A non-monotone Fraenkel-Lévy labelling for the asymmetric combinatorial game on cyclic graphs

Benedikt Löwe (bloewe@science.uva.nl) *
Institute for Logic, Language and Computation, Universiteit van Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands

April 29, 2004

Abstract. Extending results of Fraenkel, we give an algorithm that determines the value of asymmetric combinatorial games on (possibly cyclic) graphs.

2000 Mathematical Subject Classification: *Primary.* **91A46 91A43**. *Secondary.* 91A05 05C38

1. Introduction

If $\mathbf{G}$ is a graph, the (symmetric) combinatorial game on $\mathbf{G}$ is played by two players pushing a token on the graph. Whoever moves the token into a terminal node, wins. An example of a game of this type is the game of *Nim* (removing matches from a number of rows of matches until the game board is empty). If $\mathbf{G}$ was cyclic, then it is not guaranteed that one of the players will push the token into a terminal node; an infinite walk through $\mathbf{G}$ is considered a draw.[1]

Combinatorial games are perfect information games with simple payoff sets, and thus by the Gale-Stewart Theorem determined (Gale & Stewart, 1953). You can establish the winner of the game by unfolding the game into a game tree $\mathbf{T_G}$, then labelling the tree via the Gale-Stewart labelling and read off the winner from the label of the root.

However, analyzing combinatorial games via their game trees might not be optimal for several reasons:

Firstly, if $\mathbf{G}$ was cyclic, the game tree will be infinite and the labelling of the game tree will be an infinitary, possibly transfinite procedure.

Secondly, the Gale-Stewart procedure is not metamathematically parsimonious. There are computable trees with no computable winning strategy, and the Gale-Stewart theorem on determinacy of open games

* The author would like to thank Nick Bezhanishvili (Amsterdam) for fruitful discussions of topics connected to different aspects of this paper.

[1] More details can be found in the four-volume second edition of *Winning Ways* by Berlekamp, Conway and Guy.

is equivalent to a nontrivial metamathematical statement of second-order arithmetic by a theorem of Steel's.[2]

Thus, it is a *desideratum* to devise a labelling procedure directly on the graph that doesn't need unfolding into the game tree. This has been addressed by Fraenkel (Fraenkel, 1997): Following a suggestion of Azriel Lévy, Fraenkel describes a labelling on graphs that we shall call the Fraenkel-Lévy labelling, and gives an algorithm to compute this labelling with running time linear in the number of edges for finite connected graphs.

In this paper, we consider a variant of the combinatorial games where one player has to play into terminal nodes and the other has to keep the game alive for an infinite number of steps, and call it the asymmetric combinatorial game. Again, this game is a perfect information game with simple payoff, and thus could be analyzed via the Gale-Stewart technique with similar drawbacks.

It is the goal of this paper to give a finitary algorithm for asymmetric combinatorial games without the detour via the unravelled game tree. The algorithm we give is strongly influenced by the non-monotonic Gale-Stewart technique developed in (Löwe, 2003) to deal with many-player perfect information games with open and closed payoffs.

In Section 2, we define some basic graph-theoretical notions used in Section 3 where we define our games and discuss labellings and their connections to games abstractly, understanding labellings on graphs as quotients of the labellings on their associated game trees. Finally, in Section 4 we develop an algorithm for the asymmetric combinatorial games (*cf.* Figure 3) and discuss its running time.

2. Graphs

2.1. Graphs & Bisimulations

Our graphs $\mathbf{G} = \langle V_{\mathbf{G}}, E_{\mathbf{G}} \rangle$ are directed graphs (digraphs), *i.e.*, $V_{\mathbf{G}}$ is a set of vertices and $E \subseteq V_{\mathbf{G}} \times V_{\mathbf{G}}$ is an arbitrary binary relation. If $\equiv$ is an equivalence relation on $V_{\mathbf{G}}$, we can define a graph structure on the set of $\equiv$-equivalence classes $V_{\mathbf{G}/\equiv} := \{[v]_{\equiv}\,;\, v \in V_{\mathbf{G}}\}$ as follows:

$$\langle [v]_{\equiv}, [w]_{\equiv} \rangle \in E_{\mathbf{G}/\equiv} :\Longleftrightarrow \langle v, w \rangle \in E_{\mathbf{G}}.$$

[2] The statement is ATR_0, a set-theoretic existence statement for sets defined by transfinite recursion along an arithmetically defined wellorder. *Cf.* (Steel, 1976) and (Tanaka, 1990); for a detailed overview in the context of Reverse Mathematics, *cf.* (Simpson, 1999, § V.8).

We write $\mathbf{G}/{\equiv} := \langle V_{\mathbf{G}/\equiv}, E_{\mathbf{G}/\equiv}\rangle$ for the quotient graph.

If $s \in V_{\mathbf{G}}$, we call the pair $\langle \mathbf{G}, s\rangle$ a **pointed graph**. As usual, the natural numbers are identified with the sets of their predecessors, *i.e.*, $0 = \varnothing$ and $n+1 = \{0, \ldots, n\}$. If $N \in \mathbb{N} \cup \{\mathbb{N}\}$, we call a function $W : N \to V$ a **walk through $\langle \mathbf{G}, s\rangle$ of length N** if

1. for each $n+1 \in N$, we have $\langle W(n), W(n+1)\rangle \in E_{\mathbf{G}}$, and
2. $W(0) = s$.

A walk is called **finite** if $N \in \mathbb{N}$. It is called **maximal** if it is either infinite or finite of length $n+1$ where $W(n)$ is a terminal node of $\mathbf{G}$. We define the **connected component of v in $\mathbf{G}$** (in symbols: $\mathrm{ConCom}(\mathbf{G}, v)$) to be the set of vertices w such that there is a walk W of length $n+1$ through $\langle \mathbf{G}, v\rangle$ with $W(n) = w$. A pointed graph $\langle \mathbf{G}, v\rangle$ is called **connected** if $V_{\mathbf{G}} = \mathrm{ConCom}(\mathbf{G}, v)$.

If $\langle \mathbf{G}, s\rangle$ and $\langle \mathbf{H}, t\rangle$ are pointed graphs, then a function $Z : V_{\mathbf{G}} \to V_{\mathbf{H}}$ is called a **bounded epimorphism** if the following conditions hold:

1. $Z(s) = t$;
2. Z is surjective;
3. if $v_0 \in V_{\mathbf{G}}$ and $\langle v_0, v_1\rangle \in E_{\mathbf{G}}$, then $\langle Z(v_0), Z(v_1)\rangle \in E_{\mathbf{H}}$; and
4. if $w_0 \in V_{\mathbf{H}}$, $\langle w_0, w_1\rangle \in E_{\mathbf{H}}$, and $Z(v_0) = w_0$, then there is some $v_1 \in V_{\mathbf{G}}$ such that $Z(v_1) = w_1$ and $\langle v_0, v_1\rangle \in E_{\mathbf{G}}$.

