First-past-the-post Games

Roland Backhouse

School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, England, United Kingdom

Abstract

Informally, a first-past-the-post game is a (probabilistic) game where the winner is the person who predicts the event that occurs first among a set of events. Examples of first-past-the-post games include so-called block and hidden patterns and the Penney-Ante game invented by Walter Penney. We formalise the abstract notion of a first-past-the-post game, and the process of extending a probability distribution on symbols of an alphabet to the plays of a game. We establish a number of properties of such games, for example , the property that an incomplete first-past-the-post game is also a first-past-the-post game.

Penney-Ante games are multi-player games characterised by a collection of regular, prefix-free languages. Analysis of such games is facilitated by a collection of simultaneous (non-linear) equations in languages. Essentially, the equations are due to Guibas and Odlyzko. However, they did not formulate them as equations in languages but as equations in generating functions detailing lengths of words. For such games, we show how to use the equations in languages to calculate the probability of winning and how to calculate the expected length of a game for a given outcome. We also exploit the properties of first-past-the-post games to show how to calculate the probability of winning in the course of a play of the game. In this way, we avoid the construction of a deterministic finite-state machine or the use of generating functions, the two methods traditionally used for the task.

We observe that Aho and Corasick's generalisation of the Knuth-Morris-Pratt pattern-matching algorithm can be used to construct the deterministic finite-state machine that recognises the language underlying a Penney-Ante game. The two methods of calculating the probabilities and expected values, one based on the finite-state machine and the other based on the non-linear equations in languages, have been implemented and verified to yield the same results.

Keywords: Algorithmic problem solving, Penney-Ante, Regular language, Probabilistic game, Generating function

A first-past-the-post game is a game where the winner is the person who predicts the event that occurs first among a set of events. There is no limit to the examples that can be invented. One example, which was published in the Communications of the ACM just before the final version of this paper was prepared [Win13], is the following:

> "Alice and Bob roll a single die repeatedly. Alice is waiting until all six of the die's faces appear at least once. Bob is waiting for some face (any face) to appear four times. The winner is the one who gets his or her wish first; for example, if the successive rolls are 2 , 5 , 4 , 5 , 3 , 6 , 6 , 5 , 1 , then Alice wins, since all numbers have appeared, none more than three times. If the successive rolls instead happen to be 4 , 6 , 3 , 6 , 6 , 1 , 2 , 2 , 6 , then Bob wins because he has seen four 6 s and no 5 (yet)."

Different versions of this game are obtained by varying the numbers six and four in the specification; see example 6.

Email address: `roland.backhouse@nottingham.ac.uk` (Roland Backhouse)

Perhaps the most widely studied example of a first-past-the-post game is the game that is now called Penney-Ante, a game with pennies named after its inventor, Walter Penney [Pen74]. In the game, each player chooses a coin sequence (a sequence of "head"s or "tail"s); a coin is then repeatedly tossed until one of the sequences occurs. The winner is the player who chose the sequence that occurs first.

Typically in all such games, the interest is in determining the probability of winning for each of the players and the expected length of the games (assuming a fair roll of the die or toss of the coin).

The two-player Penney-Ante game has attracted much interest because it is non-transitive [Gar74]; the game is also used to demonstrate the use of generating functions in the calculation of probability distributions [GO81, GKP89]. Our interest in the game began as a simple, (for us) introductory exercise in probability generating functions. It has turned out to be an exercise in applying the calculational method to the analysis of the game in the general case of an arbitrary number of players, with the added value of new insights in the formalisation of complex probabilistic events.

Analysis of the Penney-Ante game is facilitated by a collection of simultaneous (non-linear) equations between languages. In the literature, either the equations are stated without proof [GKP89] or the equations are not given explicitly but translated directly into generating functions detailing lengths of words [GO81]. The primary contribution of this paper is to record a derivation of the equations and the associated probability distributions in which naming of word length and the use of generating functions is avoided.

Our derivation has several novel features. We introduce the abstract notion of a first-past-the-post game, and we formalise the process of extending a probability distribution on symbols of an alphabet to the plays of such a game (section 2). (Multi-player) Penney-Ante games and so-called block and hidden patterns [FS09] are shown to be instances of first-past-the-post games. Such games are characterised by a collection of regular, prefix-free languages. We derive a collection of simultaneous non-linear equations in these languages and use these to show how to calculate the probability of winning (section 4).

The equations are essentially the basis for the equations in generating functions derived by Guibas and Odlyzko [GO81]. The formula we derive generalises a formula attributed to Conway [Gar74] for the original two-player Penney-Ante game. Another instance is the formula due to Solov'ev [Sol66] for the expected number of coin tosses until a given (contiguous) pattern appears. Like Guibas and Odlyzko [GO81], we also consider the generalisation of Penney-Ante games to an arbitrary number of players; see subsection 4.2. Subsection 4.3 briefly discusses how to obtain equations for the expected length of the game that ends with a given outcome.

We show in section 4.4 that the equations in languages do not have a unique solution. Only when an additional requirement is added to the equations is the uniqueness property satisfied. This is surprising because the additional requirement is not explicitly used in constructing the generating functions or calculating probabilities.

Our work focuses on formalising and understanding the event space that underlies Penney-Ante games rather than the calculation of probabilities of winning and expected lengths of games. We think it is important to do so because then so-called "paradoxes" associated with Penney-Ante games can be explained. First-past-the-post games are characterised by prefix-free languages. We prove several different properties of such languages. For motivational purposes, these properties are interspersed throughout the text. Section 1 introduces some elementary properties in advance of the formal definition of a first-past-the-post game in section 2. Section 2.2 helps to relate our use of prefix-free languages with other accounts of the games, and section 2.3 exploits their properties to prove the fundamental property that the remainder of a game after some (typically incomplete) play of the game is a first-past-the-post game. This is used later in section 5 where we generalise our results to such situations. Subsection 5.1 generalises the construction of the equations in languages and subsection 5.2 then shows how the generating functions introduced by Guibas and Odlyzko [GO81] are constructed from these equations; in so doing, we observe an error in the statement of the central theorem of their paper.

Section 6 discusses the practical implementation of the calculations. The standard technique for calculating the probability of winning is based on the construction of a deterministic finite-state machine that recognises the language underlying a Penney-Ante game. We observe that this construction can be carried out most efficiently using an adaptation of Aho and Corasick's [AC75] generalisation of the Knuth-Morris-Pratt [KMP74] pattern-matching algorithm, rather than the standard textbook method. The two methods of calculating the probabilities and expected values, one based on Aho and Corasick's algorithm and the other based on the non-linear equations in languages, have been implemented by Ngoc Do [ND12] and verified to yield the same results.

Several intriguing challenges remain, which are discussed in section 7.

1. Preliminaries

We assume familiarity with the use of regular expressions to denote languages. To avoid confusion with ordinary addition, the usual symbol "$\cup$" is used to denote set union, and not "$+$" (as often used in regular expressions). The *alphabet* is a finite set of *symbols*, denoted henceforth by T. *Words* are (finite) sequences of symbols. The *empty word* (the word of length 0) is denoted by ε. In line with other literature on the Penney-Ante game, capital letters at the beginning of the alphabet (A, B, etc.) denote words and capital letters at the end of the alphabet (U, V, etc.) denote sets of words. An exception to this rule (also in order to be in line with other literature) is the use of X to denote a word. Sets of words are called *languages*. Symbols of the alphabet T are denoted by lower case letters (a, b, etc.). The length of word A is denoted by $\#A$. Concatenation of words and of languages is denoted by juxtaposition and (as usual) no notational distinction is made between symbols of the alphabet and words of length 1. When parsing expressions, concatenation takes precedence over other binary operators and unary operators take precedence over all binary operators.

1.1. Prefixes

For any non-empty word A, $pre.A$ is the prefix of A obtained by discarding the last symbol in A. The function pre is extended to sets by the definition: for all languages V,

$$pre.V \;=\; \{A,a \;:\; A\in T^* \wedge a\in T \wedge Aa\in V \;:\; A\} \quad .$$

(We use Eindhoven notation [Bac86, GS93] for quantifications. Moreover, we use $\{dummies\,{:}\,range\,{:}\,term\}$ to abbreviate $\langle\cup\, dummies : range : \{term\}\rangle$. In conventional notation, the dummy a in the definition of pre would be existentially quantified. Occasionally, we use the conventional notation $\{dummy\,|\,range\}$ to abbreviate $\{dummy\,{:}\,range\,{:}\,dummy\}$.)

Repeated application of pre one or more times is denoted by pre^+ and zero or more times by pre^*. Thus $pre^+.V$ is the set of all proper prefixes of words in V, and $pre^*.V$ is $V\cup pre^+.V$. Note that pre distributes through set union.

For calculational purposes the following property of pre^+ is used. For all words C and languages V,

$$C\in pre^+.V \quad\equiv\quad \{C\}T^+\cap V \;\neq\; \emptyset \quad .$$

1.2. Derivatives

For any set V of words and word X we define $X\backslash V$ by

$$X\backslash V \;=\; \{A \mid XA\in V\} \quad .$$

The set $X\backslash V$ is called the *derivative* of V with respect to X [Brz64]. It is also a *factor* of V in Conway's terminology [Con71]; the notation reflects this property. An alternative, and sometimes preferable, definition of the derivative operator is by the Galois connection: for all words X and languages U and V,

$$\{X\}U\subseteq V \;\equiv\; U\subseteq X\backslash V \quad .$$

Lemma 1 through to lemma 4 give some insight into the properties of derivatives. They are all quite straightforward and "well-known". We document them here in order to ensure complete rigour in later proofs.

Lemma 1. For all words X and sets of words V over the alphabet T,

$$\{X\}(X\backslash V) \;=\; \{X\}T^*\cap V$$

Proof Straightforward application of definitions:

$$
\begin{array}{ll}
 & B \in \{X\}(X\backslash V) \\
= & \{ \quad \text{definition of concatenation} \quad \} \\
 & \langle \exists A : A \in X\backslash V : B = XA \rangle \\
= & \{ \quad \text{definition of } X\backslash V \quad \} \\
 & \langle \exists A : XA \in V : B = XA \rangle \\
= & \{ \quad \text{one-point rule, distributivity} \quad \} \\
 & \langle \exists A :: B = XA \rangle \ \wedge \ B \in V \\
= & \{ \quad \text{definition of concatenation and intersection} \quad \} \\
 & B \in \{X\}T^* \cap V \quad .
\end{array}
$$

□

Remark: The equality symbol connecting expressions in a calculation (as used in the four steps of the above calculation) means that the expressions have equal values for all instances of the variables. Similarly, if we use an only-if symbol ("$\Rightarrow$") or an if symbol ("$\Leftarrow$") as connector, it means that the implication has the value true for all instances of the variables. The hints, enclosed in braces, justify the claim. The connectives in a sequence of steps are, of course, to be read conjunctionally. As is common in conventional calculations in mathematics, we use Robert Recorde's symbol for equality irrespective of the types of the expressions — that is, whether the expressions denote numbers, or sets, or relations, or whatever. Uncommonly, we also use Robert Recorde's symbol when the expressions denote booleans. We use the equivales symbol ("$\equiv$") to form boolean expressions, i.e. expressions that can evaluate to either true or false. The value of $p \equiv q$ is, as usual, true when p and q have the same (boolean) value and false if they have different values. Of course, when such an expression is the subject of, for example, a lemma or theorem, it means that it always evaluates to true. **End of Remark**

Corollary 2. For all words X and sets of words V,

$$X \in pre^*.V \quad \equiv \quad X\backslash V \neq \emptyset \quad .$$

Proof

$$
\begin{array}{ll}
 & X \in pre^*.V \\
= & \{ \quad \text{definition of } pre^* \quad \} \\
 & \{X\}T^* \cap V \ \neq \ \emptyset \\
= & \{ \quad \text{lemma 1} \quad \} \\
 & \{X\}(X\backslash V) \ \neq \ \emptyset \\
= & \{ \quad \text{cancellation property of concatenation} \quad \} \\
 & X\backslash V \ \neq \ \emptyset \quad .
\end{array}
$$

□

Lemma 3. For all words X and sets of words V,

$$X \backslash pre.V \ = \ pre.(X\backslash V) \quad ,$$

$$X \backslash pre^+.V \ = \ pre^+.(X\backslash V) \quad \text{, and}$$

$$X \backslash pre^*.V \ = \ pre^*.(X\backslash V) \quad .$$

Proof For any word A, we have:

$$
\begin{array}{ll}
& A \in X \backslash pre.V \\
= & \quad \{ \quad \text{definition of } X\backslash \quad \} \\
& XA \in pre.V \\
= & \quad \{ \quad \text{definition of } pre \quad \} \\
& \langle \exists a : a{\in}T : XAa \in V \rangle \\
= & \quad \{ \quad \text{definition of } X\backslash \quad \} \\
& \langle \exists a : a{\in}T : Aa \in X\backslash V \rangle \\
= & \quad \{ \quad \text{definition of } pre \quad \} \\
& A \in pre.(X\backslash V) \quad .
\end{array}
$$

Thus, by the definition of set equality, $X \backslash pre.V \;=\; pre.(X\backslash V)$. The second equality follows by induction; the third then follows from the distributivity of $X\backslash$ over set union. □

Lemma 4. For all words X and sets of words U and V,

$$X\backslash UV \;=\; (X\backslash U)V \cup \langle \cup A,B \;:\; A{\in}U \wedge X = AB \wedge 1 \leq \#B \;:\; B\backslash V \rangle \quad .$$

Proof

$$
\begin{array}{ll}
& C \in X\backslash UV \\
= & \quad \{ \quad \text{definition of } X\backslash \quad \} \\
& XC \in UV \\
= & \quad \{ \quad \text{definition of } UV \quad \} \\
& \langle \exists A,B : A{\in}U \wedge B{\in}V : XC = AB \rangle \\
= & \quad \{ \quad \text{case analysis on } \#X \leq \#A \quad \} \\
& \quad \langle \exists A,B \;:\; A{\in}U \wedge B{\in}V \wedge \#X \leq \#A \;:\; XC = AB \rangle \\
& \vee \quad \langle \exists A,B \;:\; A{\in}U \wedge B{\in}V \wedge \#X > \#A \;:\; XC = AB \rangle \\
= & \quad \{ \quad \text{definition of } X\backslash \text{ and concatenation,} \\
& \qquad\quad \#X > \#A \wedge XC = AB \;\equiv\; \langle \exists D \;:\; X = AD \wedge 1 \leq \#D \;:\; DC = B \rangle \quad \} \\
& \quad C \in (X\backslash U)V \\
& \vee \quad \langle \exists A,B,D \;:\; A{\in}U \wedge B{\in}V \wedge X = AD \wedge 1 \leq \#D \;:\; DC = B \rangle \\
= & \quad \{ \quad \text{one-point rule, set comprehension, definition of } X\backslash \quad \} \\
& \quad C \in (X\backslash U)V \\
& \vee \quad C \in \langle \cup A,D \;:\; A{\in}U \wedge X = AD \wedge 1 \leq \#D \;:\; D\backslash V \rangle \quad .
\end{array}
$$

□

2. First-past-the-post Games

In this section, we formulate the notion of a first-past-the-post game and introduce some simple examples. Subsections 2.2 and 2.3 establish some properties of such games that will be exploited later.

2.1. Definition and Examples

Penney-Ante is an instance of a class of probabilistic games for which winning is characterised by the *first* occurrence of one of a set of events, and the events are words. We begin by formalising this class of games.

Definition 5. Suppose S is a subset of T^* . The set S is said to be a *first-past-the-post game* if

(a) $pre^+.S \cap S \;=\; \emptyset$.

In words, no proper prefix of a word in S is a word in S .

(b) $pre^*.S \;=\; \{\varepsilon\} \cup (pre^+.S)T$.

In words, appending an arbitrary symbol of the alphabet T to a proper prefix of a word in S gives a word that prefixes a word in S .

(This informal statement expresses only that the righthand side of the equation is included in the lefthand side. The opposite inclusion is obvious from the definitions of pre^* and pre^+ .)

A *play* of the game is an element of $pre^*.S$. A *complete play* of the game is an element of S . An *incomplete play* of the game is an element of $pre^+.S$. □

A play of the game can be thought of as repeatedly throwing a die with sides labelled by the elements of T . The play starts with the empty word and, as the die is thrown, the symbol that occurs is appended to the end of the play. The play is complete when the play is in S . Property (a) states that no proper prefix of a word in S is an element of S . That is, the game ends —the play is complete— immediately an element of S is recognised. Property (b) states that the plays are the empty word or arbitrary continuations of an incomplete play. It has the consequence that any throw of the die continues an incomplete play of the game. A second consequence is that S is non-empty (because the righthand side of the equation is a non-empty set).

Example 6. Taking the alphabet T to be $\{a,b\}$, the table below shows examples of languages and whether or not they fulfil properties (a) and (b) of definition 5.

