A constructive game semantics for the language of linear logic

Giorgi Japaridze*

Department of Computer and Information Science
University of Pennsylvania
200 S. 33rd Street
Philadelphia, PA 19104-6389, USA
giorgi@saul.cis.upenn.edu

March 15, 1996

Abstract

I present a semantics for the language of first order additive-multiplicative linear logic, i.e. the language of classical first order logic with two sorts of disjunction and conjunction. The semantics allows us to capture intuitions often associated with linear logic or constructivism such as *sentences=games*, *sentences=resources* or *sentences=problems*, where "truth" means existence of an effective winning (resource-using, problem-solving) strategy.

The paper introduces a decidable first order logic ET in the above language and gives a proof of its soundness and completeness (in the full language) with respect to this semantics. Allowing noneffective strategies in the latter is shown to lead to classical logic.

The semantics presented here is very similar to Blass's game semantics (A.Blass, "A game semantics for linear logic", APAL, 56). Although there is no straightforward reduction between the two corresponding notions of validity, my completeness proof can likely be adapted to the logic induced by Blass's semantics to show its decidability (via equality to ET), which was a major problem left open in Blass's paper.

The reader needs to be familiar with classical (but not necessarily linear) logic and arithmetic.

*The author is grateful for support from the University of Pennsylvania, the National Science Foundation (on grants CCR-9403447 and CCR-9057570), and the Institute for Research in Cognitive Science at the University of Pennsylvania.

1 Introduction

One of the most controversial points of logical semantics is the existential quantifier,

$$\ldots \exists x \ldots,$$

read as

. . . there exists x such that . . . ,

or, sometimes, as

. . . there can be found x such that . . . .

The two readings are usually perceived as synonyms, and still the difference between them is crucial. "There exists" sounds metaphysical, whereas "can be found" means something that deals with reality. To leave alone the philosophy on the right of "existence" of the classical notion of existence, it simply has no practical meaning. Consider the sentence

For every disease there is a medicine which cures that disease.

If there is no way of finding, for each disease, a cure of it but we still know somehow that this sentence is true in the classical sense, we have no reason to be happier than we would be in the case if it were false.[1] In general, truth or falsity of a sentence can concern us only as far as this can signify something, can somehow be reflected in reality.

Of course "can be found" is a relative notion. Found by whom? If by God or another almighty being, then "exists" is really a synonym of "can be found". But the most interesting specification of "by whom" is *by a Turing machine*.

This treatment of the existential quantifier, when existence means being possible to be found by a machine, is captured by the nonclassical concept of "truth" which I suggest in this work and which will be called *effective truth*. The gist of the semantics of effective truth is that sentences are considered as certain *tasks, problems* which are to be solved by a machine, that is, by an agent who has an effective strategy for doing this; effective truth means existence of such a strategy.

The most convenient way of shaping this approach is to build all the semantics in terms of *games*:

a task (problem) = the task (problem) of winning a certain game.

[1] A naive opponent could object: The classical truth of this sentence means that we can try all the chemical stuffs, one by one, and sooner or later, one of them will work, so we do have a reason to be happy. Then I would give two answers in the same naive manner: First, what the opponent suggests already *is* a way of finding the medicine, and second: in fact this way is hardly a good way, because the poor sick man will, most likely, be poisoned and die before we reach the appropriate medicine.

There are two players in our games: Proponent, asserting a sentence, and Opponent who tries to refute it. Proponent, who represents me (us), is supposed to follow only effective strategies, whereas Opponent can use any strategy, for he is meant to represent blind forces of nature, or the devil himself.

The universal quantifier will always mean Opponent's move and the existential quantifier will mean Proponent's move. The above "medical" proposition can now be understood as the game each play of which consists of two moves: the first move is made by Opponent, who names an arbitrary disease d, and the second move is Proponent's, who must name a medicine m; the play is won by Proponent, if m really is a cure of d. Effective truth of this proposition can now really be a reason for leading a quiet life: we have an effective strategy (machine) such that, using it, we can always find a cure of any disease sent to us by the devil.

The connective $\vee$ will be treated in the same manner as the existential quantifier. Say,

$$\forall x(x \in P \vee x \notin P)$$

will be understood as the game each play of which, again, consists of two moves: first Opponent chooses an object n, which leads to the position $n \in P \vee n \notin P$, and then Proponent chooses between *left* and *right*, getting thus one of the positions $n \in P$ or $n \notin P$; the play is won by Proponent, if n belongs to P and he has chosen *left*, or n does not belong to P and *right* has been chosen. It is clear that effective truth of this proposition means nothing but decidability of the predicate $x \in P$.

In general, saying that a given sentence is effectively true, we always assert that certain relation similar to decidability (but maybe much more sophisticated than the latter) holds, as, e.g., the binary relation expressed by the sentence $\forall x(x \in P \vee x \in Q)$. Effective truth of this sentence means that there is an effective way of choosing, for each object a, one of the two sets P, Q such that a belongs to this set.

In the above examples the operator $\vee$ connects atomic formulas (games). In a more general case, for Proponent, to win the game $\alpha_1 \vee \alpha_2$ means that after he has chosen one of the components α_i, he must continue playing and win the game α_i, whereas the other component should be abandoned for ever. There is however another natural sort of disjunction, denoted by $\bigtriangledown$. The position $\alpha_1 \bigtriangledown \alpha_2$ does not oblige Proponent to choose one of the α_i and give up the other. He can make a move in one of the components, reserving at the same time the other, and switch any moment from α_1 to α_2 and back; the task is, playing in fact simultaneously in the two components, to win at least in one of them.

Strict definitions will be given in the main text (sections 2 and 3), but now, in order to develop intuition, we continue discussing some more "naive" examples.

I am in prison. My prison cell has two doors locked from the outside, the left-hand door and the right-hand door. My goal is to escape, and for that it is enough to pass through one of the doors. I happen to know that tonight one of the doors has been unlocked. Consider the proposition

The left-hand door is unlocked or *the right-hand door is unlocked.*

In order to escape it is enough for me to be able to "solve" this game (problem), in the role of Proponent, with "or" understood as $\bigtriangledown$: it is not necessary to be able to determine, at the very beginning, exactly which door is unlocked, I can simply try both and one of them will turn out to be unlocked. I write "solve" with quotation marks because in this game there are no moves at all and, under our assumption that one of the doors is really unlocked, it is trivially won.

Let us now slightly change the situation: the doors were not locked but mined, and tonight someone has removed the mine from one of the doors. Yes, we need now just $\vee$, and $\bigtriangledown$ will not do any more.

We treat negation $\neg$ in the following way: the rules of the game $\neg\alpha$ are the same as of α, only with the roles of Proponent and Opponent interchanged.

Example: Let C be a version of chess to win which for Proponent means to win a usual chess play within at most 100 moves, playing white. Then $\neg C$ will be the game to win which for Proponent means not to lose within 100 moves a chess play, playing black.[2]

Notice that the classical principle $\neg\neg\alpha \equiv \alpha$ does hold with our negation: after interchanging the roles twice, each player comes to his initial role. That means that all classical dualities work. In particular, $\forall$ can be defined in terms of $(\exists, \neg)$ by $\forall x\alpha(x) \equiv \neg\exists x\neg\alpha(x)$.

As the game of chess has been mentioned, a temptation arises to discuss one more example (which, however, must not be very original). Consider the game

$$C \bigtriangledown \neg C,$$

C being defined as above. This is in fact a play on two chessboards. On the left board Proponent plays white and on the right one he plays black. Proponent's task is to win in the sense of usual chess (well, with the within-100-moves amendment) on one of the chessboards. As switching components in a $\bigtriangledown$-play is Proponent's privilege, he has to move only in the case when the chess rules on both boards oblige him to move. I.e., as soon as Opponent has to move at least on one of the chessboards, Proponent can wait until Opponent makes this move.

I the Proponent, being not very good at chess, still can win this game even if my opponent is the world champion Kasparov, if I use the following strategy (solution): After Kasparov makes his first move on the right chessboard (where he plays white), I repeat the same move on the left chessboard (where I play white), then copy Kasparov's reply to this move back on the right chessboard

[2]In this example C is not a proposition but rather a "pure" game, and the propositional connective $\neg$ is thus an operation on games. But this is normal because propositions for us are nothing but games, and propositional connectives — operations on games.

and so on. This winning strategy can be used for any game of the type $\alpha \bigtriangledown \neg\alpha$, which means that the principle $\alpha \bigtriangledown \neg\alpha$ is valid in our sense.

As for the game $C \vee \neg C$, where at the very beginning I have to choose one of the chessboards and then win just on it, I have little chance to defeat Kasparov.[3]

To each sort of disjunction corresponds its dual conjunction, so we have two conjunctions $\wedge$ and $\bigtriangleup$, $\alpha \wedge \beta$ formally defined as $\neg(\neg\alpha \vee \neg\beta)$ and $\alpha \bigtriangleup \beta$ as $\neg(\neg\alpha \bigtriangledown \neg\beta)$.

For example, if I play with Kasparov the game $C \bigtriangledown \neg C$, to Kasparov this is the game $\neg C \bigtriangleup C$. He has to win on both chessboards. Besides, he has to move as soon as the chess rules oblige him to move on at least one of the boards.

Using the terminology of linear logic, we shall call $\vee$ the *additive disjunction*, $\wedge$ the *additive conjunction*, $\bigtriangledown$ the *multiplicative disjunction* and $\bigtriangleup$ the *multiplicative conjunction*.

That we use some terminology of linear logic is no accident. The logic of effective truth, i.e. the set of always effectively true sentences (like $\alpha \bigtriangledown \neg\alpha$), called *effective tautologies*, turns out to be an extension of Girard's [4] multiplicative-additive linear logic (MALL), in fact, a proper extension of MALL + weakening (so called BCK or Affine Logic), and the behavior of our additive and multiplicative connectives is very much similar to the behavior of those in linear logic.

Linear logic and other substructural logics are often called "resource logics". The people who first introduced logics of this type had some resource intuition in their minds, although this intuition has never been formalized as a strict semantics for the full language. And the name "resource logics" is related with some syntactic characteristics of these logics rather than justified semantically. These syntactic characteristics are determined by the forbidden rules of contraction or weakening. If we call formulas in a sequent resources, and read the sequent $A_1, \ldots, A_n \Rightarrow \Gamma$ as "The collection $A_1, \ldots, A_n$ of resources is enough for getting Γ" (let us not try to specify what "getting" means), then the contraction rule

$$\frac{A, A, \Theta \Rightarrow \Gamma}{A, \Theta \Rightarrow \Gamma}$$

says something like that you always can double any of your resources; that is, if the collection A, A, Θ was enough for getting Γ, then, this rule says, so is the collection A, Θ, because you can double A in in the latter.

And the weakening rule

$$\frac{\Theta \Rightarrow \Gamma}{A, \Theta \Rightarrow \Gamma}$$

says that you always can reduce the resources you possess; that is, if you can achieve Γ with the resource Θ, then so can you with the resource A, Θ, because

[3] However, taking into account that there is only a finite number of all possible plays of C, the game $C \vee \neg C$ has an effective solution (winning strategy for me), even if no modern machine is strong enough to follow this strategy. Still, a little bit more carefully chosen example would convince us that the game $\alpha \vee \neg\alpha$ is not always effectively solvable.

the latter can be reduced back to Θ.

Then, the justification for forbidding the contraction rule is that you cannot use more resources than you possess. The justification for forbidding the weakening rule sounds more odd: you have to use all the resources you possess.

Our logic of effective truth, too, is "resource conscious", and this is a natural consequence of the game-semantical approach.

Sentences in the classical logic are "static", they are given one of the two values 0 or 1, once and for ever. That's why a (sub)sentence, occuring more than once in a sentence or a sequent, is still, in all reasonable senses, the same in each occurrence; the quantity of these occurrences does not matter, and the rule of contraction works. As for the game semantics, sentences there are treated as something dynamic; two different occurrences of one and the same sentence denote *one and the same game*, i.e. one and the same set of potential plays, but in the process of playing up this game, the two occurrences can be realized as *different plays* of one and the same game. This is just what makes, say, $\alpha \bigtriangleup \alpha$ different from α. This is better illustrated by appealing again to chess.

In the above example with the game $C \bigtriangledown \neg C$ Proponent's winning strategy consisted in ensuring that the two occurrences of C in $C \bigtriangledown \neg C$ were realized as one and the same play. However, this trick fails to work with the game $C \bigtriangledown (\neg C \bigtriangleup \neg C)$, which is a game on three chessboards (chessboards NN 1,2 and 3, corresponding to the three occurrences of C, in their order from left to right). Kasparov can play different openings on the second and the third chessboards (where he plays white), and I can now only ensure that the play on the first chessboard coincides with the play on one of the chessboards N2 or N3, let it be N2. Then Kasparov can win on the chessboards NN 1 and 3 and, although I will have won on the chessboard N2, the whole play will be lost by me, because for winning $C \bigtriangledown (\neg C \bigtriangleup \neg C)$ it was necessary for me to win on the first chessboard or to win on both the second and the third chessboards.

Not only do resource-conscious effects arise as a consequence of the game-semantical approach, but our game semantics apparently has a chance to claim that it is a formalization of the intuitive "resource semantics". Let us speculate a little bit on this.

The things we call "resources" in everyday life are different in their nature: these can be, say, money, or electrical energy, or time and space (for computational operations). The feature which seems to be common for most things we call resources is that a resource is something necessary and/or enough for getting (achieving, accomplishing, obtaining, converting into) something. This suggests the first idea on the way of building a resource semantics: A resource must be characterized by the set of the things into which it can be *converted*.

To proceed, let us consider some examples. The style of most examples below is rather standard and Girard is the author of the sort of philosophy they support.

Today, in the situation of economic chaos which has followed the collapse of the USSR, two currencies are circulating in the ex-Soviet Republic of Georgia (my home country): Russian Rubles (RR) and Georgian coupons (GC). They are not easily convertible into each other, and it is even more problematic to convert them into dollars. However, if you are lucky enough to have a few dollars, you'll have no problems converting them into rubles or coupons, — any bank would buy dollars from you, offering for 1 dollar 1000 rubles or 1000000 coupons, — whichever you like.

Thus, the following two implications are true:

- *If you have \$1, then you can get 1000 RR.*
- *If you have \$1, then you can get 1000000 GC.*

But is then the sentence

- *If you have \$1, then you can get 1000 RR and you can get 1000000 GC*

true? Well, according to the classical logic it certainly is, but in the everyday language the above sentence, which could be shortened as

If you have \$1, then you can get 1000 RR and 1000000 GC,

would be more likely understood as that if you have \$1, then you can get both 1000 RR and 1000000 GC so that you can put the rubles into one pocket and the coupons into another. In this case the above implication is not true: you need \$2 rather than \$1 in the antecedent. Just at this point we arrive at the idea of considering two sorts of conjunction-like operations on resources: $\wedge$ and $\triangle$. To have the resource $A \wedge B$ means to have an option to convert it either into A or into B, — whichever you like, but only one of them. And to have the resource $A \triangle B$ means something more: It means to have both resources A and B and to be able to spend each of them its own way. Thus, having 1\$ implies having the resource $1000RR \wedge 1000000GC$, but not the resource $1000RR \triangle 1000000GC$.

Although coupons are the only legal currency in Georgia, because of the hyperinflation most people prefer to have rubles rather than coupons. The government still tries to strengthen coupons. It is not forbidden to accept rubles, but at least in the state-owned stores the salesmen are obliged to also accept coupons if the customer prefers to pay them instead of rubles by the rate $1RR = 1000GC$. E.g., a salesman who is selling a bottle of wine can get for it 1000 rubles or 1000000 coupons, but he can never know which of these two. This situation describes the disjunction-like operator $\vee$ on resources. The sentence

- *If the salesman has 1 bottle of wine, then he can get* $1000RR \vee 1000000GC$

is true, whereas both the sentences

- *If the salesman has 1 bottle of wine, then he can get* $1000RR$

and

- *If the salesman has 1 bottle of wine, then he can get* $1000000GC$,

where "can get" means "will get if wants" are false. We can see at this point that a resource cannot be fully characterized by the set of the things into which it can be immediately converted, — this set is the same, $\{1000RR, 1000000GC\}$, for the two resources:

- *1$*

and

- 1*bottle of wine,*

whereas the first resource is evidently stronger than the second. What makes different these two resources is the nature of this conversion. If I am the possessor, in the first case it is me who chooses between $1000RR$ and $1000000GC$, but in the second case it is not me. This is what makes the possessor of 1$ richer than the possessor of a bottle of wine.

Thus, a resource is characterized by two parameters: 1) the set of objects into which the resource can be immediately converted and 2) one of the two labels, — say, 0 and 1, where the label 0 indicates that the object into which the resource will be converted is chosen by the possessor of the resource, and the label 1 indicates that this choice is done by "somebody else". We can notice now that we have come to a game understanding of resources: resources are nothing but positions of a game; "A can be immediately converted into B" means that the transfer from the position A to the position B is a legal move; finally, the label 0 (resp. 1) for A means that it is Proponent's (resp. Opponent's) move in the position A.

Usually a resource is considered as just a means for achieving some *goal*, and the value of a resource is associated only with its potential convertibility into a (the) goal. We can consider goals as special sort of resources which cannot or should not be any more converted into anything else. If a resource is not a goal but, like a goal, is not any more convertible into anything, then it is a *dead end*: anyone who has reached a dead end has missed the possibility to reach a (the) goal.

Example: the goal is to get rid of my headache, and I have the resource of 1$ for that purpose. I can convert this dollar into an aspirin, then, taking it, I can "convert" the latter into its effect on the organism. If this relieves my headache, the goal is achieved. Otherwise I am at the situation of a dead end: the resources are spent, but the goal is missed.

We associate the label 1 with goals and the label 0 with dead ends. Reaching a goal means winning the play, and hitting a dead end means losing it. The purely "game" intuition behind this condition of losing a play is that the label 1 not only is Proponent's privilege to choose the next move, but it is also his duty to do so; however, at a dead end Proponent cannot carry out this duty because there are no more possible moves. The intuition behind the condition of winning a game is symmetric.

Which resources do we accept as "good"? — Just those which can ultimately be converted into goals. "Can be converted" here means that there is a (Proponent's) strategy which guarantees reaching a goal. And it is natural to require such a strategy to be effective.

We have not yet mentioned negation as an operator on resources. The intuition behind the negation $\neg A$ of a resource A can be characterized by saying that the following two acts are equivalent:

- to spend A;
- to get $\neg A$.

A few more words about the operator $\triangle$. Its exact behavior will be defined in Section 3, and it will be seen from that definition that in general $\triangle$ works as an operator that "adds up" resources, so that 1\$ $\triangle$ 1\$ is equal to something like 2\$. However, if A is a terminal resource such as a goal or a dead end, then our treatment of $\triangle$ will yield the equivalence of $A \triangle A$ and A. This should not confuse us. For, e.g., the situation

$$\textit{No more headache} \triangle \textit{No more headache}$$

is not any "better" than simply

No more headache.

They both mean nothing but that my goal of getting rid of the headache is achieved.

As for 1\$ $\triangle$ 1\$, it is "better" than 1\$ because it can be converted into *aspirin*$\triangle$*Tylenol*, and taking two different headache medicines gives me more chance to achieve the goal of no more headache.

I mentioned above that the logic of effective truth is a proper extension of BCK; on the other hand, it is strictly included in classical logic, where the language of the latter is meant to be augmented with $\triangledown$ and $\triangle$, which are thought of as synonyms of $\vee$ and $\wedge$, respectively.

The nice fact is that the logic of effective truth, in the full language, turns out to be decidable (in polynomial space), both at the propositional and the predicate levels, which contrasts with the undecidability of classical predicate logic. This is the focal result of the present work and its proof takes about 70% of the rest of the paper.

One more thing is worth mentioning here. As soon as we remove the requirement of effectiveness for Proponent's strategies in our games and allow any strategies, we get classical logic, where the distinction between the additive and the multiplicative versions of disjunction and conjunction simply disappears. Thus, classical logic and our variant of "linear logic" result from two special cases of one general semantical approach.

In the end, some historical remarks. Apparently Lorenzen [7] was the first to introduce a game semantics, in the late 50's. He suggested that the meaning of a proposition should be specified by establishing the rules of treating it in a debate (game) between a proponent who asserts the proposition and an opponent who denies it.

Lorenzen's approach describes logical validity exclusively in terms of rules without appealing to any kind of truth values for atoms, and this makes the semantics somewhat vicious (to my mind) as it looks more like just a "pure" syntax rather than a semantics.

Subsequently a lot of more work on game semantics was done by Lorenz [6], Hintikka and his group [3], and a number of other authors.

The notion of effective truth introduced in this paper, though defined in game-semantical terms, is in fact more similar to, say, Kleene's [5] recursive realizability, specifically, in what concerns the treatment of additive connectives and quantifiers (which are nothing but additives, again). At the same time, the predicate of recursive realizability is nonarithmetical, whereas the predicate of effective truth of an arithmetical sentence (where the latter is allowed to contain multiplicative connectives along with additives) has the complexity Σ^0_3.

To the comparison of recursive realizability and effective truth should be added that not all the recursively realizable sentences are true in the classical sense (e.g. some sentences of the form $\neg\forall x\big(\phi(x) \vee \neg\phi(x)\big)$ are recursively realizable), whereas effective truth is only a "strong version" of classical truth.

I elaborated this semantics some time before writing the present paper, and presented it in the talk "The logic of effective truth" at the *Logic and Computer Science conference* in Marseille (June 1992). At the same conference I met Andreas Blass and learned that he had found, — earlier than I, — a very similar semantics. It is described in his remarkable paper [2], where the decidability of the following two fragments of the corresponding logic is established:

1. The multiplicative propositional fragment, i.e. the fragment that uses only the connectives $\neg$, $\bigtriangledown$ and $\bigtriangleup$.

2. The fragment consisting of additive, i.e. $\bigtriangledown$- and $\bigtriangleup$-free, sequents (a sequent $\langle\phi_1, \ldots, \phi_n\rangle$ is thought of as the formula $\phi_1 \bigtriangledown \ldots \bigtriangledown \phi_n$).

However, the question whether the unrestricted logic corresponding to Blass's semantics (or, at least, the full propositional fragment of it) is decidable, recursively enumerable or even arithmetical, has not been answered so far.

Both fragments above coincide with the corresponding fragments of our logic of effective truth, and most likely this holds for the whole logic, too.

Together with the similarity between the two semantics, related with essentially identical treatments of logical connectives as operations on games, there are considerable differences between Blass's and our approaches and, especially, the consequences of these approaches:

1. Our games are finite (every play has a finite length, that is), whereas Blass's games are infinite and this fact plays a crucial role in all partial completeness proofs in [2].[4] This infiniteness makes things only second-order definable while all the theory of our games of bounded depth, including the completeness proof for the logic of effective truth, can be formalized in Peano Arithmetic.

2. At the same time, Blass's semantics does not require that Proponent's strategies be effective, while that requirement is the spirit of all our approach and its "constructivistic" effect is for us the main philosophical and practical motivation for introducing a nonclassical semantics.

3. The finiteness of our plays makes definitions simpler and more natural. E.g., a play is assumed to be lost by the player who has to move in the last position of the play, so we do not need a special parameter indicating which plays are won by which player in a game.

4. The notion of effective truth is only a strengthening of the classical notion of truth, and it is based on the traditional models for traditional languages. E.g., the standard model of arithmetic now becomes the unique game-semantical model where each atomic sentence α is a terminal position, with Opponent's obligation to move (which is though impossible to do), if α is true in the standard model, and Proponent's obligation to move otherwise. The set of effectively true arithmetical sentences is a proper subset of the set of those true in the classical sense.[5] As for Blass's approach, it hardly allows to speak about truth in the standard model of arithmetic, for, in order to maintain the difference between classical and game-semantical truth, atoms need to be interpreted as infinite games there, and then it is not clear what game should be a natural interpretation of, say, $a + b = c$.

Abramsky and Jagadeesan [1] revised Blass's game semantics by modifying game rules, and investigated the multiplicative fragment of the corresponding logic, which does not validate weakening any more and is thus closer to the original Girard's linear logic, being still stronger than the latter. This fragment is shown to be decidable, though the question on decidability of the whole logic, as well as of its full propositional fragment, remains open.

[4] E.g., the proof of nonvalidity of $\alpha \vee \neg\alpha$ in [2] uses a counterexample where α is an undetermined game, i.e. a game where none of the players has a winning strategy, and such a proof fails as soon as α is interpreted as a finite game because finite games are always determined when noneffective strategies are allowed.

[5] E.g., if α is a Π^0_3 arithmetical sentence which asserts its own not being effectively true, then $\alpha \vee \neg\alpha$ is true but not effectively true. Indeed, if we suppose that this additive disjunction is effectively true, a little analysis of our treatment of $\vee$ shows that then either α or $\neg\alpha$ should be effectively true. A further analysis of the situation reveals that in both cases we would then get an effectively true sentence which is classically false.

2 Basic notions and facts on games

Definition 2.1 A *net of games* is a triple $N = \langle W, l, R \rangle$, where:

- W is a nonempty, countable, decidable set, the elements of which are called *positions* of N.

- l is an effective function $W \to \{0, 1\}$, called the *labeling function*; for an element w of W, the value of $l(w)$ is called the *label* of w. Intuitively, $l(w) = 0$ means that Proponent has to move and $l(w) = 1$ means that Opponent has to move.

- R is a decidable binary relation, called the *development relation*, on W such that the converse of R is well-founded, i.e. there is no infinite chain $w_0 R w_1 R w_2 \ldots$ of positions. Intuitively, wRu means that the transfer from the position w to the position u is a legal move. In this case u is called a *development* of w (in N). $R(w)$ usually denotes the set of all developments of w.

Terminology 2.2 Let $N = \langle W, l, R \rangle$ be a net of games.

1. By a *legal N-sequence* we will mean any sequence of positions of N such that each $n + 1$–th term of the sequence (if it exists) is a development of the n–th term. Thus, the converse well-foundedness of R means nothing but that each legal N-sequence is finite.

2. Let $w, u \in W$. We say that u is an *rt-development* of w (in N), if the reflexive and transitive closure of R holds between w and u.

For $e \in \{0, 1\}$, we say that u is an *rt-e-development* of w, if there is a legal N-sequence $v_0, \ldots, v_n$ $(n \geq 0)$ of positions such that $v_0 = w$, $v_n = u$ and for each i with $0 \leq i < n$, v_i has the label e.

Thus, rt-e-development is a special case of rt-development.

3. As label 0 means Proponent's move and label 1 means Opponent's move, we will use the terms 0-*Player* and 1-*Player* as synonyms of Proponent and Opponent, respectively.

Definition 2.3 A *game* is a quadruple $G = \langle W, l, R, s \rangle$, where $N = \langle W, l, R \rangle$ is a net of games and s is an element of W. Usually, if N denotes a net $\langle W, l, R \rangle$ of games and $s \in W$, we use $N(s)$ to denote the game $\langle W, l, R, s \rangle$. "A position of $N(s)$" and "a position of N" are synonyms, and s is said to be the *starting position* of $N(s)$.

By a *legal G-sequence* we then mean any legal N-sequence whose first term is s.

Convention 2.4 It is not a "legal move" to speak about a function without specifying its type, as this is often done in this paper (e.g., Definitions 2.5, 2.8, 2.9). However, sometimes the type of a function really does not matter or can be seen from the context, and it would be awkward to still indicate, in each such case, a range and a domain for a function purely out of reasons of a correct style.

In order to avoid possible confusion caused by our irresponsible usage of the notion of function, let us fix a "large enough" universe U; namely, we assume that the set of positions of any game we consider is included in U; for safety we can suppose that all natural numbers (and maybe many other things) are in U. Then

- by a *function*, if not specified otherwise, we will always mean a partial function of the type $U \to U$;
- by a *finite function* we will mean a function (in the above sense) defined only for a finite number of arguments.

Definition 2.5 Let $G = \langle W, l, R, s \rangle$ be a game and f_0 and f_1 be functions.

