как последовательность соответствующих алгоритмов: $A_{13} = A_{12}\,A_{23}$ и $A_{31} = A_{32}\,A_{21}$, которые в прямом и обратном направлениях выполнят преобразование форму $F_1 \to F_2 \to F_3$ и $F_3 \to F_2 \to F_3$, а это и доказывает свойство транзитивности введенного определения эквивалентности форм представления информации.

Приведем несколько простых примеров из области бухгалтерского учета, иллюстрирующих универсальность и многообразие приложений введенного выше определения эквивалентности форм представления бухгалтерской информации.

Пример 1. Журналы операций:

Журнал операций — форма F_1

№	Дата	Корреспонденция		Сумма	Содержание
		Дебет	Кредит		
(1)	(2)	(3)	(4)	(5)	(6)

Журнал операций — форма F_2

№	Дата	Содержание	Сумма	Корреспонденция	
				Дт	Кт
(1)	(2)	(3)	(4)	(5)	(6)

В соответствии с введенным определением, эти формы журнала операций эквивалентны, так как существует прямой $F1 \xrightarrow{A_{12}} F2$ и обратный к нему алгоритм $F2 \xrightarrow{A_{21}} F1$. Прямой алгоритм A_{12} состоит в копировании содержимого граф формы F_1 в соответствующие графы формы F_1: $(1)_1 \to (1)_2$, $(2)_1 \to (2)_2$, $(3)_1 \to (5)_2$, $(4)_1 \to (6)_2$, $(5)_1 \to (4)_2$, $(6)_1 \to (3)_2$. Обратный алгоритм A_{21}: $(1)_2 \to (1)_1$, $(2)_2 \to (2)_1$, $(3)_2 \to (6)_1$, $(4)_2 \to (5)_1$, $(5)_2 \to (3)_1$, $(6)_2 \to (4)_1$. Здесь подстрочный индекс 1 или 2 указывает, соответственно, к какой форме относится показатель: к F_1 или к $F_{2.}$

Пример 2. Накладные:

Накладная — форма F_1

Товар	Ед. изм.	Кол-во	Цена	Сумма
(1)	(2)	(3)	(4)	(3) × (4) = (5)

Накладная — форма F_2

Товар	Ед. изм.	Кол-во	Сумма
(1)	(2)	(3)	(4)

Накладная — форма F_3

Товар	Ед. изм.	Цена	Сумма
(1)	(2)	(3)	(4)

Покажем эквивалентность представленных выше трех форм накладной F_1, F_2, F_3. Для этого достаточно показать, что $F_1 \leftrightarrow F_2$ и $F_2 \leftrightarrow F_3$, поскольку отсюда, согласно свойству транзитивности, следует: $F_1 \leftrightarrow F_3$.

Действительно, так как $\exists$ преобразование (алгоритм), которое сводится к копированию: $(1)_1 \to (1)_2$, $(2)_1 \to (2)_2$, $(3)_1 \to (3)_2$, $(5)_1 \to (4)_2$. Обратный алгоритм A_{21} также существует: $F2 \xrightarrow{A_{21}} F1$. Он сводится к копированию: $(1)_2 \to (1)_1$, $(2)_2 \to (2)_1$, $(3)_2 \to (3)_1$, $(4)_2 \to (5)_1$ и к расчету: $(4)_1 = (5)_1/(3)_1$.

Аналогично можно доказать эквивалентность форм F_2 и F_3. Прямой алгоритм $F2 \xrightarrow{A_{23}} F3:$ $(1)_1 \to (1)_2$, $(2)_1 \to (2)_2$, $(4)_1 / (3)_1 \to (3)_2$, $(4)_1 \to (4)_2$. Нетрудно также убедиться в существовании обратного алгоритма преобразования:

$$F3 \xrightarrow{A_{32}} F2 : (1)_2 \to (1)_1, (2)_2 \to (2)_1, (4)_2 / (3)_2 \to (3)_1, (4)_2 \to (4)_1.$$

Поскольку теперь доказано, что $F_1 \leftrightarrow F_2$ и $F_2 \leftrightarrow F_3$, то отсюда, в соответствии с доказанным выше свойством транзитивности, следует, что также и $F_1 \leftrightarrow F_3$, что, впрочем, можно доказать и непосредственно, показав, как это было сделано выше, существование прямого: $F1 \xrightarrow{A_{13}} F3$ и обратного к нему преобразования: $F3 \xrightarrow{A_{31}} F1$.