Language	(a)	(b)
$\{a\}$	$\surd$	$\times$
$\{a,ab\}$	$\times$	$\times$
T^k $(0 \le k)$	$\surd$	$\surd$
$T^{\le k}$ $(0 < k)$	$\times$	$\surd$
$\{a,ba,bb\}$	$\surd$	$\surd$
$\{b\}^*\{a\}$	$\surd$	$\surd$
$\{b\}^*\{a\}\{a\}^*\{b\}\{b\}^*\{a\}$	$\surd$	$\surd$
$\{aaa,bbb\} \cup \{ab,aab,ba,bba\}$	$\surd$	$\surd$

The set T^k , where k is some fixed natural number, exemplifies the set of complete plays in a first-past-the-post game. It is the game where a die is thrown exactly k times. The last four rows of the table also exemplify the abstract notion of a first-past-the-post game.

The final row in the table represents a simplified version of the game mentioned in the introduction. In this game a coin is tossed repeatedly, the symbols a and b representing the two possible outcomes of a single toss; one player waits for three a s or three b s, the other player waits for at least one a and at least one b . The set $\{aaa,bbb\}$ is the set of complete plays that result in a win for the first player and the set $\{ab,aab,ba,bba\}$ is the set of complete plays that result in a win for the second player; the incomplete plays are given by $\{\varepsilon,a,b,aa,bb\}$.

The penultimate three rows of the tables are used again later in the text. □

Generally, the set S may be assumed to be split into disjoint sets each of which is owned by one of the players. When the play is complete, the owner of the play is the winner. The Penney-Ante game assumes that two players each choose one word. A consequence of this assumption, and one reason the game has attracted so much attention, is that the game is then non-transitive: if one player chooses one word it is always possible for the second player to choose a word that gives a better than evens chance of winning. This, however, is not the focus of our investigation. For our purposes, the number of players can be arbitrary as can be the number of words each player chooses. There is no reason why games with fewer or more than two players should not be allowed, or why each player should choose just one word. "Games" with one player are associated with pattern-matching problems; see section 4.

The notation S for the complete plays of a game is used by Graham, Knuth and Patashnik [GKP89]. They also use N for the set of incomplete plays. We adopt this convention throughout the paper. The two clauses in definition 5 of a first-past-the-post game then become:

(7) $\quad N \cap S \;=\; \emptyset$, and

(8) $\quad N \cup S \;=\; \{\varepsilon\} \cup NT$.

2.2. Introducing Complements

In [GO81] plays are (implicitly) characterised by words that do *not* have a certain property. Theorem 12 explains the connection. Effectively, it states that, in the context of first-past-the-post games, any discussion of properties of prefix sets that do hold can be rephrased in terms of properties of other sets that do not hold.

Henceforth we overload the symbol " $\neg$ " using it both for boolean negation and for the complement of a set. That is, for all sets of words U and all words A , $A \in \neg U \equiv \neg(A \in U)$. (The lefthand occurrence of " $\neg$ " is set complement and the righthand occurrence is boolean negation.) Recall our precedence convention that unary operators always take precedence over binary operators. In particular, $\neg UV$ should be read as $(\neg U)V$ and UV^+ should be read as $U(V^+)$.

Lemma 9. If S is an arbitrary language over alphabet T ,

(10) $\quad \neg(ST^*) \cap S \;=\; \emptyset$, and

(11) $\quad \{\varepsilon\} \cup \neg(ST^*)T \;=\; \neg(ST^+)$.

Proof The proof of (10) is straightforward:

$$\begin{array}{ll}
 & \neg(ST^*) \cap S \\
\subseteq & \{ \quad S \subseteq ST^* \text{ , antimonotonicity of complement} \quad \} \\
 & \neg S \cap S \\
= & \{ \quad \text{complement} \quad \} \\
 & \emptyset \;.
\end{array}$$

The proof of (11) is by mutual inclusion. First,

$$\begin{array}{ll}
 & \{\varepsilon\} \cup \neg(ST^*)T \;\subseteq\; \neg(ST^+) \\
= & \{ \quad \text{complement, distributivity} \quad \} \\
 & (\{\varepsilon\} \cap ST^+) \;\cup\; (\neg(ST^*)T \cap ST^+) \;\subseteq\; \emptyset \\
= & \{ \quad \varepsilon \notin ST^+ \text{ , } T^+ = T^*T \quad \} \\
 & \emptyset \;\cup\; (\neg(ST^*)T \cap ST^*T) \;\subseteq\; \emptyset \\
= & \{ \quad \text{distributivity of concatenation with } T \text{ over } \cap \quad \}
\end{array}$$

$$(\neg(ST^*) \cap ST^*)T \;\subseteq\; \emptyset$$

$= \quad \{ \quad$ complement $\quad \}$

true .

Second,

$$\neg(ST^+) \;\subseteq\; \{\varepsilon\} \cup \neg(ST^*)T$$

$= \quad \{ \quad$ complement, $\{\varepsilon\} = \neg(T^+)$ $\quad \}$

$$\neg(ST^+) \cap \neg(\neg(ST^*)T) \;\subseteq\; \neg(T^+)$$

$= \quad \{ \quad$ distributivity, antimonotonicity of complement $\quad \}$

$$T^+ \;\subseteq\; ST^+ \cup \neg(ST^*)T$$

$= \quad \{ \quad$ $T^+ = T^*T$, distributivity of concatenation over $\cup$ $\quad \}$

$$T^+ \;\subseteq\; (ST^* \,\cup\, \neg(ST^*))T$$

$= \quad \{ \quad$ $T^+ = T^*T$, complement, monotonicity of concatenation $\quad \}$

true .

□

Theorem 12. S is a first-past-the-post game if and only if

(13) $\quad \neg(ST^+) \;=\; pre^*.S$, and

(14) $\quad \neg(ST^*) \;=\; pre^+.S$.

Proof The proof is by mutual implication.

First, assume that S is a first-past-the-post game. We must prove (13) and (14). We prove (13) by showing that

(15) $\quad pre^*.S \cup ST^+ \;\supseteq\; T^*$, and

(16) $\quad pre^*.S \cap ST^+ \;\subseteq\; \emptyset$.

These are equivalent to, respectively, $\neg(ST^+) \subseteq pre^*.S$ and $\neg(ST^+) \supseteq pre^*.S$ from which the equality follows by mutual inclusion.

First,

$$pre^*.S \cup ST^+ \;\supseteq\; T^*$$

$\Leftarrow \quad \{ \quad$ $T^* \;=\; \langle \mu X :: \{\varepsilon\} \cup XT \rangle$, fixed-point induction $\quad \}$

$$pre^*.S \cup ST^+ \;\supseteq\; \{\varepsilon\} \cup (pre^*.S \cup ST^+)T$$

$\Leftarrow \quad \{ \quad$ S is a first-past-the-post game: 5(b) $\quad \}$

$$\{\varepsilon\} \cup (pre^+.S)T \cup ST^+ \;\supseteq\; \{\varepsilon\} \cup (pre^*.S \cup ST^+)T$$

$= \quad \{ \quad$ $pre^*.S \;=\; S \,\cup\, pre^+.S$,

concatenation distributes over set union,

$ST^+ \;=\; ST \,\cup\, ST^+T$ $\quad \}$

true .

Second, for all words X ,

$$
\begin{array}{ll}
 & X \in pre^*.S \cap ST^+ \\
= & \quad \{ \quad \text{distributivity, definition of concatenation} \quad \} \\
 & X \in pre^*.S \ \wedge \ \langle \exists A{,}B : A{\in}S \wedge B{\in}T^+ : X{=}AB \rangle \\
\Rightarrow & \quad \{ \quad \text{definition of } pre^+ \quad \} \\
 & \langle \exists A : A{\in}S : A \in pre^+.(pre^*.S) \rangle \\
\Rightarrow & \quad \{ \quad pre^+.(pre^*.S) = pre^+.S \quad \} \\
 & S \cap pre^+.S \ \neq \ \emptyset \\
= & \quad \{ \quad S \text{ is a first-past-the-post game: 5(a)} \quad \} \\
 & \mathsf{false} \ .
\end{array}
$$

We prove (14) similarly. Note that (15) is the same as

$$pre^+.S \cup ST^* \ \supseteq \ T^*$$

since $pre^*.S = pre^+.S \cup S$ and $ST^* = S \cup ST^+$ (and, of course, set union is associative). So it remains to prove the counterpart of (16), namely

$$pre^+.S \cap ST^* \ = \ \emptyset \ .$$

We have, for all words X ,

$$
\begin{array}{ll}
 & X \in pre^+.S \cap ST^* \\
= & \quad \{ \quad \text{distributivity, definition of concatenation} \quad \} \\
 & X \in pre^+.S \ \wedge \ \langle \exists A{,}B : A{\in}S : X{=}AB \rangle \\
\Rightarrow & \quad \{ \quad \text{definition of } pre^* \quad \} \\
 & \langle \exists A : A{\in}S : A \in pre^*.(pre^+.S) \rangle \\
\Rightarrow & \quad \{ \quad pre^*.(pre^+.S) = pre^+.S \quad \} \\
 & S \cap pre^+.S \ \neq \ \emptyset \\
= & \quad \{ \quad S \text{ is a first-past-the-post game: 5(a)} \quad \} \\
 & \mathsf{false} \ .
\end{array}
$$

Now, assume (13) and (14). We must prove that S is a first-past-the-post game, i.e. S satisfies 5(a) and (b). Part (a) is proved as follows.

$$
\begin{array}{ll}
 & pre^+.S \cap S \\
= & \quad \{ \quad (14) \quad \} \\
 & \neg(ST^*) \cap S \\
= & \quad \{ \quad (10) \quad \} \\
 & \emptyset \ .
\end{array}
$$

Part (b) is similarly straightforward:

$$
\begin{array}{ll}
 & pre^*.S \ = \ \{\varepsilon\} \cup (pre^+.S)T \\
= & \quad \{ \quad (13) \text{ and } (14) \quad \} \\
 & \neg(ST^+) \ = \ \{\varepsilon\} \cup (\neg(ST^*))T
\end{array}
$$

$$
\begin{array}{ll}
= & \{ \quad (11) \quad \} \\
& \mathsf{true} \quad .
\end{array}
$$

$\square$

Although theorem 12 allows us to circumvent the use of the *pre* function altogether, we choose to do so only when it is more convenient.

2.3. Incomplete Games

In the course of a first-past-the-post game, players may want to reassess their position, asking questions like 'what is the probability of winning?' and 'what is the expected length of the remainder of the game?'. In order to answer such questions, theorem 17, below, formulates the central property that the remainder of a game after some (typically but not necessarily incomplete) play X of the game is also a first-past-the-post game.

Recall that $X\backslash S$ denotes the derivative of language S with respect to word X ; see section 1.2.

Theorem 17. $X\backslash S$ is a first-past-the-post game if S is a first-past-the-post game and X is a play of the game S. The incomplete plays of $X\backslash S$ are given by the set $X\backslash N$ where N is the set of incomplete plays of the game S.

Proof Suppose S is a first-past-the-post game and X is a play of the game S. Suppose N is the set of incomplete plays of the game S.

By definition, $N = pre^{+}.S$. Moreover, by lemma 3, $pre^{+}.(X\backslash S) = X\backslash N$. So if we prove that $X\backslash S$ is a first-past-the-post game, it follows immediately that $X\backslash N$ is the set of incomplete plays of the game $X\backslash S$.

The two clauses in the definition of first-past-the-post are proved as follows. First,

$$
\begin{array}{ll}
& X\backslash S \cap pre^{+}.(X\backslash S) \;=\; \emptyset \\
= & \{ \quad \text{lemma 3} \quad \} \\
& X\backslash S \cap X\backslash N = \emptyset \\
= & \{ \quad \text{concatenation} \quad \} \\
& \{X\}(X\backslash S \cap X\backslash N) \;\subseteq\; \emptyset \\
\Leftarrow & \{ \quad \text{monotonicity of concatenation} \quad \} \\
& \{X\}(X\backslash S) \cap \{X\}(X\backslash N) \;\subseteq\; \emptyset \\
\Leftarrow & \{ \quad \text{by lemma 1, } \{X\}(X\backslash V) \;\subseteq\; V \quad \} \\
& S \cap N \subseteq \emptyset \\
\Leftarrow & \{ \quad \text{definition: (7)} \quad \} \\
& S \text{ is a first-past-the-post game} \quad .
\end{array}
$$

So it remains to prove that

$$
\{\varepsilon\} \cup (X\backslash N)T \;=\; X\backslash N \cup X\backslash S \quad .
$$

The case when $X = \varepsilon$ is trivial (since $\varepsilon\backslash V = V$ for all V and S is a first-past-the-post game). So let us assume that $X \neq \varepsilon$. Then

$$
\begin{array}{ll}
& X\backslash N \cup X\backslash S \\
= & \{ \quad X\backslash \text{ distributes through set union} \\
& \quad \text{(used twice, once in each direction) and}
\end{array}
$$

$$
\begin{array}{lll}
 & & \quad (8){:}\ \ \{\varepsilon\}\cup NT \;=\; N\cup S \quad \} \\
 & X\backslash\{\varepsilon\}\cup X\backslash NT & \\
= & \{ \quad X\backslash\{\varepsilon\}=\emptyset \ \text{(because } X\neq\varepsilon\text{)} \quad \} & \\
 & X\backslash NT & \\
= & \{ \quad \text{lemma 4} \quad \} & \\
 & (X\backslash N)T \,\cup\, \langle\cup A,B : A\in N \wedge X=AB \wedge 1\leq\#B : B\backslash T\rangle & \\
= & \{ \quad \text{case analysis on } 1=\#B \ \vee\ 1<\#B : & \\
 & \qquad 1=\#B \ \equiv\ B\in T \wedge B\backslash T=\{\varepsilon\} & \\
 & \qquad 1<\#B \ \equiv\ B\backslash T=\emptyset \quad \} & \\
 & (X\backslash N)T \,\cup\, \langle\cup A,B : A\in N\wedge X=AB\wedge B\in T : \{\varepsilon\}\rangle & \\
= & \{ \qquad \langle\exists A,B : A\in N\wedge B\in T : X=AB\rangle & \\
 & \qquad = \quad \{ \quad \text{definition of concatenation} \quad \} & \\
 & \qquad X\in NT & \\
 & \qquad \Leftarrow \quad \{ \quad (8){:}\ \{\varepsilon\}\cup NT \;=\; N\cup S \quad \} & \\
 & \qquad X\in N\cup S \ \wedge\ X\neq\varepsilon & \\
 & \qquad = \quad \{ \quad \text{assumption: } X \text{ is a play of the game and } X\neq\varepsilon \quad \} & \\
 & \qquad \mathsf{true} \quad \} & \\
 & (X\backslash N)T\cup\{\varepsilon\} \ . &
\end{array}
$$

We conclude that $X\backslash N \cup X\backslash S \;=\; \{\varepsilon\}\cup(X\backslash N)T$, as required. □

2.4. Relating Probabilities and Expected Values

We assume that the outcome of each single throw of the die is given by some probability distribution p. The outcomes of separate throws are assumed to be independent. This suggests the following definition.

Definition 18. Let p be a function with domain T and range the set of real numbers. We define the function h_p with domain T^* inductively by

(a) $h_p.\varepsilon \;=\; 1$,

(b) $h_p.Ba \;=\; h_p.B\times p.a$, for all $B\in T^*$ and $a\in T$.

The function h_p is extended to languages by defining, for all V , where $V\subseteq T^*$,

$$h_p.V \;=\; \langle\Sigma A : A\in V : h_p.A\rangle \ .$$

The function e_p is defined on languages by, for all V , where $V\subseteq T^*$,

$$e_p.V \;=\; \langle\Sigma A : A\in V : h_p.A\times\#A\rangle \ .$$

Note: these definitions assume that the summations are well defined. In all the concrete examples discussed in this paper, this is indeed the case. □

Theorem 23 shows that, if p is a probability distribution on T , h_p is a probability distribution on a first-past-the-post game S . The value of $e_p.S$ is then interpreted as the "expected" length of the game. It is important to note, however, that definition 18 does *not* assume that p is a probability distribution. We apply definition 18 just as often when p and/or h_p cannot be viewed as probability distributions.

Typically languages are defined syntactically — by a combination of regular expressions and equations (aka grammars). "Unambiguity" of syntactic definitions is useful in the evaluation of the functions h_p and e_p. This is made precise in the following definitions and lemmas.

Definition 19 (Unambiguous Expressions). Let U and V be expressions denoting languages $L.U$ and $L.V$, respectively. We say that the union operator in the expression "$U \cup V$" is *unambiguous* if $L.U \cap L.V = \emptyset$ (i.e. the languages are disjoint). We say that the concatenation operator in the expression "UV" is *unambiguous* if, for all words A, A', B and B',

$$A,A' \in L.U \wedge B,B' \in L.V \wedge AB = A'B' \Rightarrow A = A' \wedge B = B' \quad .$$

We say that the star operator in the expression "U^*" is *unambiguous* if, for all natural numbers k and k', and sequences of words A_i ($1 \le i \le k$) and B_j ($1 \le j \le k'$) all of which are elements of $L.U$,

$$A_1 \ldots A_k = B_1 \ldots B_{k'} \Rightarrow k = k' \wedge \langle \forall i : 1 \le i \le k : A_i = B_i \rangle \quad .$$

□

Expressions and languages are, of course, different in the same way that names and people are different. ("Winston Churchill" is the name of a famous Englishman. The name consists of a forename and a surname, whilst the person has a mother and father, etc.) Definition 19 has been formulated in a way that makes the difference clear. Henceforth however, we are not so precise and we leave it to the reader to determine whether we are referring to the syntactic form of an expression or to the language that is denoted by the expression. So, for example, a less precise formulation of the first clause of definition 19 is

"the expression $U \cup V$ is *unambiguous* if $U \cap V = \emptyset$".

