LOGIC JOURNAL of the IGPL

Volume 10 Number 2 March 2002

Interest Group in Pure and Applied Logics

OXFORD UNIVERSITY PRESS

ISSN 1367-0751

Subscription Information

Volume 10, 2002 (bimonthly) Full: Europe pounds sterling 275; Rest of World US$ 450. Personal: pounds sterling 138 (US$ 225). Please note that personal rates apply only when copies are sent to a private address and payment is made by a personal cheque or credit card.

Order Information

Subscriptions can be accepted for complete volumes only. Prices include air-speeded delivery to Australia, Canada, India, Japan, New Zealand, and the USA. Delivery elsewhere is by surface post. Payment is required with all orders and may be made in the following ways:

Cheque (made payable to Oxford University Press)
National Girobank (account 500 1056)
Credit card (Access, Visa, American Express)
UNESCO Coupons
Bankers: Barclays Bank plc, PO Box 333, Oxford, UK. Code 20-65-18, Account 00715654.

Requests for sample copies, subscription enquiries, orders and changes of address should be sent to the Journals Subscriptions Department, Oxford University Press, Great Clarendon Street, Oxford OX2 6DP, UK. Tel: +44 (0) 1865 267907. Fax: +44 (0) 1865 267485. Email: jnl.orders@oup.co.uk

Advertisements

Copyright

Logic Journal of the IGPL (ISSN 1367-0751) is published bimonthly in January, March, May, July, September and November by Oxford University Press, Oxford, UK. Annual subscription price is US$ 450.00. *Logic Journal of the IGPL* is distributed by M.A.I.L. America, 2323 Randolph Avenue, Avenel, NJ 07001. Periodical postage paid at Rahway, New Jersey, USA and at additional entry points.
US Postmasters: Send address changes to *Logic Journal of the IGPL*, c/o Mercury International, 365 Blair Road, Avenel, NJ 07001, USA.

Back Issues

The current plus two back volumes are available from Oxford University Press. Previous volumes can be obtained from the Periodicals Service Company, 11 Main Street, Germantown, NY 12526 USA. Tel: +1 (518) 537 4700, Fax: +1 (518) 537 5899.

Logic Journal of the IGPL

Volume 10, Number 2, March 2002

Contents

On the Descriptive Complexity of a Simplified Game of Hex 105
A. Arratia

Labelled Natural Deduction for Conditional Logics of Normality 123
K. Broda, D. M. Gabbay, L. C. Lamb and A. Russo

The Unrestricted Combination of Temporal Logic Systems 165
M. Finger and M. A. Weiss

A Fibring Semantics for the Semantic-Morphological Interface in Natural Language 191
R. Naumann

Please visit the journal's World Wide Web site at
http://www.jigpal.oupjournals.org

Logic Journal of the Interest Group in Pure and Applied Logics

Scope of the Journal

The *Journal* is the official publication of the International Interest Group in Pure and Applied Logics (IGPL), which is sponsored by The European Foundation for Logic, Language and Information (FoLLI), and currently has a membership of over a thousand researchers in various aspects of logic (symbolic, computational, mathematical, philosophical, etc.) from all over the world.

The *Journal* is published in hardcopy and in electronic form six times per year. Publication is fully electronic: submission, refereeing, revising, typesetting, checking proofs, and publishing, all is done via electronic mailing and electronic publishing.

Papers are invited in all areas of pure and applied logic, including: pure logical systems, proof theory, model theory, recursion theory, type theory, nonclassical logics, nonmonotonic logic, numerical and uncertainty reasoning, logic and AI, foundations of logic programming, logic and computation, logic and language, and logic engineering.

The *Journal* is an attempt to solve a problem in the logic (in particular, IGPL) community:

- Long delays and large backlogs in publication of papers in current journals.
- Very tight time and page number limits on submission.

Papers in the final form should be in LaTeX. The review process is quick, and is made mainly by other IGPL members.