If Z is a bounded epimorphism between $\mathbf{G}$ and $\mathbf{H}$, we can define an equivalence relation $\equiv_Z$ on $V_{\mathbf{G}}$ by

$$v \equiv_Z w :\Longleftrightarrow Z(v) = Z(w).$$

PROPOSITION 2.1. *Let $\langle \mathbf{G}, s\rangle$ and $\langle \mathbf{H}, t\rangle$ be pointed graphs and Z a bounded epimorphism between them. Let $\equiv_Z$ be the equivalence relation on $V_{\mathbf{G}}$ defined via Z. Then $\langle \mathbf{G}/{\equiv_Z}, [s]_{\equiv_Z}\rangle \cong \langle \mathbf{H}, t\rangle$.*

Proof. Define $\widehat{Z} : V_{\mathbf{G}}/{\equiv_Z} \to V_{\mathbf{H}}$ by

$$\widehat{Z}([v]_{\equiv_Z}) := Z(v).$$

This function is clearly well-defined and a bijection. Using the fact that Z is a bounded epimorphism, it is easy to see that $\widehat{Z}$ is structure preserving. q.e.d.

2.2. The unravelled tree and the alternating graph

Let $\langle \mathbf{G}, s\rangle$ be a pointed graph. Define $V_{\mathbf{T}^s_{\mathbf{G}}}$ to be the set of finite walks through $\langle \mathbf{G}, s\rangle$. For walks W_0 of length n and W_1 of length $n+1$, we let $\langle W_0, W_1\rangle \in E_{\mathbf{T}^s_{\mathbf{G}}}$ if and only if $W_0 = W_1{\upharpoonright}n$. Furthermore, let $\mathrm{root}_{\mathbf{G},s} := \{\langle 0, s\rangle\}$ be the unique walk of length 1. Then

$$\mathbf{T}^s_{\mathbf{G}} := \langle V_{\mathbf{T}^s_{\mathbf{G}}}, E_{\mathbf{T}^s_{\mathbf{G}}}\rangle.$$

We call $\langle \mathbf{T}^s_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},s}\rangle$ the **unravelled tree of** $\langle \mathbf{G}, s\rangle$.

In the following, we will use the parity function $\mathrm{par} : \mathbb{N} \to 2$ assigning to each natural number its parity. Let $V_{\mathbf{A}_{\mathbf{G}}} := 2 \times V_{\mathbf{G}}$,

$$\langle\langle e, v\rangle, \langle 1-e, w\rangle\rangle \in E_{\mathbf{A}_{\mathbf{G}}} :\Longleftrightarrow \ \langle v, w\rangle \in E_{\mathbf{G}},$$

and call $\mathbf{A}_{\mathbf{G}} := \langle V_{\mathbf{A}_{\mathbf{G}}}, E_{\mathbf{A}_{\mathbf{G}}}\rangle$ the **alternating graph of $\mathbf{G}$**. If $s \in V_{\mathbf{G}}$, then we let $\mathbf{A}^s_{\mathbf{G}}$ be the connected component of $\langle 0, s\rangle$ in $\mathbf{A}_{\mathbf{G}}$.

PROPOSITION 2.2. *If $\langle \mathbf{G}, s\rangle$ is a pointed graph, there are bounded epimorphisms from $\langle \mathbf{T}^s_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},s}\rangle$ to $\langle \mathbf{G}, s\rangle$, from $\langle \mathbf{A}^s_{\mathbf{G}}, \langle 0, s\rangle\rangle$ to $\langle \mathbf{G}, s\rangle$ and from $\langle \mathbf{T}^s_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},s}\rangle$ to $\langle \mathbf{A}^s_{\mathbf{G}}, \langle 0, s\rangle\rangle$.*

Proof. Let $e \in 2$, $v \in V_{\mathbf{G}}$, and $\mathrm{dom}(W) = n+1$ with $W(n) = v$. Then define

$$\begin{aligned} Z_{\mathbf{T}}(W) &:= v, \\ Z_{\mathbf{A}}(\langle e, v\rangle) &:= v, \text{ and} \\ Z_{\mathbf{T},\mathbf{A}}(W) &:= \langle \mathrm{par}(n), v\rangle. \end{aligned}$$

The functions $Z_{\mathbf{T}}$, $Z_{\mathbf{A}}$ and $Z_{\mathbf{T},\mathbf{A}}$ are bounded epimorphisms. q.e.d.

If $\langle \mathbf{G}, s\rangle$ is a pointed graph, $v \in V_{\mathbf{G}}$ and W is a walk through $\langle \mathbf{G}, s\rangle$ of length $n+1$ such that $W(n) = v$, then the connected component of W in $\mathbf{T}^s_{\mathbf{G}}$ and the graph $\mathbf{T}^v_{\mathbf{G}}$ are isomorphic as graphs. As a consequence, we get a slightly more general version of Proposition 2.2:

PROPOSITION 2.3. *Let $\langle \mathbf{G}, s\rangle$ be a pointed graph, and W be a walk through $\langle \mathbf{G}, s\rangle$ of length $n+1$ such that $W(n) = v$. Then there are bounded epimorphisms from $\langle \mathbf{T}^s_{\mathbf{G}}, W\rangle$ to $\langle \mathbf{G}, v\rangle$ and from $\langle \mathbf{T}^s_{\mathbf{G}}, W\rangle$ to $\langle \mathbf{A}^v_{\mathbf{G}}, \langle 0, v\rangle\rangle$.*

Proof. Compose the bounded epimorphisms $Z_{\mathbf{T}}$ between $\mathbf{T}^v_{\mathbf{G}}$ and $\mathbf{G}$ and $Z_{\mathbf{T},\mathbf{A}}$ between $\mathbf{T}^v_{\mathbf{G}}$ and $\mathbf{A}^v_{\mathbf{G}}$ with the mentioned graph isomorphism. q.e.d.

3. Games

3.1. Games and game equivalences

Given a graph $\mathbf{G} = \langle V_{\mathbf{G}}, E_{\mathbf{G}} \rangle$ and $s \in V_{\mathbf{G}}$, we define the **(symmetric) combinatorial game on** $\langle \mathbf{G}, s \rangle$ (in symbols: $\mathsf{S}(\mathbf{G}, s)$): at the beginning of the game, a token is positioned in the vertex s; players 0 and 1 move in turn with player 0 starting by pushing the token along the edges of $\mathbf{G}$; the player making the last move wins the game. If the game goes on for infinitely many steps, the outcome of the game is a draw. We define the **inverted symmetric combinatorial game** $\overline{\mathsf{S}}(\mathbf{G}, s)$ to be the game played like the symmetric combinatorial game, just with the rôles of the two players interchanged, *i.e.*, player 1 starts.