1. *The G-play with Proponent's strategy f_0 and Opponent's strategy f_1* is a sequence P of positions of N which we construct in the following way:

   a) The first position of P is s.

   b) Suppose the first n positions of P are $w_1, \ldots, w_n$, and e is the label of w_n. Then:

   - if f_e is defined for w_n and $f_e(w_n) = u$ for some u with $w_n R u$, then the $n+1$-th position of P is u;
   - otherwise w_n is the *last position* of P.

   Notice that P is a legal G-sequence and thus P is finite.

2. *A G-play* is the G-play with Proponent's strategy f and Opponent's strategy g for some functions f and g.

   Observe that a G-play is nothing but a legal G-sequence.

3. *A G-play with e-Player's strategy f* (where $e \in \{0, 1\}$, see 2.2.3) is the G-play with e-Player's strategy f and $(1-e)$-Player's strategy g for some function g.

   In other words, a G-play with e-Player's strategy f is a legal G-sequence $\langle w_1 (= s), \ldots, w_n \rangle$ such that for any $1 \leq i \leq n$ with $l(w_i) = e$, we have:

   - if $f(w_i) = u$ for some $u \in R(w_i)$, then $i < n$ and $w_{i+1} = u$;
   - otherwise $i = n$.

Definition 2.6 The *depth* of a game $N(s) = \langle W, l, R, s \rangle$ is the least ordinal number α such that for every w with sRw, $\alpha >$ *the depth of* $N(w)$. Thus, if s has no developments, the depth of $N(s)$ is 0.

Very roughly, the depth of a game G is the maximal possible length of a G-play.

Definition 2.7 Suppose G is a game, P is a G-play, w is the last position of P and e is the label of w. Then we say that P is *lost* by e-Player and *won* by $(1-e)$-Player.

Simply the words "won" and "lost", without specifying the player, will always mean "won by Proponent" and "lost by Proponent".

Thus, every play is either won or lost. Intuitively, a play is won if a position (the last position) is reached where Opponent has to move (as the label of that position is 1) but cannot, and in a lost play we have the dual situation: Proponent has to move but cannot.

Definition 2.8 Let G be a game.

- A *solution* to G (*Proponent's winning strategy* for G) is a function f such that every G-play with Proponent's strategy f is won.
- Dually, an *antisolution* to G (*Opponent's winning strategy* for G) is a function g such that every G-play with Opponent's strategy g is lost.

Taking into account that the development relation is converse well founded, the following fact 2.9 can be considered as a correct alternative definition of the notion of solution; the form of this definition suggests that the relation "... is a solution to ..." applied later to formulas interpreted as games, belongs to the family of relations of the type "... realizes ..." which lead to diverse well-known concepts of realizability (see [8]).

Fact 2.9 (Another definition of the notion of solution) *A function f is a solution to a game $N(s) = \langle W, l, R, s\rangle$ iff the following holds:*

- *a) if $l(s) = 1$, then for all $w \in R(s)$, f is a solution to the game $N(w)$;*
- *b) if $l(s) = 0$, then $f(s) = w$ for some position w such that $w \in R(s)$ and f is a solution to the game $N(w)$.*

Proving that the above two definitions of solution are equivalent would be an easy warming-up exercise for the reader.

Fact 2.10 *A function f is a solution to a game G if and only if for any finite function* (see Convention 2.4) *g, the G-play with Proponent's strategy f and Opponent's strategy g is won.*

PROOF. Taking 2.8 as the basic definition of solution, the "only if" direction is trivial. For the "if" direction, suppose f is not a solution to G, i.e. there is a function h such that the G-play P with Proponent's strategy f and Opponent's strategy h is lost. Let then g be the function which coincides with h for the positions that participate in P and is undefined for any other object. Clearly the G-play with Proponent's strategy f and Opponent's strategy g is the selfsame P. On the other hand, since P (as well as any legal G-sequence) is finite, the function g is finite. Thus, g is a finite function such that the G-play with Proponent's strategy f and Opponent's strategy g is lost. ♣

Definition 2.11

- A game is said to be *solvable*, if it has a solution.
- A game is said to be *effectively solvable*, if it has an effective (recursive) solution.

A solution, defined in 2.8, is a function of current position and it does not see previous moves (the history of the play). However, in some situations it is more convenient to deal with a strategy which scans the whole initial segment of the play rather than the last position. Such strategies will be called "history-sensitive".

Definition 2.12 Let $G = \langle W, l, R, s\rangle$ be a game and f be a function. A G-play with Proponent's *history-sensitive* strategy f is a legal G-sequence $\langle w_1, \ldots, w_n\rangle$ such that for every $i \leq n$ with $l(w_i) = 0$ we have:

- if $f\langle w_1, \ldots, w_i\rangle = u$ and $w_i R u$, then $i < n$ and $w_{i+1} = u$;
- otherwise $i = n$.

Definition 2.13 A *history-sensitive solution* to a game G is a function f such that any G-play with Proponent's history-sensitive strategy f is won.

Theorem 2.14

1. *A game has a solution iff it has a history-sensitive solution.*
2. *A game has has an effective solution iff it has an effective history-sensitive solution.*

PROOF. We prove here only the clause 2 of the theorem. The proof of the clause 1 is simpler.

Consider a game $G = \langle W, l, R, s\rangle$.

($\Rightarrow$): Suppose f is an effective solution to G. Let g be the function defined by $g(\langle w_1, \ldots, w_n\rangle) = f(w_n)$. Evidently g is then an effective history-sensitive solution to G.

($\Leftarrow$): Suppose g is an effective history-sensitive solution to G and M_g is a machine that computes g.

- Let *a good sequence* mean a legal G-sequence $\langle w_1, \ldots, w_n\rangle$ such that for any $1 \leq i < n$, if $l(w_i) = 0$, then $g\langle w_1, \ldots, w_i\rangle = w_{i+1}$.

As W,l,R and g are recursive, the good sequences can be recursively enumerated. So, let us fix a recursive list of good sequences.

Let now f be a partial recursive function the value of which for an element w of W is computed by the following machine M_f:

- First M_f checks (from the beginning) the list of good sequences till the moment when a good sequence $\langle t_1, \ldots, t_e \rangle$ is found such that $t_e = w$. Then M_f simulates the machine M_g with $\langle t_1, \ldots, t_e \rangle$ on the input of the latter; if M_g halts and gives the output u for some $u \in R(w)$, then M_f gives the same output u.

The claim is that f is a solution to G. To show this, suppose, for a contradiction, that there is a lost G-play $\langle w_1, \ldots, w_n \rangle$ with Proponent's strategy f. Let us first verify by induction on i that

$$\textit{for any } 1 \leq i \leq n, \textit{ there is a good sequence whose last term is } w_i. \qquad (1)$$

This is trivial for $i = 1$ because $w_1 = s$ and $\langle s \rangle$ is a good sequence. Suppose now $i > 1$. Then, by the induction hypothesis, there is a good sequence $\langle u_1, \ldots, u_m \rangle$ with $u_m = w_{i-1}$. If $l(u_m) = 1$, then obviously $\langle u_1, \ldots, u_m, w_i \rangle$ is a good sequence. Suppose now that $l(u_m) = 0$, i.e. $l(w_{i-1}) = 0$. Then, as $\langle w_1, \ldots, w_n \rangle$ is a G-play with Proponent's strategy f, we have $w_i = f(w_{i-1})$. According to the definition of f, this means that for some good sequence $\langle v_1, \ldots, v_k \rangle$ with $v_k = w_{i-1}$, we have $g(\langle v_1, \ldots, v_k \rangle) = w_i$. But then $\langle v_1, \ldots, v_k, w_i \rangle$ is a good sequence, and (1) is proved.

Thus, by (1), there is a good sequence $\langle t_1, \ldots, t_e \rangle$ with $t_e = w_n$. We may suppose that $\langle t_1, \ldots, t_e \rangle$ is the first good sequence in the list of good sequences whose last term is w_n.

Observe that $\langle t_1, \ldots, t_e \rangle$ (as well as any good sequence) is an initial segment of some G-play P with Proponent's history-sensitive strategy g. Since $l(t_e) = 0$ and g is a history-sensitive solution to G, t_e cannot be the last position of P, i.e. we must have $g \langle t_1, \ldots, t_e \rangle = r$ for some $r \in R(t_e) = R(w_n)$. But then, by the definition of f, we have $f(w_n) = r \in R(w_n)$, which contradicts our assumption that w_n is the last position of a G-play (namely of $\langle w_1, \ldots, w_n \rangle$) with Proponent's strategy f: at least, the position r must follow w_n in this play. The theorem is proved. ♣

Lemma 2.15 *Suppose $N(s) = \langle W, l, R, s \rangle$ is a game such that $l(s) = 1$ and for each $u \in R(s)$, $N(u)$ is solvable. Then $N(s)$ is solvable.*

PROOF. For each $u \in R(s)$, let us fix a solution g_u to $N(u)$. We define a function f and show that it is a history-sensitive solution to $N(s)$. By Theorem 2.14, that will mean that there is a solution to $N(s)$. So, for any $u \in R(s)$ and $v_1, \ldots, v_n$ with $v_1 = u$, $n \geq 1$, let

$$f \langle s, v_1, \ldots, v_n \rangle = g_u(v_n).$$

Any $N(s)$-play with Proponent's history-sensitive strategy f looks like $\langle s, v_1, \ldots, v_n \rangle$, where, unless $n = 0$, we have $v_1 = u$ for some $u \in R(s)$. Observe that if such a play is lost, then ($n \neq 0$ and) $\langle v_1, \ldots, v_n \rangle$ is a lost $N(u)$-play with Proponent's strategy g_u. But this is impossible because g_u is a solution to $N(u)$. ♣

What follows is in fact a well known theorem which is due to Zermelo:

Theorem 2.16 *To any game there is either a solution or an antisolution, i.e. exactly one of the players has a winning strategy.*

PROOF. Before we start proving, note that almost all the definitions and facts on games enjoy perfect duality: we can always interchange "solution" and "antisolution", "Proponent" and "Opponent", "0" and "1", "won" and "lost".

Fix a game $N(s) = \langle W, l, R, s \rangle$.

First observe that both players cannot have winning strategies for $N(s)$, for otherwise the play corresponding to these two strategies should be simultaneously won and lost, which is impossible.

Let h be the depth (see 2.6) of $N(s)$. We may suppose that every $w \in W$ is an rt-development (see 2.2.2) of s, which means that the depth of $N(w)$ for any w with $s \neq w \in W$ is less than h.

By induction on depths $\leq h$ we are going to show that for an arbitrary element w of W, one of the players has a winning strategy for $N(w)$. Before using induction, we consider four cases and show that in each of them one of the players has a winning strategy.

Case 1: $l(w) = 0$ and there is $u \in R(w)$ such that Proponent has a winning strategy g for $N(u)$.

Let then $f(w) = u$ and for any $w \neq v \in W$, $f(v) = g(v)$. Since w can never appear in an $N(u)$-play (because of the converse well-foundedness of R), it is clear that f is a solution to $N(u)$, whence, by 2.9, f is a solution to $N(w)$.

Case 2: $l(w) = 1$ and there is $u \in R(w)$ such that Opponent has a winning strategy g for $N(u)$.

Dual to the previous case: we can define an antisolution f to $N(w)$.

Case 3: $l(w) = 1$ and for any $u \in R(w)$ there is a solution to $N(u)$.

Then, by Lemma 2.15, there is a solution to $N(w)$.

Case 4: $l(w) = 0$ and for any $u \in R(w)$ there is an antisolution to $N(u)$.

Dual to the case 3, with the conclusion that there is an antisolution to $N(w)$.

Now it remains to show that one of the above cases always takes place. Indeed:

Suppose $l(w) = 0$ and the case 4 is "not the case", i.e. there is $u \in R(w)$ such that $N(u)$ has no antisolution. Since the depth of $N(u)$ is less than the depth of $N(w)$, we can apply the induction hypothesis to $N(u)$ and conclude that Proponent has a winning strategy for $N(u)$, i.e. we deal with the case 1.

Suppose now $l(w) = 1$ and the case 3 does not take place, i.e. there is $u \in R(w)$ such that $N(u)$ has no solution. Then, by the induction hypothesis, there is an antisolution to $N(u)$, which means that we deal with the case 2. ♣

3 Sentences as games

Terminology and notation 3.1

1. By a "language" in this paper we mean a classical first order language without functional or individual symbols supplemented with the two additional binary connectives $\bigtriangledown$ and $\bigtriangleup$.

More precisely, a *language* is determined (and thus can be identified with) a countable set of *predicate letters* together with a function which assigns to each predicate letter P a natural number n called the *arity* of P (and P is then said to be n-ary).

Besides, the alphabet of each language consists of:

- Individual *variables*: $v_1, v_2, v_3, \ldots$; we use $x, y, z \ldots$ as metavariables for them.
- *Propositional connectives*: $\neg$ (negation), $\vee$ (additive disjunction), $\wedge$ (additive conjunction), $\bigtriangledown$ (multiplicative disjunction), $\bigtriangleup$ (multiplicative conjunction).
- *Quantifiers*: $\exists$ (existential quantifier), $\forall$ (universal quantifier).
- Technical signs: , (comma), ((left parenthesis),) (right parenthesis).

2. Throughout the paper L denotes some fixed language.

3. We define the set of *literals* of L as the union of the sets of positive and negative literals of L, defined as follows:

- α is a *positive literal* of L, if $\alpha = P(x_1, \ldots, x_n)$, where P is an n-ary predicate letter and $x_1, \ldots, x_n$ are variables (if $n = 0$, then $P(x_1, \ldots, x_n)$ is just P).
- α is a *negative literal* of L, if $\alpha = \neg\beta$ for some positive literal β of L.

The word "atom" will be used as a synonym of "positive literal".

4. *Formulas* of L are the elements of the smallest class Fm_L of expressions such that, saying "α is a formula of L" for "$\alpha \in Fm_L$", we have:

- Literals of L are formulas of L.
- If α and β are formulas of L, then $(\alpha) \vee (\beta)$, $(\alpha) \wedge (\beta)$, $(\alpha) \bigtriangledown (\beta)$, $(\alpha) \bigtriangleup (\beta)$ are formulas of L.
- If α is a formula of L and x is a variable, then $\exists x(\alpha)$ and $\forall x(\alpha)$ are formulas of L.

We often omit some parentheses in formulas, when this does not lead to any ambiguity.

5. Thus, in the formal language we prefer to restrict the scope of $\neg$ only to atoms. However, we introduce $\neg\alpha$ for complex formulas as an abbreviation defined as follows:

- $\neg(\neg\alpha) =_{df} \alpha$

- $\neg(\alpha \vee \beta) =_{df} \neg\alpha \wedge \neg\beta$
- $\neg(\alpha \wedge \beta) =_{df} \neg\alpha \vee \neg\beta$
- $\neg(\alpha \triangledown \beta) =_{df} \neg\alpha \vartriangle \neg\beta$
- $\neg(\alpha \vartriangle \beta) =_{df} \neg\alpha \triangledown \neg\beta$
- $\neg(\exists x \alpha) =_{df} \forall x \neg\alpha$
- $\neg(\forall x \alpha) =_{df} \exists x \neg\alpha$.

6. Formulas α and $\neg\alpha$ are said to be *opposite* to each other.

7. We define a *free occurrence* of a variable x in a formula in the usual way: this is an occurrence of x that is not in the scope of an occurrence of $\exists x$ or $\forall x$.

8. We will often use the standard notational convention: a formula β can be denoted by $\beta(x_1, \ldots, x_n)$, where $x_1, \ldots, x_n$ are any variables (not all of them have to occur free in β and not all the free variables of β have to be among them). Then $\beta(t_1, \ldots, t_n)$, where the t_i's are variables or any other terms (see below), denotes the result of substituting $t_1, \ldots, t_n$ for all free occurrences of $x_1, \ldots, x_n$, respectively, in β.

9. A *closed formula* or a *sentence* is a formula without free occurrences of variables.

10. Suppose $\mathcal{D}$ is a nonempty countable set (of "individuals"). A *formula of L with parameters in $\mathcal{D}$* is a pair $\langle \alpha, f \rangle$, where α is a formula of L and f is a (finite) function $V' \to \mathcal{D}$ for some subset V' of the set V of free variables of α; if $V' = V$, then we deal with a *sentence of L with parameters in $\mathcal{D}$*.

We can think of sentences with parameters in $\mathcal{D}$ as formulas in which some free variables are "substituted by elements of $\mathcal{D}$", and write, e.g., $\alpha(a_1, \ldots, a_n)$ for $\langle \alpha(x_1, \ldots, x_n), f \rangle$, if $f(x_1) = a_1$, …, $f(x_n) = a_n$.

11. We can use the words "literal" and "atom" for formulas (that is, literals or atoms) with parameters, too. If such a literal is a sentence, then we call it a *sliteral*.

Definition 3.2 A *model* for L is a triple $\mathcal{M} = \langle \mathcal{D}_\mathcal{M}, \ell_\mathcal{M}, \mathcal{R}_\mathcal{M} \rangle$ such that:

- $\mathcal{D}_\mathcal{M}$ is a nonempty countable decidable set, called the *domain of individuals*;
- $\ell_\mathcal{M}$ is an effective total function of type {atomic sentences of L with parameters in $\mathcal{D}_\mathcal{M}$} $\to \{0, 1\}$, called the *prelabeling function*;
- $\mathcal{R}_M$ is a decidable converse well-founded binary relation on {atomic sentences of L with parameters in $\mathcal{D}_\mathcal{M}$}, called the *predevelopment relation*.

Definition 3.3 A model $\mathcal{M}$ is said to be *elementary*, if the relation $\mathcal{R}_\mathcal{M}$ is empty.

Definition 3.4 Let $\mathcal{M}$ be a model for L. We define

$$N_{\mathcal{M}} = \langle W_{\mathcal{M}}, l_{\mathcal{M}}, R_{\mathcal{M}} \rangle ,$$

the *net of games induced by* $\mathcal{M}$, as follows:

- $W_{\mathcal{M}}$ is the set of all sentences of L with parameters in $\mathcal{D}_{\mathcal{M}}$.
- 1. $l_{\mathcal{M}}(\alpha) = \ell_{\mathcal{M}}(\alpha)$, if α is an atom;
  2. $l_{\mathcal{M}}(\alpha \vee \beta) = l_{\mathcal{M}}(\exists x\alpha) = 0$;
  3. $l_{\mathcal{M}}(\alpha \wedge \beta) = l_{\mathcal{M}}(\forall x\alpha) = 1$;
  4. $l_{\mathcal{M}}(\neg\alpha) = 1 - l_{\mathcal{M}}(\alpha)$;
  5. $l_{\mathcal{M}}(\alpha \triangledown \beta) = max\{l_{\mathcal{M}}(\alpha), l_{\mathcal{M}}(\beta)\}$;
  6. $l_{\mathcal{M}}(\alpha \vartriangle \beta) = min\{l_{M}(\alpha), l_{\mathcal{M}}(\beta)\}$.
- $\phi R_{\mathcal{M}} \psi$ iff one of the following holds:
  1. ϕ, ψ are atoms and $\phi \mathcal{R}_{\mathcal{M}} \psi$;
  2. $\phi = \alpha * \beta$, where $* \in \{\vee, \wedge\}$, and $\psi = \alpha$ or $\psi = \beta$;
  3. $\phi = *x\alpha(x)$, where $* \in \{\exists, \forall\}$, and $\psi = \alpha(a)$ for some $a \in \mathcal{D}_M$;
  4. $\phi = \neg\alpha$, $\alpha R_{\mathcal{M}} \alpha'$ for some α' and $\psi = \neg\alpha'$;
  5. $\phi = \alpha * \beta$, where $* \in \{\triangledown, \vartriangle\}$, and:
     - $l_{\mathcal{M}}(\alpha) = l_{\mathcal{M}}(\phi)$, $\alpha R_{\mathcal{M}} \alpha'$ for some α' and $\psi = \alpha' * \beta$, or
     - $l_{\mathcal{M}}(\beta) = l_{\mathcal{M}}(\phi)$, $\beta R_{\mathcal{M}} \beta'$ for some β' and $\psi = \alpha * \beta'$.

Now we can see how sentences are to be interpreted as games: given a model $\mathcal{M}$ for L, each sentence α of L is understood as the game $N_{\mathcal{M}}(\alpha)$. This game can be referred to as "the game corresponding to α", or, simply, "the game α" instead of $N_{\mathcal{M}}(\alpha)$.

The games of type $N_{\mathcal{M}}(\alpha)$ we call *linguistic games*.

The standard model of arithmetic defined below is an example of elementary model.

There are many versions, equivalent in expressive power, of the language of arithmetic. Here we choose one of them with infinitely many predicate letters:

$$Q_0, Q_1, Q_2, \ldots$$

In particular, let $Def_0, Def_1, Def_2, \ldots$ be the enumeration, by increasing Gödel numbers, of all primitive recursive definitions of relations. And let

$$R_0, R_1, R_2, \ldots$$

be the relations defined by $Def_0, Def_1, Def_2, \ldots$, respectively. Then, if n is the arity of R_j, we associate the same arity n with the predicate letter Q_j. Each predicate letter Q_i is interpreted as ("represents") the relation R_i, as this interpretation is set by the following definition of the standard model of arithmetic:

Definition 3.5 The *standard model of arithmetic*, denoted by $\mathcal{S}$ throughout this paper, is the following elementary model $\langle \mathcal{D}_\mathcal{S}, \ell_\mathcal{S}, \oslash \rangle$:

$\mathcal{D}_\mathcal{S}$ is the set $NAT = \{0, 1, 2, \ldots\}$ of natural numbers;

for any i and any tuple $a_1, \ldots, a_n$ of natural numbers, where n is the arity of the predicate letter Q_i, we have

$$\ell_\mathcal{S}\big(Q_i(a_1, \ldots, a_n)\big) = 1 \ \text{ iff } \ R_i(a_1, \ldots, a_n).$$

Convention 3.6 By abuse of notation, if R denotes an n-ary primitive recursive relation in our metalanguage, we will use the same expression "R" in an arithmetical formula, instead of a predicate letter Q_i which represents R. Of course, this practice induces ambiguity because $R = R_i$ for infinitely many i's (that is, infinitely many Q_i's represent R). However, we can suppose that for each relation R we deal with, we choose one fixed Q_i representing R, and everywhere in the text then "R" stands for this concrete Q_i.

This convention allows us to use standard notations for standard primitive recursive relations like "$x < y$", "$x + y = z$", "$2^x = y$" etc. without explaining their meanings, as this is done in the example below; each such expression is thought of as an atomic formula of the language of arithmetic. This allows us to pretend that in the arithmetical language we have *terms* for primitive recursive functions, and consider as formulas expressions like $\alpha(h(y,z))$, whenever $\alpha(x)$ is a formula and $h(y,z)$ is a function. Note that it is not the case that $\alpha(h(y,z))$ contains some predicate H for the graph of h and looks like, say, $\exists t(H(y,z,t) \wedge \alpha(t))$. Rather, $\alpha(x)$ and $\alpha(h(y,z))$ have exactly the same logical structure; $\alpha(h(y,z))$ is simply the result of replacing in $\alpha(x)$ each atom $Q_i(x, \vec{u})$, containing a free (in α) occurrence of x and representing some relation $R(x, \vec{u})$, by a (the) atom $Q_j(y, z, \vec{u})$, representing the relation $R(h(y,z), \vec{u})$. So $R(x, \vec{u})$ and $R(h(y,z), \vec{u})$ simply denote two different atoms of two different (in this case) arities.

The sequence of the following arithmetical sentences with parameters in NAT is a legal sequence of positions of $N_\mathcal{S}$, in fact a won play (for an explanation of the notation $N_\mathcal{S}$ see Definition 3.4):

1. $\big(0 = 1 \vee \forall v_1 \exists v_2 (v_1 = v_2)\big) \bigtriangleup \big(\exists v_1 \forall v_2 (v_1 + v_2 = v_2) \bigtriangledown 2 = 3\big)$
2. $\forall v_1 \exists v_2 (v_1 = v_2) \bigtriangleup \big(\exists v_1 \forall v_2 (v_1 + v_2 = v_2) \bigtriangledown 2 = 3\big)$
3. $\forall v_1 \exists v_2 (v_1 = v_2) \bigtriangleup \big(\forall v_2 (0 + v_2 = v_2) \bigtriangledown 2 = 3\big)$

4. $\exists v_2(124 = v_2) \bigtriangleup \big(\forall v_2(0 + v_2 = v_2) \bigtriangledown 2 = 3\big)$

5. $124 = 124 \bigtriangleup \big(\forall v_2(0 + v_2 = v_2) \bigtriangledown 2 = 3\big)$

6. $124 = 124 \bigtriangleup \big(0 + 18 = 18 \bigtriangledown 2 = 3\big)$.

Why did not we restrict our considerations to only elementary models, what do we need the predevelopment relation for? In elementary models atoms are interpreted as very specific games — games of depth 0, which are always trivially solvable or antisolvable, whereas we need to be able to interpret atoms as any possible games. Suffice it to say that otherwise the logic corresponding to our semantics would not be closed under the substitution rule. E.g., when we deal with elementary models, the game $\alpha \vee \neg\alpha$ is always effectively solvable for an atomic α, but it may be effectively unsolvable for $\alpha = \forall x \exists y \forall z \beta$. Besides, any net of games can be completely captured by our models (but hardly by elementary models), and then the logical operators appear as operations on games. E.g., one of the straightforward ways of "capturing" a net $\langle W, l, R \rangle$ is to interpret by its positions $w_0, w_1, \ldots$ the atomic sentences $P_0, P_1, \ldots$ (or $P(0), P(1), \ldots\ldots$), defining the value of the prelabeling function for P_n to be equal to $l(w_n)$ and stipulating that the predevelopment relation holds between P_n and P_m iff $w_n R w_m$.

4 Truth and effective truth

Identifying $\bigtriangledown$ with $\vee$ and $\bigtriangleup$ with $\wedge$, we can think of L as a classical first order language. A model in classical logic is understood as a pair $\mathcal{M} = \langle \mathcal{D}, \mathcal{G} \rangle$, where $\mathcal{D}$ is a nonempty set (domain of individuals) and $\mathcal{G}$ is a function which assigns to each n-ary predicate letter P of the language an n-ary relation $\mathcal{G}^P$ on $\mathcal{D}$. Then for a sentence α of L with parameters in $\mathcal{D}$, the *classical value* of α in $\mathcal{M}$, denoted by $CV_{\mathcal{M}}(\alpha)$, is defined by the following induction on the complexity:

- for an atom $P(a_1, \ldots, a_n)$, $CV_{\mathcal{M}}(P(a_1, \ldots, a_n)) = 1$, if $\mathcal{G}^P(a_1, \ldots, a_n)$ holds, and $CV_{\mathcal{M}}(P(a_1, \ldots, a_n)) = 0$ otherwise;
- $CV_{\mathcal{M}}(\neg\alpha) = 1 - CV_{\mathcal{M}}(\alpha)$;
- $CV_{\mathcal{M}}(\alpha \vee \beta) = CV_{\mathcal{M}}(\alpha \bigtriangledown \beta) = max\{(CV_{\mathcal{M}}(\alpha), CV_{\mathcal{M}}(\beta)\}$;
- $CV_{\mathcal{M}}(\alpha \wedge \beta) = CV_{\mathcal{M}}(\alpha \bigtriangleup \beta) = min\{CV_{\mathcal{M}}(\alpha), CV_{\mathcal{M}}(\beta)\}$;
- $CV_{\mathcal{M}}(\exists x \alpha(x)) = max\{CV_{\mathcal{M}}(\alpha(a)) : \ a \in \mathcal{D}\}$;
- $CV_{\mathcal{M}}(\forall x \alpha(x)) = min\{CV_{\mathcal{M}}(\alpha(a)) : \ a \in \mathcal{D}\}$.