Таким образом, внешне отличающиеся формы накладной F_1, F_2 и F_3, тем не менее, эквивалентны, поскольку содержат одну и ту же информацию. Формы F_2 и F_3 содержат минимально необходимую информацию о товарах (поступивших или выбывших). Форма F_1 избыточна по сравнению с ними, так как содержит зависимость между ценой, количеством и суммой: (5) = (3) × (4).

Пример 3 — Алгебраический и бухгалтерский балансы

Форма F_1 — Алгебраический

Статьи	Остаток
	(+,–)
Средства	+7000
Обязательства	–2000
Капитал	–5000
ИТОГО:	0

$A_{12} \rightarrow$ $\leftarrow A_{21}$

Форма F_2 — Бухгалтерский баланс

Счета	Остаток	
	Дебет	Кредит
Средства	7000	0
Обязательства	0	2000
Капитал	0	5000
ИТОГО:	7000	7000

Алгоритм перехода от алгебраического к бухгалтерскому балансу A_{12} состоит в следующем: числа со знаком «+» записываем в левую колонку по дебету, числа со знаком «–» умножаем на –1 и записываем в правую колонку по кредиту. Итоги по дебету и кредиту всегда сходятся, так как суммы положительных и отрицательных чисел по модулю равны между собой.

Обратный алгоритм A_{21}, преобразующий бухгалтерский баланс в алгебраический, также существует: ненулевые числа по дебету записываем в алгебраическом балансе со знаком «+», ненулевые числа по кредиту умножаем на –1 и записываем таким образом со знаком «–». Сумма алгебраического баланса будет равна нулю, так как итоги по дебету и кредиту равны между собой.

Доказанный факт эквивалентности двух или более форм представления информации позволяет выбрать одну из них из соображения наглядности или по другим причинам, например, из-за возможности использования более эффективного алгоритма преобразования в сравнении с другой, эквивалентной ей формой.

Эквивалентность внешне различных форм указывает на существование *инварианта*, т.е. независимой от преобразований сущности, в качестве которой, по-видимому, выступает идентичность *содержания* сравниваемых через отношение эквивалентности форм представления информации. Таким образом, через отношение эквивалентности (конгруэнтности, изоморфизма) раскрывается сущность принципа приоритета содержания над формой, но, что еще более существенно, математически формулируются критерии, позволяющие выявить идентичность, т.е. *инвариантность* — независимость содержания от формы, в которую она заключена.

В дальнейшем для обсуждения вопросов эквивалентности форм бухгалтерского учета понадобится обобщение введенного определения 1 на множество форм представления информации.

Определение 2 Два множества форм представления информации $\{F_{1i} | i = 1,2,\ldots,n\}$ и $\{F_{2j} | j = 1,2,\ldots,m\}$ эквивалентны, если существует прямой алгоритм A_{12}, преобразующий множество форм $\{F_{1i}\}$ во множество форм $\{F_{2j}\}$ и обратный к нему алгоритм A_{21}, преобразующий множество $\{F_{2j}\}$ во множество $\{F_{1i}\}$.

То же самое, но в краткой записи:

<u>Определение 2′</u>. $\{F_{1i} \mid i = 1,2,\ldots,n\} \leftrightarrow \{F_{2j} \mid j = 1,2,\ldots,m\}$, если $\exists$ алгоритмы A_{12} и A_{21}: $\{F_{1i}\} \xrightarrow{A_{12}} \{F_{2j}\}$ и $\{F_{2j}\} \xrightarrow{A_{21}} \{F_{1i}\}$.

И это определение, данное в форме 2 и 2′ и обобщенное уже на множество форм, также, как нетрудно убедиться, обладает тремя свойствами, присущими отношениям эквивалентности: *рефлексивностью*, *симметричностью* и *транзитивностью*.

Перед тем, как перейти к обсуждению проблемы эквивалентности уже самих алгоритмов преобразования информации, введем определение входных и выходных форм, т.е. терминов, которые часто используются в работах по моделированию и автоматизации учета, но обычно считаются неопределяемыми. Вместе с тем, используя предлагаемый подход, нетрудно придти к следующему их определению:

<u>Определение 3</u>. *Форма* F_2 *называется выходной по отношению к входной форме* F_1, *если существует преобразование* $F1 \xrightarrow{A_{12}} F2$.