Submissions

Submissions are made by sending a submission letter to the e-mail address: jigpl@dcs.kcl.ac.uk, giving the title and the abstract of the paper, and informing: of how to obtain the file electronically or, by sending 5 (five) hardcopies of the paper to the Editor-in-Chief.

URL: http://www.jigpal.oupjournals.org

On the Descriptive Complexity of a Simplified Game of Hex

ARGIMIRO ARRATIA, *Departamento de Matemáticas, Universidad Simón Bolívar, Apartado 89000, Caracas 1080-A, Venezuela. E-mail: arratia@ma.usb.ve*

Abstract

The game Whex is here defined, which is similar to Generalized Hex but the players are restricted to colour vertices adjacent to the vertex last coloured by one of the players. It is shown that the problem of deciding existence of winning strategies for one of the players in this game is complete for **PSPACE**, via quantifier free projections, and that the extension of first order logic with the corresponding generalized quantifier captures **PSPACE** and verifies a normal form. This problem is used to show that the problem of finding a proof in a proof system, like propositional resolution, in which the user is allowed to introduce auxiliary statements in order to help the system reach the theorem that he had set it to prove, is also complete for **PSPACE** via quantifier free projections. Also, it is established the complexity of the game Whex when restricted to graphs of outdegree at most 3, and, as a generalized quantifier, its expressive capabilities in the absence of ordering relation.

Keywords: Descriptive complexity; generalized quantifier; quantifier free projection; games; Generalized Hex; polynomial space.

1 Introduction

In Descriptive Complexity we are concerned with logical characterizations of the usual Turing machine based notion of a computational complexity class. Among the advantages of the logical approach to Computational Complexity we have: we can establish the complexity of a problem by syntactic measures, like number of quantifiers, variables and other symbols needed for describing the problem, instead of, say, designing an algorithm for it; we can define a notion of reduction among problems (the first order reduction) which is weaker than the traditional log–space; and we can sometimes discern the nature of problems (on logical grounds) that within the traditional Turing machine framework are equivalent (a fact which, often, defies our intuition).

There are various ways of constructing a logic that define exactly the properties that lie within a certain computational complexity class. The method used in this paper consists of extending the expressive power of first order logic with a uniform sequence of generalized quantifiers, corresponding to a problem which is a representative of the complexity class, in the sense of being complete for the class via logspace reducibility. It is also necessary to include some built–in ordering relation, like that of successor on the natural numbers, plus two constants that stand for the first and last element in the ordering. This is the same as saying that our logical interpretations of our complexity classes hold only over ordered finite structures with at least two elements, which seems to be a necessary restriction, at least for logics expressing polynomial time properties and of lesser complexity. The reader can find in the books [5] and [10] further details on generalized quantifiers, and an extensive discussion on the problem of the ordering in the logical characterizations of computational complexity classes.

The computational problem presented in this paper is a variation of the game of Generalized Hex (see [7]), that I have named Whex for reasons to be explained later, which illustrates the facts mentioned in the first paragraph of this introduction. In particular this new game seems easier to play in practice but, as it will be shown, enjoys similar complexity features as Hex in theory.

As an application, the new game problem Whex is used to show that the problem of finding a proof in a (mechanical) proof system with only one rule of inference like, say, propositional resolution, and in which the user is allowed to introduce auxiliary statements in order to help the system reach the theorem that we had set it to prove, is complete for the class of problems that have polynomially space bounded algorithmic solutions, via a very weak kind of reductions, namely *quantifier free projections*. On the way, here I also exhibit some tools for proving problems complete via reductions which are definable in first order logic, and illustrate their use with the particular game problem in consideration. This paper ends exhibiting the complexity of Whex, both computational and logical, when the board is restricted to be directed graphs of outdegree ≤ 3, and further, it is pointed out the expressive capabilities of Whex, as a generalized quantifier, in the absence of order.