We trust that the reader will have no difficulty in understanding what is meant.

An example of unambiguity is the expression $\{\varepsilon\} \cup (pre^+.S)T$ in definition 5. Obviously $\{\varepsilon\}$ and $(pre^+.S)T$ have an empty intersection because $\{\varepsilon\}$ is the set of words of length zero whilst $(pre^+.S)T$ contains only words of length at least one. So the "$\cup$" operator in the expression is unambiguous. Also obvious on length considerations is that the (implicit) concatenation operator in the expression $(pre^+.S)T$ is unambiguous. In general, an expression denoting the concatenation of two languages of which one is a subset of T^k for some k (i.e. all the words in the language have the same length) is unambiguous. Deterministic finite-state machines also exemplify the use of unambiguous expressions in order to define a language. A deterministic finite-state machine corresponds to a system of equations in languages; the righthand sides of the equations are disjoint unions of expressions of the form ε or aU (where U denotes the language recognised by some state of the machine). The union operators and the concatenation operators in these expressions are all unambiguous.

(Tarjan [Tar81] uses the term "non-redundant" instead of "unambiguous". We do not follow his example because it might be misleading to say that an operator is "redundant". Moreover, our terminology corresponds to the standard notion of unambiguity in the case of regular grammars.)

The following lemma is the key to evaluating probabilities and expected values in the context of first-past-the-post games. Using it, equations expressing languages are easily converted to equations expressing real numbers; see, for example, section 4.2.

Lemma 20. If $U \cup V$ is an unambiguous expression,

$$\begin{aligned} h_p.(U \cup V) &= h_p.U + h_p.V \quad \text{, and} \\ e_p.(U \cup V) &= e_p.U + e_p.V \quad . \end{aligned}$$

If UV is an unambiguous expression,

$$\begin{aligned} h_p.UV &= h_p.U \times h_p.V \quad \text{, and} \\ e_p.UV &= h_p.U \times e_p.V + e_p.U \times h_p.V \quad . \end{aligned}$$

Proof Straightforward manipulation of quantifier expressions. □

In order to remember the rules for evaluating the function e_p, it may help to note how the equations resemble the rules for differentiating a sum and product of two terms.

Although we won't use it elsewhere, it is interesting to add an additional clause to lemma 20 for the case of expressions of the form U^*. To do so, it is instructive to overload the star operator by defining x^* for real number x to be $\frac{1}{1-x}$. Then we have the lemma:

Lemma 21. If U^* is an unambiguous expression and $0 \le h_p.U < 1$,

$$h_p.U^* \;=\; (h_p.U)^* \quad \text{, and}$$

$$e_p.U^* \;=\; e_p.U \times ((h_p.U)^*)^2 \quad .$$

Proof If U^* is an unambiguous expression then so is U^i for each natural number i. Applying lemma 20, it is easy to show by induction on i that $h_p.U^i = (h_p.U)^i$ and $e_p.U^i = i \times (h_p.U)^{i-1} \times e_p.U$. The two equations are then applications of the theorems that, for real number x satisfying $0 \le x < 1$,

$$x^* \;=\; \langle \Sigma i : 0 \le i : x^i \rangle$$

and

$$(x^*)^2 \;=\; \langle \Sigma i : 0 \le i : i \times x^{i-1} \rangle \quad .$$

(The infinite summations on the righthand side of these equations are, of course, formally expressed as limits.) □

Once again, an easy way to remember the equation for the function e_p is to observe its similarity to the rule for differentiation: using the notation y^* for $\frac{1}{1-y}$,

$$\frac{\mathsf{d}}{\mathsf{d}x}(y^*) \;=\; \frac{\mathsf{d}y}{\mathsf{d}x} \times (y^*)^2 \quad .$$

Example 22. Suppose alphabet T equals $\{a,b\}$ where $p.a = q$ and $p.b = r$. The expression $\{a\}^*\{b\}$ is unambiguous so, assuming $0 \le q < 1$, we have $h_p.(\{a\}^*) = q^*$ and $e_p.(\{a\}^*) = q \times (q^*)^2$. It follows that

$$h_p.(\{a\}^*\{b\}) = q^* \times r$$

and (using the fact that $q^* \;=\; 1 + q \times q^*$)

$$e_p.(\{a\}^*\{b\}) \;=\; q \times (q^*)^2 \times r + q^* \times r \;=\; (q^*)^2 \times r \quad .$$

Note that, when $q+r = 1$, $h_p.(\{a\}^*\{b\})$ simplifies to 1 and $e_p.(\{a\}^*\{b\})$ simplifies to q^*, which equals $h_p.(\{a\}^*)$.

The same calculations can be done using a deterministic finite-state machine that recognises the language $\{a\}^*\{b\}$. Such a machine is shown in fig. 1.

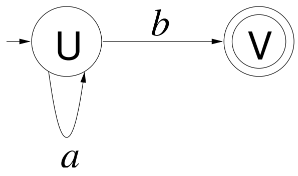


Figure 1: Deterministic finite-state machine

If we let U and V denote the languages recognised by the two states of the machine, we have

$$\mathsf{U} = \{\varepsilon\} \cup \mathsf{U}\{a\} \quad \wedge \quad \mathsf{V} = \mathsf{U}\{b\} \quad .$$

The righthand sides of these equations are unambiguous so, applying lemma 20, we have:

$$h_p.\mathsf{U} \;=\; 1 + h_p.\mathsf{U} \times h_p.\{a\} \;=\; 1 + h_p.\mathsf{U} \times q$$

and

$$h_p.\mathsf{V} \;=\; h_p.\mathsf{U} \times h_p.\{b\} \;=\; h_p.\mathsf{U} \times r \quad .$$

Solving the equations, we get

$$h_p.\mathsf{U} = q^* \;\wedge\; h_p.\mathsf{V} = q^* \times r \quad .$$

This confirms the previous calculation because $\mathsf{U} = \{a\}^*$ and $\mathsf{V} = \{a\}^*\{b\}$. Similarly, again applying lemma 20,

$$e_p.\mathsf{U} \;=\; 0 + e_p.\mathsf{U} \times h_p.\{a\} + h_p.\mathsf{U} \times e_p.\{a\} \;=\; e_p.\mathsf{U} \times q + h_p.\mathsf{U} \times q$$

and

$$e_p.\mathsf{V} \;=\; e_p.\mathsf{U} \times h_p.\{b\} + h_p.\mathsf{U} \times e_p.\{b\} \;=\; e_p.\mathsf{U} \times r + h_p.\mathsf{U} \times r \quad .$$

Solving these equations, we get

$$e_p.\mathsf{U} \;=\; (q^*)^2 \times q \wedge e_p.\mathsf{V} = (q^*)^2 \times r \quad .$$

This again confirms the previous calculations. □

We now consider the consequences of the function p being a probability distribution. Recall that, using S to denote a first-past-the-post game and N to denote $pre^+.S$, S and N satisfy (8). From this equation, it is easy to see that $h_p.T = 1 \Rightarrow h_p.S = 1$:

$h_p.S = 1$

$=$ { heading towards (8) in definition of a game, we add $h_p.N$ to both sides }

$h_p.N \;+\; h_p.S \;=\; h_p.N \;+\; 1$

$=$ { by definition, $1 = h_p.\{\varepsilon\}$; assumption: $h_p.T = 1$ }

$h_p.N \;+\; h_p.S \;=\; h_p.N \times h_p.T \;+\; h_p.\{\varepsilon\}$

$=$ { expressions $N \cup S$ and $NT \cup \{\varepsilon\}$ are unambiguous, lemma 20 }

$h_p.(N \cup S) \;=\; h_p.(NT \cup \{\varepsilon\})$

$=$ { definition of a game: (8) }

true .

This suggests that, if p is a probability distribution on T , h_p is a probability distribution on complete plays. This fact appears to be taken for granted in [GKP89] and [GO81]. (At least, we have been unable to find anything that we would recognise as a proof.) We think it is illuminating to make the theorem explicit and provide a proof. The proof is not calculational because it links the formal definitions with the informal notion of relative frequencies.

Theorem 23. If p is a probability distribution on the alphabet T (i.e. $p.a$ is the relative frequency of the occurrence of symbol a when the die is thrown and, thus, $h_p.T = 1$) and throws of the die are independent, the function h_p is a probability distribution on complete plays of a first-past-the-post game S . Specifically, for an arbitrary word A in S , $h_p.A$ is the relative frequency that the word A is a complete play of the game. Moreover, h_p is a probability distribution on 2^S (the set of subsets of S); if $U \subseteq S$, then $h_p.U$ is the relative frequency with which a word in U occurs as a complete play.

Proof Suppose $A \in pre^*.S$. We prove by induction on the length of A that $h_p.A$ is the relative frequency with which the word A occurs as a prefix of a complete play of the game.

When the length of A is zero, $A = \varepsilon$. The empty word occurs in every play of the game. That is, the relative frequency of ε as a prefix of a complete play of the game is 1 , which equals $h_p.\varepsilon$ by definition. This proves the basis.

Now suppose the length of A is at least one. Suppose $A = Ba$ for some $B \in T^*$ and $a \in T$. Since $B \in pre^*.S$, and the length of B is less than the length of A , we may assume inductively that $h_p.B$ is the relative frequency with which the word B occurs as a prefix of a complete play of the game. But $B \in pre^+.S$ and so, by definition 5(b), $h_p.B$ is the relative frequency with which words of the form Bb , for some $b \in T$, occur as a prefix of a complete play. Since $p.a$ is the relative frequency that a occurs, the independence assumption implies that $h_p.B \times p.a$ is the relative frequency with which Ba occurs as a prefix of a complete play. But $h_p.A = h_p.B \times p.a$ by definition. In this way, the induction step is verified.

A corollary of this inductive argument and definition 5(a) is that, when A is a complete play, $h_p.A$ is the relative frequency of A among complete plays. (Because of definition 5(a), a complete play only occurs as a prefix of itself and no other plays.)

By the definition of a probability distribution, it is an immediate corollary that the extension of h_p to subsets of S is a probability distribution. □

Note that h_p is just a function on arbitrary languages. As shown above, it is a probability distribution on S and on 2^S whenever p is a probability distribution on T but we apply it elsewhere to arbitrary languages. An example of where h_p is used in this way is the following lemma.

Theorem 24. Suppose S is the set of complete plays in a first-past-the-post game and N is the set of incomplete plays. Suppose the symbols in T occur with probability distribution given by p . Then

$$e_p.S \;=\; h_p.N \quad .$$

Proof

$$
\begin{array}{ll}
 & e_p.S \;=\; h_p.N \\
= & \qquad \{ \qquad \text{heading towards (8) in definition of a game,} \\
 & \qquad\qquad \text{we add } e_p.N \text{ to both sides} \quad \} \\
 & e_p.N \,+\, e_p.S \;\;=\;\; e_p.N \,+\, h_p.N \quad .
\end{array}
$$

But

$$
\begin{array}{ll}
 & e_p.N \,+\, e_p.S \\
= & \qquad \{ \qquad \text{expression } N \cup S \text{ is unambiguous, lemma 20} \quad \} \\
 & e_p.(N \cup S) \\
= & \qquad \{ \qquad (8) \quad \} \\
 & e_p.(NT \cup \{\varepsilon\}) \\
= & \qquad \{ \qquad \text{expression } NT \cup \{\varepsilon\} \text{ is unambiguous, lemma 20} \quad \} \\
 & h_p.N \times e_p.T + e_p.N \times h_p.T + e_p.\{\varepsilon\} \\
= & \qquad \{ \qquad p \text{ is a probability distribution on } T \text{, so } h_p.T = 1\,; \\
 & \qquad\qquad \text{also, for each } A \in T \text{ , } \#A = 1 \text{ . So } e_p.T = 1 \text{ .} \\
 & \qquad\qquad \text{By definition, } e_p.\{\varepsilon\} = 0 \text{ .} \quad \} \\
 & h_p.N \,+\, e_p.N \quad .
\end{array}
$$

The lemma follows by combining the two calculations (using symmetry of addition). □

Example 25. Suppose $p.a = q$ and $p.b = r$, where $q{+}r = 1$.

If $S = \{a{,}ba{,}bb\}$, then $N = \{\varepsilon{,}b\}$ and $e_p.S = 1{\times}q + 2{\times}r{\times}q + 2{\times}r{\times}r = 1{+}r = h_p.N$. Note also that $h_p.S = q + r{\times}q + r{\times}r = 1$.

If $S = \{a\}^*\{b\}$, then $N = \{a\}^*$; so $e_p.S = q^* = \frac{1}{r}$. (Recall that q^* denotes $\frac{1}{1-q}$.) Note also that $h_p.S = q^* \times r = 1$. This confirms the calculations in example 22. Note how much easier it is to use the lemma than to calculate $e_p.S$ directly from its definition. $\Box$

Although very easily proved, we have given theorem 24 the status of a "theorem" because it explains several of the so-called "paradoxes" associated with Penney-Ante games; see example 49 for further discussion. So far as we have been able to determine, the theorem has not been observed elsewhere.

The following simple corollary is fundamental to Collings' proofs [Col82] although it does not appear to be explicitly stated.

Corollary 26. Suppose p is a probability distribution on the alphabet T . Suppose S is a first-past-the-post game and X is a play of the game. Then $h_p.(X\backslash S) = 1$ and the expected length of the game $X\backslash S$ is $h_p.(X\backslash N)$.

Proof Simple combination of theorems 17, 23 and 24. $\Box$

3. Prefix-free Languages

A requirement on games is that complete plays are prefix-free languages (definition 5(a)). Any language V can be reduced to a maximal, prefix-free language by selecting the words that have no proper prefixes in V . Specifically, if V is a language, the set $PF.V$, called the *prefix-free reduction* of V , is defined by

$$PF.V \quad = \quad V \cap \neg(VT^+) \quad .$$

The element-wise formulation of $PF.V$ is that, for all languages V and all words C ,

$$C \in PF.V \quad \equiv \quad C{\in}V \;\wedge\; \neg\langle \exists\, D{,}E \;:\; D{\in}V \wedge E{\in}T^+ \;:\; DE = C\rangle \quad .$$

That is, $PF.V$ is the set of words in V that do not have a proper prefix in V .

Example 27. It is sometimes of interest to determine the expected length of a sequence of observations that culminates in a given "pattern". Patterns are classified as either *block* or *hidden* [FS09]. Formally, let A be an arbitrary word over the alphabet T . Then $PF.T^*\{A\}$ models the process of observing sequences of letters until the word A first occurs contiguously (i.e. as a "block" pattern). If $1 \leq n$ and $A = a_1a_2\ldots a_n$, then $PF.T^*\{a_1\}T^*\{a_2\}\ldots T^*\{a_n\}$ models the process of observing sequences of letters until all the letters of A occur in order but not necessarily contiguously (i.e. as a "hidden" pattern).

Lemma 32 establishes that $PF.T^*W$ is a first-past-the-post game for arbitrary non-empty set W . Thus $PF.T^*\{A\}$ and $PF.T^*\{a_1\}T^*\{a_2\}\ldots T^*\{a_n\}$ are both first-past-the-post games.

(Of course, $PF.W$ is not a first-past-the-post game for arbitrary non-empty set W . A simple counter-example is $W = \{a\}$ since $PF.\{a\} = \{a\}$. When $T \neq \{a\}$ this is not a first-past-the-post game; see example 6.) $\Box$

Several properties of the function PF will be used later. The following lemma expresses formally the process of "reducing" V to $PF.V$.

Lemma 28. For all sets of words V ,

$$V \;\subseteq\; (PF.V)\,T^* \quad .$$

Indeed, every word in V has a *unique* prefix in $PF.V$.