Для эквивалентных форм отношение «входная-выходная формы» *симметрично*, т.е. если $F_1 \leftrightarrow F_2$, то F_2 является выходной по отношению к F_1 и, наоборот F_1 является выходной по отношению к F_2. В общем случае отношение «входная-выходная форма» асимметрично, поскольку преобразование возможно только в одну сторону, т.е. отношение «входная-выходная» формы в общем случае гомоморфно. Например, преобразования: «журнал операций» $\rightarrow$ «оборотно-сальдовый баланс», а также «оборотно-сальдовый баланс» $\rightarrow$ «сальдовый баланс» — это примеры неэквивалентных, т.е. гомоморфных преобразований, поскольку обратные преобразования в общем случае не существуют. Введем теперь понятие эквивалентности двух алгоритмов A и A′.

<u>Определение 4</u>. Два алгоритма A и A′ эквивалентны, если на основе одной и той же информации с помощью каждого из них

будет получен один и тот же результат или, в краткой записи: «$A \leftrightarrow A'$, если $F1 \xrightarrow{A} F2$ и $F2 \xrightarrow{A'} F1$».

Представленный ниже ориентированный граф иллюстрирует введенное выше определение эквивалентности двух алгоритмов:

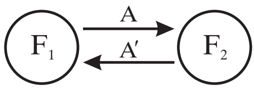

Рис. 2.1 Граф, иллюстрирующий определение эквивалентности двух алгоритмов.

Данное выше определение, как нетрудно показать, также обладает тремя необходимыми свойствами отношения эквивалентности:

Рефлексивность: «Кажый алгоритм эквивалентен сам себе, т.е всегда $A \leftrightarrow A$».

Действительно, $A \leftrightarrow A$, так как при повторном применении алгоритма А получаем одинаковые результаты: $F1 \xrightarrow{A} F2$ и $F1 \xrightarrow{A} F2$.

Симметричность: «Если $A \leftrightarrow A'$, то $A' \leftrightarrow A$».

По определению $A \leftrightarrow A'$, если $F1 \xrightarrow{A} F2$ и $F1 \xrightarrow{A'} F2$. С другой стороны, $A' \leftrightarrow A$, если $F1 \xrightarrow{A'} F2$ и $F1 \xrightarrow{A} F2$, но то же самое следует из первого, т.е. отношение эквиалентности, введенное предложенным определением, симметрично.

Транзитивность: «Если $A \leftrightarrow A'$ и $A' \leftrightarrow A''$, то $A \leftrightarrow A''$ ».

В соответствии с определением: $A \leftrightarrow A'$, поскольку существуют преобразования: $F1 \xrightarrow{A} F2$ и $F1 \xrightarrow{A'} F2$. В свою очередь, $A' \leftrightarrow A''$, поскольку $F1 \xrightarrow{A'} F2$ и $F1 \xrightarrow{A''} F2$. Но это означает, что существует пара преобразований: $F1 \xrightarrow{A} F2$ и $F1 \xrightarrow{A''} F2$, которые одинаковы по результату F_2 относительно формы F_1. Поэтому $A \leftrightarrow A''$, что и доказывает транзитивность введенного отношения эквивалентности алгоритмов.

Необходимо подчеркнуть, что об эквивалентности в смысле введенного определения можно говорить в том и только в том случае, когда сравниваемые алгоритмы имеют одну общую область определения, представленную формой F_1 и одну общую область отображения, представленную формой F_2.

Для более внимательного рассмотрения этих условий рассмотрим теперь ситуацию (см. рис. 2.2). Здесь для того, чтобы применить введенное выше определение 4, необходимо выделить соответствующие цепочки алгоритмов, исходящих из одной и входящих в одну общую форму. В данном случае можно выделить четыре пары эквивалентных (эквифинальных) алгоритмов:

- Относительно входной формы F_1 и по результату ее преобразования в выходную форму F_3: $A_{13} \leftrightarrow A_{12}A_{24}A_{43}$;
- Относительно входной формы F_1 и по результату ее преобразования в выходную форму F_4: $A_{13}\ A_{34} \leftrightarrow A_{12}A_{24}$;
- Относительно входной формы F_2 и по результату ее преобразования в выходную форму F_3: $A_{21}A_{13} \leftrightarrow A_{24}A_{43}$;
- Относительно входной формы F_2 и по результату ее преобразования в выходную форму F_4: $A_{24} \leftrightarrow A_{12}A_{24.}$

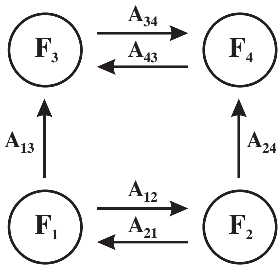

Рис. 2.2 Граф, иллюстрирующий определение эквивалентности и эквифинальности алгоритмов при условии эквивалентности входных и выходных форм представления информации.