In the asymmetric version, the rôles of player 1 and the draw are interchanged: Given a graph $\mathbf{G} = \langle V_{\mathbf{G}}, E_{\mathbf{G}} \rangle$ and $s \in V_{\mathbf{G}}$, we define the **asymmetric combinatorial game on** $\langle \mathbf{G}, s \rangle$ (in symbols: $\mathsf{A}(\mathbf{G}, s)$): at the beginning of the game, a token is positioned in the vertex s; players 0 and 1 move in turn with player 0 starting by pushing the token along the edges of $\mathbf{G}$; if player 0 pushes the token into a terminal node, he wins; if player 1 pushes the token into a terminal node, the game is a draw. If the game goes on for infinitely many steps, player 1 wins. Again, we define the **inverted asymmetric game** $\overline{\mathsf{A}}(\mathbf{G}, s)$ to be the game with the players interchanged.

Strategies in these combinatorial games are simply functions that tell the players which edge $\langle v_0, v_1 \rangle$ to use if they are presented with the token in vertex v_0. A strategy is **winning** if the player following the strategy wins the game regardless of how the other player plays, and a strategy is called **nonlosing** if the game in which one player follows the strategy results in either a win for that player or a draw.

By the determinacy theorem of Gale and Stewart (for details, *cf.* Section 3.2) each of the games G defined above will have one of the following three **values**, denoted by $\mathrm{val}(\mathsf{G})$:

W. Player 0 has a winning strategy,

D. both players have a nonlosing strategy,

L. Player 1 has a winning strategy.

On the set $L = \{\mathsf{L}, \mathsf{D}, \mathsf{W}\}$ of these values, we define a lattice structure by $\mathsf{L} \leq \mathsf{D} \leq \mathsf{W}$ and an inversion function $\mathrm{inv} : L \to L$ defined by

$$\begin{aligned} \mathsf{W} &\mapsto \mathsf{L} \\ \mathsf{D} &\mapsto \mathsf{D} \\ \mathsf{L} &\mapsto \mathsf{W}. \end{aligned}$$

We say that two games G and H are **equivalent** if they have the same value. We say that they are **anti-equivalent** if $\mathrm{val}(\mathsf{G}) = \mathrm{inv}(\mathrm{val}(\mathsf{H}))$.

If G and H are either S, $\overline{\mathsf{S}}$, A or $\overline{\mathsf{A}}$, we can define the notion of a G-H-(anti)-equivalence: Let $\mathbf{G}$ and $\mathbf{H}$ be graphs and let $f : V_{\mathbf{G}} \to V_{\mathbf{H}}$ be a function. Then f is called a **G-H-(anti)-equivalence** if for all $v \in V_{\mathbf{G}}$, we have that $\mathsf{G}(\mathbf{G}, v)$ and $\mathsf{H}(\mathbf{H}, f(v))$ are (anti)-equivalent.

There are some obvious facts about equivalence of combinatorial games:

PROPOSITION 3.1. *For every pointed graph $\langle \mathbf{G}, v\rangle$, the games $\mathsf{S}(\mathbf{G}, v)$ and $\overline{\mathsf{S}}(\mathbf{G}, v)$ are anti-equivalent. In other words,* $\mathrm{id} : V_{\mathbf{G}} \to V_{\mathbf{G}}$ *is an* S-$\overline{\mathsf{S}}$-*anti-equivalence.*

Proof. Obvious. q.e.d.

PROPOSITION 3.2. *Let $\mathbf{G}$ and $\mathbf{H}$ be graphs, let G be either S, $\overline{\mathsf{S}}$, A or $\overline{\mathsf{A}}$, and let $F : V_{\mathbf{G}} \to V_{\mathbf{H}}$ be a function. If F is a bounded epimorphism, then it is a G-G-equivalence.*

Proof. Obvious. q.e.d.

An immediate consequence of Propositions 2.2, 2.3 and 3.2 is that in order to analyze arbitrary combinatorial games, it is enough to analyze games on trees:

COROLLARY 3.3. *Let G be either S, $\overline{\mathsf{S}}$, A or $\overline{\mathsf{A}}$ and let $\langle \mathbf{G}, s\rangle$ be a pointed graph. Then the games $\mathsf{G}(\mathbf{G}, s)$ and $\mathsf{G}(\mathbf{T}^s_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},s})$ are equivalent. Also, for any walk W through $\langle \mathbf{G}, s\rangle$ with length $n + 1$ and $W(n) = v$, the games $\mathsf{G}(\mathbf{G}, v)$ and $\mathsf{G}(\mathbf{T}^s_{\mathbf{G}}, W)$ are equivalent.*

3.2. A translation into the usual Gale-Stewart theory of infinite games

Corollary 3.3 is the underlying methodology of Gale-Stewart theory (Gale & Stewart, 1953). Instead of looking at the (possibly cyclic) graph, we look at the unravelled tree and analyze the game on the tree via backwards induction with a (possibly transfinite) labelling construction.

We shall translate our tree games into the usual topological notation of Gale-Stewart theory: Look at the space ${V_{\mathbf{G}}}^{\mathbb{N}}$ of functions from $\mathbb{N}$ into $V_{\mathbf{G}}$, endowed with the product topology of the discrete topology on $V_{\mathbf{G}}$.

We define three infinite games $\mathsf{G}_0(\mathbf{G}, v)$, $\mathsf{G}_1(\mathbf{G}, v)$, and $\mathsf{H}_1(\mathbf{G}, v)$. In all of the games, players 0 and 1 play elements of $V_{\mathbf{G}}$ in turn and produce an element of ${V_{\mathbf{G}}}^{\mathbb{N}}$, let's call it X. We assume that $X(0) = v$ and that player 0 plays the odd digits and player 1 plays the even digits. The payoff sets of the games are defined as follows:

– In $\mathsf{G}_0(\mathbf{G}, v)$, player 0 wins if the least $n+1$ such that $X{\restriction}n+1$ is not a walk through $\langle \mathbf{G}, v\rangle$ exists and is odd. Otherwise, player 1 wins.

– In $\mathsf{G}_1(\mathbf{G}, v)$, player 0 wins if either X is an infinite walk through $\langle \mathbf{G}, v\rangle$, or the least $n+1$ such that $X{\restriction}n+1$ is not a walk through $\langle \mathbf{G}, v\rangle$ is odd. Otherwise, player 1 wins.

– In $\mathsf{H}_1(\mathbf{G}, v)$, player 0 wins if there is a least $n+1$ such that $X{\restriction}n+1$ is not a walk through $\langle \mathbf{G}, v\rangle$ and either $n+1$ is odd or $X(n)$ is a terminal node of $\mathbf{G}$. Otherwise, player 1 wins.