Proof Let C be a word in V. Consider a linear search of the prefixes of C, starting with the empty word and iteratively increasing the length of the prefix, to find a word that is an element of V. The search will eventually terminate successfully because C is itself such a word. An invariant of the algorithm is that the current prefix is an element of $\neg(VT^+)$. The prefix B that is found is thus an element of both V and $\neg(VT^+)$, i.e. B is an element of $PF.V$, and C is an element of $\{B\}T^*$. The word B is clearly unique because any other prefix of C is either shorter than B and so not in V, or longer than B and so in VT^+. □

Corollary 29. For all sets of words V,

$$VT^* \;=\; (PF.V)T^* \quad\wedge\quad VT^+ \;=\; (PF.V)T^+ \quad .$$

Proof For the first conjunct, we have:

$$\begin{array}{cl}
 & VT^* \\
\subseteq & \{ \quad \text{lemma 28, monotonicity of concatenation} \quad \} \\
 & (PF.V)T^*T^* \\
= & \{ \quad T^* \;=\; T^*T^* \quad \} \\
 & (PF.V)T^* \\
\subseteq & \{ \quad [\, PF.V \subseteq V \,] \text{, monotonicity of concatenation} \quad \} \\
 & VT^* \quad .
\end{array}$$

The second conjunct is an immediate corollary since $T^+ \;=\; T^*T$. □

Lemma 30. $PF.V$ is prefix-free. That is, for all sets of words V,

$$pre^+.(PF.V) \cap PF.V \quad=\quad \emptyset \quad .$$

Equivalently, for all sets of words V,

$$pre^+.(PF.V) \quad=\quad pre^*.(PF.V) \cap \neg(PF.V) \quad .$$

Proof This is, in fact, a corollary of lemma 28 but is proved directly as follows. We have, for all words C,

$$\begin{array}{cl}
 & C \in pre^+.(PF.V) \cap PF.V \\
= & \{ \quad \text{definition of } pre^+ \quad \} \\
 & \langle \exists D : D \in PF.V : D \in \{C\}T^+ \rangle \wedge C \in PF.V \\
\Rightarrow & \{ \quad PF.V \subseteq V \quad \} \\
 & \langle \exists D : D \in PF.V : D \in VT^+ \rangle \\
\Rightarrow & \{ \quad PF.V \subseteq \neg(VT^+) \quad \} \\
 & \mathsf{false} \quad .
\end{array}$$

That is, $pre^+.(PF.V) \cap PF.V \;=\; \emptyset$. The equivalent form is an immediate consequence of the definitions of pre^* and pre^+, and simple set calculus. □

Remark: The prefix-free reduction of V is a maximal prefix-free reduction in the sense that it is *prefix-free* (lemma 30) and it is the *largest* prefix-free subset of V, i.e. for all languages U,

$$(U \subseteq V \;\equiv\; U \subseteq PF.V) \;\Leftarrow\; U \cap pre^+.U \;=\; \emptyset \quad .$$

End of Remark

Lemma 31. For all languages V and U, the expression $(PF.V)U$ is unambiguous. That is, for all languages V and all words C, C', D and D',

$$CD = C'D' \wedge C \in PF.V \wedge C' \in PF.V \ \Rightarrow\ C = C' \wedge D = D' \ .$$

Proof We begin with a simple property of words.

$$
\begin{array}{ll}
 & CD = C'D' \\
\Rightarrow & \{ \quad \text{case analysis on } \#C \text{ and } \#C' \text{, definition of } pre^{+} \quad \} \\
 & C = C' \ \vee\ C \in pre^{+}.C' \ \vee\ C' \in pre^{+}.C \ .
\end{array}
$$

We now show that, assuming $C \in PF.V \wedge C' \in PF.V$, the second and third disjuncts are false.

$$
\begin{array}{ll}
 & C \in pre^{+}.C' \ \wedge\ C' \in PF.V \\
\Rightarrow & \{ \quad \text{definition of } pre^{+} \quad \} \\
 & C \in pre^{+}.(PF.V) \\
\Rightarrow & \{ \quad \text{lemma 30} \quad \} \\
 & \neg(C \in PF.V) \ .
\end{array}
$$

We conclude that

$$C \in pre^{+}.C' \ \wedge\ C \in PF.V \ \wedge\ C' \in PF.V \ \equiv\ \mathsf{false} \ .$$

Interchanging the roles of C and C', the third disjunct is also false. The lemma follows straightforwardly. $\Box$

4. Block Patterns and Penney-Ante Games

We now specialise the analysis to block patterns and Penney-Ante-type games. In Penney-Ante games, each player chooses a word. A die (with $|T|$ faces each of which bears one of the elements of T, but not necessarily fair) is then thrown repeatedly until one of the chosen words occurs as a suffix of the play. The player who made the choice is declared the winner. For example, suppose the alphabet has two symbols a and b, one player chooses the word a and the second player chooses the word bb. There are just three complete plays of this game: the words a, ba and bb. The first player wins in the first two cases and the second player wins in the third case. Note that this is a first-past-the-post game — see example 6. Recognition of a block pattern (see example 27) is a special case of a Penney-Ante game with one player.

Consider a set W of words over an alphabet T. Note that we do not assume at this stage that W is finite.

The set S is defined to be the set of minimal-length words that end in a word in W. Formally,

$$S \ = \ T^{*}W \cap \neg(T^{*}WT^{+}) \ .$$

Equivalently, $S = PF.T^{*}W$.

Returning to the example above, taking W to be $\{a,bb\}$ we have:

$$S \ = \ \{a,b\}^{*}\{a,bb\} \cap \neg(\{a,b\}^{*}\{a,bb\}\{a,b\}^{+}) \ = \ \{a,ba,bb\} \ .$$

In this very simple example, the set S is finite; this is not the case in general.

Theorem 32. For all W such that $W \subseteq T^{*}$ and $\emptyset \neq W$, $PF.T^{*}W$ is a first-past-the-post game.

Proof Let S denote $PF.T^*W$ and let N denote $pre^+.S$. Then that S satisfies 5(a) in the definition of a first-past-the-post game, i.e.

(33) $\quad N \cap S = \emptyset$,

is immediate from lemma 30 by instantiating V to T^*W.

It remains to verify the property 5(b). Now,

$$
\begin{array}{ll}
 & pre^*.S \;=\; \{\varepsilon\} \cup (pre^+.S)T \\
= & \{ \quad pre^*.S \;=\; S \cup pre^+.S \quad , \quad N = pre^+.S \quad \} \\
 & S \cup N \;=\; \{\varepsilon\} \cup NT \\
= & \{ \quad T \text{ is the alphabet, } T^* = \{\varepsilon\} \cup T^+ \quad \} \\
 & (S \cup N) \;\cap\; (\{\varepsilon\} \cup T^+) \;=\; \{\varepsilon\} \cup NT \\
= & \{ \quad \text{distributivity of intersection over union,} \\
 & \quad\;\; \text{assumption: } \emptyset \neq W \text{. So } \{\varepsilon\} \subseteq S \cup N \quad \} \\
 & \{\varepsilon\} \cup ((S \cup N) \cap T^+) \;=\; \{\varepsilon\} \cup NT \\
= & \{ \quad \text{cancellation property of languages: } \varepsilon \text{ has length } 0 \\
 & \quad\;\; \text{and words in } T^+ \text{ have length at least } 1 \quad \} \\
 & (S \cup N) \cap T^+ \;=\; NT \\
= & \{ \quad \text{definition of set concatenation and equality} \quad \} \\
 & \langle \forall B,a \;:\; B \in T^* \wedge a \in T \;:\; Ba \in S \cup N \equiv B \in N \rangle \quad .
\end{array}
$$

Now, for all $B \in T^*$ and $a \in T$, we have

$$
\begin{array}{ll}
 & Ba \in S \cup N \\
\Rightarrow & \{ \quad \text{definition of } pre \quad \} \\
 & B \in pre.(S \cup N) \\
= & \{ \quad S \cup N = pre^*.S \quad \} \\
 & B \in pre^+.S \\
= & \{ \quad N = pre^+.S \quad \} \\
 & B \in N \quad .
\end{array}
$$

For the opposite implication, choose an arbitrary word C in W. Then, for all $B \in T^*$ and $a \in T$,

$$
\begin{array}{ll}
 & B \in N \\
= & \{ \quad C \in W \quad \} \\
 & B \in N \;\wedge\; BaC \in T^*W \\
\Rightarrow & \{ \quad \text{lemma 28, definition of } S \quad \} \\
 & B \in N \;\wedge\; \langle \exists k : 0 \leq k \leq \#(BaC) : pre^k.(BaC) \in S \rangle \\
\Rightarrow & \{ \qquad pre^*.B \;\cap\; S \\
 & \qquad \subseteq \quad \{ \quad \text{assume: } B \in N \quad \} \\
 & \qquad\quad pre^*.N \;\cap\; S
\end{array}
$$

$$
\begin{array}{ll}
= & \{ \quad pre^*.N = pre^*.(pre^+.S) = pre^+.S = N \quad \} \\
& N \cap S \\
= & \{ \quad (33) \quad \} \\
& \emptyset \quad .
\end{array}
$$

That is, assuming $B{\in}N$,

$\langle \forall k \ : \ \#(aC) \le k \le \#(BaC) \ : \ \neg(pre^k.(BaC) \in S) \rangle$ }

$$
\begin{array}{ll}
& \langle \exists k : 0 \le k \le \#C : pre^k.(BaC) \in S \rangle \\
\Rightarrow & \{ \quad \text{range splitting on } k = \#C \text{, definition of } N \quad \} \\
& Ba{\in}S \ \vee \ Ba{\in}N \\
= & \{ \quad \text{definition of set union} \quad \} \\
& Ba \in S \cup N \quad .
\end{array}
$$

□

We now combine theorem 32 with theorem 12.

Lemma 34. For all W such that $W \subseteq T^*$ and $\emptyset \neq W$,

(35) $\neg(pre^+.(PF.T^*W)) \ = \ T^*WT^*$, and

(36) $\neg(pre^*.(PF.T^*W)) \ = \ T^*WT^+$.

Proof The first equation is proved as follows:

$$
\begin{array}{ll}
& \neg(pre^+.(PF.T^*W)) \\
= & \{ \quad \text{assumption: } W \subseteq T^* \text{ and } \emptyset \neq W \text{, theorems 12 and 32} \quad \} \\
& (PF.T^*W)\,T^* \\
= & \{ \quad \text{corollary 29 with } V := T^*W \quad \} \\
& T^*W\,T^* \quad .
\end{array}
$$

The second equation is proved in the same way. □

Lemma 34 gives useful insight into the (complete) plays of a Penney-Ante game but also facilitates comparison with [GO81]: Guibas and Odlyzko implicitly use $\neg(T^*\,W\,T^*)$ as the definition of a complete play. (See, for example, their definition of the function f [GO81, p.184]; see also section 5.2.)

4.1. Equations in Languages

When W is a regular language, $PF.T^*W$ is also a regular language and so can be recognised by a deterministic finite-state machine. A deterministic machine is an unambiguous representation of a regular language; it is therefore possible to compute the expected length of a Penney-Ante game by using lemma 20 to construct a system of simultaneous equations in the expected length of the set of words recognised by each state of the machine. If the set W is finite, the construction of the deterministic finite-state machine can be adapted so that the probability of winning with a given word in W can also be computed. However, the number of equations to be solved is equal to the number of states in the finite-state machine, which can be commensurate with the total length of words in W . (See section 6 for further details.) In this section, we show how to construct from a given language W a (non-linear) system of simultaneous equations in languages. The system has one equation for each word in W (which is not necessarily finite); as we show in section 4.4, these equations together with the equation 5(b) uniquely characterise $PF.T^*W$. Although W need not be finite, we do assume that it is "reduced", as defined below.

The set W is said to be *reduced* if, for all words A and B in W, A is a subword[1] of B equivales A equals B. The assumption that W is reduced is sensible because without it the game would be either unfair or ill-defined — if A is a proper suffix of B, the winner of complete play B is not well-defined, and if A is a proper subword of B and not a proper suffix, the player who chooses B can never win. For example the set $\{a,ba,bb\}$ in example 25 is not reduced. (If the complete play is ba, it is not clear whether the winner is the player choosing a or the player who chooses ba.) The need for the assumption also appears formally in our calculations.

Theorem 38 shows how to construct a (non-linear) system of equations defining the sets of complete plays, S, and incomplete plays, N, in a Penney-Ante game defined by the set of words W. An auxiliary definition is needed in ordered to formulate the theorem. Specifically, for words A and B, we define

$$(37) \qquad B \between A \;=\; \{E,F,G : B=EF \wedge A=FG \wedge 1 \leq \#F : G\} \quad .$$

We pronounce $B \between A$ as B *match* A. Note that, in spite of the symbol by which it is denoted, the match operator is not symmetric. See example 44 below for instances of the match operator and the application of theorem 38.

Theorem 38. Suppose W is a reduced set of words. Define S, N and S_A, for each word A in W, by

$$(39) \qquad S \;=\; PF.T^{*}W$$

$$(40) \qquad N \;=\; pre^{+}.S$$

$$(41) \qquad S_A \;=\; T^{*}\{A\} \cap S$$

Then

$$(42) \qquad S \;=\; \langle \cup A : A \in W : S_A \rangle$$

$$(43) \qquad N\{A\} \;=\; \langle \cup B : B \in W : S_B(B \between A) \rangle$$

Proof Equation (42) is straightforward. For the proof of (43), we first note that

$$\begin{array}{ll} & N\{A\} \;=\; \langle \cup B : B \in W : S_B(B \between A) \rangle \\ \equiv & \langle \forall C :: C \in N \;\equiv\; \langle \exists B : B \in W : CA \in S_B(B \between A) \rangle \rangle \quad . \end{array}$$

(This is a simple application of the definition of equality of sets, set concatenation and set union.) So our task is to prove

$$C \in N \;\equiv\; \langle \exists B : B \in W : CA \in S_B(B \between A) \rangle$$

for all words A in W and all words C. The proof is in two parts. In the first part, we simplify the righthand side. Then, in the second part, we prove the equivalence of the lefthandside with the simplified righthand side.

For the first part we have:

$$\begin{array}{ll} & \langle \exists B : B \in W : CA \in S_B(B \between A) \rangle \\ = & \{ \quad \text{definition of } B \between A\text{: (37)} \quad \} \\ & \langle \exists B,E,F,G : B \in W \wedge B=EF \wedge A=FG \wedge 1 \leq \#F : CA \in S_B\{G\} \rangle \\ = & \{ \quad \text{cancellation property of concatenation of words} \quad \} \\ & \langle \exists B,E,F,G : B \in W \wedge B=EF \wedge A=FG \wedge 1 \leq \#F : CF \in S_B \rangle \\ = & \{ \quad \text{case analysis on } \quad \#B < \#F \vee \#F \leq \#B : \end{array}$$

[1] A is a subword of B equivales there are words C and D such that $B=CAD$.

$\#B < \#F \ \wedge \ B{\in}W \ \wedge \ A = FG \ \wedge \ CF \in S_B$

$\Rightarrow \quad \{ \quad S_B \subseteq T^*\{B\} \quad \}$

B is a proper subword of $A \ \wedge \ B{\in}W$

$\Rightarrow \quad \{ \quad W$ is reduced, $A{\in}W \quad \}$

false .

}

$\langle \exists\, B,E,F,G \,:\, B{\in}W \wedge B = EF \wedge A = FG \wedge 1{\leq}\#F \leq \#B \,:\, CF \in S_B \rangle$

$= \quad \{ \quad \langle \exists E \,:\, B{\in}W \ \wedge \ B = EF \ \wedge \ \#F \leq \#B \,:\, CF \in S_B \rangle$

$= \quad \{ \quad S_B \subseteq T^*\{B\} \quad \}$

$\#F \leq \#B \wedge B{\in}W \ \wedge \ CF \in S_B$. }

$\langle \exists\, B,F,G \,:\, B{\in}W \wedge 1 \leq \#F \ \wedge A = FG \,:\, CF \in S_B \rangle$

$= \quad \{ \quad S \ = \ \langle \cup B : B{\in}W : S_B \rangle \quad \}$

$\langle \exists\, F,G \,:\, 1 \leq \#F \wedge A = FG \,:\, CF \in S \rangle$.

It remains to prove the equivalence of $\langle \exists B : B{\in}W : CA \in S_B(B \between A) \rangle$ and $C{\in}N$. This we do by mutual implication.

$\langle \exists\, F,G \,:\, 1 \leq \#F \wedge A = FG \,:\, CF \in S \rangle$

$\Rightarrow \quad \{ \quad$ definition of N : (40) $\quad \}$

$C{\in}N$

$= \quad \{ \quad A{\in}W \quad \}$

$C{\in}N \ \wedge \ CA \in T^*W$

$\Rightarrow \quad \{ \quad$ definition of S : (39), lemma 28 $\quad \}$

$C{\in}N \ \wedge \ \langle \exists\, E,G : CA = EG : E{\in}S \rangle$

$\Rightarrow \quad \{ \quad$ case analysis on $\quad \#A \leq \#G \ \vee \ \#G < \#A$

$\#A \leq \#G \wedge C{\in}N \ \wedge \ CA = EG \wedge E{\in}S$

$\Rightarrow \quad \{ \quad$ simple arithmetic on lengths $\quad \}$

$C{\in}N \ \wedge \ E{\in}S \ \wedge \ E \in pre^*.\{C\}$

$\Rightarrow \quad \{ \quad pre^*$ is monotonic $\quad \}$

$E \ \in \ S \cap pre^*.N$

$\Rightarrow \quad \{ \quad$ definitions of N and S : (40) and (39), $pre^* \circ pre^+ = pre^+$

S is prefix-free: lemma 30 $\quad \}$

false .

}

$C{\in}N \ \wedge \ \langle \exists\, E,G \,:\, CA = EG \wedge \#G < \#A \,:\, E{\in}S \rangle$

$\Rightarrow \quad \{ \quad$ cancellation property of concatenation of words $\quad \}$

$\langle \exists\, F,G \,:\, 1 \leq \#F \wedge A = FG \,:\, CF \in S \rangle$.

□

Example 44. Suppose the alphabet has two symbols h and t. Suppose the set W has three elements hh, ht and th. The set S is $\{hh{,}ht\}\cup\{t\}^*\{th\}$ and the sets S_{hh}, S_{ht} and S_{th} are, respectively, $\{hh\}$, $\{ht\}$ and $\{t\}^*\{th\}$; the set N is $\{\varepsilon{,}h\}\cup\{t\}^*\{t\}$.