Definition 4.1 Let $\mathcal{M}$ be a model for L. The *classical model* $\mathcal{M}^{cl} = \langle \mathcal{D}, \mathcal{G} \rangle$ *induced by* $\mathcal{M}$ is defined as follows:

- $\mathcal{D} = \mathcal{D}_{\mathcal{M}}$;
- for any n-ary predicate letter P and any $a_1, \ldots, a_n \in \mathcal{D}$, we have

$$\mathcal{G}^P(a_1, \ldots, a_n) \Leftrightarrow (N_{\mathcal{M}}(P(a_1, \ldots, a_n)) \text{ is solvable}).$$

It is easily seen that if $\mathcal{M}$ is an elementary model, $\mathcal{M}^{cl}$ is the classical model induced by $\mathcal{M}$ and α is a sliteral (with parameters in $\mathcal{D}_{\mathcal{M}}$), then $CV_{\mathcal{M}^{cl}}(\alpha) = \ell_{\mathcal{M}}(P(\alpha))$.

Theorem 4.2 *Let $\mathcal{M}$ be a model for L, $\mathcal{M}^{cl}$ be the classical model induced by $\mathcal{M}$ and ϕ be a sentence of L with parameters in $\mathcal{D}_{\mathcal{M}}$. Then $CV_{\mathcal{M}^{cl}}(\phi) = 1$ iff the game $N_{\mathcal{M}}(\phi)$ is solvable.*

PROOF. ($\Rightarrow$:) Suppose $CV_{\mathcal{M}^{cl}}(\phi) = 1$ and show, by induction on the complexity of ϕ, that $N_{\mathcal{M}}(\phi)$ is solvable.

Case 1: ϕ is an atom $P(a_1, \ldots, a_n)$. $CV_{\mathcal{M}^{cl}}(\phi) = 1$ then means that $\mathcal{G}^P_{\mathcal{M}^{cl}}(a_1, \ldots, a_n)$ holds, which, by 4.1, means that $N_{\mathcal{M}}(\phi)$ is solvable.

Case 2: $\phi = \neg\alpha$, where α is an atom. Then $CV_{\mathcal{M}^{cl}}(\alpha) = 0 \neq 1$ and, by the induction hypothesis, $N_{\mathcal{M}}(\alpha)$ is not solvable. Then, by 2.16, $N_{\mathcal{M}}(\alpha)$ has an antisolution g. Let then f be such a function that for any sentence γ, $f(\gamma) = \neg g(\neg\gamma)$ (we may suppose that g is defined for every sentence and its value is always a sentence). Now, it is easy to verify that f is a solution to $N_{\mathcal{M}}(\neg\alpha)$, for, if $\gamma_1, \ldots, \gamma_k$ is a lost $N_{\mathcal{M}}(\neg\alpha)$-play with Proponent's strategy f, then $\neg\gamma_1, \ldots, \neg\gamma_k$ is a won (by Proponent) $N_{\mathcal{M}}(\alpha)$-play with Opponent's strategy g, which is impossible because g is an antisolution to $N_{\mathcal{M}}(\alpha)$.

Case 3: $\phi = \alpha_1 \vee \alpha_2$. Then $max\{CV_{\mathcal{M}^{cl}}(\alpha_1), CV_{\mathcal{M}^{cl}}(\alpha_2)\} = 1$. We may suppose that $CV_{\mathcal{M}^{cl}}(\alpha_1) = 1$. Then, by the induction hypothesis, there is a solution g to $N_{\mathcal{M}}(\alpha_1)$. Let $f(\phi) = \alpha_1$ and for any sentence $\gamma \neq \phi$, $f(\gamma) = g(\gamma)$. We claim that f is a solution to $N_{\mathcal{M}}(\phi)$. Indeed, suppose there is a lost $N_{\mathcal{M}}(\phi)$-play with Proponent's strategy f. It will look like $\langle \phi, \alpha_1, \vec{\gamma} \rangle$ for some (possibly empty) sequence $\vec{\gamma}$ of sentences. Observe that then $\langle \alpha_1, \vec{\gamma} \rangle$ is a lost $N_{\mathcal{M}}(\alpha_1)$-play with Proponent's strategy g, which is impossible because, according to our assumption, g is a solution to $N_{\mathcal{M}}(\alpha_1)$.

Case 4: $\phi = \alpha_1 \wedge \alpha_2$. Then $CV_{\mathcal{M}^{cl}}(\alpha_1) = CV_{\mathcal{M}^{cl}}(\alpha_2) = 1$ and, by the induction hypothesis, both $N_{\mathcal{M}}(\alpha_1)$ and $N_{\mathcal{M}}(\alpha_2)$ are solvable. Now, since α_1 and α_2 are the only developments of ϕ, it follows by Lemma 2.15 that $N_{\mathcal{M}}(\phi)$ is solvable.

Case 5: $\phi = \alpha_1 \bigtriangledown \alpha_2$. Then $max\{CV_{\mathcal{M}^{cl}}(\alpha_1), CV_{\mathcal{M}^{cl}}(\alpha_2)\} = 1$. We may suppose that $CV_{\mathcal{M}^{cl}}(\alpha_1) = 1$. By the induction hypothesis, there is a solution g to $N_{\mathcal{M}}(\alpha_1)$.

Let f be such a function that for any sentence $\beta_1 \bigtriangledown \beta_2$, $f(\beta_1 \bigtriangledown \beta_2) = g(\beta_1) \bigtriangledown \beta_2$. Intuitively, to play an $N_{\mathcal{M}}(\alpha_1 \bigtriangledown \alpha_2)$-play with the strategy f, for Proponent, means that he plays, using strategy g, only in the left component of the multiplicative disjunction and does nothing in the right component.

Suppose there is a lost $N_{\mathcal{M}}(\phi)$-play

$$\langle \beta_1 \nabla \gamma_1, \ldots, \beta_n \nabla \gamma_n \rangle$$

(where $\beta_1 \nabla \gamma_1 = \alpha_1 \nabla \alpha_2 = \phi$) with Proponent's strategy f. Let $k_1 < \ldots < k_m$ be all the numbers k in the interval $1 < k \leq n$ such that $\beta_{k-1} \neq \beta_k$. Intuitively, $k \in \{k_1, \ldots, k_m\}$ means that the position $\beta_k \nabla \gamma_k$ has appeared as a result of moving in the left component of the multiplicative disjunction; all the other positions appear as a result of Opponent's move in the right component and they are not interesting for us. It is now easy to see that $\langle \alpha_1, \beta_{k_1}, \ldots, \beta_{k_m} \rangle$ is a lost $N_{\mathcal{M}}(\alpha_1)$-play with Proponent's strategy g, which is impossible because g was Proponent's winning strategy for $N_{\mathcal{M}}(\alpha_1)$. Thus no $N_{\mathcal{M}}(\phi)$-play with Proponent's strategy f can be lost, f is a solution to $N_{\mathcal{M}}(\phi)$.

Case 6: $\phi = \alpha_1 \bigtriangleup \alpha_2$. Then $CV_{\mathcal{M}}(\alpha_1) = CV_{\mathcal{M}}(\alpha_2) = 1$. By the induction hypothesis, there are solutions g_1 and g_2 to $N_{\mathcal{M}}(\alpha_1)$ and $N_{\mathcal{M}}(\alpha_2)$, respectively.

Let f be such a function that for any sentence $\beta_1 \bigtriangleup \beta_2$,

- $f(\beta_1 \bigtriangleup \beta_2) = g_1(\beta_1) \bigtriangleup \beta_2$, if $l_{\mathcal{M}}(\beta_1) = 0$;
- $f(\beta_1 \bigtriangleup \beta_2) = \beta_1 \bigtriangleup g_2(\beta_2)$, if $l_{\mathcal{M}}(\beta_1) = 1$.

Intuition: For Proponent, to follow strategy f in an $N_{\mathcal{M}}(\alpha_1 \bigtriangleup \alpha_2)$-play means to use the strategy g_1 in the first component of the multiplicative conjunction and the strategy g_2 in the second component.

Suppose there is a lost $N_{\mathcal{M}}(\phi)$-play

$$\langle \beta_1 \nabla \gamma_1, \ldots, \beta_n \nabla \gamma_n \rangle$$

(where $\beta_1 \bigtriangleup \gamma_1 = \alpha_1 \bigtriangleup \alpha_2 = \phi$) with Proponent's strategy f. As this play is lost, the label of its last position $\beta_n \bigtriangleup \gamma_n$ is 0, i.e. one of the positions β_n, γ_n has the label 0. We may suppose that $l_{\mathcal{M}}(\beta_n) = 0$. Let then $k_1 < \ldots < k_m$ be all the numbers k in the interval $1 < k \leq n$ such that $\beta_{k-1} \neq \beta_k$. Thus, $k \in \{k_1, \ldots, k_m\}$ means that the position $\beta_k \nabla \gamma_k$ has appeared as a result of moving in the left component of the multiplicative conjunction. Now it remains to verify, which can be easily done, that $\langle \alpha_1, \beta_{k_1}, \ldots, \beta_{k_m} \rangle$ is a lost $N_{\mathcal{M}}(\alpha_1)$-play with Proponent's strategy g_1, which is impossible because g_1 was Proponent's winning strategy for $N_{\mathcal{M}}(\alpha_1)$. This contradiction proves that f is a solution to $N_{\mathcal{M}}(\phi)$.

Case 7: $\phi = \exists x \alpha(x)$. Similar to the case 3.

Case 8: $\phi = \forall x \alpha(x)$. Similar to the case 4.

($\Leftarrow$:) We have just shown that if $CV_{\mathcal{M}^{cl}}(\phi) = 1$, then there is a solution to $N_{\mathcal{M}}(\phi)$. In a symmetric way we can show that if $CV_{\mathcal{M}^{cl}}(\phi) = 0$ (i.e., if $CV_{\mathcal{M}^{cl}}(\phi) \neq 1$), then there is an antisolution to $N_{\mathcal{M}}(\phi)$, which rules out solvability of $N_{\mathcal{M}}(\phi)$. ♣

Thus, identifying models with the classical models induced by them, solvability and truth appear the same. The reader can easily verify (using 4.2) that, e.g., the following holds:

Fact 4.3 *An arithmetical sentence α is true in the classical sense (in the classical standard model of arithmetic) if and only if $N_{\mathcal{S}}(\alpha)$ is solvable.*

Therefore it is safe and natural to use the word "true" for "solvable", as this usage is established in the first clause of the following definition:

Definition 4.4 Let α be a sentence of L and $\mathcal{M}$ be a model for L.

- α is said to be *true* in $\mathcal{M}$, if the game $N_{\mathcal{M}}(\alpha)$ is solvable.
- α is said to be *effectively true* in $\mathcal{M}$, if the game $N_{\mathcal{M}}(\alpha)$ is effectively solvable.

5 Tautologies and effective tautologies

Having different notions of truth, we can define different notions of tautology:[6]

Definition 5.1 Let α be a sentence of L.

- α is said to be a *tautology*, if α is true in every model for L.
- α is said to be an *effective tautology*, if α is effectively true in every model for L.

Theorem 5.4 below establishes that the usage of the traditional word "tautology" here is safe, for the set of tautologies in our sense coincides with the set of tautologies in the classical sense. In the third clause of that theorem is used the notion of arithmetical instance of a formula of L, which, roughly, means the result of substituting predicate letters of the formula by arithmetical formulas of the same arity. Here is a more precise definition:

Definition 5.2 An arithmetical *translation* from a language L is a function τ defined for some (not necessarily proper) subset S of the set of predicate letters of L such that τ assigns to each n-ary predicate letter $P \in S$ an arithmetical formula $\tau P = \phi(x_1, \ldots, x_n)$ (which may also contain parameters) with exactly n free variables.

We say that a translation τ is *good* for a formula ϕ of L (ϕ may contain natural numbers as parameters), if τ is defined for all predicate letters occuring in ϕ and, for any such letter P, τP does not contain quantifiers binding individual variables occuring in ϕ.

[6]Many authors use "tautology" to refer to valid formulas of propositional logic only, but for us $\forall x P(x) \rightarrow \exists x P(x)$ is a tautology, too.

"Translation for ϕ" means translation which is good for ϕ.

If τ is good for ϕ, we define the formula ϕ^τ by the following induction on subformulas of ϕ:

- for an atomic $\alpha = P(t_1, \ldots, t_n)$, where each t_i is either a variable or a parameter and where $\tau P = \beta(x_1, \ldots, x_n)$, we have $\alpha^\tau = \beta(t_1, \ldots, t_n)$;
- $(\neg\phi)^\tau = \neg(\phi^\tau)$, $(\phi \circ \psi)^\tau = \phi^\tau \circ \psi^\tau$, where $\circ \in \{\triangle, \triangledown, \vee, \wedge\}$, and $(Qx\phi)^\tau = Qx(\phi^\tau)$, where $Q \in \{\exists, \forall\}$.

An *arithmetical instance* of ϕ is ϕ^τ for some translation τ for ϕ.

Let CL be the classical predicate logic in language L, where the two sorts of disjunction and the two sorts of conjunction are understood as synonyms.

Remark 5.3 By a straightforward induction on the complexity of a formula β one can show that if a translation τ is good for β, then β and β^τ have exactly the same free variables.

Theorem 5.4 *For any sentence ϕ of L, the following are equivalent:*

- **(i)** $\phi \in CL$;
- **(ii)** *ϕ is a tautology;*
- **(iii)** *any arithmetical instance of ϕ is true in the standard model of arithmetic.*

PROOF. (i)⇒(ii): Suppose ϕ is not a tautology, i.e. ϕ is not true in some model $\mathcal{M}$. Then, according to 4.2, the classical value of ϕ is 0 in the classical model $\mathcal{M}^{cl}$ induced by $\mathcal{M}$, whence, by Gödel's completeness theorem for CL, $\phi \notin CL$.

(ii)⇒(iii): This immediately follows from Lemma 9.2(a), proved later in Section 9.

(iii)⇒(i): It is a well known fact that if $\phi \notin CL$, then there is an arithmetical instance ϕ^* of ϕ whose classical value in the classical standard model of arithmetic is 0. This fact can be easily seen, say, by an analysis of Henkin's proof of Gödel's completeness theorem for CL. And this, by 4.3, means nothing but that ϕ^* is not true (in our sense) in the standard model of arithmetic. ♣

The following Theorem 5.5, which is an analog of Theorem 5.4, is the main result of the present work, and most of the rest of the paper is devoted to its proof. The logic ET, mentioned below, is defined and shown to be decidable in the next section.

Theorem 5.5 *For any sentence ϕ of L, the following are equivalent:*

- **(i)** $\phi \in ET$;

- **(ii)** *ϕ is an effective tautology;*
- **(iii)** *any arithmetical instance of ϕ is effectively true in the standard model of arithmetic.*

6 Logic ET: syntactic description and decidability

In this section and throughout the rest of the paper, if not stated otherwise, "parameter" will always mean natural number and "sentence" or "sliteral" (see 3.1.11) will mean sentence or sliteral of L with parameters in NAT, the set of natural numbers.

Terminology and notation 6.1

1. When speaking about a subformula (subsentence, literal) of a formula, we are often interested in a concrete occurrence of this subformula rather than the subformula as such (which may have several occurrences). Classical logic does not care very much about distinction between subformulas and their occurrences, but we do. In order to stress that we mean a concrete occurrence, we shall use the words *osubformula, osubsentence, osliteral* ("o" for "occurrence"). E.g., if α is the first osliteral $0=0$ of the formula $0=0 \bigtriangledown 0=0$, then the result of substituting in the latter α by β is $\beta \bigtriangledown 0=0$; however, if α is the *sliteral* $0=0$, then such a result is $\beta \bigtriangledown \beta$.

2. A *surface osubsentence* of a sentence α is an osubsentence γ which is not in the scope of $\neg$, $\vee$, $\wedge$, $\exists$ or $\forall$. In this case we also say that γ has a *surface occurrence* in α.

And we say that a sentence γ has a *weak surface occurrence* in α, if γ is not in the scope of $\vee$, $\wedge$, $\exists$ or $\forall$ (but it may be in the scope of $\neg$). Since only atoms can be in the scope of $\neg$, any nonatomic sentence has a surface occurrence in α if and only if it has a weak surface occurrence in α.

3. A *multiplicative atom*, or *multiplicatively atomic sentence* is a sentence which is either a sliteral or has one of the forms $\alpha \vee \beta, \alpha \wedge \beta, \exists x\alpha, \forall x\alpha$. In other words, multiplicatively atomic is a sentence which is the only surface osubsentence of itself.

4. Every formula is a *multiplicative* ($\bigtriangledown, \bigtriangleup$-) *combination* of its surface osubformulas. E.g., the formula $(\alpha \bigtriangleup \beta)\bigtriangledown\alpha$ is the combination "$(-_1 \bigtriangleup -_2)\bigtriangledown -_3$" of α, β and α, or the combination "$(-_2 \bigtriangleup -_1)\bigtriangledown -_3$" of β, α and α, or the combination "$-_1 \bigtriangledown -_2$" of $\alpha \bigtriangleup \beta$ and α, or the combination "$-_1$" of $(\alpha \bigtriangleup \beta)\bigtriangledown\alpha$. We shall usually use capital Latin letters for multiplicative combinations. Say, if A is "$(-_2 \bigtriangleup -_1)\bigtriangledown -_3$", then $A(\beta,\alpha,\alpha)$ means $(\alpha \bigtriangleup \beta)\bigtriangledown\alpha$ and $A(\alpha,\beta,\alpha)$ means $(\beta \bigtriangleup \alpha)\bigtriangledown\alpha$. By using the sign "!" in an expression like $A!(\alpha_1,\ldots,\alpha_n)$, we shall indicate that each α_i is a multiplicative atom. Thus, the sentence $A!(\alpha_1,\ldots,\alpha_n)$ contains exactly n multiplicatively atomic surface osubsentences,

whereas $A(\alpha_1, \ldots, \alpha_n)$ may contain more than n multiplicatively atomic osubsentences.

5. A *hypersentence* is a sentence ϕ together with a (possibly empty) set of disjoint pairs (α_0, α_1) of opposite (recall 3.1.6) surface osliterals of ϕ; such pairs will be called *married couples* (of the hypersentence), and α_0 and α_1 are said to be *spouses* to each other. As these pairs are disjoint, every osliteral can have at most one spouse. If an osliteral has a spouse, it is said to be *married*; otherwise it is *single*.

6. A hypersentence is said to be *clean*, if there are no married couples in it. Every sentence α will, at the same time, be understood as the corresponding clean hypersentence denoted by the same letter α, and vice versa: every clean hypersentence α will be identified with the sentence α.

Remark 6.2 In order to relax terminology and notation, it is convenient to assume that the set of married couples is somehow graphically "built in" the hyperformula: say, spouses are connected with curved lines. Then we can freely use such terms as, " the result of replacing in a hypersentence α the osubsentence β by γ", or "the result of replacing in α the parameter a by the parameter b". True, this "result" may not always remain a hypersentence: say, if a married osliteral was replaced by another (different) sentence and its spouse was left unchanged, then the new "spouses" will not be opposite any more. However, there are at least two interesting us cases when replacement is safe:

1. When the replaced osubsentence does not contain a married osliteral;

2. When all occurrences of some parameter a in the hypersentence are replaced by another parameter b. Clearly in this case married couples containing a will remain opposite to each other, for a will be changed to b in both of them.

Definition 6.3 A *hyperlabeling* for a hypersentence α is a function l: {surface osubsentences of α} $\rightarrow \{0, 1\}$ such that, calling the value of l for a sentence β the *hyperlabel* of β, we have:

1. Spouses have different (opposite) hyperlabels;
2. Single osliterals have the hyperlabel 0;
3. $l(\alpha \vee \beta) = l(\exists x \alpha) = 0$;
4. $l(\alpha \wedge \beta) = l(\forall x \alpha) = 1$;
5. $l(\alpha \triangledown \beta) = max\{l(\alpha), l(\beta)\}$;
6. $l(\alpha \bigtriangleup \beta) = min\{l(\alpha), l(\beta)\}$.

Definition 6.4 A hypersentence α is said to be *1-like*, if for any hyperlabeling l, we have $l(\alpha) = 1$; otherwise α is said to be *0-like*.

Of course the question whether a hypersentence is 0- or 1-like is decidable.

Definition 6.5 A hypersentence β will be said to be a *marriage-extension* of a hypersentence α, if α and β are identical as sentences and the set of married couples of α is a proper subset of that of β.

Definition 6.6 Let α and β be hypersentences.

1. We say that β is a *1-hyperdevelopment* of α iff one of the following holds:

   - a) β is the result of replacing in α some surface osubsentence $\gamma \wedge \delta$ by γ or δ, or
   - b) β is the result of replacing in α some surface osubsentence $\forall x\gamma(x)$ by $\gamma(a)$ for some parameter a.

   To get the definition of *strict 1-hyperdevelopment*, we add to the clause (b) the condition that a is the smallest parameter not occuring in α.

2. We say that β is a *0-hyperdevelopment* of α iff one of the following holds:

   - a) β is the result of replacing in α some surface osubsentence $\gamma \vee \delta$ by γ or δ, or
   - b) β is the result of replacing in α some surface osubsentence $\exists x\gamma(x)$ by $\gamma(a)$ for some parameter a, or
   - c) β is a marriage-extension of α.

   To get the definition of *strict 0-hyperdevelopment*, we add to the clause (b) the condition that either a occurs in α, or α does not contain parameters and $a = 0$.

3. Finally, we say that β is (simply) a *hyperdevelopment* (resp. strict hyperdevelopment) of α, if β is a 1- or 0-hyperdevelopment (resp. strict 1- or 0-hyperdevelopment) of α.

Lemma 6.7
a) *There is no infinite chain $\alpha_0, \alpha_1, \ldots$ of hypersentences such that for any i, α_{i+1} is a hyperdevelopment of α_i; moreover, for any fixed α_0, there is a finite upper bound on the lengths of all such chains.*
b) *The set of all strict hyperdevelopments of any hypersentence α is finite.*

PROOF. (a): In fact the length of each such chain is $\leq m+(n/2)$, where m is the number of occurrences of $\vee, \wedge, \exists, \forall$ in α_0 and n is the number of occurrences of predicate letters in α_0 except occurrences in married osliterals. It suffices to observe that each transfer from α_i to α_{i+1} (where α_{i+1} is a hyperdevelopment of α_i) means "spending" in α_i either one of the occurrences of one of the operators $\vee, \wedge, \exists, \forall$ (this occurrence disappears in α_{i+1}) or a pair of single osliterals (which become married in α_{i+1}).
(b): Evident.

Definition 6.8 The *hypercomplexity* of a hypersentence α_1 is the length n of the longest chain $\alpha_1, \dots, \alpha_n$ of hypersentences such that for each i with $1 \leq i < n$, α_{i+1} is a hyperdevelopment of α_i.

So, if β is a hyperdevelopment of α, then the hypercomplexity of β is less than that of α.

Definition 6.9 We now define the set ET of hypersentences by stipulating that $\alpha \in ET$ iff one of the following holds:

1. α is 1-like and any 1-hyperdevelopment of it belongs to ET;
2. α is 0-like and there is a 0-hyperdevelopment of it which belongs to ET.

The above definition is correct, because it defines $\alpha \in ET$ in terms of $\beta \in ET$ for those βs whose hypercomplexity is less than that of α.

Notation 6.10 $\alpha[a/b]$ denotes the result of replacing in α every occurrence of the parameter a by b (see Remark 6.2).

Lemma 6.11 *Suppose b is a parameter not occuring in a hypersentence α, and a is any parameter. Then $\alpha \in ET$ iff $\alpha[a/b] \in ET$.*

PROOF. Indeed, if b does not occur in α, then α and $\alpha[a/b]$ are congruent in the sense that the only difference between these two sentences is that the first uses a and the second b instead. Therefore there is no reason why one hypersentence should be in ET and the other not. ♣

Lemma 6.12 *For any sentence α and any parameters a and b, if $\alpha \in ET$, then $\alpha[a/b] \in ET$.*

PROOF. Assume $\alpha \in ET$. Let α be $A!(\gamma_0, \gamma_1, \dots, \gamma_n)$. Then $\alpha[a/b] = A!(\gamma_0[a/b], \gamma_1[a/b], \dots, \gamma_n[a/b])$.

First note that

$$\alpha \textit{ is 0-like iff } \alpha[a/b] \textit{ is 0-like.} \tag{2}$$

Indeed, suppose α is 0-like, i.e. for some hyperlabeling l for α, $l(\alpha) = 0$. Let l' be the hyperlabeling for $\alpha[a/b]$ such that for any married osliteral $\gamma_i[a/b]$ of $\alpha[a/b]$, $l'(\gamma_i[a/b]) = l(\gamma_i)$. Clearly $l(\alpha) = l'(\alpha[a/b])$ $(= 0)$ and, consequently, $\alpha[a/b]$ is 0-like. And in a similar way we can show that if $\alpha[a/b]$ is 0-like, then so is α.

To prove the lemma, we proceed by induction on the hypercomplexity of α.

Case 1: α is 0-like. Then there is a 0-hyperdevelopment $\beta \in ET$ of α. By the induction hypothesis, $\beta[a/b] \in ET$. Since, by (2), $\alpha[a/b]$ is 0-like, it is enough to show that $\beta[a/b]$ is a 0-hyperdevelopment of $\alpha[a/b]$.

According to Definition 6.6.2, the fact that β is a 0-hyperdevelopment of α means that one of the following three subcases takes place:

Subcase 1: β is the result of replacing in α some γ_i of the form $\psi_0 \vee \psi_1$ by ψ_j $(j = 0, 1)$. We may suppose that $i, j = 0$. So, we have

$$\alpha = A!(\psi_0 \vee \psi_1, \gamma_1, \ldots, \gamma_n)$$

and

$$\beta = A!(\psi_0, \gamma_1, \ldots, \gamma_n).$$

Therefore,

$$\alpha[a/b] = A!(\psi_0[a/b] \vee \psi_1[a/b], \gamma_1[a/b], \ldots, \gamma_n[a/b])$$

and

$$\beta[a/b] = A!(\psi_0[a/b], \gamma_1[a/b], \ldots, \gamma_n[a/b]).$$

As we see, $\beta[a/b]$ is then a 0-hyperdevelopment of $\alpha[a/b]$.

Subcase 2: β is the result of replacing in α some γ_i of the form $\exists x \psi(x)$ by $\psi(c)$ for some parameter c. We may suppose that $i = 0$. So, we have

$$\alpha = A!(\exists x \psi(x), \gamma_1, \ldots, \gamma_n)$$

and

$$\beta = A!(\psi(c), \gamma_1, \ldots, \gamma_n).$$

Denote the formula $\psi(x)[a/b]$ by $\phi(x)$. Then

$$\alpha[a/b] = A!(\exists x \phi(x), \gamma_1[a/b], \ldots, \gamma_n[a/b])$$

and, as it is easy to see,

$$\beta[a/b] = A!(\phi(d), \gamma_1[a/b], \ldots, \gamma_n[a/b]),$$

where $d = c$, if $c \neq a$, and $d = b$, if $c = a$. In either case, $\beta[a/b]$ is a 0-hyperdevelopment of $\alpha[a/b]$.

Subcase 3: β is a marriage-extension of α. It is obvious that $\beta[a/b]$ is then a marriage-extension of $\alpha[a/b]$ and thus, $\beta[a/b]$ is a 0-hyperdevelopment of $\alpha[a/b]$.

Case 2: α is 1-like. Then, by (2) (as being 1-like means nothing but not being 0-like), $\alpha[a/b]$ is 1-like. Consider any 1-hyperdevelopment δ of $\alpha[a/b]$. We need to show that $\delta \in ET$.

Subcase 1: δ is the result of replacing in $\alpha[a/b]$ some $\gamma_i[a/b]$ of the form $\psi_0 \wedge \psi_1$ by ψ_j. We may suppose that $i, j = 0$. It is clear that $\gamma_0 = \psi'_0 \wedge \psi'_1$ for some ψ'_0, ψ'_1 such that $\psi_0 = \psi'_0[a/b]$ and $\psi_1 = \psi'_1[a/b]$. Let

$$\delta' = [A!(\psi'_0, \gamma_1, \ldots, \gamma_n).$$

Then δ' is a 1-hyperdevelopment of α. Notice that $\delta = \delta'[a/b]$. Since $\alpha \in ET$, $\delta' \in ET$, whence, by the induction hypothesis, $\delta \in ET$.