The payoff sets for player 0 in the defined three games are either open (G_0 and H_1) or closed (G_1) in the topology defined on ${V_{\mathbf{G}}}^{\mathbb{N}}$, and by the usual Gale-Stewart theorem for open and closed games without draw, one of the two players has a winning strategy, *i.e.*, the values are either W or L.

It is easy to see that these infinite Gale-Stewart games correspond to the combinatorial games as follows:

PROPOSITION 3.4. *For every pointed graph $\langle \mathbf{G}, v\rangle$, the following equivalences hold:*

$$
\begin{array}{rcl}
\mathrm{val}(\mathsf{G}_0(\mathbf{G}, v)) = \mathsf{W} & \iff & \mathrm{val}(\mathsf{S}(\mathbf{T}^v_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},v})) = \mathsf{W} \\
& \iff & \mathrm{val}(\mathsf{A}(\mathbf{T}^v_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},v})) = \mathsf{W} \\
\mathrm{val}(\mathsf{G}_0(\mathbf{G}, v)) = \mathsf{L} & \iff & \textit{player 1 has a nonlosing strategy for} \\
& & \mathsf{S}(\mathbf{T}^v_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},v}) \\
& \iff & \textit{player 1 has a nonlosing strategy for} \\
& & \mathsf{A}(\mathbf{T}^v_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},v}) \\
\mathrm{val}(\mathsf{G}_1(\mathbf{G}, v)) = \mathsf{W} & \iff & \textit{player 0 has a nonlosing strategy for} \\
& & \mathsf{S}(\mathbf{T}^v_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},v}) \\
\mathrm{val}(\mathsf{G}_1(\mathbf{G}, v)) = \mathsf{L} & \iff & \mathrm{val}(\mathsf{S}(\mathbf{T}^v_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},v})) = \mathsf{L} \\
\mathrm{val}(\mathsf{H}_1(\mathbf{G}, v)) = \mathsf{W} & \iff & \textit{player 0 has a nonlosing strategy for} \\
& & \mathsf{A}(\mathbf{T}^v_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},v}) \\
\mathrm{val}(\mathsf{H}_1(\mathbf{G}, v)) = \mathsf{L} & \iff & \mathrm{val}(\mathsf{A}(\mathbf{T}^v_{\mathbf{G}}, \mathrm{root}_{\mathbf{G},v})) = \mathsf{L}
\end{array}
$$

As a consequence, we get a proof of the claim (see above) that the values W, L and D are the only possible values for our games.

3.3. Sound labellings

Let $L := \{\mathsf{L}, \mathsf{D}, \mathsf{W}\}$, $\mathbf{G}$ be a graph and G be one of S, $\overline{\mathsf{S}}$, A and $\overline{\mathsf{A}}$. An L-labelling $\ell : V_{\mathbf{G}} \to L$ is called **G-sound** if it is a total function and if for each vertex $v \in V_{\mathbf{G}}$, we have

$$\ell(v) = \mathrm{val}(\mathsf{G}(\mathbf{G}, v)).$$

Because of Proposition 3.1, the notions of S-soundness and $\overline{\mathsf{S}}$-soundness are closely connected:

COROLLARY 3.5. *Let* $\mathbf{G}$ *be a graph and* $\ell : V_{\mathbf{G}} \to L$ *be* S*-sound. Then* $\overline{\ell}$ *defined by* $\overline{\ell}(v) := \mathrm{inv}(\ell(v))$ *is* $\overline{\mathsf{S}}$*-sound.*

The Gale-Stewart analysis for games on trees gives a (possibly transfinite) recursive procedure to actually compute an S-sound labelling:

THEOREM 3.6 (Gale & Stewart; 1953). *If* $\mathbf{T}$ *is a tree, then there is recursive procedure that computes (in less than* $\mathrm{Card}(V_{\mathbf{T}})^{+}$ *steps)*[3] *the* S*-sound labelling* ℓ*.*

PROPOSITION 3.7. *Let* $\langle \mathbf{G}, s\rangle$ *be a connected pointed graph and let* ℓ *be the* S*-sound labelling on* $\mathbf{T}^{s}_{\mathbf{G}}$*. For* $v \in V_{\mathbf{G}}$ *and* W *such that* $Z_{\mathbf{T}}(W) = v$*, we define*

$$\ell/{\equiv_{Z_{\mathbf{T}}}}(v) := \ell(W).$$

This quotient labelling is well-defined and is the S*-sound labelling for* $\mathbf{G}$*.*

Proof. Let us show that $\equiv_{Z_{\mathbf{T}}}$ respects $\ell_{\mathbf{T}}$:

Suppose $W \equiv_{Z_{\mathbf{T}}} W'$, *i.e.*, $Z_{\mathbf{T}}(W) = Z_{\mathbf{T}}(W') = v$ for some v. Corollary 3.3 tells us that $\mathsf{S}(\mathbf{T}^{s}_{\mathbf{G}}, W)$ and $\mathsf{S}(\mathbf{T}^{s}_{\mathbf{G}}, W')$ are both equivalent to $\mathsf{S}(\mathbf{G}, v)$, so in particular, $\ell(W) = \ell(W')$, and the quotient labelling is S-sound. q.e.d.

For asymmetric combinatorial games, we don't have the symmetry of Corollary 3.5:

PROPOSITION 3.8. *If* $\mathrm{val}(\mathsf{A}(\mathbf{G}, v)) = \mathsf{W}$*, then* $\mathrm{val}(\overline{\mathsf{A}}(\mathbf{G}, v)) \neq \mathsf{L}$*. All other combinations are possible.*

Proof. For player 0, a winning strategy is a strategy that forces the token into a terminal node in an odd number of moves. Such a strategy is a nonlosing strategy for player 1 in the inverted game.

In Figure 1, examples for all eight combinatorially possible situations are given. q.e.d.

Because of the asymmetry indicated by Proposition 3.8, we define the following notion: A (partial) function $\ell : V_{\mathbf{G}} \to L^2$ is called a **(partial)** L**-bilabelling on** $\mathbf{G}$. We write $\ell(v) = \langle \ell_0(v), \ell_1(v)\rangle$. An L-bilabelling $\ell : V_{\mathbf{G}} \to L^2$ is called **A-sound** if it's total and for each vertex $v \in V$, we have

$$\ell_0(v) = \mathrm{val}(\mathsf{A}(\mathbf{G}, v)) \text{ and } \ell_1(v) = \mathrm{val}(\overline{\mathsf{A}}(\mathbf{G}, v)).$$

[3] Note that there is a possibly confusing typo in (Fraenkel, 1997, Definition 1): the transfinite recursion is bounded by $\mathrm{Card}(V)^{+}$, not $\mathrm{Card}(V)$; run on a countable graph, it will be bounded by a countable ordinal (i.e., $\alpha < \omega_1$).