Таким образом, можно считать доказанным следующее утверждение:

<u>Утверждение 1</u>. Если $F1 \leftrightarrow F2$ и $F3 \leftrightarrow F4$, и при этом существуют алгоритмы A_{13} и A_{24} такие, что $F1 \xrightarrow{A_{13}} F3$ и $F2 \xrightarrow{A_{24}} F4$, то существуют четыре пары эквивалентных алгоритмов:

1) $A_{13} \leftrightarrow A_{12}A_{24}A_{43}$;
2) $A_{13}A_{34} \leftrightarrow A_{12}A_{24}$;
3) $A_{21}A_{13} \leftrightarrow A_{24}A_{43}$;
4) $A_{24} \leftrightarrow A_{12}A_{24}$.

В то же время, следует понимать, что с позиций введенного определения некорректно говорить, например, об эквивалентности алгоритма A_{13}: $F1 \xrightarrow{A_{13}} F3$ и алгоритма $A_{23} = A_{24}A_{43}$: $F2 \xrightarrow{A_{24}} F4 \xrightarrow{A_{43}} F3$, несмотря на то, что результаты преобразований с помощью этих алгоритмов совпадают.

С точки зрения введенного выше определения 4 эквивалентности алгоритмов рассматриваемая пара алгоритмов A_{13} и $A_{23} = A_{24}A_{43}$ не может сравниваться по результату — выходной форме F_3, поскольку преобразуются *различные* входные формы

F_1 и F_2. Но здесь как раз и видна разница между *эквивалентностью* алгоритмов и их *эквифинальностью*: в этом смысле алгоритмы A_{13} и A_{23} *эквифинальны*, поскольку результаты их преобразований совпадают, но они *неэквивалентны*, поскольку эти результаты получены на основе разнопредставленных исходных данных.

В целях иллюстрации рассмотрим следующий простой пример: сравним два алгоритма — формулы A, A′ расчета налога на добавленную стоимость от исходной суммы X без включения в нее НДС, и суммы Y, включающей НДС. В обоих случаях используется одна и та же ставка *c*, равная, например, 0.2 (20%) или 0.1 (10%). В первом случае — алгоритм A, как известно, используется формула НДС $= X \cdot c$, во втором случае после преобразования: $Y = X + X \cdot c = X \cdot (1 + c)$ получаем расчетную формулу — алгоритм A′: НДС $= Y \cdot c'$, где $c' = c/(1 + c)$ — ставка включенного НДС. В каждом из этих случаев расчет ведется от *различных исходных данных*: в первом случае от формы представления $F_1 = \{X,c\}$, во втором случае — от формы представления $F_2 = \{Y,c'\}$. Соответственно, и результаты в общем виде будут представлены в двух внешне различных формах: $F_3 = X \cdot c$ и $F_4 = Y \cdot c'$. Рассматриваемой ситуации соответствует граф, ранее представленный в общем виде (см. рис.2.2).

Поскольку входные и выходные формы представления информации в данном случае эквивалентны, то в соответствии с *утверждением* 1 (см. выше) можно заранее предсказать, что существуют перечисленные выше четыре пары эквивалентных алгоритмов расчета НДС:

1. относительно данных в форме $F_1 = \{X,c\}$ и результата в форме $F_3 = X \cdot c$:

$$A_{13} \leftrightarrow A_{12}\,A_{24}\,A_{43}$$

2. относительно данных в форме $F_1 = \{X,c\}$ и результата в форме $F_4 = Y \cdot c'$:

$$A_{12}\,A_{24} \leftrightarrow A_{24}$$

3. относительно данных в форме $F_2 = \{Y,c'\}$ и результата в форме $F_3 = X \cdot c$:

$$A_{24}\,A_{43} \leftrightarrow A_{21}\,A_{13}$$

4. относительно данных в форме F_2 и результата в форме F_4:

$$A_{24} \leftrightarrow A_{21}\,A_{13}\,A_{34}$$

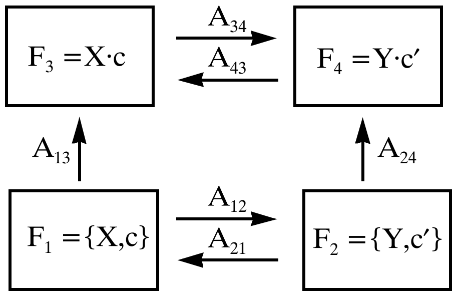

Рис. 2.3 Граф, иллюстрирующий эквивалентность и эквифинальность алгоритмов расчета НДС.

Остается только определить каждую из этих пар алгоритмов. Например, для первой пары: $A_{13} \leftrightarrow A_{12}A_{24}A_{43}$ имеем: A_{13}: $F_3 = X \cdot c$ и эквивалентную ему цепочку алгоритмов (формул):

A_{12}: $F_2 = \{Y = X/(1 + c), c' = c/(1 + c)\}$;

A_{24}: $F_4 = Y \cdot c'$;

A_{43}: $F_4 = Y \cdot c' = X \cdot (1 + c) \cdot c/(1 + c) = X \cdot c = F_3$,
поскольку по определению: $Y = X \cdot (1 + c)$ и $c' = c/(1 + c)$.

Аналогично можно определить и три остальные эквивалентные пары алгоритмов. Следует еще раз подчеркнуть, что в данном случае некорректно было бы говорить об эквивалентности алгоритмов расчета от стоимости с включенным и невключенным НДС, т.е. о непосредственной эквивалентности формул: $F_3 = X \cdot c$ и $F_4 = Y \cdot c'$, поскольку расчеты велись от различной базы исходных данных, соответственно от $F_1 = \{X, c\}$ и от $F_2 = \{Y, c'\}$, несмотря на то, что результаты этих расчетов совпадают, поскольку сравниваемые алгоритмы в соответствии с вышеизложенным эквифинальны, но неэквивалентны.

Здесь также можно привести пример, когда сгруппированные по-разному одни и те же данные в форме F_1 и F_2 дадут один и тот же результат: $F1 \xrightarrow{A} F2$ и $F2 \xrightarrow{A'} F3$, но поскольку формы F_1 и F_2 неэквивалентны, т.е. не могут быть непосредственно преобразованы друг в друга, то согласно введенному определению алгоритмы A_{13} и A_{23} не будут эквивалентными.

Если это одни и те же данные, то при наличии исходной базы данных — формы F_0 можно показать, что существуют алгоритмы A и A', которые эквивалентны, хотя формы F_1 и F_2 и не могут быть непосредственно преобразованы друг в друга. Сказанное выше можно проиллюстрировать следующим графом:

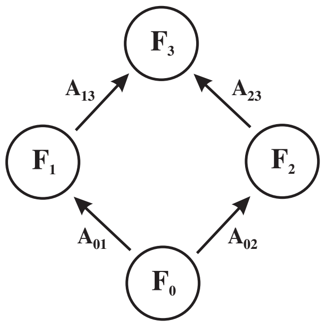

Рис.2.4 Граф, иллюстрирующий эквивалентность алгоритмов при наличии общей базы исходных данных.

На представленном выше рис. 2.4 эквивалентными следует считать алгоритмы: $A = A_{01}A_{13}$ и $A' = A_{02}A_{23}$, т.е. $A \leftrightarrow A'$, то время, как алгоритмы A_{13} и A_{23}, входящие в состав алгоритмов A и A', таковыми не являются, но они эквифинальны.

Это обстоятельство согласуется с *принципом осторожности*, поскольку при отсутствии общей исходной базы нет возможности проверить, что формы F_1 и F_2 получены преобразованием одних и тех же данных, т.е., что они имеют одну и ту же область определения (см. ниже рис.2.5).

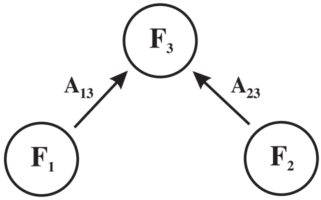

Рис. 2.5 Граф, иллюстрирующий неэквивалентность, но эквифинальность алгоритмов A_{13} и A_{23} при отсутствии общей базы исходных данных.