Subcase 2: δ is the result of replacing in $\alpha[a/b]$ some $\gamma_i[a/b]$ of the form $\forall x\psi(x)$ by $\psi(c)$ for some parameter c. We may suppose that $i = 0$. In view of Lemma 6.11 (as $\psi(x)$ does not contain a), we may suppose that $c \neq a$. Then $\gamma_0 = \forall x\psi'(x)$ for some ψ' such that $\psi(x) = \psi'_i[a/b](x)$. Let

$$\delta' = A!(\psi'(c), \gamma_1, \ldots, \gamma_n).$$

Notice that δ' is a 1-hyperdevelopment of α and $\delta = \delta'[a/b]$. Since $\alpha \in ET$, $\delta' \in ET$, whence, by the induction hypothesis, $\delta \in ET$.

Lemma 6.13 (Another definition of ET) *For any hypersentence α, we have $\alpha \in ET$ iff:*

a) α is 1-like and any strict 1-hyperdevelopment of it belongs to ET, or

b) α is 0-like and there is a strict 0-hyperdevelopment of it which belongs to ET.

PROOF. (a): Suppose α is 1-like. If $\alpha \in ET$, then any 1-hyperdevelopment of α is in ET and, — as a strict 1-hyperdevelopment is at the same time a 1-hyperdevelopment, — any strict 1-hyperdevelopment of α is in ET. Assume now $\alpha \notin ET$. Then there is a 1-hyperdevelopment γ of α with $\gamma \notin ET$. If γ is at the same time a strict hyperdevelopment of α we are done. Otherwise, γ is the result of replacing in α an osubsentence $\forall x\delta(x)$ by $\delta(a)$, for some parameter a. Let b be the smallest parameter not occuring in α, and let γ' be the result of replacing in α the osubsentence $\forall x\delta(x)$ by $\delta(b)$. Note that γ' is a strict 1-hyperdevelopment of α. To show that $\gamma' \notin ET$, notice that $\gamma = \gamma'[b/a]$ (the fact that b does not occur in α and that therefore b occurs only in the osubsentence $\delta(b)$ of the sentence γ is essential here), whence, as $\gamma \notin ET$, 6.12 implies that $\gamma' \notin ET$.

(b): Suppose α is 0-like. If $\alpha \notin ET$, then no 0-hyperdevelopment of α is in ET and therefore there is no strict 0-hyperdevelopment of α in ET. Assume now $\alpha \in ET$. Then there is a 0-hyperdevelopment β of α with $\beta \in ET$. If β is not at the same time a strict 0-hyperdevelopment of α, then β is the result of replacing in α an osubsentence $\exists x\delta(x)$ by $\delta(a)$ for a not occuring in α. Let b be any parameter occuring in α or, — if α does not contain parameters, let b be 0. As it is easily seen, $\beta[a/b]$ is the result of replacing in α the osubsentence $\exists x\delta(x)$ by $\delta(b)$, which means — in view of our assumptions about b — that $\beta[a/b]$ is a 0-hyperdevelopment of α. And, since $\beta \in ET$, 6.12 implies that $\beta[a/b] \in ET$. ♣

Theorem 6.14 *ET is decidable. In fact, it is decidable in polynomial space.*

PROOF. The decidability of ET immediately follows from Lemmas 6.13 and 6.7, and a straightforward analysis of the appropriate definitions and proofs convinces us that a reasonable decision algorithm needs at most polynomial space. ♣

Lemma 6.15

1. *if $\alpha \in ET$ and β is a 1-hyperdevelopment of α, then $\beta \in ET$;*
2. *if $\alpha \notin ET$ and β is a 0-hyperdevelopment of α, then $\beta \notin ET$.*

PROOF. We consider only the first clause, the case when β is a 1-hyperdevelopment of α on the basis of 6.6.1a. The other cases (of both clauses) are handled in a quite similar way. We use induction on the hypercomplexity of α.

Suppose $\alpha = A(\delta_1 \wedge \delta_2, \vec{\xi}) \in ET$ and $\beta = A(\delta_i, \vec{\xi})$ $(i \in \{1, 2\})$.

If α is 1-like, then, by Definition 6.9, $\beta \in ET$.

Suppose now α is 0-like. A little analysis of Definition 6.3 (together with 6.4) convinces us that then $A(\delta_i, \vec{\xi})$ is 0-like. According to 6.9.2, there is a 0-hyperdevelopment of α which belongs to ET. Observe that this 0-hyperdevelopment has the form $A(\delta_1 \wedge \delta_2, \vec{\xi'})$ and $A(\delta_i, \vec{\xi'})$ is a 0-hyperdevelopment of $A(\delta_i, \vec{\xi})$. But, by the induction hypothesis (as $A(\delta_1 \wedge \delta_2, \vec{\xi'}) \in ET$), we have $A(\delta_i, \vec{\xi'}) \in ET$, which means that $A(\delta_i, \vec{\xi}) \in ET$. ♣

7 Relaxed linguistic games

As we require the domain of a model for L to be countable, we shall assume that the domain $\mathcal{D}_{\mathcal{M}}$ of any model we consider is NAT. True, a finite model cannot be isomorphic to a model with domain NAT. However, any finite model can be viewed as a countably infinite model where we have infinitely many "copies" of one of the elements of the domain. Therefore, the assumption that the domain of every model is NAT in fact does not lead to any loss of generality.

As we agreed in the previous section, by a "sentence" we always mean a sentence with parameters in NAT.

Throughout this section we assume that a model $\mathcal{M}$ for L is fixed.

I suggest to the reader to recall Definition 3.4 and our terminological convention according to which we can identify a sentence α with the game $N_{\mathcal{M}}(\alpha)$.

Definition 7.1 We define $N^{\circ}_{\mathcal{M}}$ to be the net $\langle W_{\mathcal{M}}, l_{\mathcal{M}}, R^{\circ}_{\mathcal{M}} \rangle$ of games, where $W_{\mathcal{M}}$ and $l_{\mathcal{M}}$ are defined as in 3.4 and for any $\alpha, \beta \in W$, we have $\alpha R^{\circ}_{\mathcal{M}} \beta$ iff β is the result of replacing in α a surface multiplicatively atomic osubsentence δ, which has the same label as α, by a sentence δ' such that $\delta R_{\mathcal{M}} \delta'$.

Terminology and notation 7.2 In order to distinguish the two versions $N_{\mathcal{M}}$ and $N^{\circ}_{\mathcal{M}}$ of the net of games induced by $\mathcal{M}$, from now on we call the former the *regular* version and the latter the *relaxed* version. We also apply the adjectives "regular" and "relaxed", respectively, to the development relations $\mathcal{R}_{\mathcal{M}}$ and $\mathcal{R}^{\circ}_{\mathcal{M}}$, the games $N_{\mathcal{M}}(\alpha)$ and $N^{\circ}_{\mathcal{M}}(\alpha)$ or solutions to them, etc. However, we may omit these adjectives in cases when it does not matter which version we deal with, when the version can be seen from the context, or when we consider a variable version.

Of course every regular development of α is, at the same time, a relaxed development, but vice versa does not generally hold. E.g.,

$$\exists x(x=x) \bigtriangleup (\beta(a) \bigtriangledown \forall y \neg\beta(y)) \tag{3}$$

is a relaxed, but not regular, development of

$$\exists x(x=x) \bigtriangleup (\exists y \beta(y) \bigtriangledown \forall y \neg\beta(y)), \tag{4}$$

whatever the model is.

Intuitively, the difference between the regular and the relaxed versions of linguistic games is that in relaxed games players may make "ahead-of-time", or "impatient" moves. The main task of this section is to establish that such impatient moves of one player cannot affect the chance of the other player to win. Moreover: the other player may even benefit by the impatience of his adversary. Going back to the above example, in the position (4) it was Proponent's move because of the 0-labeled multiplicative conjunct $\exists x(x=x)$; the other conjunct was 1-labeled and he did not have (and was not allowed in the regular case) to move in it. However, by going to the position (3), Proponent has made an impatient move in the second conjunct of (4). This did not release him from the duty to move in the first conjunct (one can show that a properly impatient move never changes the label), so he still has to replace in (3) the osubsentence $\exists x(x=x)$ by $(b=b)$ for some b, which shows that Proponent did not benefit by postponing this regular move; on the other hand, he missed the possibility to use, in the second conjunct of (4), the strategy described in Introduction that enabled a bad chess player to defeat the world champion in the game $C \bigtriangledown \neg C$. It would be more clever of Proponent to go from the position (4) to

$$(b=b) \bigtriangleup (\exists y \beta(y) \bigtriangledown \forall y \neg\beta(y)),$$

then wait until Opponent makes his move in $\forall y \neg\beta(y)$, and only after that make a move in $\exists y \beta(y)$, choosing the same substitution for y as Opponent will have chosen.

Definition 7.3 Let $e \in \{0,1\}$. An *e-trace* from a sentence α to a sentence β is a legal $N^{\circ}_{\mathcal{M}}$-sequence $\gamma_1, \ldots, \gamma_n$ with $n \geq 1$ such that $\gamma_1 = \alpha$, $\gamma_n = \beta$ and for every $1 \leq i < n$ (if $n > 1$), $l(\gamma_i) = e$; such a trace is said to be *trivial*, if $n = 1$; otherwise the trace is *nontrivial*.

An *e-tracing* for a sentence β is a function t defined on some (sub)set $\{\beta_1, \ldots, \beta_n\}$ of surface osubsentences of β which assigns to each β_i an e-trace (from some sentence) to β_i. And an *e-traced sentence* (β, t) is a sentence β given together with an e-tracing t for it.

A trace will usually serve as a piece of information on the history of a play used by Proponent for making a successful next move.

Although Lemma 7.7 below is in good accordance with intuition, a rigorous proof of it takes quite a space, and an "impatient" reader prone to trust us can just memorize Lemma 7.7 and skip the rest of this section.

Definition 7.4 Let $e = 0$ or $e = 1$ and $\alpha = A!(\alpha_1, \ldots, \alpha_n)$. Then an *$e$-expansion* of α is a traced sentence (β, t), where β has the form $A(\beta_1, \ldots, \beta_n)$ and t is an e-tracing for β which assigns, to each β_i with $1 \leq i \leq n$, a (possibly trivial) e-trace from α_i to β_i.

Such an expansion is said to be *pulling*, if

- the label of α is e and
- there is $1 \leq i \leq n$ such that the trace $t\beta_i$ from α_i to β_i is nontrivial and, for the second term σ of this trace, the sentence

  $$A(\alpha_1, \ldots, \alpha_{i-1}, \sigma, \alpha_{i+1}, \ldots, \alpha_n)$$

  is a regular development of α.

Then the osubsentence β_i is said to be a *pulling* osubsentence of this expansion.

Intuitively, an e-expansion of α is a pair (β, t), where β is the result of a series of "superimpatient" e-Player's moves made in some surface components of α, and t is a record of the history of these moves ("superimpatient", because this player does not even care whether it is his move in the whole position or not). A pulling expansion contains a hint for a "patient" e-Player how to make a move in α which would be really legal in a regular play and which would take us closer to β (β "pulls" α towards itself). Namely, this move should be a repetition of the first move made by the superimpatient e-Player in a pulling component.

Although an expansion of α is a traced sentence, i.e. a sentence β together with a tracing, in some contexts we identify it with just β.

Lemma 7.5 Suppose (β, t) is an e-expansion of α and β is e-labeled. Then α is e-labeled, too.

PROOF. Let α, β and t be as in the definition of e-expansion. Notice that then for each $1 \leq i \leq n$, if β_i is e-labeled, then so is α_i. Then it follows easily by Definition 3.4 that if β is e-labeled, then so is α. ♣

Lemma 7.6 Suppose α and β have the labels e and $1 - e$, respectively, and (β, t) is an e-expansion of α. Then (β, t) is a pulling e-expansion of α.

PROOF. We consider the case $e = 0$; the case $e = 1$ is symmetric. So, assume the conditions of the lemma with $e = 0$. Let $\alpha = A!(\alpha_1, \ldots, \alpha_n)$ and $\beta = A(\beta_1, \ldots, \beta_n)$. We proceed by induction on the complexity of A, that is, the "multiplicative complexity" of α. First of all note that as each α_i is multiplicatively atomic, any relaxed development of α_i is, at the same time, a regular development of α_i. In view of this, the case when the sentence α is multiplicatively atomic, is straightforward.

Suppose $\alpha = \phi_1 \bigtriangledown \phi_2$. We may suppose that $\phi_1 = A_1!(\alpha_1, \ldots, \alpha_m)$ and $\phi_2 = A_2!(\alpha_{m+1}, \ldots, \alpha_n)$ for some $1 \leq m < n$. Then $\beta = \psi_1 \bigtriangledown \psi_2$, where $\psi_1 = A_1(\beta_1, \ldots, \beta_m)$ and $\psi_2 = A_2(\beta_{m+1}, \ldots, \beta_n)$. Let t_1 and t_2 be the restrictions of t to $\{\beta_1, \ldots, \beta_m\}$ and $\{\beta_{m+1}, \ldots, \beta_n\}$, respectively. Notice that (ψ_1, t_1) and (ψ_2, t_2) are 0-expansions of ϕ_1 and ϕ_2, respectively. As α is 0-labeled, both its multiplicative disjuncts are 0-labeled. And as β is 1-labeled, we may suppose that ψ_1 is 1-labeled. Thus, ψ_1 is 1-labeled and (ψ_1, t_1) is a 0-expansion of the 0-labeled ϕ_1. Then, by the induction hypothesis, this expansion is pulling, i.e. there is $1 \leq i \leq m$ such that the trace $t_1\beta_i$ (which $= t\beta_i$) from α_i to β_i is not trivial and for the second term σ of this trace, the sentence

$$A_1(\alpha_1, \ldots, \alpha_{i-1}, \sigma, \alpha_{i+1}, \ldots, \alpha_m)$$

is a regular development of ϕ; but then, by Definition 3.4, the sentence

$$A_1(\alpha_1, \ldots, \alpha_{i-1}, \sigma, \alpha_{i+1}, \ldots, \alpha_m) \bigtriangledown \phi_2,$$

i.e. the sentence

$$A(\alpha_1, \ldots, \alpha_{i-1}, \sigma, \alpha_{i+1}, \ldots, \alpha_n),$$

is a regular development of α. This means that (β, t) is a pulling 0-expansion of α.

The case $\alpha = \phi_1 \bigtriangleup \phi_2$ can be handled in a similar manner. ♣

Lemma 7.7 *For any sentence ϕ of L,*
a) $N_{\mathcal{M}}(\phi)$ is solvable iff $N^{\circ}_{\mathcal{M}}(\phi)$ is;
b) $N_{\mathcal{M}}(\phi)$ is effectively solvable iff $N^{\circ}_{\mathcal{M}}(\phi)$ is.

PROOF. We prove only the clause (b); the clause (a) is easier to prove.

($\Leftarrow$):

Assume h is an effective solution to $N^{\circ}_{\mathcal{M}}(\phi)$. By induction on the length of a ϕ-play we simultaneously define Proponent's effective history-sensitive strategy f and another effective function g which assigns to every initial segment $\vec{\xi} = \xi_1, \ldots, \xi_n$ of an $N_{\mathcal{M}}(\phi)$-play with this strategy of Proponent's a traced sentence $g\vec{\xi}$; and we verify, at each step, that the following conditions are satisfied:

Condition 1. $g\vec{\xi}$ is a 0-expansion of ξ_n.

Condition 2. h is a relaxed solution to $g\vec{\xi}$ (see the paragraph preceding 7.5).

Here we go. We define $g\langle\phi\rangle$ to be ϕ, where every multiplicatively atomic osubsentence has a trivial trace. Of course, both conditions 1 and 2 are satisfied (for $\langle\phi\rangle$ in the role of $\langle\vec{\xi}\rangle$).

Now, suppose $\vec{\xi} = \xi_1, \ldots, \xi_n$ is an initial segment of a ϕ-play with Proponent's strategy f, $g\vec{\xi}$ is defined and the conditions 1-2 are satisfied. Let $\xi_n = A!(\alpha_1, \ldots, \alpha_k)$ and $g\vec{\xi}$ be $A(\beta_1, \ldots, \beta_k)$ with relaxed 0-traces $tr_1, \ldots, tr_k$ from the osubsentences $\alpha_1, \ldots, \alpha_k$ to $\beta_1, \ldots, \beta_k$, respectively.

Case 1: ξ_n is 1-labeled. Then f does not have to be defined for $\vec{\xi}$ and we only need to define g for $\langle\vec{\xi}, \xi_{n+1}\rangle$, where ξ_{n+1} is an arbitrary regular development of ξ_n. A regular development is always, at the same time, a relaxed development, so we have

$$\xi_{n+1} = A(\alpha_1, \ldots, \alpha_{i-1}, \sigma, \alpha_{i+1}, \ldots, \alpha_k)$$

for some i, where σ is a development of the 1-labeled α_i. That is, ξ_{n+1} has the form

$$B!(\alpha_1, \ldots, \alpha_{i-1}, \sigma_1, \ldots, \sigma_m, \alpha_{i+1}, \ldots, \alpha_k),$$

where $\sigma_1, \ldots, \sigma_m$ are all surface multiplicatively atomic osubsentences of σ. Then we define

$$g\langle\vec{\xi}, \xi_{n+1}\rangle = B(\beta_1, \ldots, \beta_{i-1}, \sigma_1, \ldots, \sigma_m, \beta_{i+1}, \ldots, \beta_k).$$

We want to make this sentence a traced one which would be a 0-expansion of ξ_{n+1}. It is easily seen that this goal is achieved if we let each β_j have the same 0-trace as it had in $g\vec{\xi}$, and every σ_j have a trivial trace. So, the condition 1 is satisfied.

As there was a 0-trace from the 1-labeled α_i to β_i, this trace could only be a trivial one, i.e. $\alpha_i = \beta_i$. As ξ_n is 1-labeled, so is (by Lemma 7.5 and the condition 1) $g\vec{\xi}$, which easily implies that $g\langle\vec{\xi}, \xi_{n+1}\rangle$ is a relaxed development of $g\vec{\xi}$ and, as h is a relaxed solution to the latter, h must be a relaxed solution to $g\langle\vec{\xi}, \xi_{n+1}\rangle$ as well (see 2.9). Thus, the condition 2, too, is satisfied.

Case 2: ξ_n is 0-labeled. Then we need to define f for $\vec{\xi}$ and g for $\langle\vec{\xi}, f\vec{\xi}\rangle$. We use the notation $h^x(\eta)$, defined by $h^0(\eta) = \eta$ and $h^{x+1}(\eta) = h(h^x(\eta))$. Let p be the least number such that $h^p(g\vec{\xi})$ is (defined and) 1-labeled. Such a p exists, for, if the label of $g\vec{\xi}$, is 1, then $p = 0$ will do, and if this label is 0, then the existence of p follows from the fact that h is a relaxed solution to $g\vec{\xi}$. By the same reason,

$$h \text{ is a relaxed solution to } h^p(g\vec{\xi}). \tag{5}$$

We have:

$$(g\vec{\xi}=)\ \ h^0(g\vec{\xi}) \ = \ A(\delta_1^0, \ldots, \delta_k^0) \quad (= A(\beta_1, \ldots, \beta_k))$$

$$h^1(g\vec{\xi}) = A(\delta^1_1,\ldots,\delta^1_k)$$
$$\ldots$$
$$h^p(g\vec{\xi}) = A(\delta^p_1,\ldots,\delta^p_k)$$

For each $1 \leq i \leq k$, let $\vec{tr}'_i$ be the result of concatenating $\vec{tr}_i$ (see the paragraph preceding Case 1) with $\langle\delta^1_i,\ldots,\delta^p_i\rangle$ and then repeatedly deleting each term δ^j_i equal to its left neighbor in the sequence.

Now, let ω be the traced sentence $A(\delta^p_1,\ldots,\delta^p_k)$, with the trace $\vec{tr}'_i$ for each osubsentence δ^p_i. It is evident that ω, just like $g\vec{\xi}$, is a 0-expansion of ξ_n. And as ω is 1-labeled, Lemma 7.6 gives that it is a pulling 0-expansion of ξ_n. Let then δ^p_j be the leftmost pulling osubsentence of this expansion, and let σ be the second term of the trace $\vec{tr}'_j$. Then we define

$$f\vec{\xi} = A(\alpha_1,\ldots,\alpha_{j-1},\sigma,\alpha_{j+1},\ldots,\alpha_k).$$

Note that

$$\omega \textit{ is a 0-expansion of } f\vec{\xi}. \tag{6}$$

As the second term σ of a 0-trace from α_j is a relaxed development of α_j and as the latter is multiplicatively atomic, σ a regular development of it as well. This means that $f\vec{\xi}$ is a regular development of ξ_n.

And we define $g\langle\vec{\xi}, f\vec{\xi}\rangle$ to be the traced sentence ω. According to 5 and 6, the conditions 1 and 2 are then satisfied for $\langle\vec{\xi}, f\vec{\xi}\rangle$.

Thus, f, which evidently is effective, is defined for any initial segment $\vec{\xi}$ of a $N_{\mathcal{M}}(\phi)$-play with Proponent's strategy f, as soon as the last sentence ξ_n of $\vec{\xi}$ is 0-labeled, and the value of f is then a regular development of ξ_n. This means nothing but that f is an effective solution to $N_{\mathcal{M}}(\phi)$.

($\Rightarrow$):

Assume h is an effective solution to $N_{\mathcal{M}}(\phi)$. By induction on the length of a relaxed ϕ-play we simultaneously define Proponent's effective history-sensitive relaxed strategy f together with two other effective functions t and g, where g assigns to every initial segment $\vec{\xi} = \xi_1,\ldots,\xi_n$ of an $N^{\circ}_{\mathcal{M}}(\phi)$-play with this strategy of Proponent's a sentence $g\vec{\xi}$, and t, applied to $\vec{\xi}$, returns a tracing for ξ_n (makes the sentence ξ_n traced), so that the following conditions are satisfied:

Condition 1. $(\xi_n, t\vec{\xi})$ is a 1-expansion of $g\vec{\xi}$.

Condition 2. h is a regular solution to $g\vec{\xi}$.

Here we go. We define $g\langle\phi\rangle$ to be ϕ and $t\langle\phi\rangle$ to be the tracing which assigns the trivial trace to every surface osubsentence of ϕ. Evidently both conditions 1 and 2 are satisfied (for $\langle\phi\rangle$ in the role of $\langle\vec{\xi}\rangle$).

Now, suppose $\vec{\xi} = \xi_1, \ldots, \xi_n$ is an initial segment of a relaxed ϕ-play with Proponent's strategy f, g and t are defined for $\vec{\xi}$ and the conditions 1-2 are satisfied. Let $g\vec{\xi}$ be $A!(\beta_1, \ldots, \beta_k)$, ξ_n be $A(\alpha_1, \ldots, \alpha_k)$ and $t\vec{r}_1, \ldots, t\vec{r}_k$ be the 1-traces assigned to $\alpha_1, \ldots, \alpha_k$ by the tracing $t\vec{\xi}$.

Case 1: ξ_n is 1-labeled. Then f does not have to be defined for $\vec{\xi}$ and we only need to define g and t for $\langle\vec{\xi}, \xi_{n+1}\rangle$, where ξ_{n+1} is an arbitrary relaxed development of ξ_n.

The value of g we leave unchanged: $g\langle\vec{\xi}, \xi_{n+1}\rangle = g\vec{\xi}$.

We have

$$\xi_{n+1} = A(\alpha_1, \ldots, \alpha_{i-1}, \sigma, \alpha_{i+1}, \ldots, \alpha_k),$$

for some i, where σ is a relaxed development of the 1-labeled α_i. Then we define $t\langle\vec{\xi}, \xi_{n+1}\rangle$ as the tracing that leaves unchanged the trace $t\vec{r}_j$ for each α_j with $j \neq i$, and assigns the trace $\langle t\vec{r}_i, \sigma\rangle$ to σ. Conditions 1 and 2 evidently remain satisfied.

Case 2: ξ_n is 0-labeled. Then we need to define f for $\vec{\xi}$ and g and t for $\langle\vec{\xi}, f\vec{\xi}\rangle$. First we define the sequence $\omega_0, \ldots, \omega_p$ of sentences and the sequence $r_0, \ldots, r_p$ of tracings for ξ_n as follows:

- $\omega_0 = g\vec{\xi}$ and $r_0 = t\vec{\xi}$. Note that, by the condition 1, (ξ_n, r_0) is a 1-expansion of ω_0.

- Suppose now ω_m and r_m are defined, ω_m has the form $B!(\eta_1, \ldots, \eta_l)$ and (ξ_n, r_m), where ξ_n has the form $B(\zeta_1, \ldots, \zeta_l)$, is a 1-expansion of ω_m. Then:

  1. If ω_m has the label 0, then $p = m$.
  2. Suppose now ω_m has the label 1. Then, by Lemma 7.6, (ξ_n, r_m) is a pulling 1-expansion of ω_m, and let ζ_j be the leftmost pulling osubsentence of this expansion. Let

     $$\sigma = C!(\sigma_1, \ldots, \sigma_d)$$

     be the second term of the trace assigned by r_m to ζ_j. Then we define ω_{m+1} to be

     $$B(\eta_1, \ldots, \eta_{j-1}, \sigma, \eta_{j+1}, \ldots, \eta_l),$$

     whose !-form is

     $$D!(\eta_1, \ldots, \eta_{j-1}, \sigma_1, \ldots, \sigma_d, \eta_{j+1}, \ldots, \eta_l)$$

     for a corresponding D.

     The trace assigned by r_m to ζ_j looks like

     $$\langle\eta_j, C(\sigma_1^0, \ldots, \sigma_d^0), \ldots, C(\sigma_1^b, \ldots, \sigma_d^b)\rangle,$$

where $C(\sigma_1^0, \ldots, \sigma_d^0) = \sigma = C(\sigma_1, \ldots, \sigma_d)$ and $C(\sigma_1^b, \ldots, \sigma_d^b) = \zeta_j$. So, ξ_n is $D(\zeta_1, \ldots, \zeta_{j-1}, \sigma_1^b, \ldots, \sigma_d^b, \zeta_{j+1}, \ldots, \zeta_l$. Now, we define r_{m+1} as follows: to each ζ_i with $1 \leq i \leq l,\ \ i \neq j,\ \ r_{m+1}$ assigns the same trace as assigned by r_m, and to each σ_i $(1 \leq i \leq d)$, r_{m+1} assigns the trace which is the result of repeatedly deleting terms equal to their left neighbors in the sequence $\langle \sigma_0^i, \ldots, \sigma_b^i \rangle$.

It is not hard to verify that (ξ_n, r_{m+1}) is a 1-expansion of ω_{m+1}.

Notice that each ω_{m+1} is a regular development of ω_m (recall what pulling expansion means), and the number p, sooner or later, will be reached. ω_p is a regular rt-1-development (recall 2.2.2) of $\omega_0 = g\vec{\xi}$ and, as h is a regular solution to the latter (the condition 2), by 2.9, we have

$$h \textit{ is a regular solution to } \omega_p. \tag{7}$$

And, as we noted above,

$$(\xi_n, r_p) \textit{ is a 1-expansion of } \omega_p. \tag{8}$$

ω_p is 0-labeled and, as h is a regular solution to it, h must be defined for ω_p and its value must be a regular development of ω_p. That is, we have:

$$\omega_p = G!(\lambda_1, \ldots, \lambda_c)$$

for some G and $\vec{\lambda}$,

$$\xi_n = G(\mu_1, \ldots, \mu_c)$$

for some $\vec{\mu}$ and (as every development is a relaxed development)

$$h\omega_p = G(\lambda_1, \ldots, \lambda_{i-1}, \sigma, \lambda_{i+1}, \ldots, \lambda_c)$$

for some σ. We then define

$$f\vec{\xi} = G(\mu_1, \ldots, \mu_{i-1}, \sigma, \mu_{i+1}, \ldots, \mu_c).$$

Taking into account that λ_i is 0-labeled, which means that the 1-trace from λ_i to μ_i determined by r_p is a trivial one, i.e. $\lambda_i = \mu_i$, it is clear that

$$f\vec{\xi} \text{ (is defined and) is a relaxed development of } \xi_n. \tag{9}$$

We define

$$g\langle \vec{\xi}, f\vec{\xi} \rangle = h\omega_p.$$

And we define $t\langle \vec{\xi}, f\vec{\xi} \rangle$ to be the tracing for $f\vec{\xi}$ that assigns to each μ_j $(1 \leq j \leq c$, $j \neq i)$ the same 1-trace to μ_j as r_p, and assigns the trivial trace to σ. Then (8) implies that the condition 1 is satisfied for $\langle \vec{\xi}, f\vec{\xi} \rangle$, and that the condition 2 is satisfied follows from (7) by 2.9.