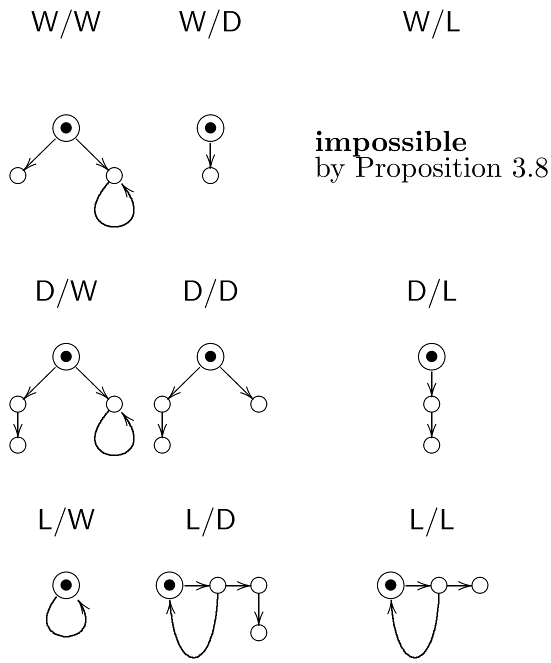

Figure 1. Examples for the eight possible value combinations in the asymmetric and the inverted asymmetric combinatorial game.

In analogy to Theorem 3.6, the Gale-Stewart analysis gives us the existence of A-sound labellings for trees $\mathbf{T}$. Among several ways to produce such a labelling, there is one procedure that was the motivation for the algorithm described in Section 4: in (Löwe, 2003), the present author gives a non-monotone variant of the Gale-Stewart procedure that is able to deal with a much more general situation (many-player games with reasonably simple payoffs); this non-monotonic Gale-Stewart procedure can be used to construct the A-sound labelling for trees.

PROPOSITION 3.9. *Let $\langle \mathbf{G}, s\rangle$ be a connected pointed graph and let ℓ be the A-sound labelling on $\mathbf{T}^s_{\mathbf{G}}$. Then for $v \in V_{\mathbf{G}}$, $e \in 2$ and W such that $Z_{\mathbf{T},\mathbf{A}}(W) = \langle e, v\rangle$, we define*

$$\ell/{\equiv_{Z_{\mathbf{T},\mathbf{A}}}}(\langle e, v\rangle) := \ell(W).$$

Then this quotient labelling is well-defined and is the A-sound labelling for $\mathbf{A}^s_{\mathbf{G}}$.

We would like to extend this to an A-sound bilabelling, but not all nodes in $\mathbf{G}$ are necessarily reachable by both players. The following simple construction helps:

If $\langle \mathbf{G}, s\rangle$ is a connected pointed graph, let $\mathbf{G}^*_s$ be defined by

$$\begin{aligned} V_{\mathbf{G}^*_s} &:= V_{\mathbf{G}} \cup \{x\}, \\ E_{\mathbf{G}^*_s} &:= E_{\mathbf{G}} \cup \{\langle x, s\rangle\} \end{aligned}$$

(where $x \notin V_{\mathbf{G}}$). The connectedness of $\langle \mathbf{G}, s\rangle$ implies that

$$\mathrm{ConCom}(\mathbf{A}_{\mathbf{G}}, \langle 0, s\rangle) \cup \mathrm{ConCom}(\mathbf{A}_{\mathbf{G}^*_s}, \langle 0, x\rangle) = 2 \times V_{\mathbf{G}};$$

moreover, if W is a walk through $\langle \mathbf{G}, s\rangle$ and W' is a walk through $\langle \mathbf{G}^*, x\rangle$, and $Z_{\mathbf{T},\mathbf{A}}(W) = Z_{\mathbf{T},\mathbf{A}}(W')$, then they represent exactly the same position in the game on $\mathbf{G}^*$ (albeit with different game histories), so if ℓ is A-sound on $\mathbf{T}^s_{\mathbf{G}}$ and ℓ^* is A-sound on $\mathbf{T}^x_{\mathbf{G}^*_s}$, then $\ell(W) = \ell^*(W')$. As a consequence, we get:

PROPOSITION 3.10. *If $\langle \mathbf{G}, s\rangle$ is a connected pointed graph, ℓ is an A-sound labelling on $\mathbf{T}^s_{\mathbf{G}}$ and ℓ^* is an A-sound labelling on $\mathbf{T}^x_{\mathbf{G}^*_s}$, then the bilabelling $\ell^\dagger$ defined by*

$$\ell^\dagger_e(v) := \begin{cases} \ell/{\equiv_{Z_{\mathbf{T},\mathbf{A}}}}(\langle e, v\rangle) & \textit{if } \langle e, v\rangle \in \mathrm{ConCom}(\mathbf{A}_{\mathbf{G}}, \langle 0, s\rangle), \textit{ or} \\ \ell^*/{\equiv_{Z_{\mathbf{T},\mathbf{A}}}}(\langle e, v\rangle) & \textit{otherwise} \end{cases}$$

is welldefined, total and an A-sound bilabelling on $\mathbf{G}$.

In the following section, we shall give an algorithm to compute $\ell^\dagger$ directly without going through the tree unravelling.

4. The algorithm

We fix a graph $\mathbf{G}$. For the purpose of this section, we assume that $V_{\mathbf{G}}$ is finite. Using the lattice structure on L, we can define an ordering $\leq^*$ on L^2 as the product ordering of $\langle L, \overline{\leq}\rangle$ with $\langle L, \leq\rangle$ as depicted in Figure 2.[4]

As mentioned, Fraenkel (following a suggestion of Azriel Lévy; *cf.* (Fraenkel, 1997, p. 15)) has published a labelling procedure that works directly on the graph instead of the unravelled tree. We call it the **Fraenkel-Lévy procedure**:

We let $\mathrm{FL}^0(v) := \mathsf{L}$ for all terminal nodes v. After that, for all v such that $v \notin \mathrm{dom}(\mathrm{FL}^n)$, we define

$$\mathrm{FL}^{n+1}(v) := \begin{cases} \mathsf{W} & \text{if there is } \langle v, w\rangle \in E \text{ and } \mathrm{FL}^n(w) = \mathsf{L}, \text{ or} \\ \mathsf{L} & \text{if for all } \langle v, w\rangle \in E, \text{ we have } \mathrm{FL}^n(w) = \mathsf{W}. \end{cases}$$

[4] Here $\overline{\leq}$ denotes the inverse ordering of $\leq$. By Proposition 3.8, no A-sound bilabelling can take the value W/L (*cf.* Figure 1).

$$
\begin{tikzcd}[column sep=small]
 & & \mathsf{L}/\mathsf{W} \arrow[dl] \arrow[dr] & & \\
 & \mathsf{D}/\mathsf{W} \arrow[dl] \arrow[dr] & & \mathsf{L}/\mathsf{D} \arrow[dl] \arrow[dr] & \\
\mathsf{W}/\mathsf{W} \arrow[dr] & & \mathsf{D}/\mathsf{D} \arrow[dl] \arrow[dr] & & \mathsf{L}/\mathsf{L} \arrow[dl] \\
 & \mathsf{W}/\mathsf{D} \arrow[dr] & & \mathsf{D}/\mathsf{L} \arrow[dl] & \\
 & & [\mathsf{W}/\mathsf{L}] & &
\end{tikzcd}
$$

Figure 2. The ordering $\leq^*$ on L^2.