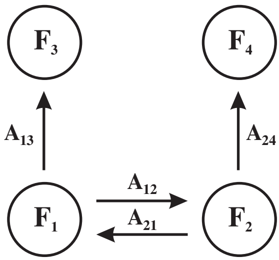

Рис. 2.6 Граф, иллюстрирующий неэквивалентность, но эквифинальность алгоритмов для случая эквивалентности входных и неэквивалентности выходных форм

Все сказанное относительно эквивалентности алгоритмов с общей исходной базой данных можно сформулировать в виде следующего утверждения:

<u>Утверждение 2</u>. Если $F1 \xrightarrow{A_{13}} F3$ и $F2 \xrightarrow{A_{23}} F3$ и при этом имеется общая исходная база данных — форма F_0, такая, что: $F0 \xrightarrow{A_{01}} F1$ и $F0 \xrightarrow{A_{02}} F2$, то существует одна и только одна пара эквивалентных алгоритмов: $A_{01}A_{13} \leftrightarrow A_{02}A_{23}$.

<u>Следствие из утверждения 2</u>. Если же исходная база — форма F_0 отсутствует, то в соответствии с принципом осторожности нельзя говорить о существовании эквивалентных алгоритмов даже, если существует преобразование:

$F1 \xrightarrow{A_{13}} F3$ и $F2 \xrightarrow{A_{23}} F3$.

В то же время, если даже при отсутствии общей базы исходных данных формы F_1 и F_2 допускают взаимно-однозначное преобразование, то имеем две пары алгоритмов, которые являются эквивалентными, что иллюстрируется следующим графом:

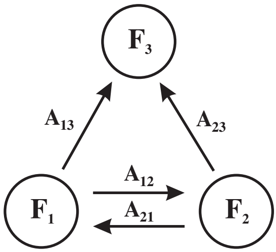

Рис. 2.7 Граф, иллюстрирующий эквивалентность алгоритмов при условии эквивалентности входных форм.

Здесь эквивалентными будут следующие две пары алгоритмов:

- Алгоритмы эквивалентные относительно формы F_1 по результату — форме F_3: $A_{13} \leftrightarrow A_{12}A_{23}$;
- Алгоритмы эквивалентные относительно формы F_2 по результату — форме F_3: $A_{23} \leftrightarrow A_{21}A_{23}$.

Таким образом, можно сформулировать следующее утверждение:

Утверждение 3. Если $F1 \xrightarrow{A_{13}} F3$ и $F2 \xrightarrow{A_{23}} F3$ и при этом $F1 \leftrightarrow F2$, то существуют две пары эквивалентных алгоритмов: 1) $A_{13} \leftrightarrow A_{12}A_{23}$ и 2) $A_{23} \leftrightarrow A_{21}A_{23}$.

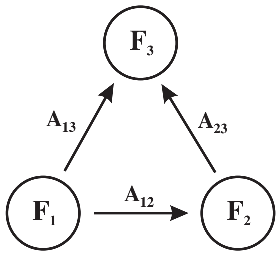

Рис. 2.8 Граф, иллюстрирующий эквивалентность алгоритмов при условии гомоморфного преобразования исходной формы

И, наконец, рассмотрим ситуацию эквивалентности двух алгоритмов при условии, что исходная форма F_1 гомоморфно, т.е. в одну сторону, преобразуется в промежуточную форму F_2. Приведенный выше рис. 2.8 иллюстрирует следующее ниже утверждение:

Утверждение 4. Если $F1 \xrightarrow{A_{13}} F3$ и $F2 \xrightarrow{A_{23}} F3$ и при этом $F1 \xrightarrow{A_{12}} F2$ гомоморфно, т.е. обратного ему преобразования A_{21} не существует, то существует одна и только одна пара эквивалентных алгоритмов: $A_{13} \leftrightarrow A_{12}A_{23}$.

Отметим, что и в последнем случае алгоритмы $A = A_{13}$ и $A' = A_{12}A_{23}$ эквивалентны относительно одной и той же исходной формы F_1.