The following table shows $B \between A$ for each of the 9 combinations of B and A. (Rows are indexed by B and columns by A.)

$\between$	hh	ht	th
hh	$\{\varepsilon{,}h\}$	$\{t\}$	$\emptyset$
ht	$\emptyset$	$\{\varepsilon\}$	$\{h\}$
th	$\{h\}$	$\{t\}$	$\{\varepsilon\}$

The appropriate instances of (43) are thus as follows:

$$\begin{array}{rcl} N\{hh\} & = & S_{hh}\{\varepsilon{,}h\}\cup S_{th}\{h\} \\ N\{ht\} & = & S_{hh}\{t\}\cup S_{ht}\cup S_{th}\{t\} \\ N\{th\} & = & S_{ht}\{h\}\cup S_{th} \end{array}$$

(Some simplification has been applied to these equations. So, for example, in the first equation the term $S_{ht}\emptyset$ has been omitted and, in the second equation, $S_{ht}\{\varepsilon\}$ has been simplified to S_{ht}.)

These equations are complemented by the equations:

$$\begin{array}{rcl} N\cup S & = & \{\varepsilon\}\cup N\{h{,}t\} \\ S & = & S_{hh}\cup S_{ht}\cup S_{th} \end{array}$$

The combination of the two sets of equations is the basis for calculating the probabilities of winning a game with three players who each choose the three words hh, ht and th as the eventual outcome of the game, as we discuss in the next section. □

4.2. Solov'ev's Equation and Conway's Equation

Suppose we are given a probability distribution p on the elements of the alphabet T. Suppose W is reduced and S equals $PF.T^*W$. Then, for each word A in W, $h_p.S_A$ is the relative frequency of words ending in A among all complete plays of the game (theorem 23). We show how to use (43) to evaluate $h_p.S_A$ for each A. In the case that W has one element, this gives Solov'ev's equation for the expected length of a sequence of observations culminating in (the "block pattern") A; see theorem 48. In the case that W has two elements, this gives Conway's formula for the probability that each person wins in a two-person Penney-Ante game; see theorem 50.

Lemma 45. Suppose V is a family of languages V_B indexed by words B in W. Suppose W is reduced and finite. Then $\langle\cup B : B{\in}W : S_B V_B\rangle$ is unambiguous.

Proof By lemma 31, each term $S_B V_B$ is unambiguous. Also, for all words D, D', E and E', and all words B and C in W,

$$\begin{array}{ll} & DE = D'E' \;\wedge\; D{\in}S_B \;\wedge\; D'{\in}S_C \\ \Rightarrow & \{ \quad S_B{\cup}S_C \subseteq PF.T^*W \text{ , lemma 31} \quad \} \\ & DE = D'E' \;\wedge\; D{\in}S_B \;\wedge\; D'{\in}S_C \;\wedge\; D = D' \\ \Rightarrow & \{ \quad W \text{ is reduced, } S_B \subseteq T^*\{B\} \text{ , } S_C \subseteq T^*\{C\} \quad \} \\ & D = D' \;\wedge\; E = E' \;\wedge\; B = C \quad . \end{array}$$

□

We are now in a position to formulate the theorems attributed to Solov'ev and Conway. In the statement of the theorems, the binary operator " : " is defined on pairs of words by, for all C and D,

$$(46) \qquad C : D \quad = \quad \frac{h_p.(C \between D)}{h_p.D} \quad .$$

This operator generalises the one with the same name[2] in [GKP89]. See theorem 50, below, for further explanation of the generalisation. When applying the definition we assume, of course, that $h_p.D$ is non-zero.

First, we combine lemmas 20 and 45 with theorem 38 to obtain a linear system of equations in the quantities $h_p.N$ and $h_p.S_A$, for each A in W.

Theorem 47. For all A in W,

$$h_p.N \;=\; \langle \Sigma B : B{\in}W : h_p.S_B \times (B:A) \rangle \quad .$$

Also, for all A and B in W,

$$B:A \;=\; \left\langle \Sigma\, E,F,G \,:\, B{=}EF \wedge A{=}FG \wedge 1{\le}\#F \,:\, (h_p.F)^{-1} \right\rangle \quad .$$

Finally,

$$1 \;=\; \langle \Sigma B : B{\in}W : h_p.S_B \rangle \quad .$$

Proof The expression $N\{A\}$ is obviously unambiguous. So, by lemma 20, $h_p.N\{A\}$ is the product of $h_p.N$ and $h_p.\{A\}$. Applying h_p to both sides of (43) and dividing through by $h_p.A$, this gives the left side of the first equation above. The righthand side is immediate from lemma 20 and lemma 45.

For the second equation we have:

$$\begin{array}{ll}
& B:A \\
= & \quad \{ \quad \text{definition} \quad \} \\
& \dfrac{h_p.(B \between A)}{h_p.A} \\
= & \quad \{ \quad \text{(37) and lemma 20} \\
& \qquad\quad \text{(obviously, righthand side of (37) is unambiguous),} \\
& \qquad\quad \text{distributivity, Leibniz} \quad \} \\
& \left\langle \Sigma\, E,F,G \,:\, B{=}EF \wedge A{=}FG \wedge 1{\le}\#F \,:\, \dfrac{h_p.G}{h_p.FG} \right\rangle \\
= & \quad \{ \quad h_p.FG = h_p.F \times h_p.G \quad \} \\
& \left\langle \Sigma\, E,F,G \,:\, B{=}EF \wedge A{=}FG \wedge 1{\le}\#F \,:\, (h_p.F)^{-1} \right\rangle \quad .
\end{array}$$

The final equation is a consequence of theorem 23. (This is the only equation that assumes that p is a probability distribution on T.) □

Theorem 47 is invalid when, for some $A{\in}W$, $h_p.A{=}0$. The simple reason is that division by zero is not allowed. The equations can obviously be reformulated so that division by zero is avoided. Even so, the equations have no solution in the case that $h_p.A{=}0$ for all $A{\in}W$. (This will happen if $h_p.a{=}0$ for some symbol $a{\in}T$ that occurs in every word in W.) This is to be expected: such a game involves waiting for an event that will never occur. Similarly, theorem 23 is invalid in such a case since the relative frequency of a particular complete play occurring is zero divided by zero.

Typically $A:A$ is greater than 1; it is the expected length of the first occurrence of block pattern A, as shown in the next theorem.

Theorem 48 (Solov'ev's formula). Suppose $S{=}PF.T^*\{A\}$. Then

$$e_p.S \;=\; A:A \quad .$$

[2] The notation used by Gardner [Gar74] in what appears to be the first publication of Conway's formula is CD. In later publications, several authors use Gardner's notation.

Proof

$$
\begin{array}{ll}
& e_p.S \\
= & \{ \quad \text{theorem 24} \quad \} \\
& h_p.N \\
= & \{ \quad \text{theorem 47 with } W := \{A\} \text{ (using } h_p.S = 1 \text{)} \quad \} \\
& A:A \ .
\end{array}
$$

□

Example 49. Suppose the alphabet has two symbols h and t (for heads and tails). Suppose k is a natural number and A is the word $h^k t$ and B is the word h^{k+1} . Then

$$
\begin{array}{rcl}
A \between A & = & \{\varepsilon\} \\
B \between B & = & \left\{ j : 0 \le j \le k : h^j \right\}
\end{array}
$$

Suppose further that $p.h = q$ and $p.t = r$, where $q+r = 1$. It follows that

$$
\begin{array}{rcl}
h_p.(A \between A) & = & 1 \\
h_p.(B \between B) & = & \dfrac{1-q^{k+1}}{1-q}
\end{array}
$$

Since $h_p.A$ is $q^k \times r$ and $h_p.B$ is q^{k+1} ,

$$
A:A \quad = \quad \frac{1}{q^k \times r}
$$

and

$$
B:B \quad = \quad \frac{1-q^{k+1}}{(1-q) \times q^{k+1}} \ .
$$

It follows from theorem 48 that

$$
\frac{e_p.(PF.T^*\{B\})}{e_p.(PF.T^*\{A\})} \quad = \quad \frac{1-q^{k+1}}{q} \ .
$$

In the literature, this property is sometimes described as "paradoxical" [Col82, GKP89]. An informal summary might be: "the expected number of coin-tosses before h^{k+1} is encountered is approximately $\frac{1}{q}$ times greater than the expected number of coin-tosses before $h^k t$ is encountered", or as summarised in [GKP89]: "patterns with no self-overlaps occur sooner than overlapping patterns do!". Great care must be taken, however, not to be misled by such informal statements. It does *not* mean that in the game defined by $W = \{h^k t, h^{k+1}\}$ the expected number of coin-tosses before $h^k t$ is encountered will be in the ratio $\frac{1-q^{k+1}}{q}$ to the expected number of coin-tosses before h^{k+1} is encountered. This is because, in general, the set $PF.T^*\{A,B\} \cap T^*\{A\}$ is not equal to $PF.T^*\{A\}$. (For example, $hthh \in PF.T^*\{hh\}$ but $hthh \notin (PF.T^*\{hh,ht\} \cap T^*\{hh\})$.) In fact, in the Penney-Ante game defined by $W = \{h^k t, h^{k+1}\}$, we have

$$
\frac{h_p.(PF.T^*\{h^k t, h^{k+1}\} \cap T^*\{h^k t\})}{h_p.(PF.T^*\{h^k t, h^{k+1}\} \cap T^*\{h^{k+1}\})} \quad = \quad \frac{r}{q}
$$

and

$$
\frac{e_p.(PF.T^*\{h^k t, h^{k+1}\} \cap T^*\{h^k t\})}{e_p.(PF.T^*\{h^k t, h^{k+1}\} \cap T^*\{h^{k+1}\})} \quad = \quad \frac{r}{q} \ .
$$

Both equations are just what one would expect and not at all "paradoxical". The "paradox" is caused by comparing events in two quite different event spaces. □

Theorem 50. Suppose $W=\{A,B\}$. Suppose W is reduced. Then

$$\frac{h_p.S_A}{h_p.S_B} \quad = \quad \frac{B:B \;-\; B:A}{A:A \;-\; A:B} \quad .$$

Proof Instantiating theorem 47, we get two equations, one for each word in W :

$$h_p.N \quad = \quad h_p.S_A \times (A:A) + h_p.S_B \times (B:A) \quad \text{, and}$$

$$h_p.N \quad = \quad h_p.S_A \times (A:B) + h_p.S_B \times (B:B) \quad .$$

Eliminating $h_p.N$ gives the theorem. □

Corollary 51 (Conway's formula). If A and B have equal length, and p assigns equal values to each element of T then

$$\frac{h_p.S_A}{h_p.S_B} \quad = \quad \frac{h_p.(B \between B) - h_p.(B \between A)}{h_p.(A \between A) - h_p.(A \between B)} \quad .$$

The latter is equivalent to the formula attributed to John Horton Conway in [Gar74] for the odds of A winning against B in a Penney-Ante game where a coin is tossed and the probability of a head or tail occurring is $\frac{1}{2}$. In Conway's formula, the notation $B:A$ is used for $h_p.(B \between A) \times 2^{\#A\,-\,1}$. It is not clear from the published literature whether or not Conway derived the general formula given in theorem 50. □

The examples below test the use of theorem 50 on cases where it is easy to predict the relative frequency of occurrence of words in S_A and in S_B .

Example 52. Suppose the alphabet has two symbols h and t (for heads and tails). Suppose k is a natural number and A is the word $h^k t$ and B is the word h^{k+1} . Suppose further that $p.h=q$ and $p.t=r$, where $q+r=1$ and both q and r are non-zero. A simple argument establishes that the relative frequency of A compared to B in a Penney-Ante game is $\frac{r}{q}$. We can check that this is predicted by theorem 50 as follows. We first calculate that

$$\begin{array}{lcl} A \between A & = & \{\varepsilon\} \\ A \between B & = & \emptyset \\ B \between A & = & \{j : 0 \leq j < k : h^j t\} \quad . \end{array}$$

Then.

$$\begin{array}{lcl} h_p.(A \between B) & = & 0 \\ h_p.(B \between A) & = & \dfrac{(q^k - 1) \times r}{q-1} \quad . \end{array}$$

Combining these with the calculations in example 49 and substituting in theorem 50, we get, for example,

$$B:A \quad = \quad \frac{(q^k-1)\times r}{(q-1)\times(q^k \times r)}$$

Hence, applying theorem 50 (top formula) (and a lot of simplification), we get

$$\frac{h_p.S_A}{h_p.S_B} \quad = \quad \frac{r}{q}$$

as expected. □

Example 53. Suppose the alphabet has two symbols a and b. Suppose the set W has two elements, A and B, equal to a and bb, respectively. Suppose $p.a=q$ and $p.b=r$, where $q+r=1$. As observed earlier, $PF.(\{a,b\}^*\{a,bb\})=\{a,ba,bb\}$. If q and r model the relative frequency of occurrences of a and b, respectively, it is clear that the relative frequency of S_A, which equals $\{a,ba\}$, is $q+r\times q$ and the relative frequency of S_B, which equals $\{bb\}$, is r^2. Let us check that this is what is predicted by theorem 50.

We calculate that $A \mathbin{\overline{\underline{\circ}}} A$ equals $\{\varepsilon\}$, $B \mathbin{\overline{\underline{\circ}}} B$ equals $\{\varepsilon,b\}$ and both $A \mathbin{\overline{\underline{\circ}}} B$ and $B \mathbin{\overline{\underline{\circ}}} A$ equal the empty set. The h_p values are now easily calculated. Applying theorem 50, we get

$$\frac{h_p.S_A}{h_p.S_B} \quad = \quad \frac{(1{+}r)\times q-0}{1\times r^2-0}$$

which simplifies to $\dfrac{(1{+}r)\times q}{r^2}$. Exploiting the fact that $q+r=1$ and $h_p.S_A+h_p.S_B=1$, it follows that $h_p.S_A$ equals $1-r^2$ and $h_p.S_B$ equals r^2. □

The next example is of a game with an infinite number of players.

Example 54. Suppose $T=\{a,b,c\}$ and $W=\{a\}\{b\}^*\{c\}$. (So each word in W is of the form ab^kc for some k, $0\le k$. Note that W is not finite but it is reduced.) It is easy to verify that $ab^kc \mathbin{\overline{\underline{\circ}}} ab^kc=\{\varepsilon\}$ and, when $j\neq k$, $ab^jc \mathbin{\overline{\underline{\circ}}} ab^kc = \emptyset$. Thus:

$$\begin{array}{rcl} N\{ab^kc\} & = & S_{ab^kc} \\ N\cup S & = & \{\varepsilon\}\cup N\{a,b,c\} \\ S & = & \langle\cup k : 0\le k : S_{ab^kc}\rangle \end{array}$$

It is immediate from these equations that $S=N\{a\}\{b\}^*\{c\}$. However, it is difficult to “solve” them in the sense of determining a regular expression defining N. Indeed, it is not even clear that there is a unique solution for N; see section 4.4.

Suppose now that $p.a=q$, $p.b=r$ and $p.c=s$, where $q+r+s=1$. Then, exploiting the above equation for S, we obtain:

$$\begin{array}{rcl} h_p.N\times q\times r^k\times s & = & h_p.S_{ab^kc} \\ h_p.N + h_p.N\times q\times r^*\times s & = & 1 + h_p.N\times(q{+}r{+}s) \end{array}$$

(where we write r^* for $\frac{1}{1-r}$). It follows that $h_p.N=\frac{1-r}{q\times s}$ and $h_p.S_{ab^kc}=(1-r)\times r^k$. So the expected length of a game is $\frac{1-r}{q\times s}$ (which equals $\frac{1}{q}+\frac{1}{s}$) and the probability that the recognised pattern is ab^kc is $(1-r)\times r^k$. □

4.3. Expected Lengths of Complete Plays

In this subsection we show how the above analysis can be repeated for calculating expected values. Specifically, lemmas 20 and 45 are used to simplify the result of applying the function e_p to both sides of (43). This yields a system of equations that can be solved to obtain $e_p.N$ and $e_p.S_A$, for each A in W. Formally, the analogue to theorem 47 is that

$$\begin{array}{rl} & h_p.N\times\#A + e_p.N \\ = & \langle\Sigma B\ :\ B{\in}W\ :\ e_p.S_B\times(B:A) + h_p.S_B\times(B::A)\rangle \end{array}$$

where

$$B::A \quad = \quad \left\langle\Sigma\, E,F,G\ :\ B=EF\wedge A=FG\wedge 1{\le}\#F\ :\ (h_p.F)^{-1}\times\#G\right\rangle$$

and

$$h_p.N \;=\; e_p.S \;=\; \langle\Sigma A : A{\in}W : e_p.S_A\rangle \quad .$$

The interpretation of $e_p.S_A$ is the expected length of complete plays that end in the word A.