Thus, f, which evidently is effective, is defined for any initial segment $\vec{\xi}$ of a $N^{\circ}_{\mathcal{M}}(\psi)$-play with Proponent's strategy f, as soon as the last sentence ξ_n of $\vec{\xi}$ is 0-labeled, and the value of f is then a relaxed development of ξ_n. This means nothing but that f is an effective history-sensitive solution to $N^{\circ}_{\mathcal{M}}(\psi)$. Therefore, by Theorem 2.14.2, $N^{\circ}_{\mathcal{M}}(\psi)$ is effectively solvable. Lemma 7.7 is proved. ♣

8 Proof of Theorem 5.5(i)⇒(ii)

The definition of ET very much resembles our definition of (relaxed) games. 1-hyperdevelopments correspond to Opponent's moves and 0-hyperdevelopments correspond to Proponent's moves. An essential difference arises only when it comes to atoms of the initial sentence: in real games a play may continue beyond these atoms, which is not the case with "ET-games"; on the other hand, there is nothing in real games directly corresponding to marriage-extension "moves" in ET.

In this section we are going to show that if a sentence α is in ET, then, in every model, Proponent has an effective winning strategy for α. Roughly, Proponent acts as follows:

Whenever Opponent makes a move in α, Proponent finds the corresponding 1-hyperdevelopment α' of α and treats it as a "counterpart" of current position. To determine how to move now, Proponent finds a 0-hyperdevelopment $\alpha'' \in ET$ of α' and tries to "copy" in the real play the "ET-move" corresponding to the transfer from α' to α'', and so on. Of course this is not always possible. Namely, if the above "ET-move" consists in going from a hypersentence to its marriage-extension, it cannot be "copied". However, marriage between osliterals γ and δ is a signal for Proponent to try, from now on, to keep the counterparts (in the real play) of γ and δ opposite to each other. So, if Opponent moves in the counterpart of γ, Proponent tries to make a dual move in δ, and vice versa. As long as Proponent succeeds in doing so, the label of a given position of the real game is 1 whenever its counterpart is 1-like, and this is what ensures winning.

The precise definition of this strategy and the proof of its correctness below are technically involved and pretty boring, so the reader not willing to fight through them can pass on to the next section.

To sentences, hypersentences and many other objects we deal with, can be assigned Gödel numbers. Then, when we say, e.g., "the smallest hypersentence such that...", we mean "smallest by its Gödel number".

Terminology and notation 8.1 We define an operation X on hypersentences:

> $X(\beta) = \beta'$, *where* β' *is the smallest marriage-extension of* β *such that* $\beta' \in ET$*; if such a* β' *does not exist, then* $X(\beta) = \beta$.

Next, we define $X^0(\beta) = \beta$ and $X^{m+1}(\beta) = X(X^m(\beta))$. Clearly for any hypersentence β there is m such that $X^m(\beta) = X^{m+1}(\beta) = X^{m+2}(\beta) = \ldots$ We then call $X^m(\beta)$ the X-*closure* of β. And we say that a sentence is X-*closed*, if it is its own X-closure.

Note that

$$\textit{If } \beta \textit{ belongs to } ET\textit{, then so does its } X\textit{-closure.} \tag{10}$$

Now the proof begins. Assume $\mathcal{M}$ is a model for L and ϕ is sentence of L such that $\phi \in ET$. We will identify each sentence β of L with the corresponding relaxed game $N^{\circ}_{\mathcal{M}}(\beta)$ (see 7.1). As we are going to deal only with this relaxed game, the word "relaxed" will be omitted before "game", "trace", "development", "strategy", etc.

By induction on the length of a ϕ-play we simultaneously define Proponent's effective history-sensitive strategy f together with two other effective functions t and g, where g assigns to every initial segment $\vec{\xi} = \xi_1, \ldots, \xi_n$ of a ϕ-play with this strategy of Proponent's a hypersentence $g\vec{\xi}$, and t assigns to $\vec{\xi}$ a 1-tracing (see 7.3) for ξ_n, so that the following conditions are satisfied:

Condition 1. $g\vec{\xi} \in ET$.

Condition 2. $g\vec{\xi}$ is X-closed.

Condition 3. If $\xi_n = A!(\alpha_1, \ldots, \alpha_k)$, then $g\vec{\xi} = A!(\beta_1, \ldots, \beta_k)$, where:

1. If β_i is not an osliteral, then $\alpha_i = \beta_i$.
2. If β_i is a 0-labeled single osliteral, then, again, $\alpha_i = \beta_i$.
3. If β_i is a 1-labeled single osliteral, then $t\vec{\xi}$ assigns to α_i a 1-trace from β_i to α_i.
4. If β_{j_0} and β_{j_1} are spouses to each other, then $t\vec{\xi}$ assigns to (at least) one of the sentences α_{j_i}, $i \in \{0, 1\}$ a 1-trace from $\neg\alpha_{1-j_i}$ to α_{j_i}.

Before we start defining these three functions, let us prove the following fact:

$$\textit{If the condition 3 holds for a hypersentence } \beta \textit{ in the role of } g\vec{\xi} \textit{ and } \beta' \textit{ is an } X\textit{-closure of } \beta\textit{, then that condition holds for } \beta'\textit{, too.} \tag{11}$$

To see this, it is enough to consider the case when β' is a marriage-extension of β which contains just one additional married couple. That is, the only difference between β and β' then is that for some $1 \leq l, m \leq k$, β_l and β_m were single osliterals in β and they are spouses to each other in β'. Therefore, the conditions 3.1, 3.2 and 3.3 trivially continue to be satisfied for β'. So does the condition 3.4 for any married couple different from (β_l, β_m). So, we only need to verify that the condition 3.4 holds for the couple (β_l, β_m). As β_l and β_m are opposite to each other, one of them — let it be β_l — must have the $l_{\mathcal{M}}$-label 0, and then, by the condition 3.2 for β, $\alpha_l = \beta_l$. According to the condition 3.3 for β, $t\vec{\xi}$ assigns to α_m a 1-trace from β_m; thus, by the chain $\beta_m = \neg\beta_l = \neg\alpha_l$, $t\vec{\xi}$ assigns to α_m a 1-trace from $\neg\alpha_l$, which means that the condition 3.4 is satisfied. (11) is proved.

We define $g\langle\phi\rangle$ to be the X-closure of ϕ. And $t\langle\phi\rangle$ is the tracing that assigns the trivial trace to each 1-labeled surface osliteral of ϕ. It is easy to see that the conditions 1-3 are satisfied (for $\langle\phi\rangle$ in the role of $\langle\vec{\xi}\rangle$). Now, suppose $\langle\vec{\xi}\rangle = \langle\xi_1, \ldots, \xi_n\rangle$ is an initial segment of a ϕ-play with Proponent's strategy f, the functions g and t are defined for $\vec{\xi}$ and the conditions 1-3 are satisfied. According to the condition 3, we have $\xi_n = A!(\alpha_1, \ldots, \alpha_k)$ and $g\vec{\xi} = A!(\beta_1, \ldots, \beta_k)$.

Case 1: ξ_n is 1-labeled. Then f does not need to be defined for $\vec{\xi}$; we must only define g and t for $\langle\vec{\xi}, \xi_{n+1}\rangle$, where ξ_{n+1} is an arbitrary development of ξ_n. We have

$$\xi_{n+1} = A(\alpha_1, \ldots, \alpha_{i-1}, \sigma, \alpha_{i+1}, \ldots, \alpha_k)$$

for some $1 \leq i \leq k$, where σ is a development of the 1-labeled α_i. Then one of the following three cases takes place:

Subcase 1a: α_i is a sliteral. Then, by the condition 3.1, β_1 is a sliteral, too. We define $g\langle\vec{\xi}, \xi_{n+1}\rangle = g\vec{\xi}$, which guarantees that the conditions 1 and 2 remain to be satisfied. Now we need to define t for $\langle\vec{\xi}, \xi_{n+1}\rangle$ and verify that the condition 3, too, is satisfied.

Subsubcase 1a(i): β_i is single. Then, by the condition 3.3, $t\vec{\xi}$ assigns to α_i a 1-trace $\vec{tr}$ from β_i to α_i. Then we define $t\langle\vec{\xi}, \xi_{n+1}\rangle$ to be the tracing that assigns the trace $\langle\vec{tr}, \sigma\rangle$ to σ; to any other osliteral of ξ_{n+1} this tracing assigns the same trace as $t\vec{\xi}$. It is easy to see that then the condition 3 is satisfied.

Subsubcase 1a(ii): β_i is married to some β_j. If α_i and α_j are opposite, we define $t\langle\vec{\xi}, \xi_{n+1}\rangle$ to be the tracing that assigns the trace $\langle\alpha_i, \sigma\rangle$ to σ, is undefined for α_j and assigns to any other osliteral of ξ_{n+1} the same trace as $t\vec{\xi}$. Suppose now α_i and α_j are not opposite. Note that $\neg\alpha_i$ is 0-labeled and therefore there cannot exist a (nontrivial) trace from $\neg\alpha_i$ to α_j. Therefore, by the condition 3.4, $t\vec{\xi}$ assigns to α_i a 1-trace from $\neg\alpha_j$ to α_i. Then we define $t\langle\vec{\xi}, \xi_{n+1}\rangle$ to be the tracing that assigns the trace $\langle\vec{tr}, \sigma\rangle$ to σ and assigns the same trace as $t\vec{\xi}$ to any other osliteral of ξ_{n+1} (namely, is undefined for α_j). It is easily seen that in this subsubcase, too, the condition 3 is satisfied.

Subcase 1b: $\alpha_i = \gamma \wedge \delta$. Note that then, by the condition 3.1, $\beta_i = \alpha_i$. We may suppose that $\sigma = \gamma$. Then we define $g\langle\vec{\xi}, \xi_{n+1}\rangle$ to be the X-closure of $A(\beta_1, \ldots, \beta_{i-1}, \sigma, \beta_{i+1}, \ldots, \beta_k)$. And $t\langle\vec{\xi}, \xi_{n+1}\rangle$ is the tracing that assigns to any surface osliteral α_j $(1 \leq j \leq k, \ j \neq i)$ of ξ_{n+1} the same trace as $t\vec{\xi}$, and assigns a trivial trace to any other surface osliteral of ξ_{n+1} (i.e. to those that are surface osliterals of σ). The condition 1 follows now from (10) and Lemma 6.15a, and the condition 2 is trivially satisfied; as for the condition 3, it is evident for $A(\beta_1, \ldots, \beta_{i-1}, \sigma, \beta_{i+1}, \ldots, \beta_k)$ in the role of $g\vec{\xi}$, whence, by (11), it holds for its X-closure $g\langle\vec{\xi}, \xi_{n+1}\rangle$.

Subcase 1c: $\alpha_i = \forall x\gamma(x)$. This subcase is similar to the previous one.

Case 2: ξ_n is 0-labeled.

Subcase 2a: $g\vec{\xi}$ is 0-like. Then, as $g\vec{\xi} \in ET$, there is a strict 0-hyperdevelopment $\eta \in ET$ of $g\vec{\xi}$ (if there are many such hyperdevelopments, we suppose that η is the smallest among them). Since $g\vec{\xi}$ is X-closed, the relation between $g\vec{\xi}$ and η is determined by 6.6.2a or 6.6.2b. We consider only the case 6.6.2a; the case 6.6.2b is similar. So, for some i, $\beta_i = \gamma \vee \delta$ and — we may suppose —

$$\eta = A(\beta_1, \ldots, \beta_{i-1}, \gamma, \beta_{i+1}, \ldots, \beta_k).$$

Then we define

$$f\vec{\xi} = A(\alpha_1, \ldots, \alpha_{i-1}, \gamma, \alpha_{i+1}, \ldots, \alpha_k).$$

We define $g\langle\vec{\xi}, f\vec{\xi}\rangle$ to be the X-closure of η. And, $t\langle\vec{\xi}, f\vec{\xi}\rangle$ is the tracing that assigns to any surface osliteral α_j $(1 \leq j \leq k,\ j \neq i)$ of $f\vec{\xi}$ the same trace as $t\vec{\xi}$, and to any other surface osliteral of $f\vec{\xi}$ (i.e. those that are surface osliterals of γ) assigns a trivial trace. As $\eta \in ET$ and $g\langle\vec{\xi}, f\vec{\xi}\rangle$ is an X-closure of η, the condition 1 is satisfied (see (10)). The condition 2 holds trivially, and the condition 3 is evident in view of (11).

Subcase 2b: $g\vec{\xi}$ is 1-like.

Subsubcase 2b(i): Suppose there is a married couple (β_l, β_m) in $g\vec{\xi}$ such that $t\vec{\xi}$ assigns to α_l a nontrivial 1-trace $\omega_1, \ldots, \omega_p$ with $p \geq 2$, $\omega_1 = \neg\alpha_m$, $\omega_p = \alpha_l$. If there are many such couples, we suppose that (β_l, β_m) is the smallest among them. Then we define

$$f\vec{\xi} = A(\alpha_1, \ldots, \alpha_{m-1}, \neg\omega_2, \alpha_{m+1}, \ldots, \alpha_k).$$

Note that α_m is 0-labeled and $\neg\omega_2$ is its development, so $f\vec{\xi}$ is a development of ξ_n. We define $g\langle\vec{\xi}, f\vec{\xi}\rangle = g\vec{\xi}$. As for the tracing $t\langle\vec{\xi}, \xi_{n+1}\rangle$, it assigns the trace $\omega_2, \ldots, \omega_p$ to α_l and remains the same as $t\vec{\xi}$ for any other osubsentence of $f\vec{\xi}$. The conditions 1 and 2 trivially continue to hold for $\langle\vec{\xi}, f\vec{\xi}\rangle$, and the condition 3 is also easy to verify.

Subsubcase 2b(ii): If the married couple described in the subsubcase 2b(i) does not exist, then f is undefined for $\vec{\xi}$ and g, t are undefined for $\langle\vec{\xi}, \xi_{n+1}\rangle$.

Thus, in every (sub)subcase of the case when ξ_n is 0-labeled, except 2b(ii), $f\vec{\xi}$ is defined and its value is a development of ξ_n. To conclude that f is a solution to ϕ, it remains to show that the subsubcase 2b(ii) never takes place.

But indeed: suppose that the married couple described in 2b(i) does not exist. Consider an arbitrary married couple (β_l, β_m) of $g\vec{\xi}$. According to the condition 3.4, one of the osubsentences α_l, α_m of ξ_n — let it be α_l — comes with a 1-trace from $\neg\alpha_m$ to α_l. Consequently, this trace must be a trivial one, which means that $\alpha_l = \neg\alpha_m$. Thus, for each married couple (β_l, β_m) of $g\vec{\xi}$, α_l and α_m are opposite. Let then h be a hyperlabeling such that for any married osliteral β_e of $g\vec{\xi}$, $h(\beta_e)$ equals the label of α_e. Then for every $1 \leq e \leq k$, $h(\beta_e) \leq$ the label of α_e, whence $h(g\vec{\xi}) \leq$ the label of ξ_n, which means that either ξ_n is not 0-labeled, or $g\vec{\xi}$ is not 1-like, and this contradicts the conditions of the subcase 2b.

The part $(i) \Rightarrow (ii)$ of Theorem 5.5 is proved. ♣

An analysis of the proof of 5.5(i)⇒(ii) (and the proofs of all the lemmas employed in that proof) can convince us that in fact the following strong version of the soundness of ET holds:

Theorem 8.2 *There is an effective function which, for any model $\mathcal{M}$ for L (where $\mathcal{D}_\mathcal{M}, \ell_\mathcal{M}, \mathcal{R}_\mathcal{M}$ are given as Turing machines), returns an effective function (Turing machine) f which is a solution to any game $N_\mathcal{M}(\phi)$ with $\phi \in ET$.*

I wouldn't like to waste the reader's time on any more details or comments on the proof of this theorem.

9 Proof of Theorem 5.5(ii)⇒(iii)

The part (ii)⇒(iii) of Theorem 5.5 immediately follows from the clause (b) of Lemma 9.2. Before we start proving the latter, we need the following auxiliary lemma:

Lemma 9.1 *Suppose $\beta(x)$ is a formula of L, \$ is an arithmetical translation good* (see 5.2) *for $\beta(x)$, and $(\beta(x))^\$ = \gamma(x)$. Then $(\beta(a))^\$ = \gamma(a)$ for any parameter a.*

PROOF. Assume \$ is good for $\beta(x)$ and $(\beta(x))^\$ = \gamma(x)$. According to 5.3, x is free in $\beta(x)$ iff it is free in $\gamma(x)$. If x is not free in these formulas, then $\beta(x) = \beta(a)$, $\gamma(x) = \gamma(a)$ and we are done. So, suppose x is free in $\beta(x), \gamma(x)$.

We proceed by induction on the complexity of $\beta(x)$.

Suppose $\beta(x)$ is an atom $P(x, y_1, \ldots, y_n)$, all the variables of which are explicitly indicated, and $\$P = \delta(x_0, \ldots, x_n)$. Then, by the definition of translation, $(\beta(x))^\$ = \delta(x, y_1, \ldots, y_n)$ and $(\beta(a))^\$ = \delta(a, y_1, \ldots, y_n)$. Since x is free in $\gamma(x) = \delta(x, y_1, \ldots, y_n)$, we have $\gamma(a) = \delta(a, y_1, \ldots, y_n)$. Thus, $(\beta(a))^\$ = \gamma(a)$.

The cases when $\beta(x)$ is a complex formula are pretty straightforward, and we consider only one of them. Suppose $\beta(x) = \forall z\delta(x, z)$. Then $\gamma(x) = \forall z\delta'(x, z)$, where $\delta'(x, z) = (\delta(x, z))^\$$, whence, by the induction hypothesis, $\delta'(a, z) = (\delta(a, z))^\$$, whence $\forall z\delta'(a, z) = \forall z((\delta(a, z))^\$) = (\forall z\delta(a, z))^\$ = (\beta(a))^\$$. But $\forall z\delta'(a, z) = \gamma(a)$. Thus, $(\beta(a))^\$ = \gamma(a)$. ♣

Lemma 9.2 *For any sentence ϕ of L and any arithmetical translation τ good for ϕ, there is a model $\mathcal{M}$ with $\mathcal{D}_{\mathcal{M}} = NAT$ for L such that:*

a) if $N_{\mathcal{M}}(\phi)$ is solvable, then so is $N_{\mathcal{S}}(\phi^{\tau})$;

b) if $N_{\mathcal{M}}(\phi)$ is effectively solvable, then so is $N_{\mathcal{S}}(\phi^{\tau})$.

(Recall that $\mathcal{S}$ is the standard model of arithmetic.)

PROOF. Taking Lemma 7.7 into account, we'll deal only with relaxed plays, and omit everywhere the word "relaxed", as well as the superscript "$\circ$".

Let us fix a recursive list

$$\alpha_0, \alpha_1, \ldots$$

of all arithmetical sentences and a recursive list

$$Q_0, Q_1, \ldots$$

of zero place predicate letters of L not occuring in ϕ. Then, for any arithmetical sentence $\beta = \alpha_i$, let Q_β denote the sliteral Q_i.

Let τ be an arithmetical translation good for ϕ. We may suppose that each predicate letter of L either occurs in ϕ, or is one of the Q_i. We define a new arithmetical translation \$ as follows:

- For each predicate letter Q_i from the above list, $\$Q_i = \alpha_i$;
- for any other predicate letter P, $\$P = \tau P$.

Thus, every arithmetical sentence β is $\gamma^{\$}$ for some (at least one) sentence γ of L. Note that the translation \$ is good for ϕ.

We now define a model $\mathcal{M}$ for L: for any atomic sentences γ, δ of L,

- $\ell_{\mathcal{M}}(\gamma) = l_{\mathcal{S}}(\gamma^{\$})$;
- $\gamma R_{\mathcal{M}} \delta$ iff $\delta = Q_i$ for some i and $\gamma^{\$} R_{\mathcal{S}} \delta^{\$}$.

A straightforward induction on the complexity of γ convinces us that

$$l_{\mathcal{M}}(\gamma) = l_{\mathcal{S}}(\gamma^{\$}) \quad (\text{all } \gamma). \tag{12}$$

Assume h is a solution to $N_{\mathcal{M}}(\phi)$. We define a Proponent's history-sensitive strategy f and a function g which assigns to every initial segment $\vec{\xi} = \langle \xi_1, \ldots, \xi_n \rangle$ of an $N_{\mathcal{S}}(\phi^{\tau})$-play with this strategy a sentence $g\vec{\xi}$ of L such that the following three conditions are satisfied:

Condition 1: h is a solution to $N_{\mathcal{M}}(g\vec{\xi})$.

Condition 2: $\xi_n = (g\vec{\xi})^{\$}$.

Condition 3: \$ is good for every surface osubsentence of $g\vec{\xi}$.

We define $g\langle\phi^\tau\rangle = \phi$. Thus, the condition 1 is satisfied. As ϕ does not contain any of the atoms $Q_0, Q_1, \ldots$, evidently $\phi^\tau = \phi^\$$, so the condition 2, too, is satisfied. And as \$ is good for ϕ, it is good for each subsentence of it, which means that the condition 3 is satisfied as well.

Suppose now $\vec{\xi} = \langle\xi_1, \ldots, \xi_n\rangle$ is an initial segment of an $N_{\mathcal{S}}(\phi^*)$-play, $g\vec{\xi}$ is defined and the conditions 1 and 2 are satisfied. In view of the condition 2, we must have

$$g\vec{\xi} = A!(\beta_1, \ldots, \beta_k)$$

and

$$\xi_n = A(\beta_1^\$, \ldots, \beta_k^\$).$$

Case 1: ξ_n is 1-labeled (in the standard model of arithmetic). Note that, by (12) and the condition 2, $l_{\mathcal{M}}(g\vec{\xi}) = 1$. This position obliges Opponent to move, and we only need to define g for $\langle\vec{\xi}, \xi_{n+1}\rangle$, where ξ_{n+1} is an arbitrary development of ξ_n. ξ_{n+1} must be the result of replacing in ξ_n a 1-labeled osubsentence $\beta_i^\$$ by an (arithmetical) development σ of it.

Subcase 1a: $\beta_i = \gamma\wedge\delta$. Then $\beta_i^\$ = \gamma^\$\wedge\delta^\$$. And we may suppose that $\sigma = \gamma^\$$. Then we define $g\langle\vec{\xi}, \xi_{n+1}\rangle$ to be the result of replacing in $g\vec{\xi}$ the osubsentence β_i by γ. Of course $\xi_{n+1} = (g\langle\vec{\xi}, \xi_{n+1}\rangle)^\$$. Note also that $g\langle\vec{\xi}, \xi_{n+1}\rangle$ is a development of the 1-labeled $g\vec{\xi}$ and, as h is a solution to the latter, it must be a solution to $g\langle\vec{\xi}, \xi_{n+1}\rangle$, too. Thus, the conditions 1 and 2 are satisfied. Since \$ was good for β_i and γ is a subsentence of β_i, \$ remains good for γ. As no other osubsentences of $g\vec{\xi}$ have been changed, the condition 3, too, continues to hold.

Subcase 1b: $\beta_i = \forall x\gamma(x)$. Let $\delta(x) = (\gamma(x))^\$$. Then $\beta_i^\$ = \forall x\delta(x)$ and $\sigma = \delta(a)$ for some parameter a. Since \$ is good for $\gamma(x)$, by Lemma 9.1, $\delta(a)$, i.e. σ, is then equal to $(\gamma(a))^\$$. We define $g\langle\vec{\xi}, \xi_{n+1}\rangle$ to be the result of replacing in $g\vec{\xi}$ the osubsentence β_i by $\gamma(a)$. Clearly $(g\langle\vec{\xi}, \xi_{n+1}\rangle)^\$ = \xi_{n+1}$, i.e. the condition 2 holds. And an argument similar to that from the previous subcase can convince us that the conditions 1 and 3 hold as well.

Subcase 1c: β_i is a sliteral. Then we define $g\langle\vec{\xi}, \xi_{n+1}\rangle$ to be the result of replacing in $g\vec{\xi}$ the osubsentence β_i by Q_σ. Evidently $\xi_{n+1} = (g\langle\vec{\xi}, \xi_{n+1}\rangle)^\$$; also, $g\langle\vec{\xi}, \xi_{n+1}\rangle$ is a development of the 1-labeled $g\vec{\xi}$ and, as h is a solution to the latter, it must be a solution to $g\langle\vec{\xi}, \xi_{n+1}\rangle$, too. Again, both conditions 1 and 2 are satisfied. Also, since Q_σ does not contain any variables, \$ is good for it and this implies that the condition 3, too, is satisfied.

Case 2: ξ_n is 0-labeled. Then, by (12), so is $g\vec{\xi}$ and, as h is a solution to the latter, $h(g\vec{\xi})$ is a development of $g\vec{\xi}$ and

$$h \textit{ is a solution to } h(g\vec{\xi}). \tag{13}$$

$h(g\vec{\xi})$ must be the result of replacing in $g\vec{\xi}$ a 0-labeled osubsentence β_i by a development σ of β_i. Then we define $f\vec{\xi}$ to be the result of replacing in ξ_n the corresponding osubsentence $\beta_i^\$$ by $\sigma^\$$. Of course,

$$f\vec{\xi} = (h(g\vec{\xi}))^\$. \tag{14}$$

Note that, by (12), $\beta_i^\$$ is 0-labeled. It is also easy to verify that $\sigma^\$$ is a development of $\beta_i^\$$ (if β_i has the form $\exists\ldots$, we'll need to use Lemma 9.1). Therefore,

$$f\vec{\xi} \ \textit{ is a development of } \ \xi_n. \tag{15}$$

We define $g\langle\vec{\xi}, f\vec{\xi}\rangle = h(g\vec{\xi})$. According to (13) and (14), the conditions 1 and 2 are then satisfied for $\langle\vec{\xi}, f\vec{\xi}\rangle$ in the role of $\vec{\xi}$. If β_i is not a sliteral, then σ is a subsentence of β_i and therefore \$ remains good for it. And if β_i is a sliteral, then so is σ and, as σ then does not contain variables, \$ is good for it. This implies that the condition 3 holds as well.

Thus, as soon as ξ_n is 0-labeled, f is defined for $\vec{\xi}$ and its value is a development of ξ_n (by (15)). We conclude that f is a solution to ϕ^τ. The clause (a) of the lemma is proved.

Finally, notice that f is primitive recursive relative to h, whence, if h is effective, so is f. This proves the clause (b). ♣

10 Generalized modus ponens for effective truth

Lemma 10.1 *Suppose $A(\alpha,\vec{\beta})$ and $\neg\alpha$ are effectively true in a model $\mathcal{M}$. Then, for any sentence γ, $A(\gamma,\vec{\beta})$ is effectively true in $\mathcal{M}$.*[7]

PROOF. Fix a model $\mathcal{M}$. We shall deal only with the relaxed net of games induced by $\mathcal{M}$, and omit the word "relaxed" everywhere.

Assume that h is an effective solution to $A(\alpha,\vec{\beta})$ and r is an effective solution to $\neg\alpha$.

The intuition behind Proponent's strategy for $A(\gamma,\vec{\beta})$ we are going to define can be described as follows: Proponent plays only in the component $\vec{\beta}$ of $A(\gamma,\vec{\beta})$. To determine what moves to make in this play, Proponent also plays an "experimental" $A(\alpha,\vec{\beta})$-play, where he uses the strategy h. He has an assistant who is in the role of opponent in this play. The assistant uses the strategy r for the component α and repeats Opponent's moves, made in the component $\vec{\beta}$ of the $A(\gamma,\vec{\beta})$-play, in the component $\vec{\beta}$ of the experimental play. Then, the strategy h's replies to these moves in the experimental play show Proponent how to act in the component $\vec{\beta}$ of the $A(\gamma,\vec{\beta})$-play. The strategy r allows the assistant to always make the component α 0-labeled, and this means that the strategy h solves $A(\alpha,\vec{\beta})$ only at the expense of a successful play in the component $\vec{\beta}$.