For some $N \leq |V_{\mathbf{G}}|$, we have $\mathrm{FL}^N = \mathrm{FL}^{N+1}$; and we let

$$\mathrm{FL}(v) := \begin{cases} \mathrm{FL}^N(v) & \text{if } v \in \mathrm{dom}(\mathrm{FL}^N), \text{ or} \\ \quad \mathsf{D} & \text{otherwise.} \end{cases}$$

This algorithm produces the S-sound labelling in $O(|V_{\mathbf{G}}| + |E_{\mathbf{G}}|)$ steps, and its methodology is essentially that of the quotient labelling of the Gale-Stewart labelling on $\mathbf{T_G}$: we label a vertex $v \in V_{\mathbf{G}}$ as soon as some W with $Z_{\mathbf{T}}(W) = v$ is labelled in the Gale-Stewart construction. As the Gale-Stewart procedure, this labelling is monotonic in the sense that whenever a vertex is labelled, it will retain that label for ever.

Following this idea and injecting non-monotonicity in the spirit of (Löwe, 2003, § 5) into the procedure, we shall now give an algorithm `NMFL` (for "non-monotonic Fraenkel-Lévy") that produces the A-sound bilabelling.

We give the algorithm in pseudocode in Figure 3. Let us explain the two special datatypes `label` and `graph` used in the pseudocode:

Variables of type `label` can take the values `W`, `D`, and `L` representing W, D, and L. We have a binary relation $<$ defined for variables of type `label`, and `X` $<$ `Y` is `TRUE` if and only if $\mathsf{X} < \mathsf{Y}$. In addition, there is a unary function `INV` defined on `label` corresponding to the inversion function inv.

The datatype `graph` encodes the bilabelled graph structure. Let $\langle \mathbf{G}, \ell \rangle$ be a bilabelled graph with $V_{\mathbf{G}} = \{v_i\,; 0 \leq i \leq N\}$. If `G` is a variable of type `graph` representing $\langle \mathbf{G}, \ell \rangle$, then the following objects are defined:

- `Nvert`[`G`], the number of vertices of the graph, *i.e.*, $N + 1$;
- for each $i \leq N$, the objects `Outbound`[`G`, `i`] and `Inbound`[`G`, `i`] representing the sets $\{j\,; \langle v_i, v_j \rangle \in E_{\mathbf{G}}\}$ and $\{j\,; \langle v_j, v_i \rangle \in E_{\mathbf{G}}\}$, respectively;

– for each $i \leq N$ and $e \in 2$, an object $\mathtt{Ell[G, i, e]}$ of type `label`, representing $\ell_e(v_i)$.

We run the algorithm `NMFL` on (a representation of) $\mathbf{G}$. For any $t \in \mathbb{N}$, we let $\ell_e^t(v_i)$ be the value of $\mathtt{Ell[G, i, e]}$ at time t of the algorithm,[5] and

$$\ell^t(v_i) := \langle \ell_0^t(v_i), \ell_1^t(v_i) \rangle.$$

PROPOSITION 4.1. *For each $i \leq N$, the sequence $\langle \ell^t(v_i)\,;\, t \in \mathbb{N} \rangle$ is $\leq^*$-decreasing in L^2. In other words, the sequence $\langle \ell_0^t(v_i)\,;\, t \in \mathbb{N} \rangle$ is $\leq$-decreasing and the sequence $\langle \ell_1^t(v_i)\,;\, t \in \mathbb{N} \rangle$ is $\leq$-decreasing.*

Proof. Let $t+1$ be the least number such that

$$\ell^{t+1}(v_i) <^* \ell^t(v_i) \tag{$\star$}$$

for some i. Clearly, the value of the bilabeling can only be changed by code lines `110`, `140` or `255` of the algorithm.

Since the procedure `MAIN` can only change the values at terminal nodes from $\mathsf{L/W}$ to $\mathsf{D/W}$ and then to $\mathsf{D/L}$, the lines `110` and `140` cannot create a decrease in $\leq^*$. Note that if at time $t+1$ we are executing code line `255`, this implies that v_i is not a terminal node since $\mathtt{Label(G, i, e)}$ is only called if there is an edge from v_i to somewhere.

Since the bilabelling is initialized with $\mathsf{L/W}$ (which is the top element of $\langle L^2, \leq^* \rangle$), the first call of $\mathtt{Label(G, i, e)}$ cannot result in situation ($\star$), Consequently, there are some $s_0 < s_1 < t$ such that at both s_0 and s_1, the procedure $\mathtt{Label(G, i, e)}$ is called. Let s_0 be largest with that property. By ($\star$), we have that $\ell^{t+1}(v_i) <^* \ell^t(v_i) = \ell^{s_1}(v_i)$. By code lines `205` to `225`, we have

$$\ell_e^{t+1}(v_i) := \sup_{\leq}\{\mathrm{inv}(\ell_{1-e}^{s_1}(w))\,;\, \langle v_i, w \rangle \in E_{\mathbf{G}}\}, \text{ and}$$

$$\ell_e^{s_1}(v_i) := \sup_{\leq}\{\mathrm{inv}(\ell_{1-e}^{s_0}(w))\,;\, \langle v_i, w \rangle \in E_{\mathbf{G}}\}.$$

But this means that there is some w such that $\ell^{s_0}(w) <^* \ell^{s_1}(w)$, contradicting the choice of $t+1$ as minimal. q.e.d.