4.4. Uniqueness

In the case that a Penney-Ante game has just two players, theorem 50 together with the equation $h_p.S_A + h_p.S_B = 1$ enables one to calculate both $h_p.S_A$ and $h_p.S_B$. In other words, it is possible to determine the probability that each of the players wins. This raises the question whether or not the system of equations

$$
\begin{array}{lrcl}
(55) & N \cup S & = & \{\varepsilon\} \cup NT \\
(56) & S & = & \langle \cup A : A{\in}W : S_A \rangle \\
(57) & N\{A\} & = & \langle \cup B : B{\in}W : S_B(B \between A) \rangle
\end{array}
$$

(the combination of definition 5(b) and (43)) viewed as equations in the unknowns N , S and S_A (for each $A{\in}W$), has a unique solution independently of the size of W .

The answer is no. A simple example demonstrates that there may be more than one solution. Suppose $T = \{a\} = W$. Then, since $a \between a = \{\varepsilon\}$, we get just two equations (equation (56) is trivial):

$$
\begin{array}{rcl}
N \cup S & = & \{\varepsilon\} \cup N\{a\} \\
N\{a\} & = & S
\end{array}
$$

As is easily checked, one solution to these equations is $N = \{\varepsilon\}$ and $S = \{a\}$. (This is the desired solution.) A second solution is $N = \{a\}^*$ and $S = \{a\}^+$.

Note that, although these two equations do not have a unique solution, we can use them to determine $h_p.N$ and $h_p.S$. Specifically, since inevitably $h_p.a = 1$, we get the equations:

$$
\begin{array}{rcl}
h_p.N + h_p.S & = & 1 + h_p.N \\
h_p.N & = & h_p.S
\end{array}
$$

Unsurprisingly, the expected length of a complete play is 1 . (Apply theorem 24.) Note, however, that $h_p.\{a\}^*$ is undefined. (Recall that $\{a\}^*$ is a solution for N .)

It was a surprise to us that the equations in languages do not have a unique solution since Guibas and Odlyzko [GO81] prove that the derived equations in generating functions do have unique solutions. Their argument is based on the fact that, when $\{A,B\}$ is reduced, $\varepsilon \in A \between B \equiv A = B$ for all words A and B .

In this section, we show that the system of equations (55), (56) and (57) have both a least solution and a greatest solution! This raises a doubt about their use in calculating probabilities. A step towards resolving the doubt is to show that the least solution is the unique solution when we add the requirement that $N \cap S = \emptyset$ (cf. clause (a) in definition 5 of a first-past-the-post game).

Lemmas 58, 60 and 61 below do not assume that $N \cap S = \emptyset$; theorem (64) does make the assumption.

Lemma 58. Suppose N , S and S_A ($A{\in}W$) solve the system of equations (55), (56) and (57). Then

$$\langle \forall A \ : \ A{\in}W \ : \ S_A \subseteq T^*\{A\} \rangle \ \wedge \ S \subseteq T^*W \quad .$$

Proof

$$
\begin{array}{ll}
 & S_A \text{ solves (57)} \\
\Rightarrow & \{ \quad \text{equality of sets} \quad \} \\
 & S_A(A \between A) \subseteq N\{A\} \\
\Rightarrow & \{ \quad \{\varepsilon\} \subseteq A \between A \ , \ N \subseteq T^* \text{ , monotonicity of concatenation} \quad \} \\
 & S_A\{\varepsilon\} \subseteq T^*\{A\} \\
= & \{ \quad \{\varepsilon\} \text{ is the unit of concatenation} \quad \} \\
 & S_A \subseteq T^*\{A\} \quad .
\end{array}
$$

The inclusion $S \subseteq T^*W$ follows immediately from equation (56). □

Corollary 59. When W is non-empty and $\varepsilon \notin W$, the system of equations (55), (56) and (57) has greatest solutions

$$
\begin{aligned}
N &= T^* \\
S &= T^* W \\
S_A &= T^* \{A\}
\end{aligned}
$$

Proof It is easy to check that the given solutions do indeed satisfy (55) and (56). To check that (57) is satisfied we calculate as follows:

$$
\begin{array}{ll}
 & \langle \cup B : B \in W : T^*\{B\}(B \sqsupset\!\sqsubset A)\rangle \\
= & \{ \quad \text{range splitting; } \varepsilon \notin W \text{ so } \{\varepsilon\} \subseteq A \sqsupset\!\sqsubset A \quad \} \\
 & T^*\{A\} \cup \langle \cup B,C : B \in W \wedge \varepsilon \neq C \wedge C \in B \sqsupset\!\sqsubset A : T^*\{BC\}\rangle \\
= & \{ \quad B \in W \wedge C \in B \sqsupset\!\sqsubset A \Rightarrow BC \in T^*\{A\}\,,\ T^* = T^* T^*\,, \text{ set calculus} \quad \} \\
 & T^*\{A\} \ .
\end{array}
$$

That $N = T^*$ is the largest solution for N is obvious. That $S = T^* W$ and $S_A = T^*\{A\}$ are the largest solutions for S and S_A (for each $A \in W$) is immediate from lemma 58. □

Corollary 59 is disturbing because it is the equations (55), (56) and (57) that underlie the construction of the generating functions for the languages $N = \neg(T^* W T^*)$ and $S = T^* W \cap \neg(T^* W T^+)$. In [GKP89, eqns. (8.67) and (8.68)], for example, the equations are instantiated for the language $W = \{\texttt{THTTH}\}$ and then used to derive generating functions. The question is: why don't the generating functions so constructed correspond to the solutions $N = T^*$ and $S = T^*\{\texttt{THTTH}\}$ instead? The answer to this question is that the construction of the generating functions assumes distributivity properties that only hold for unambiguous expressions, in particular, the generating function of the language $U \cup V$ is the sum of the generating function of U and the generating function of V only if $U \cap V = \emptyset$. If we add to the equations (55), (56) and (57) the requirement that $N \cap S = \emptyset$, the equations do have a unique solution. This is proved in theorem 64.

Lemma 60. Suppose W is reduced. Suppose N, S and S_A ($A \in W$) solve the system of equations (55), (56) and (57). Then the statements $\varepsilon \in S$, $\{\varepsilon\} = S$, $\{\varepsilon\} = W$ and $\emptyset = N$ are all equivalent.

Proof

$$
\begin{array}{ll}
 & \varepsilon \in S \\
\Rightarrow & \{ \quad \text{lemma 58} \quad \} \\
 & \varepsilon \in T^* W \\
= & \{ \quad T^* W = W \cup T^+ W\,,\ \neg(\varepsilon \in T^+ W) \quad \} \\
 & \varepsilon \in W \\
= & \{ \quad W \text{ is reduced, } \varepsilon \text{ is a subword of all words} \quad \} \\
 & \{\varepsilon\} = W \\
\Rightarrow & \{ \quad N \text{ and } S \text{ satisfy (57), } \varepsilon \sqsupset\!\sqsubset \varepsilon = \emptyset \quad \} \\
 & \emptyset = N \\
\Rightarrow & \{ \quad N \text{ and } S \text{ satisfy (55)} \quad \} \\
 & \varepsilon \in S \ .
\end{array}
$$

By mutual implication, it follows that the statements in the above calculation are all equivalent. □

Lemma 61. Suppose W is reduced and non-empty. Suppose N , S and S_A ($A \in W$) solve the system of equations (55), (56) and (57). Then

$$PF.T^*W \subseteq S \quad \wedge \quad pre^+.(PF.T^*W) \subseteq N \quad .$$

Proof It is useful to separate the cases that $\varepsilon \in W$ and $\neg(\varepsilon \in W)$. If W is reduced $\varepsilon \in W \equiv \{\varepsilon\} = W$ (because ε is a subword of every word). But then, by lemma 60, the unique solution to (55), (56) and (57) is $S = \{\varepsilon\}$ and $N = \emptyset$ and the theorem holds. From here on, we assume that $\neg(\varepsilon \in W)$.

Suppose N , S and S_A ($A \in W$) satisfy (55), (56) and (57). The induction hypothesis is that for natural number n ,

(62) $$pre^+.(PF.T^*W) \cap T^{\leq n} \subseteq N \cap T^{\leq n}$$

and

(63) $$PF.T^*W \cap T^{\leq n} \subseteq S \cap T^{\leq n} \quad .$$

In the base case, $T^{\leq 0} = \{\varepsilon\}$. The property (63) is trivially true because $T^*W \cap \{\varepsilon\}$ is the empty set by assumption. The property (62) is equivalent to $\varepsilon \in N$, which we prove as follows.

$$\begin{array}{ll}
 & \varepsilon \in N \\
\Leftarrow & \{ \quad N \text{ and } S \text{ satisfy (55)} \quad \} \\
 & \neg(\varepsilon \in S) \\
= & \{ \quad \text{lemma 60} \quad \} \\
 & \neg(\{\varepsilon\} = W) \\
= & \{ \quad \text{assumption} \quad \} \\
 & \mathsf{true} \quad .
\end{array}$$

This establishes the base case of (62) and (63).

For the induction step, assume the induction hypothesis and suppose X has length n and $a \in T$. First,

$$\begin{array}{ll}
 & Xa \in pre^+.(PF.T^*W) \\
\Rightarrow & \{ \quad \text{lemma 30} \quad \} \\
 & X \in pre^+.(PF.T^*W) \quad \wedge \quad Xa \in \neg(PF.T^*W) \\
\Rightarrow & \{ \quad \text{induction hypothesis: (62) applied to } X \quad \} \\
 & X \in N \quad \wedge \quad Xa \in \neg S \\
= & \{ \quad \text{cancellation property of concatenation of words} \quad \} \\
 & Xa \in NT \quad \wedge \quad Xa \in \neg S \\
\Rightarrow & \{ \quad N \text{ and } S \text{ satisfy (55)} \quad \} \\
 & Xa \in N \quad .
\end{array}$$

This establishes the induction step for (62). Now

$$\begin{array}{ll}
 & Xa \in PF.T^*W \\
= & \{ \quad \text{definition of } pre \quad \} \\
 & X \in pre.(PF.T^*W) \quad \wedge \quad Xa \in PF.T^*W \\
\Rightarrow & \{ \quad PF.T^*W \subseteq T^*W \quad \}
\end{array}$$

$X \in pre.(PF.T^*W) \;\wedge\; \langle \exists A,C : A{\in}W : Xa = CA \rangle$

$\Rightarrow \quad \{ \quad$ assumption: $\neg(\varepsilon{\in}W)$. So $\langle \forall A : A{\in}W : 1 \leq \#A \rangle \quad \}$

$X \in pre.(PF.T^*W) \;\wedge\; \langle \exists A,C : A{\in}W \wedge C \in pre^+.(PF.T^*W) : Xa = CA \rangle$

$\Rightarrow \quad \{ \quad (35) \quad \}$

$X \in \neg(T^*WT^*) \;\wedge\; \langle \exists A,C : A{\in}W \wedge C \in pre^+.(PF.T^*W) : Xa = CA \rangle$

$= \quad \{ \quad$ induction hypothesis: (62), applied to $C \quad \}$

$X \in \neg(T^*WT^*) \;\wedge\; \langle \exists A : A{\in}W : Xa \in N\{A\} \rangle$

$= \quad \{ \quad (57) \quad \}$

$X \in \neg(T^*WT^*) \;\wedge\; \langle \exists A,B : A{\in}W \wedge B{\in}W : Xa \in S_B(B \between A) \rangle$

$\Rightarrow \quad \{ \quad W$ is reduced. So $\varepsilon \in B \between A \;\equiv\; B = A \quad \}$

$\langle \exists A : A{\in}W : Xa \in S_A \rangle$

$= \quad \{ \quad (56) \quad \}$

$Xa \in S$.

This establishes the induction step for (63). □

Theorem 64. Suppose W is reduced and non-empty. Suppose N , S and S_A ($A{\in}W$) solve the system of equations (55), (56) and (57) together with the equation

(65) $\quad N{\cap}S = \emptyset$.

Then $N = pre^+.(PF.T^*W)$, $S = PF.T^*W$ and (for each $A{\in}W$) $S_A = T^*\{A\} \cap S$. That is, this is the unique solution to the combined system of equations. Also (by lemma 61) it is the least solution of the system of equations (55), (56) and (57).

Proof It is useful to separate the cases that $\varepsilon{\in}W$ and $\neg(\varepsilon{\in}W)$. When $\varepsilon{\in}W$, we have already shown that the unique solution to (55), (56) and (57) is $S{=}\{\varepsilon\}$ and $N{=}\emptyset$. Noting that (65) is also satisfied, it is also the the unique solution to the combination of (55), (56) and (57) with (65). From here on, we assume that $\neg(\varepsilon{\in}W)$.

Suppose N , S and S_A ($A{\in}W$) satisfy the given system of equations. We first prove by induction on the length of words that, for all words X ,

$$(X{\in}S \;\equiv\; X \in PF.T^*W) \;\wedge\; (X{\in}N \;\equiv\; X \in pre^+.(PF.T^*W)) \quad .$$

The induction hypothesis is thus for natural number n ,

(66) $\quad S \cap T^{\leq n} \;=\; PF.T^*W \cap T^{\leq n}$.

and

(67) $\quad N \cap T^{\leq n} \;=\; pre^+.(PF.T^*W) \cap T^{\leq n}$.

Lemma 61 establishes that, in each case, the lefthand side contains the righthand side so that only the inclusion of the lefthand side in the righthand side has to be proved. Nevertheless the stronger induction hypothesis is useful.

In the base case, $T^{\leq 0} = \{\varepsilon\}$. We have:

$\varepsilon \in pre^+.(PF.T^*W)$

$= \quad \{ \quad (36) \quad \}$

$\varepsilon \in \neg(T^*WT^*)$

$= \quad \{ \quad \varepsilon$ has length $0 \quad \}$

$\neg(\varepsilon \in W)$

$= \quad \{ \quad$ assumption $\quad \}$

true .

Also,

$\varepsilon \in PF.T^*W$

$= \quad \{ \quad$ definition of $PF \quad \}$

$\varepsilon \in T^*W \cap \neg(T^*WT^+)$

$\Rightarrow \quad \{ \quad$ set calculus $\quad \}$

$\varepsilon \in T^*W$

$= \quad \{ \quad T^*W = W \cup T^+W$

assumption: $\neg(\varepsilon \in W)$,

ε has length 0 , words in T^+W have length at least $1 \quad \}$

false

and

$\varepsilon \in S$

$= \quad \{ \quad$ lemma 60 $\quad \}$

$\varepsilon \in W$

$= \quad \{ \quad$ assumption $\quad \}$

false .

Consequently,

$\varepsilon \in N$

$= \quad \{ \quad \neg(\varepsilon \in S) \quad \}$

$\varepsilon \in N \cup S$

$= \quad \{ \quad S$ and N satisfy (55) $\quad \}$

true .

These four equations establish the base case of (66) and (67).

For the induction step, assume the induction hypothesis and suppose X has length n and $a \in T$. We have to prove that

(68) $\quad Xa \in S \ \equiv \ PF.T^*W$

and

(69) $\quad Xa \in N \ \equiv \ Xa \in pre^+.(PF.T^*W)$.

In order to prove (69), we appeal to lemma 61: the lemma establishes that $Xa \in N$ if $Xa \in pre^+.(PF.T^*W)$. For the opposite implication, we exploit lemma 36 to replace $pre^+.(PF.T^*W)$ by $\neg(T^*WT^*)$. Specifically, we prove that

$$Xa \in N \Rightarrow (Xa \in T^*WT^* \Rightarrow \mathsf{false}) \ .$$

As a preliminary, we observe that:

$$Xa \in N$$

$$\Rightarrow \quad \{ \quad N \text{ and } S \text{ satisfy (55), } Xa \neq \varepsilon \quad \}$$

$$Xa \in NT$$

$$= \quad \{ \quad \text{cancellation property of concatenation of words} \quad \}$$

$$X \in N$$

$$= \quad \{ \quad \text{induction hypothesis: (67), applied to } X \quad \}$$

$$X \in pre^{+}.(PF.T^{*}W)$$

$$= \quad \{ \quad pre^{*} \circ pre^{+} \ = \ pre^{*} \quad \}$$

$$\langle \forall C \, : \, C \in pre^{*}.\{X\} \, : \, C \in pre^{+}.(PF.T^{*}W) \rangle$$

$$= \quad \{ \quad \text{induction hypothesis: (67), applied to } C \quad \}$$

$$\langle \forall C \, : \, C \in pre^{*}.\{X\} \, : \, C \in N \rangle \quad .$$

So, if we assume $Xa \in N$, we may also assume every property in the above calculation. (That is we may also assume $Xa \in NT$, $X \in N$, $X \in pre^{+}.(PF.T^{*}W)$, etc.) So, let us assume all these properties. Then

$$Xa \in T^{*}WT^{*}$$

$$= \quad \{ \quad \text{definition of } T^{*} \quad \}$$

$$Xa \in T^{*}W \ \vee \ Xa \in T^{*}WT^{+}$$

$$= \quad \{ \quad T^{+} = T^{*}T \text{, cancellation property of concatenation of words} \quad \}$$

$$Xa \in T^{*}W \ \vee \ X \in T^{*}WT^{*}$$

$$= \quad \{ \quad (36) \quad \}$$

$$Xa \in T^{*}W \ \vee \ X \in \neg(pre^{+}.(PF.T^{*}W))$$

$$= \quad \{ \quad \text{assumption} \quad \}$$

$$Xa \in T^{*}W$$

$$= \quad \{ \quad \text{definition of } T^{*}W \quad \}$$

$$\langle \exists A,C : A \in W : Xa = CA \rangle \quad .$$

We continue the calculation with just the term $Xa = CA$ and the additional assumption $A \in W$.

$$Xa = CA$$

$$= \quad \{ \quad A \in W \text{ so, by assumption, } 1 \leq \#A \quad \}$$

$$Xa = CA \ \wedge \ C \in pre^{*}.\{X\}$$

$$= \quad \{ \quad \text{assumption: specifically, } C \in N \quad \}$$

$$Xa = CA \ \wedge \ CA \in N\{A\}$$

$$\Rightarrow \quad \{ \quad (57) \quad \}$$

$$\langle \exists B : B \in W : Xa \in S_{B}(B \between A) \rangle$$

$$= \quad \{ \quad \text{by assumption (} Xa \in N \text{) and (65), } Xa \in \neg S \quad \}$$

$$\langle \exists B,D,E \, : \, B \in W \wedge D \in S_{B} \wedge E \in B \between A \wedge E \neq \varepsilon \, : \, Xa = DE \rangle$$

$$\Rightarrow \quad \{ \quad \text{weakening, definition of } pre \quad \}$$

$\langle \exists B,D \; : \; B\in W \wedge D\in S_B \; : \; D \in pre^*.\{X\}\rangle$

$\Rightarrow \quad \{ \quad$ assumption: specifically, $\langle \forall C \; : \; C \in pre^*.\{X\} \; : \; C\in N\rangle \quad \}$

$\langle \exists B,D \; : \; B\in W \wedge D\in S_B \; : \; D\in N\rangle$

$\Rightarrow \quad \{ \quad S_B \subseteq S$, (65) $\quad \}$

false .