[7] To see why this lemma should be considered as generalized modus ponens, take $A(\alpha,\beta)$ to be $\alpha\nabla\beta$ and γ to be $\perp$, identifying $\perp\nabla\beta$ with β.

And as moves in this component are copied in the $A(\gamma, \vec{\beta})$-play, Proponent is guaranteed to win in the latter.

The reader satisfied by this explanation can skip the rest of this section which is devoted to a formal implementation of this intuition.

Let

$$\vec{\beta} = \beta_1, \ldots, \beta_m.$$

Before we describe a solution to $A(\gamma, \vec{\beta})$, we need some local terminology and notation.

For a function s and argument B, $s^0(B) = B$ and $s^{k+1}(B) = s(s^k(B))$.

A sentence λ will be said to be *interesting*, if $\lambda = A(\delta_0, \delta_1, \ldots, \delta_m)$ for some $\delta_0, \ldots, \delta_m$ such that h is a solution to λ and r is a solution to $\neg\delta_0$; the osubsentence δ_0 will be said to be the *main osubsentence* of λ. Thus, $A(\alpha, \vec{\beta})$ is an interesting sentence and α is its main osubsentence.

The *r-closure* of such an interesting sentence λ is λ, if λ is 0-labeled, and is

$$A(\neg r^k(\neg\delta_0), \delta_1 \ldots, \delta_m)$$

otherwise, where k is the least number such that $r^k(\neg\delta_0)$ is 1-labeled.

Notice that the r-closure of λ exists (is correctly defined) because of the assumption that r solves $\neg\delta_0$, and note also that the r-closure of λ continues to be interesting.

And the *h-closure* of λ is $h^k(\lambda)$, where k is the least number such that $h^k(\lambda)$ is 1-labeled or $h^{k+1}(\lambda)$ has the same main osubsentence as $h^k(\lambda)$. Note that the h-closure of λ (is well defined and) continues to be interesting.

We say that λ is r- (resp. h-) *closed*, if its r- (resp. h-) closure is λ; and λ is *rh-closed*, if it is both r- and h-closed.

Now, to get the sentence that we call the *rh-closure* of λ, we take the r-closure of λ, then the h-closure of this r-closure, then the r-closure of this h-closure, $\ldots$, — until we reach an rh-closed sentence.

Since the operations of r- and h-closure preserve the property of being interesting and they can change only the main osubsentence, we have:

The rh-closure of an interesting sentence $A(\delta_0, \delta_1, \ldots, \delta_m)$ remains interesting and it is $A(\delta'_0, \delta_1, \ldots, \delta_m)$ for some δ'_0. (16)

We simultaneously define Proponent's effective history-sensitive strategy f together with two other effective functions t and g, where g assigns to every initial segment $\vec{\xi} = \xi_1, \ldots, \xi_n$ of an $A(\gamma, \vec{\beta})$-play with this strategy of Proponent's, with $\xi_n = A(\delta_0, \ldots, \delta_m)$, a sentence $g\vec{\xi} = A(\delta'_0, \ldots, \delta'_m)$, and $t\vec{\xi}$ determines traces for some surface osubsentences of ξ_n, so that the following conditions are satisfied:

Condition 1. $g\vec{\xi}$ is interesting.

Condition 2. $g\vec{\xi}$ is rh-closed.

Condition 3. $t\vec{\xi}$ assigns to each δ_i $(1 \leq i \leq m)$ a 1-trace from δ'_i to δ_i.

Here is the description of f, g and t; at each step of this definition we verify that the above three conditions are satisfied.

We define $g\langle A(\gamma, \vec{\beta})\rangle$ to be the rh-closure of $A(\alpha, \vec{\beta})$. And $t\langle A(\gamma, \vec{\beta})\rangle$ is the tracing that assigns the trivial trace to each osubsentence β_i $(1 \leq i \leq m)$. The conditions 1-3 are evidently satisfied (for $\langle A(\gamma, \vec{\beta})\rangle$ in the role of $\langle\vec{\xi}\rangle$).

Now, suppose $\langle\vec{\xi}\rangle = \langle \xi_1, \ldots, \xi_n\rangle$, where

$$\xi_n = A(\delta_0, \ldots, \delta_m),$$

is an initial segment of a $A(\gamma, \vec{\beta})$-play with Proponent's strategy f, the functions g and t are defined for $\vec{\xi}$,

$$g\vec{\xi} = A(\delta'_0, \ldots, \delta'_m)$$

and the conditions 1-3 are satisfied.

Case 1: ξ_n is 1-labeled. We must only define g and t for $\langle\vec{\xi}, \xi_{n+1}\rangle$, where ξ_{n+1} is an arbitrary development of ξ_n. We have

$$\xi_{n+1} = A(\delta_0, \ldots, \delta_{i-1}, \sigma, \delta_{i+1}, \ldots, \delta_m)$$

for some $0 \leq i \leq m$, where σ is a development of the 1-labeled δ_i. Then we define $g\langle\vec{\xi}, \xi_{n+1}\rangle = g\vec{\xi}$. And we define $t\langle\vec{\xi}, \xi_{n+1}\rangle$ to be the tracing that assigns to each δ_j with $1 \leq i \leq m$, $j \neq i$ the same trace as $t\vec{\xi}$, and, if $i \neq 0$, assigns the trace $\langle\vec{tr}, \sigma\rangle$ to σ, where $\vec{tr}$ is the trace assigned to δ_i by $t\vec{\xi}$. The conditions 1 and 2 trivially continue to be satisfied, and evidently the condition 3, too, remains to be satisfied.

Case 2: ξ_n is 0-labeled. We define the sequence $\omega_0, \ldots, \omega_p$ of sentences and the sequence $s_0, \ldots, s_p$ of tracings for ξ_n as follows:

- $\omega_0 = g\vec{\xi}$ and $s_0 = t\vec{\xi}$. By our assumption, the conditions 1-3 are satisfied for ω_0 and s_0 in the roles of $g\vec{\xi}$ and $t\vec{\xi}$.
- Suppose now ω_l and s_l are defined, and the conditions 1-3 are satisfied for ω_l and s_l in the roles of $g\vec{\xi}$ and $t\vec{\xi}$. Then:
    1. If ω_l has the label 0, then $p = m$.
    2. Suppose now $\omega_l = A(\theta_0, \ldots, \theta_m)$ has the label 1. Note that, as ω_l is r-closed, its main osubsentence θ_0 is 0-labeled. Therefore the only reason for ω_l's being 1-labeled (whereas ξ_n is 0-labeled) can be that for some $1 \leq i \leq m$, θ_i is 1-labeled whereas δ_i is 0-labeled (if there are many such i, we suppose that our i is the smallest among them). This means that the 1-trace $\langle \eta_1, \ldots, \eta_d\rangle$ from θ_i to δ_i, assigned to δ_i

by s_l, is not trivial ($d \geq 2$). Then we define ω_{l+1} to be the rh-closure of

$$A(\theta_0, \dots, \theta_{i-1}, \eta_2, \theta_{i+1}, \dots, \theta_m).$$

And we define s_{l+1} to be the tracing that assigns the trace $\langle \eta_2, \dots, \eta_d \rangle$ to δ_i and assigns the same trace as s_l to each osubsentence δ_j of ξ_n with $1 \leq j \leq m$, $j \neq i$. It is not hard to verify that the conditions 1-3 are satisfied for ω_{l+1} and s_{l+1} in the roles of $g\vec{\xi}$ and $t\vec{\xi}$.

Each ω_{m+1} is a development of ω_m, so the number p will sooner or later be reached. Thus,

$$\text{\textit{ω_p is 0-labeled and the conditions 1-3 are satisfied for ω_p and s_p in the roles of $g\vec{\xi}$ and $t\vec{\xi}$.}} \tag{17}$$

Then, as h solves $g\vec{\xi}$ (the condition 1), the value of h for ω_p (is defined and) is a development of ω_p. Assume

$$\omega_p = A(\rho_0, \dots, \rho_m).$$

The h-closedness of ω_p implies that $h\omega_p$ is the result of replacing in ω_p one of the 0-labeled ρ_i, $i \neq 0$, by a development σ of ρ_i. Then we define

$$f\vec{\xi} = A(\delta_0, \dots, \delta_{i-1}, \sigma, \delta_{i+1}, \dots, \delta_m).$$

We also define $g\langle \vec{\xi}, f\vec{\xi} \rangle$ as the rh-closure of $h\omega_p$, and we define $t\langle \vec{\xi}, f\vec{\xi} \rangle$ as the tracing that assigns the trivial trace to σ and assigns the same trace as s_p to each δ_j with $1 \leq j \leq m$, $j \neq i$.

Note that as ρ_i is 0-labeled and there is a 1-trace from ρ_i to δ_i (determined by s_p), we have $\delta_i = \rho_i$. Therefore $f\vec{\xi}$ is a development of ξ_n.

It is easy to verify that $h\omega_p$ is interesting (taking into account that ω_p is so), whence, by (16), $g\langle \vec{\xi}, f\vec{\xi} \rangle$ is interesting. That is, the condition 1 is satisfied for $\langle \vec{\xi}, f\vec{\xi} \rangle$. The condition 2 is trivially satisfied because $\langle \vec{\xi}, f\vec{\xi} \rangle$ is an rh-closure, and the condition 3 is also evident.

Thus, f is defined for $\vec{\xi} = \langle \xi_1, \dots, \xi_n \rangle$ as soon as ξ_n is 0-labeled, and $f\vec{\xi}$ is a (relaxed) development of ξ_n. This means that f is a history-sensitive relaxed solution to $A(\gamma, \vec{\beta})$, whence, by 2.14 and 7.7, $A(\gamma, \vec{\beta})$ is effectively true in $\mathcal{M}$. ♣

11 Arithmetization of the game semantics of arithmetic

We shall use $\alpha \to \beta$ to abbreviate $\neg\alpha \vee \beta$ and $\exists x_{x \leq y} \alpha$ to abbreviate $\exists x(x \leq y \wedge \alpha)$. The meanings of the abbreviations $\exists x_{x<y}$, $\forall x_{x \leq y}$, etc. should also be clear.

Definition 11.1

1. The set of Δ_0-*formulas* is the smallest set of arithmetical formulas such that:

- atomic formulas are Δ_0-formulas;
- if α and β are Δ_0-formulas, then $\neg\alpha$, $\alpha \vee \beta$ are Δ_0-formulas;
- if α is a Δ_0-formula and t is a term (see Convention 3.6) not containing x, then $\exists x_{x \leq t}\alpha$ is a Δ_0-formula.

2. "Σ_0"-formula and "Π_0"-formula are synonyms of "Δ_0"-formula, and for $n \geq 1$, we say that a formula α is a Σ_n- (resp. Π_n-) formula, if $\alpha = \exists x_1 \ldots \exists x_k \beta$ (resp. $\alpha = \forall x_1 \ldots \exists x_k \beta$) for some Π_{n-1}- (resp. Σ_{n-1}-) formula β and some, possibly empty, sequence $x_1, \ldots, x_k$ of variables.

Note that, according to the above definition, every Σ_n- or Π_n-formula is, at the same time, a Σ_{n+1}- and a Π_{n+1}-formula as well.

The meaning of the notion of Σ_n- (resp. Π_n-, Δ_0-) *sentence* must be clear: this is the result of replacing each free variable in a Σ_n- (resp. Π_n-, Δ_0-) formula by a natural number (see the beginning of Section 7).

The following fact does not require any comments:

Fact 11.2 *Truth for Δ_0 sentences is decidable.*

This implies a more interesting fact:

Fact 11.3 *There is an effective strategy $DZERO$ which is a solution to any true Δ_0-sentence.*

PROOF. Consider an arbitrary 0-labeled true Δ_0-sentence δ. We define $DZERO(\delta)$ as follows:

Case 1: $\delta = \alpha_1 \vee \alpha_2$. As δ is a true Δ_0 sentence, so is at least one disjunct α_i of it, and this disjunct, in view of 11.2, can be determined in an effective way. Then let $DZERO(\delta) = \alpha_i$.

Case 2: $\delta = \exists x_{x \leq t}\alpha(x)$. Again, in an effective way can be found b with $b \leq t$ such that $\alpha(b)$ is a true Δ_0-sentence. We then define $DZERO(\delta) = \alpha(b)$.

Thus, $DZERO$ is defined for any true 0-labeled Δ_0-sentence δ and the value of $DZERO$ for δ is a development of δ which remains a true Δ_0-sentence. This, together with the observation that any development of a 1-labeled true Δ_0-sentence remains a true Δ_0-sentence, implies that $DZERO$ is a solution to any true Δ_0-sentence. ♣

Fact 11.4 *There is an effective strategy $UNIV$ which is a solution to any true Π_2-sentence.*

PROOF. Consider an arbitrary true Π_2-sentence π. It has the form

$$\forall x_1, \ldots, \forall x_m \exists y_1 \ldots \exists y_n \alpha(x_1, \ldots, x_m, y_1, \ldots, y_n),$$

(possibly $m, n = 0$) for some Δ_0-formula $\alpha(\vec{x}, \vec{y})$. The first m moves in a π-play are to be made by Opponent, and these moves lead to the true Σ_1-sentence

$$\sigma = \exists y_1 \ldots \exists y_n \alpha(a_1, \ldots, \alpha_m, y_1, \ldots, y_n)$$

for some $a_1, \ldots, a_m$. At this point Proponent starts checking, one after one, all possible n-tuples of numbers until he reaches a tuple $b_1, \ldots, b_n$ such that the Δ_0-sentence $\delta = \alpha(a_1, \ldots, a_m, b_1, \ldots, b_n)$ is true. Such a tuple exists because σ is true, and this checking is effective according to 11.2. After the tuple is found, Proponent makes n consecutive moves, each of which consists in the deleting of "$\exists y_i$" and replacing "y_i" by "b_i". The play comes to the sentence (position) δ. Now Proponent can switch to the strategy $DZERO$ which guarantees that Proponent wins the play. ♣

Corollary 11.5 *Every true Σ_3-sentence is effectively true.*

PROOF. Indeed, suppose σ is a true Σ_3-sentence. We may suppose that σ is not a Π_2-sentence, for otherwise Proponent can use the strategy $UNIV$ for it. Then $\sigma = \exists x_1, \ldots, \exists x_n \pi(x_1, \ldots, x_n)$ for some Π_2-formula $\pi(\vec{x})$ and some $n \geq 1$, and there is an n-tuple $a_1, \ldots, a_n$ of numbers such that $\pi(\vec{a})$ is true. Proponent starts the play by making n consecutive moves, consisting in deleting the "$\exists x_i$" and replacing the "x_i" by "a_i", and this ultimately leads to the position $\pi(\vec{a})$. Now Proponent can switch to the strategy $UNIV$. This guarantees a win for him. ♣

Definition 11.6 We say that an arithmetical formula $\alpha(x_1, \ldots, x_n)$, with exactly $x_1, \ldots, x_n$ free, *represents* (in the standard model) an n-place relation A on natural numbers, if for any natural numbers $a_1, \ldots, a_n$, the relation $A(a_1, \ldots, a_n)$ holds if and only if $\alpha(a_1, \ldots, a_n)$ is true in the standard model.

It is well known that

Fact 11.7 *Any decidable relation can be represented by a Σ_1-formula.*

As the negation of a decidable relation remains decidable, it is clear that "Σ" can be replaced by "Π" above.

Let us fix some good coding of finite functions of the type $NAT \to NAT$. E.g., if such a function f is defined exactly for $a_1, \ldots, a_n$ and its values for these arguments are $b_1, \ldots, b_n$, then let the code of f be a (the) standard code of the sequence $\langle (a_1, b_1), \ldots, (a_n, b_n) \rangle$.

And we understand a partial recursive function f as a Turing machine, which allows us to speak about the code of such a function. For a machine f and a natural number i, we define a one-place function $f^{\leq i}$ as follows: For any argument $n \in NAT$,

- if $n \leq i$ and for the input n the machine f halts within $\leq i$ steps and outputs a natural number m, then $f^{\leq i}(n) = m$;
- otherwise $f^{\leq i}$ is undefined for n.

Note that $f^{\leq i}$ is a finite function. Note also that for any machine f and any numbers n, m,

$$f(n) = m \text{ iff } f^{\leq i}(n) = m \text{ for some } i. \tag{18}$$

Convention 11.8 Under the standard Gödel numbering, not every natural number is the code of something. However, it is convenient to sometimes abuse terminology and say "the x-coded sentence (formula, finite function, ...)", as this is done in the following "definition":

> FOO(x) is the function that returns the code of the negation of the x-coded sentence.

In such a case we will mean that FOO returns 0, if x is not the code of a sentence. Similarly, if FOO is defined as a predicate and the body of the definition somewhere says "...the x-coded sentence...", we suppose that FOO returns "false" as soon as the argument x is not the code of a sentence.

We fix the following four arithmetical formulas:

- $Won4(x, y, z, i)$ is a formula defining the following 4-place relation: "(the x-coded sentence)-play with Proponent's strategy (the y-coded machine)$^{\leq i}$ and Opponent's strategy (the z-coded finite function) is won".

As for any y and i the one-place functions (the y-coded machine)$^{\leq i}$(...) and (the y-coded finite function)(...) are both finite and $l_{\mathcal{S}}$(...) and ...$R_{\mathcal{S}}$... are decidable, it is easy to see that the predicate $Won4$ is decidable. Therefore we suppose that $Won4(x, y, z, i)$ is a Σ_1-formula.

- $Won3(x, y, z) = \exists i Won4(x, y, z, i)$; $Won3$ is a Σ_1-formula.
- $Solves(y, x) = \forall z Won3(x, y, z)$; $Solves$ is a Π_2-formula.
- $Eftrue(x) = \exists y Solves(y, x)$; $Eftrue$ is a Σ_3-formula.

In view of (18), it is evident that $Won3(x, y, z)$ represents the 3-place predicate "(the x-coded sentence)-play with Proponent's strategy (the y-coded machine) and Opponent's strategy (the z-coded finite function) is won".

And in view of 2.10, $Solves(y, x)$ represents the predicate "(the y-coded machine) is a solution to (the x-coded sentence)".

Finally, $Eftrue(x)$ clearly represents the predicate "(the x-coded sentence) is effectively true in the standard model".

Thus, the predicate of effective truth of an arithmetical sentence has the complexity Σ_3, which contrasts with the well-known nonarithmeticity of classical truth.

Definition 11.9 Let $\alpha(x)$ be an arithmetical formula with only x free, and $h(y_1, \ldots, y_n)$ be an n-ary primitive recursive function. We say that a Proponent's strategy f is *stable* for $(\alpha(x), h(y_1, \ldots, y_n))$, if for any numbers $a, b_1, \ldots, b_n$ such that $a = h(b_1, \ldots, b_n)$, whenever f solves the sentence $\alpha(a)$, it also solves the sentence $\alpha(h(b_1, \ldots, b_n))$ (recall 3.6).

Lemma 11.10 *Let $\alpha(x)$ be an arithmetical formula with only x free, $h(y_1, \ldots, y_n)$ be an n-ary primitive recursive function and f be a Proponent's effective strategy. Then f "can be made" stable for $(\alpha(x), f(y_1, \ldots, y_n))$. More precisely, there is a Proponent's effective strategy f' stable for $(\alpha(x), f(y_1, \ldots, y_n))$ such that for any number a, f' solves $\alpha(a)$ whenever f does.*

PROOF. Assume $a = h(b_1, \ldots, b_n)$. Here is an informal description of Proponent's strategy f':

In a $\alpha(a)$-play, f' acts exactly as f. In a $\alpha(h(b_1, \ldots, b_n))$-play, Proponent pretends that he sees "a" instead of "$h(b_1, \ldots, b_n)$", and again acts exactly as he would act in a $\alpha(a)$-play following strategy f: chooses the same disjunct when he sees an additive disjunction, chooses the same parameter when he sees an existential (osub)sentence, etc. Note that this is possible because the sentences $\alpha(a)$ and $\alpha(h(b_1, \ldots, b_n))$ have exactly the same logical structure and they only differ by their atoms (see the end of the second paragraph of 3.6); also, any atomic sentence which emerges in this $\alpha(h(b_1, \ldots, b_n))$-play with strategy f' has the same truth value (label) as the corresponding atom in the corresponding $\alpha(a)$-play with strategy f. Clearly then f' is an effective solution to $\alpha(h(b_1, \ldots, b_n))$ whenever f (and f') is so to $\alpha(a)$. ♣

12 Safe countertrees

Throughout this section "(hyper)sentence" means that of L.

Terminology and notation 12.1 By a *tree* we mean a set T of natural numbers, whose elements are called *nodes* of T, together with a binary relation $\prec$ on T such that:

- $\prec$ is transitive and irreflexive;
- if $a \prec c$ and $b \prec c$, then either $a \prec b$ or $b \prec a$ or $a = b$ (all $a, b, c \in T$);
- there is a node $r \in T$, called the *root* of T, such that $r \prec a$ for all $r \neq a \in T$.

If there are no infinite or arbitrarily long $\prec$-chains in T, we say that T has a *finite height* and call the length h of the longest (a longest) chain $a_1 \prec \ldots \prec a_h$ the *height* of the tree.

$a \preceq b$ is defined as $a \prec b$ *or* $a = b$. If $a \preceq b$ (resp. $a \prec b$), we say that a is an *ancestor* (resp. *proper ancestor*) of b, and b is a *descendant* (resp. *proper descendant*) of a.

If $a \prec b$ and there is no c with $a \prec c \prec b$, we say that a is the *parent* of b and b is a *child* of a.

We say that a and b are *siblings*, if a and b have a common parent. A *proper sibling* of a is any sibling of a except a itself.

A *tree of sentences* is a tree T with each node a of which is associated a sentence $\widehat{a}$, called the *content* of a. We often identify nodes with their contents, although it should be remembered that different nodes do not necessarily have different contents.

Definition 12.2 A sentence is said to be *safe*, if no atom has two weak surface occurrences in it (recall 6.1.2).

E.g., $P(3) \bigtriangleup \neg P(2)$ is safe, but $P(3) \bigtriangleup \neg P(3)$ is not.

Definition 12.3 A *countertree* for a sentence α is a tree T of sentences (which are viewed as clean hypersentences, — see 6.1.6) with the root α such that for any node β:

- if β is 1-like, then it has exactly one child, and this child is a 1-hyperdevelopment of β;
- if β is 0-like, then every clean 0-hyperdevelopment of β is a child of β and every child of β is a clean 0-hyperdevelopment of β.

Such a tree is said to be *safe*, if every node of it is safe. And this tree is said to be *primitive recursive*, if the relations "... is a node of T", "... is the content of ...", "... is the parent of ...", "... is a sibling of ..." are primitive recursive.

Remark 12.4 Note that a child is always a hyperdevelopment of its parent which, by 6.7a, implies that a countertree always has a finite height.

In the notion of translation defined in Section 5, instead of the language of arithmetic can be taken any other language, including the language L whose formulas are to be translated. Below we define a translation from L into L:

Definition 12.5 Let α be a sentence with a hypercomplexity n, and $z_1, \ldots, z_{2n}$ be the first $2n$ variables (in the alphabetical list of variables) not occuring in α. Then we define the *plus-translation* $+$ for α by stipulating that for any m-ary predicate letter P of α,

$$+P = \forall z_1 \exists z_2 \ldots \forall z_{2n-1} \exists z_{2n} P(x_1, \ldots, x_m, z_1, \ldots, z_{2n}).$$

(We assume here that if P is an m-ary predicate letter of L, then it is an $m+2n$-ary predicate letter as well; if this seems confusing, we can choose any "safe" $(m+2n)$-ary predicate letter P' instead of P in the right-hand side of the above equation).

Lemma 12.6 *If a sentence α does not belong to ET, then there is a safe primitive recursive countertree for α^+, where $+$ is the plus-translation for α.*

PROOF.[8] Assume $\alpha \notin ET$. We simultaneously define a tree T of sentences and a function which assigns to each node β of T a hypersentence β' called the *image* of β, such that the following two conditions are satisfied:

Condition 1. If $\beta = A!(\gamma_1, \ldots, \gamma_k)$, then $\beta' = A!(\delta_1, \ldots, \delta_k)$, where for each $1 \leq i \leq k$:

- a) if δ_i is not a sliteral, then $\gamma_i = \delta_i^+$;
- b) if δ_i is a sliteral, then γ_i is an rt-development of δ_i^+ (see 2.2.2; note that in which model we take an rt-development does not matter here).

Condition 2. $\beta' \notin ET$.

Extending the usage of the word "image", we also say that each δ_i in the condition 1 above is the image of γ_i; if the image of γ_i is an osliteral, then we say that γ_i is a *quasiosliteral*. We can say that a quasiosliteral is positive, negative, single, or married, if its image is so, and that two quasiosliterals are spouses to each other, if their images are so.

According to the condition 1b, every quasiosliteral with an image $P(\vec{a})$ (where $P(\vec{a})$ is a positive or negative sliteral), is

$$Q_{i+1}z_{i+1} \ldots Q_{2n}z_{2n}P(\vec{a}, b_1, \ldots, b_i, z_{i+1}, \ldots, z_{2n})$$

for some $0 \leq i \leq 2n$ and $b_1, \ldots, b_i$ (where the Q_j are alternating quantifiers). We call the number i the *age* of the quasiosliteral, and if $i \geq 2$, we say that the quasiosliteral is *aged*. And we call a parameter b_j $(1 \leq j \leq i)$ *the z_j-parameter* of the quasiosliteral.

We put α^+ to be the root of T and α to be its image. Clearly the conditions 1 and 2 are satisfied.

Now, suppose $\beta = A!(\gamma_1, \ldots, \gamma_k)$ is a node of T, $\beta' = A!(\delta_1, \ldots, \delta_k)$ is its image and the conditions 1 and 2 are satisfied for β and β'. Below we define the set of children of β and the images of these children; at the same time we verify that β satisfies the conditions of Definition 12.3 and the children of β together with their images satisfy the conditions 1 and 2. Here we go:

Case 1: Suppose β is 1-like and:

[8] This proof takes all the rest of this section. Reading it is not necessary for understanding the material in the remaining sections.

Subcase 1a: β' is 1-like. Then (as $\beta' \notin ET$) there is a 1-hyperdevelopment $\phi' \notin ET$ of β'; if there is more than one such hyperdevelopment, we choose the smallest one. We have

$$\phi' = A(\delta_1, \ldots, \delta_{i-1}, \delta, \delta_{i+1}, \ldots, \delta_k)$$

for some $1 \leq i \leq k$, where δ is a (hyper)development of the $\wedge$- or $\forall$-sentence δ_i. Then, as it is easily seen, the formula

$$\phi = A(\gamma_1, \ldots, \gamma_{i-1}, \delta^+, \gamma_{i+1}, \ldots, \gamma_k)$$

is a 1-hyperdevelopment of β. We set ϕ to be the only child of β and ϕ' to be its image. Note that the conditions 1 and 2 are satisfied for ϕ and ϕ', and β satisfies the conditions of Definition 12.3.

Subcase 1b: β' is 0-like. Let then l be the smallest hyperlabeling for β' such that $l(\beta') = 0$. It is easy to see that there is then a smallest $1 \leq i \leq k$ such that δ_i is an osliteral with $l(\delta_i) = 0$, and γ_i has the opposite hyperlabel (for any hyperlabeling, because β is clean), which means — taking into account that γ_i is a rt-development of δ_i^+ and the latter is a sliteral with a quantifier prefix — that γ_i has the form $\forall z_j \gamma(z_j)$ for some $1 \leq j \leq 2n$. We then choose the least parameter a not occuring in β and stipulate that

$$\phi = A(\gamma_1, \ldots, \gamma_{i-1}, \gamma(a), \gamma_{i+1}, \ldots, \gamma_k)$$

is the only child of β and the selfsame β' is its image. Clearly the conditions 1 and 2 continue to hold for ϕ and β', and β satisfies the conditions of Definition 12.3.