PROPOSITION 4.2. *The procedure* `Label` *is called at most $4 \cdot |E_{\mathbf{G}}|$ times.*

[5] Code lines `050` and `060` make sure that the values $\ell_e(t)(v_i)$ are defined early on in the algorithm (in step $e \cdot N + i$), so from step $2 \cdot N$ onwards, ℓ^t is a total function. In the following description, we shall mostly ignore these first $2 \cdot N$ steps of the algorithm.

```
ALGORITHM NMFL(G:graph)

PROCEDURE MAIN(G:graph)
  BEGIN
    FOR i:=1 TO Nvert[G] DO
      BEGIN
        Ell[G,i,0]:=L;
        Ell[G,i,1]:=W;
      END;
    FOR i:=1 TO Nvert[G] DO
      IF Outbound[G,i] = EMPTY THEN
        BEGIN
          Ell[G,i,0]:=D;
            FOR j IN Inbound[G,i] DO
              Label(G,j,1);
          Ell[G,i,1]:=L;
            FOR j IN Inbound[G,i] DO
              Label(G,j,0);
        END;
  END.


PROCEDURE Label(G:graph;i:integer;e:binary)
  BEGIN
    aux:=L;
    FOR j IN Outbound[G,i] DO
        aux := aux + INV[Ell[G,j,1-e]];
    IF Ell[G,i,e] != aux THEN
      BEGIN
        Ell[G,i,e]:=aux;
        FOR j IN Inbound[G,i] DO
          Label(G,j,1-e);
      END;
  END.
```

Figure 3. The algorithm for the non-monotonic Fraenkel-Lévy labelling in pseudocode.

Proof. By code lines `120`, `150`, and `265`, each call of $\texttt{Label}(G, i, e)$ is associated with an edge $\langle v_i, w\rangle$, and by code lines `110`, `140`, and `235`, preceded by a change of $\ell(w)$. By Proposition 4.1, this means that each such an edge can be used for calls of the procedure `Label` at most four times. q.e.d.

THEOREM 4.3. *The running time of* `NMFL` *is* $O(|V_{\mathbf{G}}| + |E_{\mathbf{G}}|^2)$. *If* $\mathbf{G}$ *is connected, this is* $O(|E_{\mathbf{G}}|^2)$.

Proof. The procedure `NMFL` itself (without the recursive calls of `Label`) takes at most $4 \cdot |V_{\mathbf{G}}| + 2 \cdot |E_{\mathbf{G}}|$ steps. Each call of `Label` has running time $O(|E_{\mathbf{G}}|)$, so by Proposition 4.2, the entire running time is $O(|V_{\mathbf{G}}| + |E_{\mathbf{G}}|^2)$. q.e.d.

The critical lines in the algorithm that push the running time from linear to quadratic are `215` and `225`: every time we run `Label` for v_i, we have to check the current values of all its successors. Let $\mathbf{G}_n$ be the graph with a root v_0 and n immediate successors of the root which are terminal nodes, *i.e.*, $|E_{\mathbf{G}_n}| = n$. Then `Label(G,1,e)` is called n times (once for each terminal node) and each time, its running time is at least n because it has to check each of the terminal nodes, so the total running time is at least $|E_{\mathbf{G}_n}|^2$.

If you allow a set operation `CAP` (intersection) as basic step of computation, then you can make the algorithm into a linear time algorithm by doing some bookkeeping of the current labelling values of vertices with set objects `Wverts[G,e]`, `Dverts[G,e]`, and `Lverts[G,e]` collecting all vertices currently e-labelled W, D or L, respectively. Then, the `FOR` loop in code lines `215` and `225` can be reduced to

```
211 IF Lverts[G,1-e] CAP Outbound[G,i] != EMPTY THEN
216   aux:=W;
221   ELSE
226     IF Dverts[G,1-e] CAP Outbound[G,i] != EMPTY THEN
231       aux:= D;
```

which is just a fixed finite amount of steps if you consider `CAP` to be a single operation.

THEOREM 4.4. *If run on the graph* $\mathbf{G}$, *the algorithm* `NMFL` *computes the* A*-sound bilabelling on* $\mathbf{G}$.

Proof. Again, let $\ell_e^t(v_i)$ be the value of `Ell[G,i,e]` at time t. By Theorem 4.3, these values stabilize at some finite time N. Let $\ell_e(v_i) := \ell_e^N(v_i)$ be the eventual value.

This proof follows essentially the idea of the Gale-Stewart proof (a.k.a. "backwards induction"). The main ingredient of that idea is that players have strategies that force the following to be true: if W is a run of the game according to the strategy, then the sequence of time indices of the label assignments of $W(n)$ during the algorithm is a decreasing sequence of integers. By wellfoundedness of $\mathbb{N}$, it can be deduced that we hit one of the basic cases eventually. Unfortunately, in our case, the nonmonotonicity of the algorithm causes some problems: it is possible that vertex v is labelled at time t, but some successors receive their label later. In order to deal with this, we have to go through the different cases in detail.

For any $v \in V_{\mathbf{G}}$, let

$$\mathrm{ind}_e(v) := \min\{t\,;\, \ell_e^t(v) = \ell_e(v)\}$$

be the **e-index** of v_i. (Note that by Proposition 4.1, for all $\mathrm{ind}_e(v) \leq t \leq N$, we have that $\ell_e^t(v) = \ell_e(t)$.)

We shall discuss the properties of the six possible cases:

Case 1. If $\ell_0(v_i) = \mathsf{W}$, then there is some w with $\langle v_i, w\rangle \in E_{\mathbf{G}}$, $\ell_1(w) = \mathsf{L}$, and $\mathrm{ind}_1(w) < \mathrm{ind}_0(v_i)$.

[The vertex v_i has been labelled by code line `255`, and this means that in the preceding call of code line `225` some successor was 1-labelled L. By Proposition 4.1, a 1-label L can never be changed anymore.]

Case 2. If $\ell_0(v_i) = \mathsf{D}$, then there is no w with $\langle v_i, w\rangle \in E_{\mathbf{G}}$ and $\ell_1(w) = \mathsf{L}$. Also, either v_i is terminal or there is some w with $\langle v_i, w\rangle \in E_{\mathbf{G}}$ and $\ell_1(w) = \mathsf{D}$, and for all such w, $\mathrm{ind}_1(w) < \mathrm{ind}_0(v_i)$.

[If v_i is terminal, the claim is trivial, so let v_i be nonterminal. Let $t := \mathrm{ind}_0(v_i)$ which is the time of a call of code line `255`. Therefore, at time t, we have

$$\forall w\,(\langle v_i, w\rangle \in E_{\mathbf{G}} \rightarrow \ell_1^t(w) \neq \mathsf{L})\ \&\ \exists w\,(\langle v_i, w\rangle \in E_{\mathbf{G}}\,\&\,\ell_1^t(w) = \mathsf{D}). \qquad (\star_t)$$

By Proposition 4.1, none of the vertices that are 1-labelled L can change their labelling anymore and the vertices 1-labelled D can only change their label to L. Suppose that there is some $t < s \leq N$ such that $(\star_s)$ is not true anymore. Then at the least such time s, we have a call of code line `255` and all successors of v_i are 1-labelled L. In the subsequent call of `Label[G,i,0]`, the label of v_i will be changed to W in contradiction to the assumption.]

Case 3. If $\ell_0(v_i) = \mathsf{L}$, then for all w with $\langle v_i, w\rangle \in E_{\mathbf{G}}$, we have $\ell_1(w) = \mathsf{W}$. Moreover, both v_i and all of its successors have received that label at the beginning of the algorithm (code lines `050` and `060`).