This concludes the proof of (69). Now we must prove (68). Again we appeal to lemma 61: the lemma establishes that $Xa\in S \Leftarrow Xa \in PF.T^*W$ and we have to prove the opposite implication. We have,

$Xa \in S$

$\Rightarrow \quad \{ \quad$ set calculus $\quad \}$

$Xa \in N\cup S$

$= \quad \{ \quad S$ and N satisfy (55) $\varepsilon \neq Xa \quad \}$

$Xa \in NT$

$= \quad \{ \quad$ cancellation property of concatenation of words $\quad \}$

$X \in N$

$= \quad \{ \quad$ induction hypothesis: (66) $\quad \}$

$X \in pre^+.(PF.T^*W)$

$= \quad \{ \quad (36) \quad \}$

$X \in \neg(T^*WT^*)$

$= \quad \{ \quad$ cancellation property of concatenation of words $\quad \}$

$Xa \in \neg(T^*WT^+)$

$= \quad \{ \quad (35) \quad \}$

$Xa \in pre^*.(PF.T^*W)$.

But also,

$Xa \in S$

$\Rightarrow \quad \{ \quad (65) \quad \}$

$\neg(Xa\in N)$

$= \quad \{ \quad$ (69), which was proved above $\quad \}$

$\neg(Xa \in pre^+.(PF.T^*W))$.

Combining these two calculations, we have;

$Xa \in S$

$\Rightarrow \quad \{ \quad$ two calculations above $\quad \}$

$Xa \in pre^*.(PF.T^*W) \wedge \neg(Xa \in pre^+.(PF.T^*W))$

$= \quad \{ \quad$ for all sets of words V , $pre^*.V \; = \; V \cup pre^+.V \quad \}$

$Xa \in PF.T^*W$.

This concludes the proof of (68) and the proof of the theorem. □

Theorem 64 does not guarantee that the system of equations in theorem 47 has a unique solution. Indeed, we have unable to prove that this is the case, and we must leave this as an open problem.

5. Generalisation

Suppose X is a play of the game S, where $S = PF.(T^*W)$. We saw in theorem 17 that $X\backslash S$ is a first-past-the-post game. This suggests that it should be straightforward to generalise theorem 38 to equations characterising $X\backslash S$, thus enabling the calculation of the probability of winning and the expected length of the remainder of the game. This we do below in theorem 71.

In section 5.2, we show how the equations are used to derive probabilistic generating functions. This enables us to correct errors in [GO81, theorem 2.1].

5.1. Equations

Lemma 70. Suppose $S = PF.(T^*W)$ and $N = pre^+.S$. Then

$$X\backslash S = \emptyset \quad \equiv \quad X \in T^*WT^+ \quad .$$

Proof The converse of the lemma is

$$X\backslash S \neq \emptyset \quad \equiv \quad \neg(X \in T^*WT^+) \quad .$$

But $\neg(X \in T^*WT^+)$ equivales $X \in N \cup S$ by lemma 36 (and $N \cup S = pre^*.(PF.T^*W)$). That is, we have to prove that

$$X\backslash S \neq \emptyset \quad \equiv \quad X \in N \cup S \quad .$$

This is straightforward:

$$
\begin{array}{cl}
 & X\backslash S \neq \emptyset \\
= & \{ \quad \text{lemma 1} \quad \} \\
 & \{X\}T^* \cap S \neq \emptyset \\
= & \{ \quad \text{definition of } pre^* \quad \} \\
 & X \in pre^*.S \\
= & \{ \quad \text{definition of } N \text{ and } S \quad \} \\
 & X \in N \cup S \quad .
\end{array}
$$

□

Theorem 71. Let X be a word and W be a non-empty set of words over the alphabet T. Suppose X is a play of the game (possibly not complete). Then

(72) $\quad \{\varepsilon\} \cup (X\backslash N)T \;=\; X\backslash N \,\cup\, X\backslash S \quad .$

Also, for all words A in W,

(73) $\quad (X\backslash N)\{A\} \,\cup\, X \between A \;=\; \langle \cup B : B \in W : (X\backslash S_B)(B \between A) \rangle \quad .$

Finally,

(74) $\quad X\backslash S \;=\; \langle \cup B : B \in W : X\backslash S_B \rangle \quad .$

Proof The property (72) was proved in theorem 17.

To prove (73), suppose $A \in W$. Then

S is a first-past-the-post game

$\Rightarrow$ { (43) }

$N\{A\} = \langle \cup B : B \in W : S_B(B \between A) \rangle$

$\Rightarrow$ { Leibniz }

$X \backslash (N\{A\}) = X \backslash \langle \cup B : B \in W : S_B(B \between A) \rangle$

$=$ { distributivity of $(X \backslash)$ over set union }

$X \backslash (N\{A\}) = \langle \cup B : B \in W : X \backslash (S_B(B \between A)) \rangle$.

We now simplify the subexpressions involving $(X \backslash)$. First we show that

(75) $X \backslash (N\{A\}) = (X \backslash N)\{A\} \cup X \between A$.

We have, for all words B,

$B \in X \backslash (N\{A\})$

$=$ { lemma 4, $B \in D \backslash \{A\} \equiv DB = A$ }

$B \in (X \backslash N)\{A\}$

$\vee \langle \exists C,D : X = CD \wedge DB = A \wedge 1 \leq \#D : C \in N \rangle$

Now,

$\langle \exists C,D : X = CD \wedge DB = A \wedge 1 \leq \#D : C \in N \rangle$

$=$ { trading and range disjunction, definition of N }

$\langle \exists C : \langle \exists D : X = CD \wedge 1 \leq \#D : DB = A \rangle : C \in pre^+.S \rangle$

$=$ { $\langle \exists D : X = CD : 1 \leq \#D \rangle$

$=$ { definition of pre^+ }

$C \in pre^+.\{X\}$

$\Rightarrow$ { $\{X\} \subseteq pre^*.S$, since X is a play of the game }

$C \in pre^+.(pre^*.S)$

$\Rightarrow$ { $pre^+ \circ pre^* = pre^+$ }

$C \in pre^+.S$,

Leibniz

}

$\langle \exists C : \langle \exists D : X = CD \wedge 1 \leq \#D : DB = A \rangle : \mathsf{true} \rangle$

$=$ { trading and range disjunction }

$\langle \exists C,D : X = CD \wedge DB = A \wedge 1 \leq \#D : \mathsf{true} \rangle$

$=$ { definition of $X \between A$: (37) }

$B \in X \between A$.

We have thus proved (75). Now we have to show that

$$X \backslash (S_B(B \between A)) = (X \backslash S_B)(B \between A) \ .$$

We have:

$$
\begin{array}{ll}
& X\backslash(S_B(B \between A)) \;=\; (X\backslash S_B)(B \between A) \\
\Leftarrow & \{ \quad \text{lemma 4} \quad \} \\
& \langle \cup C,D \;:\; C\in S_B \wedge X = CD \wedge 1 \leq \#D \;:\; D\backslash(B \between A)\rangle \;=\; \emptyset \\
\Leftarrow & \{ \quad S_B \subseteq S \subseteq T^*W \quad \} \\
& \neg(X \in T^*WT^+) \\
= & \{ \quad \text{lemma 70} \quad \} \\
& X\backslash S \neq \emptyset \\
= & \{ \quad \text{corollary 2, definition of “play”} \quad \} \\
& X \text{ is a play of the game } S\,.
\end{array}
$$

Equation (74) is immediate from (42) and the distributivity of $X\backslash$ over set union. □

Lemma 76. For all languages U and V and all words X, if UV is unambiguous then $(X\backslash U)V$ is unambiguous. Also, if $U{\cup}V$ is unambiguous then $(X\backslash U)\cup(X\backslash V)$ is unambiguous.

Proof

$$
\begin{array}{ll}
& (X\backslash U)V \text{ is unambiguous} \\
= & \{ \quad \text{definition} \quad \} \\
& \langle \forall A,B,C,D \;:\; A\in X\backslash U \wedge B\in X\backslash U \wedge C\in V \wedge D\in V \wedge AC = BD \;:\; A = B\rangle \\
= & \{ \quad \text{cancellation property of concatenation of words} \quad \} \\
& \langle \forall A,B,C,D \;:\; A\in X\backslash U \wedge B\in X\backslash U \wedge C\in V \wedge D\in V \wedge XAC = XBD \;:\; XA = XB\rangle \\
\Leftarrow & \{ \quad A,B := XA,XB\,,\; \{X\}X\backslash U \subseteq U \quad \} \\
& \langle \forall A,B,C,D : A\in U \wedge B\in U \wedge C\in V \wedge D\in V \wedge AC = BD : A = B\rangle \\
= & \{ \quad \text{definition} \quad \} \\
& UV \text{ is unambiguous.}
\end{array}
$$

That $(X\backslash U)\cup(X\backslash V)$ is unambiguous if $U{\cup}V$ is unambiguous is proved similarly. □

Corollary 77. Suppose p is a probability distribution on the elements of T and suppose the function h_p is defined on subsets of T^* as in definition 18. Suppose W is a reduced subset of T^*. Then, for all A in W and all X in $\neg(T^*WT^*)$ (i.e. all plays of the game),

$$h_p.(X\backslash N) + (X : A) \quad = \quad \langle \Sigma B : B\in W : h_p.(X\backslash S_B) \times (B : A)\rangle$$

and

$$1 \quad = \quad \langle \Sigma B \;:\; B\in W \;:\; h_p.(X\backslash S_B)\rangle \quad .$$

Proof As in the proof of theorem 47, the corollary is a combination of theorem 71, lemma 76 and lemmas 20 and lemma 45. □

The following property was observed by [Col82].

Example 78. Suppose $S = PF.T^*\{A\}$ and X is a play of S. Then the expected length of the game $X\backslash S$ is $(A : A) - (X : A)$.

Proof We instantiate corollary 77 with W equal to $\{A\}$:

$$
\begin{array}{ll}
 & e_p.(X\backslash S) \\
= & \qquad \{ \qquad \text{theorem 17 and theorem 24} \quad \} \\
 & h_p.(X\backslash N) \\
= & \qquad \{ \qquad \text{corollary 77} \quad \} \\
 & h_p.(X\backslash S_A)\times(A:A)\ -\ (X:A) \\
= & \qquad \{ \qquad \text{corollary 26, } S\!=\!S_A \quad \} \\
 & (A:A)-(X:A) \quad .
\end{array}
$$

□

5.2. Generating Functions

We now turn to the construction of the generating functions and the error in [GO81, theorem 2.1]. For any set of words V , define the generating function $F.V$ by

(79) $$F.V \;=\; \left\langle \Sigma A : A{\in}V : z^{-(\#A)} \right\rangle \quad .$$

(The coefficient of z^{-n} in $F.V$ is the number of words of length n in V .)

We note that, for all sets of words U and V such that $U{\cap}V=\emptyset$

$$F.(U{\cup}V) \;=\; F.U+F.V \quad .$$

Also, for arbitrary set of words U and V such that UV is unambiguous,

$$F.(UV) \;=\; F.U\times F.V \quad .$$

For brevity, let us write N_X for $F.(X\backslash N)$ and S_X for $F.(X\backslash S)$. Then, distributing the function F over both sides of (72), we obtain:

(80) $$1+N_X{\times}|T|{\times}z^{-1} \;=\; N_X{+}S_X \quad .$$

Also, abbreviating $F.(X\backslash S_B)$ to $S_{X,B}$, by applying F to equation (73) and applying the above distributivity properties, we get

(81) $$N_X{\times}z^{-(\#A)}+F.(X \between A) \;=\; \langle \Sigma B \,:\, B{\in}W \,:\, S_{X,B}\times F.(B \between A)\rangle \quad .$$

(Distributivity is allowed on the left side because the length of all words in $X \between A$ is less than the length of all words in $(X\backslash N)\{A\}$; on the righthand side it is allowed because of lemmas 76 and 45.)

Finally, by applying F to equation (74), we get

(82) $$S_X \;=\; \langle \Sigma B : B{\in}W : S_{X,B}\rangle \quad .$$

(Distribution of F over the set union is allowed so long as W is reduced.)

In order to allow direct comparison with [GO81, (2.2) on p.191], let us now relate Guibas and Odlyzko's notation to ours. They use the notation “ $f(n)$ ” for the “number of strings of length n over our alphabet which start with X and do not contain any of A , B , $\ldots$, T ”. (Here “ A , B , $\ldots$, T ” is the set of words W ; the symbol “ T ” should not be confused with our use of T for the alphabet.) The notation $F(z)$ is the corresponding generating function $\langle \Sigma n : 0{\le}n : f(n){\times}z^{-n}\rangle$. Since the set of strings underlying F is $\{X\}T^*\cap\neg(T^*WT^*)$, we have

$$F(z) \;=\; N_X{\times}z^{-(\#X)} \quad .$$

(In more detail, $\neg(T^*WT^*)$ equals N by lemma 35 and so $\{X\}T^*\cap\neg(T^*WT^*)$ equals $\{X\}(X\backslash N)$ by lemma 1. The term $z^{-(\#X)}$ is $F.\{X\}$ and N_X is $F.(X\backslash N)$.)

Guibas and Odlyzko also use the notation “ $f_H(n)$ ” to denote “the number of strings of length n which start with X , end with H , and do not contain any of A , B , ... , T except for that single occurrence of H at the end of the string”. The corresponding generating function is denoted by “ $F_H(z)$ ”. Thus the set of strings underlying F_H is $\{X\}(X\backslash S_H)$ and $F_H(z)$ equals $S_{X,H}{\times}z^{-(\#X)}$.

The final item of notation used by Guibas and Odlyzko is “ XY_z ”. The equivalence with our formulation is given by the equation:

$$XY_z \times z \;\; = \;\; z^{\#Y} \times F.(X \between Y) \quad .$$

The example used by Guibas and Odlyzko is when X is $hthtth$ and Y is $httht$. Then $X \between Y$ equals $\{t,ttht\}$ and $F.(X \between Y)$ equals $z^{-1}{+}z^{-4}$. Guibas and Odlyzko define XY_z in this case to be $z^3{+}z^0$, which agrees with the above equation.

Using Guibas and Odlyzko’s notation and the above equalities, the equation (80) becomes

$$1 \; + \; F(z) \times z^{\#X} \times |T| \times z^{-1} \;\; = \;\; F(z) \times z^{\#X} \; + \; \left\langle \Sigma B : B{\in}W : F_A(z) \times z^{\#X} \right\rangle \quad .$$

This is equivalent to the first equation in the set of simultaneous equations [GO81, (2.2) on p.191].

On the other hand, the equation (81) becomes, for each A in W ,

$$F(z) \; - \; \langle \Sigma B : B{\in}W : F_B(z) \times z \times BA_z \rangle \;\; = \;\; -(z^{1\,-\,\#X} \times XA_z) \quad .$$

This equation differs from the second set of simultaneous equations [GO81, (2.2) on p.191] in the righthand side: both the sign of the term and the exponent of z are incorrect. (The latter is undoubtedly a typographical error: their exponent is “ $1{-}|H|$ ” and in this context “ H ” has no meaning; the omission of the minus sign may also be a typographical error but, so far as we are aware, the error has never been corrected in any subsequent publication.)