Case 2: Suppose β is 0-like. Then we stipulate that all clean 0-hyperdevelopments of β, and only those, are children of β. Thus, β satisfies the conditions of 12.3.

Consider any clean 0-hyperdevelopment (child) ϕ of β. We have

$$\phi = A(\gamma_1, \ldots, \gamma_{i-1}, \gamma, \gamma_{i+1}, \ldots, \gamma_k)$$

for some i, where γ_i is a $\vee$- or $\exists$-sentence and γ is its development. In order to define the image ϕ' of ϕ, we need to consider the following subcases:

Subcase 2a: γ_i is not a quasiosliteral. Then it is clear that there is δ such that $\gamma = \delta^+$ and δ is a 0-hyperdevelopment of δ_i. Then we define

$$\phi' = A(\delta_1, \ldots, \delta_{i-1}, \delta, \delta_{i+1}, \ldots, \delta_k).$$

Subcase 2b: γ_i is a single quasiosliteral of age 1, and there is an (exactly one single) aged quasiosliteral γ_j in ϕ such that δ_i and δ_j are opposite to each other and γ_j and γ have the same z_1- and z_2-parameters. Then ϕ' is the marriage-extension of β' by means of adding to the set of married couples of the latter the couple (δ_i, δ_j).

Subcase 2c. In all the remaining cases we define $\phi' = \beta'$.

The condition 1, as it is easily seen, is satisfied for ϕ and ϕ' in each of the above subcases (a), (b) and (c); as $\beta' \notin ET$, the condition 2 is trivially satisfied in the subcase (c), and it is also satisfied in the subcases (a) and (b) because ϕ' is a 0-hyperdevelopment of β' (see 6.15).

Since α is the root of T and, as we established above, every node β of T satisfies the conditions of 12.3, T is a countertree for α. And it is evident that this tree is primitive recursive. It remains to show that T is safe.

Suppose $\alpha = A!(\alpha_1, \ldots, \alpha_k)$ is a node of T, α_i is a quasiosliteral of α and β is a descendant of α. The definition of countertree easily implies that then β has the form $A(\beta_1, \ldots, \beta_k)$; we extend the usage of the words "ancestor" and "descendant" to osubsentences of the nodes of T and say that α_i $(1 \leq i \leq k)$ is an ancestor of β_i and β_i is a descendant of α_i. In a similar way, we extend the usage of the words "parent" and "child" to quasiosliterals.

An analysis of our step-by-step construction of T easily convinces as that the following lemma holds for T:

Lemma 12.7

a) *If a node α is a child of a node β and α', β' are the images of α, β, respectively, then either $\alpha' = \beta'$ or α' is a hyperdevelopment of β'.*

b) *If γ is a (surface) quasiosliteral of a node, then so are all its descendants, and they all have the same image.*

c) *A surface osubsentence γ of a node is a quasiosliteral iff γ is $Q_i z_i \ldots$-$Q_{2n} z_{2n} \delta$ for some literal δ and some $1 \leq i \leq 2n$, where the Q_j are alternating quantifiers.*

d) *If two quasiosliterals of a node are spouses to each other, then so are their descendants.* (This follows from (a) and (b))

Lemma 12.8 *Suppose β is a node of T and γ and δ are different positive (resp. negative) aged quasiosliterals of β. Then the z_1- (resp. z_2-) parameters of γ and δ are different.*

PROOF. Taking 12.7c into account, one can verify that if γ and β are positive (resp. negative) aged quasiosliterals, then the z_1- (resp. z_2-) parameters have appeared in them — that is, in their ancestors — by the subcase 1b; but, on one hand, a z_i-parameter, once it has appeared, never disappears in the descendants, and, on the other hand, the subcase 1b never introduces an already existing parameter, so we conclude that the z_1- (resp. z_2-) parameters of γ and δ must be different. ♣

Lemma 12.9 *Suppose β is a node of T and γ and δ are quasiosliterals of β. Then γ and δ are spouses to each other if and only if the following two conditions are satisfied:*

a) both γ and δ are aged;

b) γ has the same z_1- and z_2-parameters as δ.

PROOF. ($\Rightarrow$:) It is only the subcase 2b where marriages happen. In view of 12.7d and 12.7a, the conditions of the subcase 2b then immediately imply both clauses (a) and (b) of the lemma.

($\Leftarrow$:) We may suppose that γ became aged earlier than δ. More precisely, for some $\beta', \beta'', \gamma', \gamma'', \delta', \delta''$ we have: β', β'' are ancestor nodes of β, β'' is a child of β', γ' and γ'' are the ancestors of γ in β' and β'', respectively, and δ' and δ'' are the ancestors of δ in β' and β'', respectively; $\gamma' = \gamma''$ and they are aged; δ' is of age 1 and δ'' is of age 2. By 12.7c, $\gamma', \gamma'', \delta', \delta''$ are quasiosliterals, and by 12.7b, $\gamma', \gamma'', \gamma$ have a common image and so do $\delta', \delta'', \delta$. Note that the subcases 1a and 2a never deal with quasiosliterals and the subcase 1b never introduces an already existing parameter. Therefore, the transfer from β' to β'' can only be determined either by the subcase 2b or by the subcase 2c. If this is the subcase 2b, we are done, because that means that γ'' and δ'' marry each other in β'' and therefore, by 12.7d, γ and δ are spouses to each other. It remains to show that the subcase 2c is ruled out, i.e. that the transfer from β' to β'' is really determined by the subcase 2b. Suppose not. Then an analysis of the conditions of the subcase 2b shows that we should have one of the following:

i) There is a quasiosliteral $\xi \neq \gamma'$ in β' with the same z_1- and z_2-parameters as δ'' such that the image of ξ is opposite to the image of δ', or

ii) Either γ' or δ' is married.

In the case (i), note that ξ and γ' have the same z_1- and z_2-parameters and equal images, which, by 12.8, is a contradiction.

Now, to rule out the case (ii), first notice that since δ' is not aged, by 12.9a, δ' is not married. And if γ' is married, then (the child of) its spouse has the same (same as a sentence) image and (by 12.9b) the same z_1- and z_2-parameters as δ'', which, again by 12.8, is a contradiction. ♣

Let β be a node of T and (γ_0, γ_1) — a married couple of quasiosliterals of β. Then we'll say that this couple, as well as each γ_i, is *spoiled* (in β), if for some m, one of the following holds:

a) both γ_i are of (not necessarily the same) age $\geq m$ and their z_m-parameters are different, or

b) γ_i is of age $\geq m$, γ_{1-i} of age $< m$ and the variable z_m in γ_{1-i} is bound by $\forall$.

Lemma 12.10 *Let β be a node of T, δ a child of β and (γ_0, γ_1) a married couple of quasiosliterals of β. Then, if this couple is spoiled in β, so is it (i.e. the couple of the children of γ_0,γ_1) in δ.*

PROOF. Indeed, if the reason of γ_0 and γ_1's being spoiled in β is the clause (a) of the definition of "spoiled", then clearly the same reason will work in δ, too. And if the reason for being spoiled in β is the clause (b) of the definition of "spoiled", then, again, the same reason remains in δ unless δ results from β by deleting $\forall z_m$ in γ_{1-i} and replacing the variable z_m by a parameter a (this can only happen in the subcase 1b of the definition of T). But then, according to the conditions of the subcase 1b, this z_m-parameter a is different from any already existing parameter of β, including the z_m-parameter of γ_i. This means that the interesting us couple is spoiled in δ on the basis of the clause (a) of the definition of "spoiled". ♣

Lemma 12.11 *Suppose β is a node of T and γ is a married quasiosliteral of β whose first quantifier is $\forall$ and whose age is m. Suppose also that β' is a descendant of β and the descendant γ' of the quasiosliteral γ in β' is of age $m+2$. Then β and β' have different images unless γ' is spoiled.*

PROOF. For a contradiction, deny this. γ and γ' must have the forms $\forall z_{m+1}\exists z_{m+2}\omega(z_{m+1}, z_{m+2})$ and $\omega(a,b)$, respectively. We must have nodes β_1, β_2 with $\beta \preceq \beta_1 \prec \beta_2 \prec \beta'$, where β_2 is a child of β_1, such that, if we denote by γ_1 and γ_2 the descendants of γ in β_1 and β_2, respectively, we have $\gamma_1 = \gamma = \forall z_{m+1}\exists z_{m+2}\omega(z_{m+1}, z_{m+2})$ and $\gamma_2 = \exists z_{m+2}\omega(a, z_{m+2})$. Note that all the four nodes: $\beta, \beta_1, \beta_2, \beta'$ have the same image I_β, and that, by 12.7c and 12.7b, all the four osliterals $\gamma, \gamma_1, \gamma_2, \gamma'$ are quasiosliterals with a common image I_γ. Note also that the transfer from β_1 to β_2 can only be determined by the subcase 1b, which means that I_β is 0-like. Let then l be the smallest hyperlabeling with $l(I_\beta) = 0$. Let δ and δ' be the spouses of γ and γ', respectively. By 12.9a, 12.7c and 12.7b, these two osliterals are quasiosliterals with a common image I_δ which clearly (as a sentence) equals $\neg I_\gamma$. The subcase 1b implies that $l(I_\gamma) = 0$ and hence $l(I_\delta) = 1$.

As we assume, γ' is not spoiled. Then 12.10 implies that γ is not spoiled, either. Remembering that γ is an rt-development of I_γ^+ and δ is an rt-development of $I_\delta^+ = \neg I_\gamma^+$, a little analysis of the definition of "spoiled" convinces us that then $\delta = \exists z_{m+1}\forall z_{m+2}\neg\omega(z_{m+1}, z_{m+2})$ and $\delta' = \neg\omega(a,b)$. This means that there are nodes β_3, β_4 with $\beta \prec \beta_3 \prec \beta_4 \preceq \beta'$, where β_4 is a child of β_3, such that, if we denote by δ_3 and δ_4 the descendants of δ in β_3 and β_4, respectively, we have $\delta_3 = \forall z_{m+2}\neg\omega(a, z_{m+2})$ and $\delta_4 = \neg\omega(a,b)$. Clearly I_δ is the image of both δ_3, δ_4. Again, the transfer from β_3 to β_4 could only have taken place by the subcase 1b, with δ_3 in the role of γ_i of the latter. But this is impossible because, as we already know, $l(I_\delta) = 1$, whereas 1b requires that $l(I_\delta) = 0$.

Lemma 12.12 *T is safe.*

PROOF. Suppose, for a contradiction, β is a node of T containing two surface occurrences of one and the same atom. Denote the corresponding osubsentences — of course they must be quasiosliterals — by γ and δ. The age of these quasiosliterals is $2n$ and, as $n \neq 0$, they are aged, which, by 12.9 means that they are spouses to each other. Let β_1 be the most remote ancestor of β in which (the ancestors of) γ and δ first became spouses. The age of at least one of these two spouses in β_1 should be exactly 2 (and 1 in the parent of β_1), for otherwise 12.9 implies that their parents (already) were spouses to each other. We may suppose that γ is the quasiosliteral whose ancestor in β_1 was of age 2. Let us then fix proper descendants $\beta_2 \prec \ldots \prec \beta_n \preceq \beta$ of the node β_1 such that for each $2 \leq j \leq n$, the age of the ancestor of γ in β_j is $2j$. Since γ is not spoiled in β, by 12.10, its ancestors in $\beta_1, \ldots, \beta_n$ are not spoiled either, whence 12.11 easily implies that the image β'_j of each such β_j is different from the image β'_{j-1} of β_{j-1}. Then, by 12.7a, we have that $\beta'_1 H \ldots H \beta'_n$, where H is the transitive closure of the hyperdevelopment relation. This means that the hypercomplexity of β'_1 is $\geq n$. But the hypercomplexity of β'_1 is less than that of the original formula α (for, β'_1 contains a married couple but α does not). Consequently, the hypercomplexity of α is greater than n, which is a contradiction (recall that n was just the hypercomplexity of α). ♣

Lemma 12.12 completes the proof of Lemma 12.6. ♣

Now we are ready for a finishing stroke.

13 Proof of Theorem 5.5(iii)⇒(i)

This part of Theorem 5.5 could be called the arithmetical completeness of ET.

Lemma 13.1 *If there is a safe primitive recursive countertree for a sentence α of L, then there is an arithmetical translation τ for α such that α^τ is not effectively true in the standard model of arithmetic.*

PROOF. Let us fix a safe primitive recursive countertree T for α. For simplicity and without loss of generality we assume that T is infinite, every natural number is its node and 0 is its root. From now on, “node”, “parent” etc. will mean those in T.

We also fix the set $\{P_1, \ldots, P_n\}$ of all predicate letters of α. And for each i with $1 \leq i \leq n$ we say that a sentence β of arithmetic is P_i*-appropriate*, if β contains exactly as many free variables as the arity of P_i, and if, at the same time, β does not contain variables (no matter free or bound) occuring in α.

It is convenient to assume throughout this proof that there are no predicate letters or variables in our language L other than those ocurring in α.

First we need to introduce notations for some functions and relations.

- By abuse of notation, we use $\widehat{x}$ to denote the function of x that returns the code of the content of node x (recall from 12.1 that we also use the same expression to denote this content itself).

- $transl(x, t_1, \ldots, t_n)$ is the $n+1$-place function that returns the code of the sentence S^τ, where S is the x-coded sentence and τ is the translation that is defined exactly for the predicate letters $P_1, \ldots, P_n$ and assigns to each P_i the t_i-coded P_i-appropriate formula (according to Convention 11.8, if t_i does not code such a formula, then the function returns 0).

- $Parent(x, y)$ is the relation "x is the parent of y".

- $Sibl(x, y)$ is the relation "x and y are siblings".

- $Contains(x, y)$ is the relation "The x-coded sentence has a surface occurrence of the y-coded sentence".

- For a formula $\beta(x_1, \ldots, x_k)$ of arithmetic or of L with exactly $x_1, \ldots, x_k$ free, we use $[\beta]$ (or $[\beta(x_1, \ldots, x_k)]$) to denote the k-place primitive recursive function that assigns to each k-tuple $a_1, \ldots, a_k$ of numbers the code of the sentence $\beta(a_1, \ldots, a_k)$; we assume that the *term* (see 3.6) $[\beta(x_1, \ldots, x_k)]$ has exactly the same free variables $x_1, \ldots, x_k$ as the formula $\beta(x_1, \ldots, x_k)$. Note that if β has no free variables, then $[\beta]$ is simply the code of β.

- $su(x, y_1, \ldots, y_n)$ is the $(n+1)$-place function that returns the code of the sentence that is the result of respectively substituting the numbers $y_1, \ldots, y_n$ for the variables "t_1",…, "t_n" in the x-coded formula.

- $(s)_i$ is the 2-place function that returns the ith term of the s-coded finite nonempty sequence of numbers.

- $lh(s)$ is the function that returns the length of the s-coded finite nonempty sequence of numbers.

It is easily seen that these functions and relations are primitive recursive.

Now, recalling the definition of the predicate $Solves$ from Section 11, abbreviating $t_1, \ldots, t_n$ by $\vec{t}$ and using "$\phi \to \psi$" for "$\neg\phi \vee \psi$", we define

$$Spec(v, \vec{t}) \equiv \exists s\Big(Solves(s, transl(\widehat{v}, \vec{t})) \wedge \forall v' \forall s' \big(2^{v'} \cdot 3^{s'} < 2^v \cdot 3^s \wedge Sibl(v, v') \to \neg Solves(s', transl(\widehat{v'}, \vec{t}))\big)\Big).$$

Explanation: The function $2^v \cdot 3^s$ is used to encode pairs (v, s) of natural numbers. And $\vec{t}$ is in fact a variable for translations. Introducing some apparently not very adequate jargon, we can say that $Spec(v, \vec{t})$ asserts that, under the translation $\vec{t}$, the node v has the shortest effective solution among its siblings. More precisely this means that the $\vec{t}$-translation of $\widehat{v}$ has an effective solution s such that there is no pair (v', s') smaller than (s, v) where v' is a sibling of v and s' is an effective solution to the $\vec{t}$-translation of $\widehat{v'}$.

Next, we define

$$
\begin{aligned}
Superspec(v,\vec{t}) \quad \equiv \quad & \exists s\Big((s)_1 = 0 \wedge (s)_{lh(s)} = v \wedge \\
& \forall i_{1 \le i < lh(s)} Parent((s)_i,(s)_{i+1}) \wedge \\
& \forall i_{1 \le i \le lh(s)} Spec((s)_i,\vec{t})\Big).
\end{aligned}
$$

Using our jargon, $Superspec(v,\vec{t})$ asserts that under the translation $\vec{t}$, every ancestor of the node v (including v) has the shortest effective solution among its siblings.

For every i with $1 \le i \le n$, let m_i be the code of the formula

$$
\begin{aligned}
& \exists v\Big(Contains\big(\widehat{v},[\neg P_i(z_1,\ldots,z_{k_i})]\big) \wedge \\
& Superspec\big(v, su(t_1,t_1,\ldots,t_n),\ldots,su(t_n,t_1,\ldots,t_n)\big)\Big),
\end{aligned} \tag{19}
$$

where k_i is the arity of P_i and where we assume that none of the variables of this formula occurs in α.

We now define the translation τ: τ is a finite function defined just for $P_1,\ldots,P_n$ such that for every i with $1 \le i \le n$,

$$
\begin{aligned}
\tau(P_i) \quad = \quad & \exists v\Big(Contains(\widehat{v},[\neg P_i(z_1,\ldots,z_{k_i})]) \wedge \\
& Superspec\big(v, su(m_1,m_1,\ldots,m_n),\ldots,su(m_n,m_1,\ldots,m_n)\big)\Big).
\end{aligned}
$$

Lemma 13.2 *For each i with $1 \le i \le n$,*

$$
su(m_i,m_1,\ldots,m_n) = [\big(P_i(z_1,\ldots,z_{k_i})\big)^{\tau}].
$$

PROOF. Indeed, $su(m_i,m_1,\ldots,m_n)$ represents the code of the formula which is the result of respectively substituting the numbers $m_1,\ldots,m_n$ for the variables "t_1",…,"t_n" in the m_i-coded formula, i.e. in the formula (19), and $\big(P_i(z_1,\ldots,z_{k_i})\big)^{\tau}$ is just such a formula. ♣

We now introduce the following abbreviation:

$$
x^{\tau} \equiv transl\big(x, su(m_1,m_1,\ldots,m_n),\ldots,su(m_n,m_1,\ldots,m_m)\big).
$$

Lemma 13.3 *For any sentence γ, $[\gamma]^{\tau} = [\gamma^{\tau}]$.*

PROOF. Indeed, in view 13.2, x^{τ} is the function of x that returns the code of S^{τ}, where S is the x-coded sentence. ♣

Let now

$$Special(v) \equiv$$
$$Spec\big(v, su(m_1, m_1, \ldots, m_n), \ldots, su(m_n, m_1, \ldots, m_n)\big)$$

and

$$Superspecial(v) \equiv$$
$$Superspec\big(v, su(m_1, m_1, \ldots, m_n), \ldots, su(m_n, m_1, \ldots, m_n)\big).$$

That is, if we unfold the abbreviations *Spec* and *Superspec* in the above formulas and use the abbreviation x^τ, we have:

$$\begin{aligned} Special(v) \;\; \equiv \;\; & \exists s \Big(Solves(s, \widehat{v}^\tau) \wedge \forall v' \forall s' \\ & \big(2^{v'} \cdot 3^{s'} < 2^v \cdot 3^s \wedge Sibl(v, v') \rightarrow \neg Solves(s', \widehat{v'}^\tau)\big)\Big) \end{aligned}$$

and

$$\begin{aligned} Superspecial(v) \;\; \equiv \;\; \exists s \Big(& (s)_1 = 0 \wedge \\ & (s)_{lh(s)} = v \wedge \\ & \forall i_{1 \leq i < lh(s)} Parent\big((s)_i, (s)_{i+1}\big) \wedge \\ & \forall i_{1 \leq i \leq lh(s)} Special\big((s)_i\big)\Big). \end{aligned}$$

We will say that a node c is *special*, if $Special(c)$ is true which, using our jargon, means that under the translation τ, c has the shortest effective solution among its siblings.

And we say that a node c is *superspecial*, if $Superspecial(c)$ is true, i.e. c and all its ancestors are special.

Then, for each $1 \leq i \leq n$ and $a_1, \ldots, a_{k_i}$, $P_i^\tau(a_1, \ldots, a_{k_i})$ asserts that there is a superspecial node with a surface occurrence of the sliteral $\neg P_i(a_1, \ldots, a_{k_i})$.

Lemma 13.4 *For any number p_v, if $Special(p_v)$ is true, then it is effectively true.*

PROOF. Suppose $Special(p_v)$ is true. Let then p_s be a parameter for which the sentence

$$\begin{aligned} & Solves(p_s, \widehat{p_v}^\tau) \wedge \forall v' \forall s' \\ & \big(2^{v'} \cdot 3^{s'} < 2^{p_v} \cdot 3^{p_s} \wedge Sibl(p_v, v') \rightarrow \neg Solves(s', \widehat{v'}^\tau)\big) \end{aligned} \tag{20}$$

is true. Then both its conjuncts are true. Let

$$(a_1, b_1), \ldots, (a_k, b_k)$$

be all pairs (a,b) such that $2^a \cdot 3^b < 2^{p_v} \cdot 3^{p_s}$ and a is a sibling of p_v. The truth of the second conjunct of (20) means that for each $1 \leq i \leq k$, the Σ_2-sentence $\neg Solves(b_i, \widehat{a_i}^{\tau})$ is true and hence, by 11.5, has an effective solution g_i. Fix this finite list

$$g_1, \ldots, g_k$$

of effective solutions to

$$\neg Solves(b_1, \widehat{a_1}), \ldots, \neg Solves(b_k, \widehat{a_k}).$$

Now we describe Proponent's strategy for $Special(p_v)$: First, Proponent goes from the position $Special(p_v)$ to the position (20) (i.e. deletes "$\exists s$" and in what remains substitutes the parameter p_s for the variable s). By 2.9, it suffices to show that Proponent's strategy solves (20), and this means that it solves both its conjuncts. But the first conjunct is a true Π_2-sentence and, therefore, Proponent can use strategy $UNIV$ for it (see 11.4). As for the second conjunct, Proponent's strategy should be able to solve the sentence

$$\neg 2^a \cdot 3^b < 2^{p_v} \cdot 3^{p_s} \vee \neg Sibl(p_v, a) \vee \neg Solves(b, \widehat{a}^{\tau}) \tag{21}$$

for any parameters a, b. If one of the first two disjuncts of this disjunction is true, let Proponent go from (21) to this disjunct and then use strategy $UNIV$. Otherwise, i.e. if $2^a \cdot 3^b < 2^{p_v} \cdot 3^{p_s}$ and a and p_v are siblings, then for some $1 \leq i \leq k$, $(a,b) = (a_i, b_i)$. Let in this case Proponent go to the third conjunct and then use the strategy g_i for it. This strategy is an effective solution to $Special(p_v)$. ♣

Lemma 13.5 *For any node a, if $Superspecial(a)$ is true, then it is effectively true.*

PROOF. Suppose a is superspecial. Let the nodes $a_1, \ldots, a_m$ be such that a_1 is the root, $a_m = a$ and for each $1 \leq i < m$, a_i is the parent of a_{i+1}. And let b be the code of the sequence $a_1, \ldots, a_m$. For each $1 \leq i \leq m$, a_i is special and hence, by 13.4, $Special(a_i)$ is effectively true, so let us fix effective solutions

$$g_1, \ldots, g_m$$

to

$$Special(a_1), \ldots, Special(a_m).$$

Here is Proponent's strategy for $Superspecial(a)$, i.e. for

$$\begin{aligned} \exists s\Big(\quad & (s)_1 = 0 \wedge \\ & (s)_{lh(s)} = a \wedge \\ & \forall i_{1 \leq i < lh(s)} Parent\big((s)_i, (s)_{i+1}\big) \wedge \\ & \forall_{1 \leq i \leq lh(s)} Special\big((s)_i\big) \quad \Big): \end{aligned}$$

First, Proponent deletes "$\exists s$" and substitutes the parameter b for the variable s, coming to the position

$$\begin{aligned}&(b)_1 = 0 \wedge \\ &(b)_{lh(b)} = a \wedge \\ &\forall i_{1\le i<lh(b)} Parent\big((b)_i, (b)_{i+1}\big) \wedge \\ &\forall i_{1\le i\le lh(b)} Special\big((b)_i\big).\end{aligned} \tag{22}$$

Now Proponent must have a winning strategy for each of the four conjuncts of this sentence. The first three conjuncts are true Δ_0-sentences and strategy $UNIV$ (as well as $DZERO$) can be used for them. As for the fourth conjunct, i.e.

$$\forall i\big(\neg 1 \le i \le lh(b) \vee Special((b)_i)\big),$$

to solve it means to solve, for each c, the true sentence

$$\neg 1 \le c \le lh(b) \vee Special((b)_c).$$

This is done as follows: if the first conjunct, which is a Δ_0-sentence, is true, then Proponent chooses it and uses for it strategy $UNIV$. Otherwise Proponent chooses the second conjunct and uses the strategy g_c for it; in view of 11.10, we may suppose that g_c is stable for $(Special(x), (y)_z)$, and this means that g_c solves $Special((b)_c)$. ♣

Lemma 13.6 *If a node b is special, then there is an effective function which is a solution to every sentence $\neg Special(a)$ where a is a proper sibling of b.*

PROOF. Suppose b is special. Let us then fix p such that

$$Solves(p, \widehat{b}) \text{ is true} \tag{23}$$

and for any sibling a of b,

$$\forall s'\big(\neg 2^a \cdot 3^{s'} < 2^b \cdot 3^p \vee \neg Solves(s', \widehat{a}^\tau)\big) \text{ is true.} \tag{24}$$

Let then $(a_1, d_1), \ldots, (a_k, d_k)$ be all pairs (a, d) such that $2^a \cdot 3^d < 2^b \cdot 3^p$. By (24), for each $1 \le i \le k$, the Σ_2-sentence $\neg Solves(d_i, \widehat{a_i}^\tau)$ is true and, by 11.5, has an effective solution g_i. Fix these solutions

$$g_1, \ldots, g_k$$

to

$$\neg Solves(d_1, \widehat{a_1}^\tau), \ldots, \neg Solves(d_k, \widehat{a_k}^\tau).$$

We now describe a Proponent's strategy which is a solution to every sentence $\neg Special(a)$ where a is a sibling of b. A strategy solves $\neg Special(a)$, i.e.

$$\forall s\Big(\neg Solves(s,\widehat{a}^{\tau}) \vee$$
$$\exists v'\exists s'\big(2^{v'}\cdot 3^{s'} < 2^{a}\cdot 3^{s} \wedge Sibl(a,v') \wedge Solves(s',\widehat{v'}^{\tau})\big)\Big),$$

if it solves the sentence

$$\neg Solves(d,\widehat{a}^{\tau}) \vee \exists v'\exists s'$$
$$\big(2^{v'}\cdot 3^{s'} < 2^{a}\cdot 3^{d} \wedge Sibl(a,v') \wedge Solves(s',\widehat{v'}^{\tau})\big) \tag{25}$$

for any d.

This is how Proponent should act for (25):

If the pair (a,d) is such that a is a sibling of b and $2^b\cdot 3^p < 2^a\cdot 3^d$, then let Proponent choose the right disjunct of (25), then delete in it "$\exists v'\exists s'$" and in what remains substitute b and p for v' and s', respectively. The play comes to the position

$$2^b\cdot 3^p < 2^a\cdot 3^d \wedge Sibl(a,b) \wedge Solves(p,\widehat{b}^{\tau}). \tag{26}$$

Now Proponent must be able to solve each of the three conjuncts of (26). But these conjuncts are true Π_2-sentences, so Proponent can use $UNIV$.