Все это позволяет сделать следующий вывод: ставить вопрос об эквивалентности (подобии, изоморфизме) двух алгоритмов (преобразований или учетных схем) можно в том и только в том случае, если они имеют общую базу данных — общую форму представления, определенную на одних и тех же данных. Однако это условие необходимое для *постановки*, но недостаточное для *решения* вопроса их эквивалентности. Достаточным является только условие, что оба алгоритма приводят к одному и тому же результату.

Отметим, что провести такое доказательство для любого числового наполнения формы, достаточно сложно и требует в каждом частном случае определенных усилий. Поэтому, как правило, доказательство того, что получен один и тот же результат, сводится к решению соответствующего числового примера, но такие действия не должны, очевидно, считаться доказательством, хотя именно бухгалтерский учет изобилует примерами такого подхода. При этом под одним и тем же результатом не следует в общем случае понимать простое совпадение числовых значений, а именно, и это особенно характерно для бухгалтерского учета, результатом может быть *инвариант*, например, тождественное равенство (баланс) оборотов по дебету и кредиту (первый постулат Пачоли), баланс итогов дебетовых и кредитовых остатков (второй постулат Пачоли), баланс валюты актива и пассива и т.п. Термин «инвариант» или точнее, «инвариантное соотношение», употребляется в настоящей работе для обозначения того фундаментального для бухгалтерского учета факта, что *инвариант* — в нашем примере балансовое равенство — не зависит от конкретного числового наполнения исходной формы и не зависит от того, какие в результате получились числа, представляющие, например, слева и справа факт балансового равенства актива и пассива.

С другой стороны, проблему эквивалентности, на наш взгляд, не следует отождествлять с проблемой сопоставимости (сравнимости) форм представления и алгоритмов преобразования информации. Однако эквивалентность для сопоставимости имеет такое же значение как знак равенства при сравнении двух величин, т.е. эквивалентность играет роль границы — точки отсчета, относительно которой определяется сходство и различие двух форм представления информации или алгоритмов ее преобразования.При этом, чтобы быть корректным, при сопоставлении, например, двух алгоритмов, следует исходить, по крайней мере, из следующих посылок:

- Алгоритмы созданы для решения одной и той же задачи;
- Они должны при сопоставлении иметь одну и ту же исходную базу данных;
- И, наконец, самое главное — необходим критерий, по которому должны сравниваться результаты алгоритмов.

Например, при сопоставлении двух учетных схем списания расходов, например, LIFO и FIFO, эти условия выполняются:

- Алгоритмы — учетные схемы созданы для решения одной и той же задачи списания расходов;
- Всегда можно при сопоставлении использовать одни и те же данные — общую исходную базу данных;
- Критерием сопоставления этих учетных схем является влияние их на финансовый результат — прибыль предприятия при соблюдении принципа «прочих равных условий».

Отметим, что те же самые требования должны соблюдаться и при сопоставлении учетных политик предприятия в целях выбора одной из них по сформулированному критерию, а также и в других сопоставлениях.

Вместе с тем, во многих случаях, например, при сопоставлении отечественной и зарубежной систем бухгалтерского учета, не все требования корректности сопоставления могут быть, строго говоря, соблюдены. Здесь не могут быть выполнены, прежде всего, первые два условия:

- правила учета, определяющие алгоритмы формирования финансовых отчетов России и МСФО (или GAAP), различны, т.е. алгоритмы предназначены для решения разных задач;
- базы данных, если они записаны как проводки, также различны.

В то же время, при должной организации учета, как показано в работе З.В. Кирьяновой и Е.В. Однинушкиной удается избежать параллельного ведения учета по российской системе и по системе GAAP. Для этого в российский регистр заносятся консолидированные бухгалтерские записи с помощью довольно оригинальной системы кодирования счетов. Например, в российской системе общепроизводственные расходы (зарплата) имеет аналитический код 252, в GAAP — 630. Отсюда консолидированный код для России будет 252630, для GAAP — консолидированный код получаем инверсией: 630252. Поскольку бухгалтерские программы различают счета только по первым трем цифрам, то консолидированный код России будет восприниматься как обычный трехзначный код счета «Общепроизводственные расходы (зарплата)», аналогично программа, обрабатывающая данные по GAAP воспримет только 630, т.е. код который обозначает аналогичный счет в GAAP. Таким образом, благодаря формированию общей базы данных достигается консолидация, а точнее сказать, трансформация российской главной книги в международную главную книгу, а затем и в финансовые отчеты GAAP или МСФО.