A very simple example demonstrates the error. Suppose $T = \{a\} = W$ and $X = A = a$. Note that $a \between a = \{\varepsilon\}$. So, instantiating (80), (81) and (82), we get

$$1 + N_a{\times}z^{-1} \;\; = \;\; N_a + S_a \quad ,$$

$$N_a{\times}z^{-1} + 1 \;\; = \;\; S_{a,a} \quad \text{and}$$

$$S_a \;\; = \;\; S_{a,a} \quad .$$

These equations are easily solved to get $N_a = 0$ and $S_a = S_{a,a} = 1$. This is as expected since $a\backslash\{a\} = \{\varepsilon\}$ and $PF.(\{a\}^*\{a\}) = \{a\}$. On the other hand, the equations predicted by [GO81, Theorem 2.1] are

$$1 + F(z) \;\; = \;\; F(z) \times z \; + \; F_a(z) \times z \quad ,$$

$$F(z) \; - \; F_a(z) \times z \;\; = \;\; 1 \quad .$$

These equations have solution $F(z) = \frac{2}{z}$ and $F_a(z) = \frac{2-z}{z^2}$, which is clearly incorrect.

When the equations are specialised to the case that $X = \varepsilon$, the error disappears (the term $F.(X \between A)$ in (81) simplifies to 0). Guibas and Odlyzko never use their theorem 2.1 except in this case so the error is isolated; subsequent publications that refer to their paper also exploit only the special case, where there is no error. But Guibas and Odlyzko do not give an independent proof of the special case: they only provide a proof of theorem 2.1.

6. Implementation

This section is about our practical experience with calculating probabilities and expected values when playing the Penney-Ante game, in particular when the number of players and/or the set W is finite but large. We compare two methods. The first is to calculate $A \between B$ for each pair of words A and B in W and then solve the system of equations given by theorem 47 (or, more generally, corollary 77 when a play X is also given) together with equation (42). The second is to construct a deterministic finite-state machine that recognises the language $PF.T^*W$ and then exploit the fact that such a machine is an unambiguous expression of the language in order to construct and then solve a similar system of equations in the probabilities $h_p.S_A$. (Equations in the expected lengths $e_p.S_A$, for each A in W , can also be computed from a deterministic finite-state machine as we illustrated in example 22. In general, the equations can be formulated in the same way that we formulated such equations from theorem 47 in section 4.3.)

With one exception, all publications that we are aware of that discuss the calculation of the probabilities and expected values do so by first constructing a finite-state machine. The exception is the paper by Noonan and Zeilberger [NZ99] which describes a Maple package to solve for the generating functions; these are then instantiated in order to construct probabilities and expected values. Publications that exploit the finite-state machine typically do not give details of how it is constructed; in almost all cases (for example, [Col82, Nic07]), it appears to be done in an entirely ad hoc fashion. In one publication [CWT05], the standard textbook algorithm to construct a finite-state machine from a given (arbitrary) regular expression is used.

A much simpler, and more effective, way to construct the finite-state machines is to exploit Aho and Corasick's [AC75] generalisation of the well-known Knuth-Morris-Pratt algorithm [KMP74]. Aho and Corasick show how to construct a finite-state machine that recognises the language $T^*W\,T^*$ for a given alphabet T and finite set of words W . The construction has three stages: construct a tree recognising W , construct the Knuth-Morris-Pratt "failure function" for the tree and finally convert this to a deterministic finite-state machine. Fig. 2 illustrates this construction for the alphabet $\{a,b\}$ and set of words $\{aaa,abb,baa\}$.

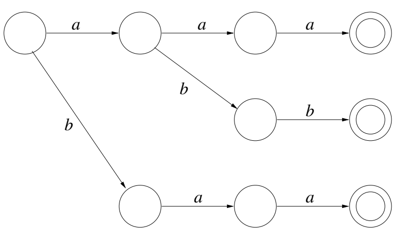
The construction shown in fig. 2 differs from the one described by Aho and Corasick [AC75] in that the failure function has not been constructed for the final nodes (fig. 2(b)) and, in the deterministic finite-state machine (fig. 2(c)), there are transitions from each final state to itself for each symbol of the alphabet (rather than the transitions that would be constructed by their algorithm). Although Aho and Corasick do state in the Concluding Remarks of their paper that their algorithm constructs recognisers of languages of the form $T^*W\,T^*$, their remarks are incorrect without this modification to their algorithm — which they do not make explicit.

The construction is linear in the sum of the lengths of the words in W. As is well-known, it is a trivial task to convert a deterministic finite-state machine recognising a language L to one recognising $\neg L$: just switch final to non-final states and vice-versa. So Aho and Corasick's algorithm describes directly how to construct a deterministic finite-state machine that recognises the set $\neg(T^*W\,T^*)$, the set of incomplete plays of the game defined by W . A simple modification of the algorithm —specifically, eliminate transitions from final states of the recogniser for $T^*W\,T^*$ — also yields a recogniser of $PF.T^*W$. Note that the recogniser so obtained is not reduced (in the standard textbook sense of "reduced"); importantly, it has the property that there is a one-to-one correspondence between the words in W and the final states.

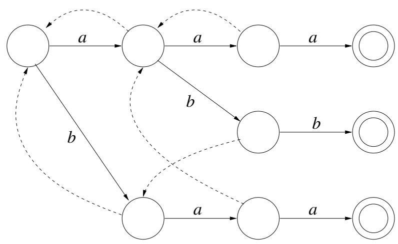
Having constructed the deterministic finite-state machine for $PF.T^*W$, a standard textbook method is used to construct a (linear) system of equations in the languages L_q , for each state q , consisting of the set of words recognised by q . Using lemma 20 and the fact that the righthand side of each equation is unambiguous, a (linear) system of equations is constructed in the unknowns $h_p.L_q$. Solving these equations, using standard methods, enables one to determine $h_p.S_A$ for each word A in W : just exploit the one-to-one correspondence between the words in W and the final states.

In passing, it may be worth noting that the construction of the failure function is equivalent to the construction of a so-called "factor graph" [BL77] of the language $PF.T^*W$; a "factor graph" [Bac75] is the transitive reduction of the "factor matrix", a beautiful but little-known concept introduced by Conway [Con71] in his book on regular algebra. Given the relative coincidence of the publication of Penney's paper and the publication of Conway's book, it may be that Conway's understanding of factor theory helped him to construct the formula for calculating the probability of winning in a 2-person Penney-Ante game (see theorem 50 and corollary 51).

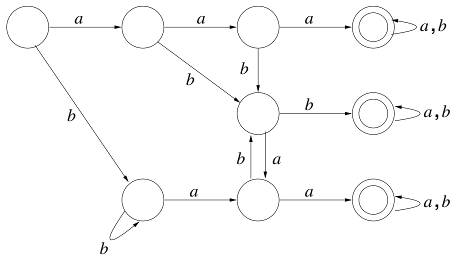
The solution of the equations given by theorem 47 and Aho and Corasick's construction of a deterministic finite-state recogniser of $PF.T^*W$ give two methods of calculating the probability of winning for each choice of a word in W as well as the expected length of a game. More generally, corollary 77 enables the calculations to be made at an arbitrary intermediate point in the game. Similarly, the finite-state machine can also be



(a) Tree of Words



(b) Failure Function



(c) Deterministic Finite-State Machine

Figure 2: Construction of a Recogniser of $\{a,b\}^*\{aaa,abb,baa\}\{a,b\}^*$.

used at an arbitrary intermediate point in the game: for given play X , calculate the probabilities of reaching each of the final states from the current state (the state reached from the start state given input X).

Both methods have been implemented by Ngoc Do [ND12]. The relative independence of the two methods gives a useful practical way of testing the theory: the calculations should always give the same answer! Ngoc Do confirmed that this is indeed the case for randomly generated test data. The maximum number of words and maximum word length in her testing were 25 and 20 , respectively.

The two methods have different space and time complexities. When using the deterministic finite-state machine, the dominant factor is the number of terms in the equations that must be solved, which equals the number of transitions in the finite-state machine. When using the equations given by corollary 77, the number of equations is approximately the number of words in the set W , and the number of terms in the equations is typically much smaller. (The number of terms in the two sets of equations is only equal when $W=\{\varepsilon\}$.) The dominant factor is, however, the calculation of $A ⌶ B$ for pairs of words A and B . We suspect that Aho and Corasick's algorithm can also be used to improve the efficiency of this calculation for large words and/or large sets of words but have not investigated it in detail.

Ngoc Do [ND12] has experimented with sets of up to 300 words each of length up to 100 . For such very large input values, she was unable to compute the probabilities of winning but she was able to compute the size of the equations that must be solved. For small input values, Ngoc Do's results show little overall difference between the two methods but for larger input values, the size of the deterministic finite-state machine soon becomes extremely large.

Of course, these experiments do not reflect a practical application. In the case of such an application, we recommend repeating the experiments to determine which method is better.

7. Discussion and Conclusion

The Penney-Ante game has attracted a lot of interest as an example of so-called "paradoxes" in probability theory and also as a (textbook) example of the use of generating functions. This paper has a very different focus to earlier publications: our focus is on formalising and reasoning about the event space that underlies plays of the game rather than the probabilities or expected lengths. We have shown how our equational characterisation of plays of the game underlies the equational characterisation of appropriate generating functions, probabilities of winning and expected lengths of games.

Probability theory is a well-established and well-understood area of mathematics but it is also notorious for the misinterpretations and mistakes that are frequently made. Because of the theory's central importance in many applications, great care is needed to properly formalise the event space under consideration. An informal description of the event space is easily misunderstood and, in many cases, several different event spaces can be confused.

In the analysis of Penney-Ante games, there are two event spaces: a single throw of a die and a sequence of throws of a die comprising a complete play of the game. However, several of the publications on the game seem to have other event spaces in mind without being explicit about what they are. For example, Guibas and Odlyzko [GO81, p.195] state:

> The probability that none of A , ..., T occurs in the first n throws and that the following $|H|$ throws will produce $H = h_1, \ldots, h_{|H|}$ is just $s(n)Pr(H)$ ... .

In this statement, three different event spaces can be identified: the event space consisting of words in which none of A , ..., T occurs, the event space consisting of words of length $|H|$, and the concatenation of the latter two event spaces. The last of these is *not* the event space consisting of complete plays of the game since a word in A , ..., T may be recognised when a proper prefix of H has been thrown. Formally, using our notation, their arguments appear to ascribe meaning to $h_p.S_B \times h_p.(B \between A)$ as a probability, whereas, for $B \neq A$, the frequency that a word in $S_B(B \between A)$ ever occurs is 0 — the game would be terminated before such an event occurs. If it can indeed be interpreted as a probability, the event space should be made clear. Similarly, in [GKP89] equations are formulated for $h_p.N$ and $h_p.S_A$ (for each A) —albeit using a different notation— and it is claimed that $h_p.S_A$ is the probability that the event A occurs. However, this claim does not appear to be properly justified, as evidenced by the fact that no claim is made about the meaning of $h_p.N$. Here we have made clear that h_p is a probability distribution on the event space $PF.T^*W$. In the derivation of theorem 50, the function h_p is also applied to languages not in this event space, in which case it is typically not a probability distribution: $h_p.N$, for example, is the expected length of a complete play of the game.

The confusion of different event spaces also explains some of the so-called "paradoxes" of Penney-Ante games. For example, for given words A and B over alphabet T , the event spaces $PF.T^*\{A\}$, $PF.T^*\{B\}$ and $PF.T^*\{A,B\}$ are different and not related by a simple distributivity property of the function PF . So it is not surprising that the expected length of an event in $PF.T^*\{A\}$ is unrelated to the expected length of an event in $PF.T^*\{A,B\} \cap T^*\{A\}$. The "paradoxical" situation that "patterns with no self-overlaps occur sooner than overlapping patterns do" is no longer a paradox if the event spaces are made explicit — see example 49.

We make no use whatsoever of generating functions. Generating functions enable one to derive numerical properties related to word length; our derivations show that word length is irrelevant to deriving Conway's formula (and also Solov'ev's formula). Even in the case of calculating the expected length of a complete play of a game, where word length is part of the definition, theorem 24 is all that is needed. Generating functions offer a very powerful tool for reasoning about a sequence of numbers. Their role in our analysis has been taken by the event space itself: the set of complete plays of the game. The algebra of languages is not as rich as the algebra of generating functions (for example, multiplication of generating functions is commutative but concatenation of languages is not) but greater understanding can be achieved by exploiting the algebra to explore the relations between different event spaces. On the other hand, we have been unable to calculate a formula for the standard deviation of the length of complete games (or for other higher-order cumulants). The conclusion would appear to be that the versatility of generating functions is best demonstrated by their use in determining higher-order cumulants.

Some challenges remain:

We have established in theorem 64 that the set of complete plays of a Penney-Ante game can be uniquely characterised by a system of (non-linear) equations combined with an unambiguity requirement but have been unable to translate this theorem to a proof that the equations in probabilities (theorem 47) also have a unique solution. We suspect that the problem can be overcome by a deeper analysis of the relation between path-finding in a graph and solving linear equations (in real arithmetic). Essentially, the problem is to show that the matrix defined by the colon operator —see (46)— is non-singular. It should be possible to do so by showing that the matrix defined by the match operator —see (37)— is also "non-singular" using techniques similar to those used by Tarjan [Tar81, lemma 2].

We have observed how Aho and Corasick's algorithm can be used to calculate the probability of winning (and also expected values, although not in detail) but we have not shown how it might also be used to compute the matrix of match values. We strongly suspect that this is possible but further investigation is required.

Finally, we have only just begun an exploration of first-past-the-post games since much of the paper is about one particular subclass of such games. It would be interesting to extend the exploration to other subclasses, for example to the class of "hidden" patterns mentioned in example 27.

Acknowledgement. I am very grateful to Loan Ngoc Do for her comprehensive literature search on the Penney-Ante game and for her implementation work discussed in section 6. Many thanks also to the anonymous referees (both of the current paper and the earlier conference paper) and the editors, Jeremy Gibbons and Pablo Nogueira, for their very thorough and detailed comments.

References

[AC75] Alfred V. Aho and Margaret J. Corasick. Efficient string matching: An aid to bibiliographic search. *Communications of the ACM*, 18(6):333–340, 1975.

[Bac75] R.C. Backhouse. *Closure algorithms and the star-height problem of regular languages.* PhD thesis, University of London, 1975.

[Bac86] R.C. Backhouse. *Program Construction and Verification.* Prentice-Hall International, 1986.

[BL77] R.C. Backhouse and R.K. Lutz. Factor graphs, failure functions and bi-trees. In A. Salomaa and M. Steinby, editors, *Fourth Colloquium on Automata, Languages and Programming*, pages 61–75. Springer-Verlag, LNCS 52, July 1977.

[Brz64] J.A. Brzozowski. Derivatives of regular expressions. *Journal of the ACM*, 11(4):481–494, October 1964.

[Col82] Stanley Collings. Coin sequence probabilities and paradoxes. *Bulletin of the Institute of Mathematics and its Applications*, 18:227–232, 1982.

[Con71] J.H. Conway. *Regular Algebra and Finite Machines.* Chapman and Hall, London, 1971.

[CWT05] Piotr Chrząstowski-Wachtel and Jerzy Tyszkiewicz. A Maple package for conditional event algebras. In G. Kern-Isberner, W.Rödder, and F.Kulmann, editors, *WCII 2002*, volume LNAI 3301, pages 131–151. Springer-Verlag, 2005.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics.* Cambridge University Press, 2009.

[Gar74] Martin Gardner. On the paradoxical situations that arise from nontransitive relations. *Scientific American*, 231(4):120–124, October 1974. Reprinted with additions in his book *Time Travel and Other Mathematical Bewilderments*, 1988, 55–69.

[GKP89] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics : a Foundation for Computer Science.* Addison-Wesley Publishing Company, 1989.

[GO81] L.J. Guibas and A.M. Odlyzko. String overlaps, pattern matching and nontransitive games. *Journal of Combinatorial Theory, Series A30*, pages 183–208, 1981.

[GS93] David Gries and Fred B. Schneider. *A Logical Approach to Discrete Math.* Springer-Verlag, 1993.

[KMP74] D.E. Knuth, J.H. Morris, and V.R. Pratt. Fast pattern matching in strings. *SIAM Journal of Computing*, 6:325–350, 1974.

[ND12] Nhu Loan Ngoc Do. Evaluation of methods for first-past-the-post games. Master's thesis, School of Computer Science, University of Nottingham, 2012.

[Nic07] R. S. Nickerson. Penney Ante: Counterintuitive probabilities in coin tossing. *Undergraduate Mathematics and its Applications (UMAP)*, 28:503–532, 2007.

[NZ99] John Noonan and Doron Zeilberger. The Gouldon-Jackson cluster method: Extensions, applications and implementations. *Journal of Difference Equations and Their Applications*, 5:355–377, 1999.

[Pen74] Walter Penney. Problem 95: Penney-Ante. *Journal of Recreational Mathematics*, 7:321, 1974.

[Sol66] A.D. Solov'ev. A combinatorial identity and its application to the problem concerning the first occurrence of a rare event. *Theory of Probability and its Applications*, 11:276–282, 1966.

[Tar81] Robert Endre Tarjan. A unified approach to path problems. *Journal of the ACM*, pages 577–593, 1981.

[Win13] Peter Winkler. Puzzled tumbling dice. *Communications of the ACM*, 56(2):112, 2013.