Suppose now a is a sibling of b and not $2^b\cdot 3^p < 2^a\cdot 3^d$, which means that (as $a\neq b$) $2^a\cdot 3^d < 2^b\cdot 3^p$ and thus, for some $1\leq i\leq k$, $(a,d)=(a_i,c_i)$. Then Proponent chooses the left disjunct of (25) and then uses the strategy g_i for it.

It is easily seen that we have just described an effective solution to any $\neg Special(a)$ where a is a sibling of b. ♣

We say that a node a of T is a *nonbranch relative* of a node b, if $a\neq b$ and a is neither an ancestor nor a descendant of b, i.e. a and b don't belong to one branch of the tree. Otherwise a and b are *branch relatives*.

Lemma 13.7 *If a node b is superspecial, then there is an effective function which is a solution to every sentence $\neg Superspecial(a)$ where a is a nonbranch relative of b.*

PROOF. Let $b_1,\ldots,b_m$ be such that b_1 is the root, $b_m=b$ and, for each e with $1\leq e<m$, b_e is the parent of b_{e+1}. Since b is superspecial, all the b_e are special, which, by 13.6, means that for each $1\leq e\leq m$, there is an effective function g_e which solves $\neg Special(d)$ for every proper sibling d of b_e. Let us fix these functions

$$g_1,\ldots,g_m.$$

Our strategy must solve the sentence

$$\neg(c)_1 = 0 \vee$$

$$\neg(c)_{lh(c)} = a \vee$$
$$\exists i_{1\leq i<lh(c)} \neg Parent((c)_i, (c)_{i+1}) \vee$$
$$\exists i_{1\leq i\leq lh(c)} \neg Special((c)_i)$$

for any nonbranch relative a of b and any c. Note that if a is a nonbranch relative of a superspecial node, then it is not superspecial, and the above sentence is true. This is how Proponent acts in this case to solve that sentence:

If one of the first three disjuncts is true, then Proponent chooses this disjunct and uses strategy $UNIV$ for it.

Suppose now the first three disjuncts are false. Then Proponent goes to the position

$$\exists i\big(1 \leq i \leq lh(c) \wedge \neg Special((c)_i)\big). \tag{27}$$

Note that in this case c really codes a sequence $c_1, \ldots, c_k$ such that c_1 is the root, $c_k = a$ and for each $1 \leq i < k$, c_i is the parent of c_{i+1}. Let j be the biggest number such that c_j is a common ancestor of a and b (which means that $c_j = b_j$), and let $p = j+1$. Clearly $p \leq m, k$ and b_p and c_p are proper siblings. Proponent effectively finds p and goes from (27) to the sentence

$$1 \leq p \leq lh(c) \wedge \neg Special((c)_p). \tag{28}$$

The first conjunct of (28) is now solved by $UNIV$. And for the second conjunct Proponent uses strategy g_p. In view of 11.10, we may suppose that g_p is stable for $(\neg Special(x), (y)_z)$, and then we have that g_p solves $\neg Special((c)_p)$. ♣

Lemma 13.8 *If ξ is a surface osliteral of a superspecial node, then $\neg\xi^\tau$ is effectively true.*

PROOF. Suppose a is a superspecial node and ξ is a surface osliteral of (the content of) a. We need to consider two cases. Begin with the simpler one.

Case 1: ξ is a negated atom $\neg P(b_1, \ldots, b_m)$. Then $\neg\xi^\tau = (P(b_1, \ldots, b_m))^\tau$ is the true sentence

$$\exists v\big(Contains(\widehat{v}, [\neg P(b_1, \ldots, b_m)]) \wedge Superspecial(v)\big). \tag{29}$$

By 13.5, there is an effective solution g (fix it) to $Superspecial(a)$. The following effective strategy of Proponent's is a solution to (29): first Proponent goes from (29) to

$$Contains(\widehat{a}, [\neg P(b_1, \ldots, b_m)]) \wedge Superspecial(a). \tag{30}$$

Then Proponent uses $UNIV$ for the left conjunct and the strategy g for the right one.

Case 2: ξ is an atom $P(b_1, \ldots, b_m)$. Then $\neg\xi^\tau$ is the sentence

$$\forall v\big(\neg Contains(\widehat{v}, [\neg P(b_1, \ldots, b_m)]) \vee \neg Superspecial(v)\big). \tag{31}$$

First let us check that

$$\textit{Any node with a surface occurrence of } \neg P(b_1,\ldots,b_m) \textit{ is a nonbranch relative of } a. \tag{32}$$

Indeed, suppose, for a contradiction, there is a branch relative c of a with a surface occurrence of $\neg P(b_1,\ldots,b_m)$. It is an evident property of countertrees that if a literal has a surface occurrence in a node, it has a surface occurrence in all descendants of that node. So, if c is an ancestor of a, then $\neg P(b_1,\ldots,b_m)$ must have a surface occurrence in a, too, which is impossible because a is safe and it has a surface occurrence of $P(b_1,\ldots,b_m)$. The case when a is an ancestor of c is symmetric, and (32) is proved.

Proponent acts as follows: after Opponent goes from (31) to the position

$$\neg Contains(\widehat{c},[\neg P(b_1,\ldots,b_m)]) \vee \neg Superspecial(c)$$

for some c, in case the left disjunct is true, Proponent goes to this disjunct and uses $UNIV$ for it; otherwise, by (32), c is a nonbranch relative of the superspecial a, and then Proponent goes to $\neg Superspecial(c)$ and uses the strategy defined in 13.7 for it. ♣

Lemma 13.9 *Suppose a node a is superspecial. Then there is a child b of a such that $\widehat{b}^{\tau}$ is effectively true.*

PROOF. Suppose a is superspecial. Let $\widehat{a} = A!(\beta_1,\ldots,\beta_m,\gamma_1,\ldots,\gamma_k)$, where each β_i is a literal and each γ_i is a nonliteral. By the above lemma, for each $1 \leq i \leq m$, $\neg\beta_i^{\tau}$ is effectively true. And $\widehat{a}^{\tau}$ is effectively true because a is special. Then, using the generalized modus ponens lemma (Lemma 10.1) m times, we get that ψ, where $\psi = A!(0=1,\ldots,0=m,\gamma_1^{\tau},\ldots,\gamma_k^{\tau})$, is effectively true. Note that the arithmetical label of ψ and the hyperlabel of $\widehat{a}$ are equal ("the" hyperlabel because $\widehat{a}$ is clean).

Suppose the hyperlabel of $\widehat{a}$, and hence the label of ψ, is 0. Then there is a development ψ' of ψ which is effectively true, and ψ' must be the result of replacing in ψ some 0-labeled osubsentence $(\gamma_j)^{\tau}$ $(1 \leq j \leq k)$ by a development $(\gamma'_j)^{\tau}$ of it, so we have that

$$A!(0=1,\ldots,0=m,\gamma_1^{\tau},\ldots,\gamma_{j-1}^{\tau},\gamma_j'^{\tau},\gamma_{j+1}^{\tau},\ldots,\gamma_k^{\tau})$$

is effectively true. Then, by generalized modus ponens again (taking into account that the sentences $\neg 0 = 1,\ldots,\neg 0 = m$ are effectively true), we have that

$$A!(\beta_1^{\tau},\ldots,\beta_m^{\tau},\gamma_1^{\tau},\ldots,\gamma_{j-1}^{\tau},\gamma_j'^{\tau},\gamma_{j+1}^{\tau},\ldots,\gamma_k^{\tau})$$

is effectively true. But notice that

$$A!(\beta_1,\ldots,\beta_m,\gamma_1,\ldots,\gamma_{j-1},\gamma'_j,\gamma_{j+1},\ldots,\gamma_k)$$

is a child of a. Thus, a has a child whose τ-translation is effectively true.

Suppose now that the hyperlabel of $\widehat{a}$, and hence the label of ψ, is 1. Then, by the definition of countertree, a has a child c such that $\widehat{c}$ is a 1-hyperdevelopment of $\widehat{a}$. Thus,

$$\widehat{c} = A!(\beta_1, \ldots, \beta_m, \gamma_1, \ldots, \gamma_{j-1}, \gamma'_j, \gamma_{j+1}, \ldots, \gamma_k)$$

for some $1 \leq j \leq k$, where γ_j is 1-hyperlabeled and γ'_j is its 1-hyperdevelopment. Now notice that $\widehat{c}^\tau$ is a relaxed development of $\widehat{a}^\tau$ and, as the latter is effectively true and 1-labeled, $\widehat{c}^\tau$ is effectively true. ♣

Lemma 13.10 *There are no superspecial nodes.*

PROOF. According to the previous lemma, if a is a superspecial node, then it has a child b such that $\widehat{b}^\tau$ has an effective solution p. We may suppose that for any pair (b', p'), if b' is a sibling of b and p' is an effective solution to $\widehat{b'}^\tau$, we have $2^b \cdot 3^p < 2^{b'} \cdot 3^{p'}$. This means that b is superspecial. Thus, any superspecial node must have a superspecial child, and then, if there is a superspecial node, there is an infinite chain of superspecial nodes in which each node is the parent of the next node in the chain, which is impossible because, by 12.4, the tree T has a finite height. ♣

We now can complete the proof of Lemma 13.1. According to Lemma 13.10, the root, whose content is the initial sentence α, is not superspecial, which, by definition (since the root has no predecessors and no siblings) means that simply α^τ is not effectively true. ♣

To complete the proof of Theorem 5.5(iii)⇒(i), suppose $\phi \notin ET$. Then, by Lemma 12.6, there is a primitive recursive countertree for ϕ^+; and then, by Lemma 13.1, there is a translation τ such that $(\phi^+)^\tau$ is not effectively true. Take the composition $*$ of $+$ and τ. Then ϕ^* is not effectively true.

14 ET is strictly between BCK and classical logic

The author has failed to find an axiomatization for ET, and most likely this task cannot be accomplished unless the language of ET is either restricted, as this is done in the next section, or extended by means of new multiplicative-style operators.

In this section we locate an interval to which ET belongs in the hierarchy of known sequential calculi.

Below "(hyper)sentence" will always mean (hyper)sentence of our language L and "parameter", as in the previous sections, will mean natural number.

The reader should have noticed that in our semantics (and in the syntactic description of ET) the formulas $\alpha \bigtriangledown (\beta \bigtriangledown \gamma)$ and $(\alpha \bigtriangledown \beta) \bigtriangledown \gamma$ are the same in all reasonable senses. This enables us to relax the formation rules and allow formulas like $\alpha_1 \bigtriangledown \ldots \bigtriangledown \alpha_n$, dropping all the vacuous parentheses between the α_i (of course, we can do the same with $\bigtriangleup, \vee$ and $\wedge$, too). We also agree to identify $\alpha \bigtriangledown \beta$ with $\beta \bigtriangledown \alpha$. That is, we look at $\alpha_1 \bigtriangledown \ldots \bigtriangledown \alpha_n$ as the application of the operator $\bigtriangledown$ to the *multiset* $\{\alpha_1, \ldots, \alpha_n\}$ rather than to the sequence $\langle \alpha_1, \ldots, \alpha_n \rangle$. A multiset is a set which may contain more than one (but a finite number of) copies of its elements; in other words, a multiset is a sequence in which the order, — but not the quantity, — of its elements is disregarded.

In what follows, Greek capital letters always denote the multiplicative disjunction $\alpha_1 \bigtriangledown \ldots \bigtriangledown \alpha_n$ of some sentences $\alpha_1, \ldots, \alpha_n$, for some $n \geq 0$. If $n = 1$, such a "disjunction" is just α_1, and if $n = 0$, the "disjunction" is empty; $\Theta \bigtriangledown \beta$, where Θ is the empty disjunction, is understood as β.

Lemma 14.1 *For any hypersentences $\Phi \bigtriangledown \phi \in ET$ and $\Psi \bigtriangledown \psi \in ET$, we have $\Phi \bigtriangledown \Psi \bigtriangledown (\phi \bigtriangleup \psi) \in ET$.*

(We mean that all the married couples of $\Phi \bigtriangledown \phi$ and $\Psi \bigtriangledown \psi$ are preserved in $\Phi \bigtriangledown \Psi \bigtriangledown (\phi \bigtriangleup \psi)$ and there are no other married couples in the latter).

PROOF. Assume $\Phi \bigtriangledown \phi \in ET$ and $\Psi \bigtriangledown \psi \in ET$. We proceed by induction on the sum of the hypercomplexities of these two hypersentences.

Case 1: $\Phi \bigtriangledown \Psi \bigtriangledown (\phi \bigtriangleup \psi)$ is 0-like. A little analysis of the case convinces us that then one of the hypersentences $\Phi \bigtriangledown \phi$, $\Psi \bigtriangledown \psi$, — say, the first one, — is 0-like. This means that there is a 0-hyperdevelopment $\Phi' \bigtriangledown \phi' \in ET$ of $\Phi \bigtriangledown \phi$; the hypercomplexity of the former is less than the hypercomplexity of the latter, and we can use the induction hypothesis and conclude that $\Phi' \bigtriangledown \Psi \bigtriangledown (\phi' \bigtriangleup \psi) \in ET$. Now it remains to notice that the latter is a 0-hyperdevelopment of $\Phi \bigtriangledown \Psi \bigtriangledown (\phi \bigtriangleup \psi)$.

Case 2: $\Phi \bigtriangledown \Psi \bigtriangledown (\phi \bigtriangleup \psi)$ is 1-like. Consider an arbitrary 1-hyperdevelopment θ of this hypersentence. We want to show that $\theta \in ET$.

θ must result from $\Phi \bigtriangledown \Psi \bigtriangledown (\phi \bigtriangleup \psi)$ by replacing a surface multiplicatively atomic osubsentence η by a 1-hyperdevelopment η' of η. η occurs in one of the osubsentences Φ, ϕ, Ψ or ψ, — say, in ϕ, and let ϕ' be the result of replacing in ϕ the osubsentence η by η'. Then $\Phi \bigtriangledown \phi'$ is a 1-hyperdevelopment of $\Phi \bigtriangledown \phi$ and, by 6.15.1, $\Phi \bigtriangledown \phi' \in ET$. Then, by the induction hypothesis, $\Phi \bigtriangledown \Psi \bigtriangledown (\phi' \bigtriangleup \psi)$, i.e. θ, belongs to ET. ♣

Remember that in our language $\neg$ is applicable only to atoms, and in other cases it is used just as an abbreviation.

A *sequent* for us is a multiset $\{\alpha_1, \ldots, \alpha_n\}$ of sentences, written in the form $\alpha_1 \bigtriangledown \ldots \bigtriangledown \alpha_n$. So, we use "$\bigtriangledown$" instead of ",". The other technical difference from the common presentation of sequential calculi is that we use parameters (as we deal with sentences only) instead of free variables.

Here is a list of (an axiom and) some sequential rules of inference, where:

- ξ in (33) is atomic;
- $\alpha(a)$ in (37) and (38) is the result of substituting in $\alpha(x)$ all free occurrences of the variable x by the parameter a;
- In (38) the parameter a does not occur in $\Theta, \alpha(x)$.

$$\frac{}{\Theta \triangledown \xi \triangledown \neg\xi}; \tag{33}$$

$$\frac{\Theta_1 \triangledown \alpha_1 \qquad \Theta_2 \triangledown \alpha_2}{\Theta_1 \triangledown \Theta_2 \triangledown (\alpha_1 \vartriangle \alpha_2)}; \tag{34}$$

$$\frac{\Theta \triangledown \alpha_i}{\Theta \triangledown (\alpha_1 \vee \alpha_2)},\ i = 1, 2; \tag{35}$$

$$\frac{\Theta \triangledown \alpha_1 \qquad \Theta \triangledown \alpha_2}{\Theta \triangledown (\alpha_1 \wedge \alpha_2)}; \tag{36}$$

$$\frac{\Theta \triangledown \alpha(a)}{\Theta \triangledown \exists x \alpha(x)}; \tag{37}$$

$$\frac{\Theta \triangledown \alpha(a)}{\Theta \triangledown \forall x \alpha(x)}; \tag{38}$$

$$\frac{\Theta \triangledown \alpha \triangledown \alpha}{\Theta \triangledown \alpha}. \tag{39}$$

Notice that as we deal with multisets (i.e. multiplicative disjunctions of multisets of disjuncts), the exchange rule

$$\frac{\Theta \triangledown \alpha_1 \triangledown \alpha_2 \triangledown \Theta'}{\Theta \triangledown \alpha_2 \triangledown \alpha_1 \triangledown \Theta'}$$

is senseless because the premise and the conclusion of this rule are simply thought to be identical.

The *logic BCK*, also called *Affine logic*, is given by the axiom (33) and the rules (34)-(38).[9] It is known that the addition of the rule of cut to this system does not really extend it.

And the whole list (33)-(39) gives *classical logic CL*.

The *length of a BCK-derivation* of ϕ is the length of the longest branch of the BCK-derivation tree for ϕ.

Lemma 14.2 *For any formula ϕ and any parameters c and b, if ϕ has a BCK-proof of length l, then so does $\phi[c/b]$.*

[9]BCK is an extension of the multiplicative-additive linear logic ($MALL$). The latter is given by the rules (34)-(38) and the axiom $\alpha \triangledown \neg\alpha$, where α is atomic.

(Recall that $\phi[c/b]$ denotes the result of replacing in ϕ all the occurrences of the parameter c by the parameter b.)

PROOF. Induction on the length of the BCK-derivation.

If ϕ is an axiom (of the form (33)), clearly so is $\phi[c/b]$.

Among the rules (34)-(38) we consider only the last one; the others are more or less straightforward.

So, suppose $\Theta\bigtriangledown\alpha(a)$, where a has no occurrence in $\Theta, \alpha(x)$, has a BCK-proof of length l. We want to show that then $(\Theta\bigtriangledown\forall x\alpha(x))[c/b]$ has a BCK-proof of length $l+1$. Let a' be a parameter not occuring in $\Theta, \alpha(x)$ and different from c and b. By the induction hypothesis, $(\Theta\bigtriangledown\alpha(a))[a/a']$, i.e. $\Theta\bigtriangledown\alpha(a')$, has a BCK-proof of length l. Again, by the induction hypothesis, $\Theta\bigtriangledown\alpha(a')[c/b]$, i.e. $(\Theta[c/b])\bigtriangledown(\alpha(a')[c/b])$, has a BCK-proof of length l. Notice that a' does not occur in $\Theta[c/b], \alpha(x)[c/b]$ and $\alpha(a')[c/b]$ is the result of replacing in $\alpha(x)[c/b]$ the variable x by a'. Then, applying (38) to $(\Theta[c/b])\bigtriangledown(\alpha(a')[c/b])$, we get that $(\Theta[c/b])\bigtriangledown\forall x(\alpha(x)[c/b])$, i.e. $(\Theta\bigtriangledown\forall x\alpha(x))[c/b])$, has a BCK-proof of length $l+1$. ♣

Lemma 14.3 *Suppose $BCK \vdash \phi$ and ϕ' is a 1-hyperdevelopment of ϕ. Then $BCK \vdash \phi'$.*

PROOF. We consider the case when ψ' is the result of replacing in ϕ an osubsentence $\forall y\beta(y)$ by $\beta(d)$. The other case, when the replaced subsentence is an additive conjunction, is simpler.

We proceed by induction on the length of the BCK-proof of ϕ. The only non-straightforward case the rule (38), when $\phi = \Theta\bigtriangledown\forall x\alpha(x)$ and $BCK \vdash \Theta\bigtriangledown\alpha(a)$ for a not occuring in $\Theta, \alpha(x)$. In view of 14.2, we may suppose that $a \neq d$. There are two subcases to be considered:

Subcase 1: $\forall y\beta(y)$ and $\forall x\alpha(x)$ are different osubsentences of ϕ. Then $\phi' = \Theta'\bigtriangledown\forall x\alpha(x)$, where Θ' is the result of replacing in Θ the osubsentence $\forall y\beta(y)$ by $\beta(d)$. $\Theta'\bigtriangledown\alpha(a)$ is then a 1-hyperdevelopment of $\Theta\bigtriangledown\alpha(a)$ and, by the induction hypothesis, $BCK \vdash \Theta'\bigtriangledown\alpha(a)$. $a \neq d$ implies that a does not occur in Θ'; therefore, by the rule (38), $BCK \vdash \Theta'\bigtriangledown\forall x\alpha(x)$, i.e. $BCK \vdash \phi'$.

Subcase 2: $\forall y\beta(y)$ and $\forall x\alpha(x)$ are one and the same osubsentence of ϕ. Then, as $BCK \vdash \Theta\bigtriangledown\alpha(a)$, we have by 14.2 that $BCK \vdash \Theta\bigtriangledown\alpha(d)$, i.e. $BCK \vdash \phi'$. ♣

Fact 14.4 $BCK \subset ET \subset CL$ ($\subset$ means proper inclusion).

PROOF. The inclusion $ET \subseteq CL$ is evident: Suppose $\alpha \in ET$. Then, by 5.5(i)⇒(ii), α is effectively true in every model; effective truth means the existence of an effective solution, and existence of a solution means truth; thus, α is true in every model, whence, by 5.4(ii)⇒(i), $\alpha \in CL$. And to see that this inclusion is proper, it is enough to check that either of the three classical tautologies listed below, where α is a zero-place predicate letter and β is a 1-place predicate letter, does not belong to ET:

- $\alpha \vee \neg\alpha$;
- $\alpha \bigtriangledown (\neg\alpha \bigtriangleup \neg\alpha)$;
- $\exists x \forall y (\beta(x) \bigtriangledown \neg\beta(y))$ (although $\forall y \exists x (\beta(x) \bigtriangledown \neg\beta(y)) \in ET$).

The fact that $BCK \neq ET$ can be verified by a routine checking that either of the following two formulas[10] belongs to ET but is not derivable in the cut-free BCK:

$$\Big((\neg\alpha_1 \bigtriangledown \neg\alpha_2) \bigtriangleup (\neg\beta_1 \bigtriangledown \neg\beta_2)\Big) \bigtriangledown \Big((\alpha_1 \bigtriangledown \beta_1) \bigtriangleup (\alpha_2 \bigtriangledown \beta_2)\Big);$$

$$\Big((\neg\alpha_1 \wedge \neg\alpha_2) \bigtriangleup (\neg\beta_1 \wedge \neg\beta_2)\Big) \bigtriangledown$$
$$\Big(\big(\alpha_1 \bigtriangledown (\beta_1 \vee \beta_2)\big) \vee \big(\alpha_2 \bigtriangledown (\beta_1 \vee \beta_2)\big) \vee \big((\alpha_1 \vee \alpha_2) \bigtriangledown \beta_1\big) \vee \big((\alpha_1 \vee \alpha_2) \bigtriangledown \beta_2\big)\Big).$$

It remains to show that $BCK \subseteq ET$. Assume $BCK \vdash \phi$. To show that $\phi \in ET$, we use induction on the *complexity* of ϕ, under which we mean the number of occurrences of logical operators in ϕ. $BCK \vdash \phi$ means that one of the following five cases takes place:

Case 1: ϕ is an axiom $\Theta \bigtriangledown \xi \bigtriangledown \neg\xi$. Let ϕ' be the marriage-extension (and thus a 0-hyperdevelopment) of ϕ in which the osliterals ξ and $\neg\xi$ are spouses to each other. Clearly ϕ' is 1-like, and so is any 1-hyperdevelopment of ϕ', any 1-hyperdevelopment of any 1-hyperdevelopment of ϕ', ... This means that $\phi \in ET$.

Case 2: $BCK \vdash \phi$ follows from $BCK \vdash \alpha$ and $BCK \vdash \beta$ by the rule (34). The complexities of α and β are less than that of ϕ, which enables us to use the induction hypothesis and conclude that $\alpha, \beta \in ET$. Then, by Lemma 14.1, $\phi \in ET$.

Case 3: $BCK \vdash \phi$ follows from $BCK \vdash \alpha$ by one of the rules (35) or (37). The complexity of α is less than that of ϕ and, by the induction hypothesis, $\alpha \in ET$. But α is a 0-hyperdevelopment of ϕ and, by 6.15, the latter belongs to ET.

Case 4: $BCK \vdash \phi$ follows from $BCK \vdash \alpha$ by one of the rules (36) or (38). Then, observe, ϕ is 1-like. Consider an arbitrary 1-hyperdevelopment ϕ' of ϕ. By 14.3, $BCK \vdash \phi'$, whence, by the induction hypothesis (as the complexity of ϕ' is less than that of ϕ), $\phi' \in ET$. Thus, every 1-hyperdevelopment of the 1-like ϕ belongs to ET, and this means that $\phi \in ET$. ♣

15 The $\bigtriangleup$-free fragments of BCK and ET are the same

Let L^- denote the fragment of our language L defined by:

[10] The first formula is taken from [2].

$\alpha \in L^-$ iff $\alpha \in L$ and α does not contain $\triangle$.

And we define BCK^- as BCK without the rule (34).

Theorem 15.1 *For any sentence ϕ of L^-, $BCK^- \vdash \phi \Leftrightarrow \phi \in ET$.*

PROOF. The ($\Rightarrow$) part immediately follows from Fact 14.4.

We prove the ($\Leftarrow$) part by induction on the complexity of ϕ. Assume ϕ is a sentence of L^- and $\phi \in ET$.

Case 1: ϕ is 1-like. Taking into account that ϕ is clean (see 6.1.6), this implies that ϕ contains a surface osubsentence of the form $\alpha \wedge \beta$ or $\forall x \alpha(x)$, that is, one of the following two subcases takes place:

Subcase 1a: $\phi = \Theta \triangledown \forall x \alpha(x)$. Let a be a parameter not occuring in ϕ. $\Theta \triangledown \alpha(a)$ is a 1-hyperdevelopment of ϕ and, by 6.15.1, as the latter belongs to ET, so does the former. But the complexity of $\Theta \triangledown \alpha(a)$ is less than that of ϕ and then, by the induction hypothesis, $BCK \vdash \Theta \triangledown \alpha(a)$; then, by the rule (38), $BCK \vdash \phi$.

Subcase 1b: $\phi = \Theta \triangle (\alpha \wedge \beta)$. This subcase is similar to the previous one.

Case 2: ϕ is 0-like. Then $\phi \in ET$ means that there is a 0-hyperdevelopment $\phi' \in ET$ of ϕ, i.e. one of the following three subcases takes place:

Subcase 2a: ϕ' is a marriage-extension of ϕ. This means that ϕ contains surface osliterals ξ and $\neg\xi$, and as ϕ is $\triangle$-free, ϕ has the form $\Theta \triangledown \xi \triangledown \neg \xi$, i.e. it is an axiom.

Subcase 2b: $\phi = \Theta \triangledown (\alpha_1 \vee \alpha_2)$ and $\phi' = \Theta \triangledown \alpha_i$ ($i = 1$ or $i = 2$). By the induction hypothesis, $BCK \vdash \Theta \triangledown \alpha_i$, whence, by the rule (35), $BCK \vdash \phi$.

Subcase 2c: $\phi = \Theta \triangledown \exists x \alpha(x)$ and $\phi' = \Theta \triangledown \alpha(a)$. This subcase is similar to the previous one. ♣

References

[1] S.Abramsky and R.Jagadeesan, *Games and full completeness for multiplicative linear logic.* Journal of Symbolic Logic 59 (1994), no. 2, pp.543-574.

[2] A.Blass, *A game semantics for linear logic.* Annals of Pure and Applied Logic, v.56 (1992), pp. 183-220.

[3] *Game-Theoretical Semantics: Essays on Semantics by Hintikka, Carlson, Peacocke, Rantala and Saarinen* (edited by E.Saarinen). Dordrecht, Holland, 1979.

[4] J.Y.Girard, *Linear logic.* Theoretical Computer Science, v.50-1 (1987), pp. 1-102.

[5] S.C.Kleene, *Introduction to Metamathematics.* New York, 1952.

[6] K.Lorenz, *Dialogspiele als semantische Grundlage von Logikkalkülen.* Arch. Math. Logik Grundlag., v.11 (1968), pp.32-55, 73-100.

[7] P.Lorenzen, *Ein dialogisches Konstruktivitätskriterium.* In: Infinitistic Methods (PWN, Warsaw, 1959), pp.193-200.

[8] A.S.Troelstra, *Realizability.* ILLC Prepublication Series for Mathematical Logic and Foundations ML-92-09. Amsterdam, 1992.