[Obvious from Proposition 4.1 and code lines `225` and `255`.]

Case 4. If $\ell_1(v_i) = \mathsf{W}$, then there is some w with $\langle v_i, w\rangle \in E_{\mathbf{G}}$ such that $\ell_0(w) = \mathsf{L}$ and both v_i and w have been labelled at the beginning of the algorithm (code lines `050` and `060`).

[This is dual to **Case 3.**]

Case 5. If $\ell_1(v_i) = \mathsf{D}$, then there is no w with $\langle v_i, w\rangle \in E_{\mathbf{G}}$ and $\ell_0(w) = \mathsf{L}$. Also, there is some w with $\langle v_i, w\rangle \in E_{\mathbf{G}}$ and $\ell_0(w) = \mathsf{D}$, and for all such w, $\mathrm{ind}_0(w) < \mathrm{ind}_1(v_i)$.

[This is dual to **Case 2.**, except that terminal nodes cannot be 1-labelled D.]

Case 6. If $\ell_1(v_i) = \mathsf{L}$, then for all w with $\langle v_i, w\rangle \in E_{\mathbf{G}}$, we have $\ell_0(w) = \mathsf{W}$ and $\mathrm{ind}_0(w) < \mathrm{ind}_1(v_i)$.

[This is dual to **Case 1.** (Note that this includes the possibility that v_i is terminal.)]

With our six cases in mind, we can now define strategies for player $e \in 2$ as follows:

The strategy σ_e plays from v into some successor w such that

$$\ell_{1-e}(w) = \mathrm{inv}(\ell_e(v)),$$

and –whenever possible– such that

$$\mathrm{ind}_{1-e}(w) < \mathrm{ind}_e(v).$$

We shall show that σ_0 and σ_1 are witnesses for value $\ell_0(v)$ in $\mathsf{A}(\mathbf{G}, v)$ and value $\ell_1(v)$ in $\overline{\mathsf{A}}(\mathbf{G}, v)$, respectively. This will finish the proof of Theorem 4.4.

Case A: $\ell_0(v) = \mathsf{W}$.

Let W be a maximal walk through $\langle \mathbf{G}, v\rangle$ where player 0 follows σ_0 (*i.e.*, $W(2n+1) = \sigma_0(W(2n))$). By **Case 1.** and **Case 6.**, we have $\ell_0(W(2n)) = \mathsf{W}$ and $\ell_1(W(2n+1)) = \mathsf{L}$. If W is finite (say, of length $n+1$), then $W(n)$ is terminal, so $\ell(W(n)) = \mathsf{D}/\mathsf{L}$, hence n is odd and player 0 has won the game with run W.

If W is infinite, then define

$$i_k := \mathrm{ind}_{\mathrm{par}(k)}(W(k)). \tag{†}$$

By definition of σ_0 and by **Case 1.** and **Case 6.**, this is a strictly decreasing sequence of natural numbers which is absurd.

Together, we get that $\mathrm{val}(\mathsf{A}(\mathbf{G}, v)) = \mathsf{W}$.

Case B: $\ell_1(v) = \mathsf{W}$.

We are now playing the inverted game, *i.e.*, player 1 starts. Let W be a maximal walk through $\langle \mathbf{G}, v\rangle$ where player 1 follows σ_1 in the inverted game. By **Case 3.** and **Case 4.**, we have $\ell_0(W(2n+1)) = \mathsf{L}$ and $\ell_1(W(2n+2)) = \mathsf{W}$. This implies that none of the vertices in W can be terminal, and thus W is infinite and player 1 wins the game with run W.

Consequently, $\mathrm{val}(\overline{\mathsf{A}}(\mathbf{G}, v)) = \mathsf{W}$.

Case C: $\ell_0(v) = \mathsf{D}$.

Let W be a maximal walk through $\langle \mathbf{G}, v \rangle$ where player 0 follows σ_0 in the regular (non-inverted) asymmetric game. By **Case 2.** and **Case 5.**, the following three subcases cover all possibilities:

Subcase B1. There is some n such that $\ell_0(W(2n)) = \mathsf{W}$. By **Case A**, player 0 wins.

Subcase B2. For all k, $\ell_{\mathrm{par}(k)}(W(k)) = \mathsf{D}$ and W is finite (say, of length $n+1$). Then $W(n)$ is a terminal node, and since $\ell_{\mathrm{par}(n)}(W(n)) = \mathsf{D}$, we have that n is even, so the game is a draw.

Subcase B3. For all k, $\ell_{\mathrm{par}(k)}(W(k)) = \mathsf{D}$ and W is infinite. Now by **Case 2.** and **Case 5.** and the definition of σ_0, the sequence i_k as defined in (†) is a strictly descending sequence of natural numbers, yielding a contradiction.

Similarly (using **Case B** instead of **Case A**), we can show that σ_1 is a nonlosing strategy for player 1. Together, σ_0 and σ_1 witness that $\mathrm{val}(\mathsf{A}(\mathbf{G}, v)) = \mathsf{D}$.

Case D: $\ell_0(v) = \mathsf{L}$.

By **Case 3.**, this means that player 0 is forced into a position w with $\ell_1(w) = \mathsf{W}$. Now apply **Case B**.

Case E: $\ell_1(v) = \mathsf{D}$.

This is dual to **Case B**.

Case F: $\ell_1(v) = \mathsf{L}$.

By **Case 6.**, this means that player 1 is forced into a position w with $\ell_0(w) = \mathsf{W}$. Now apply **Case A**.

q.e.d.

References

Aviezri S. **Fraenkel**, Combinatorial game theory foundations applied to digraph kernels, **Electronic Journal of Combinatorics** 4 (1997), Research Paper 10 (17 pages), [The Wilf Festschrift (Philadelphia, PA, 1996)]

David **Gale**, Frank M. **Stewart**, Infinite Games with Perfect Information, *in:* Harold W. Kuhn, Albert W. Tucker (*eds.*), Contributions to the Theory of Games II, Princeton 1953 [Annals of Mathematical Studies 28], p. 245–266

Benedikt **Löwe**, Determinacy for infinite games with more than two players with preferences, *submitted* (**ILLC Publication Series** PP-2003-19)

Stephen G. **Simpson**, Subsystems of Second Order Arithmetic, Berlin 1999 [Perspectives in Mathematical Logic]

John R. **Steel**, Determinateness and Subsystems of Analysis, Ph.D. Thesis, University of California at Berkeley, 1976

Kazuyuki **Tanaka**, Weak axioms of determinacy and subsystems of analysis, I: Δ^0_2 games, **Zeitschrift für Mathematische Logik und Grundlagen der Mathematik** 36 (1990), p. 481–491