2 Background on Descriptive Complexity

Some needed notions from Descriptive Complexity are reviewed in this section. More details can be found in the books [5], [10]. A vocabulary $\tau = \{R_1, \ldots, R_r, C_1, \ldots, C_s\}$ is a finite set of relation and constant symbols, where each relation R_i has arity n_i. A finite τ–structure $\mathcal{A} = \langle A, R_1^{\mathcal{A}}, \ldots, R_r^{\mathcal{A}}, C_1^{\mathcal{A}}, \ldots, C_s^{\mathcal{A}} \rangle$ consists of a universe $A = \{0, \ldots, n-1\}$, relations $R_i^{\mathcal{A}} \subseteq A^{n_i}$, and constants $C_j^{\mathcal{A}}$ ($1 \leq i \leq r$, $1 \leq j \leq s$). STRUCT(τ) denotes the set of all finite τ–structures. A problem over vocabulary τ is a subset of STRUCT(τ) closed under isomorphisms.

A logic $\mathcal{L}$ consists of: 1) all sets $\mathcal{L}(\tau)$ of formulas over each finite vocabulary τ, build up from the symbols in τ, variables, boolean operations $\wedge$, $\vee$, $\neg$, and quantifiers $\forall$, $\exists$, following certain syntactic rules, together with a satisfaction relation $\models$ which establishes the meaning of formulas in (finite) models (possibly extended with interpretations for free variables). In particular FO denotes First Order logic, and FO_s is First Order logic with built–in symbols $suc(\cdot,\cdot)$, 0 and max, which are always interpreted on finite structures as the successor relation over the naturals, the first and the last element. Hence, a sentence in FO_s describes a property of *ordered structures*.

If $\mathcal{L}$ is a logic and ϕ a sentence in $\mathcal{L}(\tau)$, for some vocabulary τ, then $MOD(\phi) := \{\mathcal{A} \in \mathrm{STRUCT}(\tau) : \mathcal{A} \models \phi\}$ denotes the set of finite models that satisfy ϕ.

Problems as traditionally considered in Computational Complexity are sets of strings, usually of 0's and 1's, so that they can be inputted into Turing machines. To transform finite structures into strings of $\{0,1\}^*$, and viceversa, some forms of encodings are established, and, furthermore, these encodings can be made very efficiently (e.g., first order definable, see [10]). Hence, if τ is some vocabulary and $\mathcal{A} \in \mathrm{STRUCT}(\tau)$, $e_\tau(\mathcal{A})$ denotes an encoding of $\mathcal{A}$ as a string over the set of symbols $\{0,1\}$, which is first order definable. If Ω is some problem over τ (in the sense of being a set of finite τ-structures closed under isomorphism), then the set of strings over $\{0,1\}$ corresponding to encodings of elements of Ω is denoted $e_\tau(\Omega) := \{e_\tau(\mathcal{A}) : \mathcal{A} \in \Omega\} \subseteq \{0,1\}^*$.

Thus, given a complexity class $\mathcal{C}$, defined in terms of the sets of strings over $\{0,1\}$ accepted by some kind of Turing machines, when it is said that $\mathcal{C}$ *is captured by a logic* $\mathcal{L}$ it is meant formally that

1. for each set of strings $S \in \mathcal{C}$, there is some vocabulary τ and a sentence ϕ in $\mathcal{L}(\tau)$, such that $S = e_\tau(MOD(\phi))$; and
2. for each vocabulary τ and each sentence ϕ in $\mathcal{L}(\tau)$, $e_\tau(MOD(\phi)) \in \mathcal{C}$.

Whenever class $\mathcal{C}$ is captured by logic $\mathcal{L}$ (or $\mathcal{L}$ captures $\mathcal{C}$), it will be denoted $\mathcal{C} = \mathcal{L}$. Moreover, $\mathcal{C} \leq \mathcal{L}$ denotes that item 1. above is satisfied, and $\mathcal{L} \leq \mathcal{C}$ denotes that item 2. is satisfied. So, $\mathcal{C} = \mathcal{L}$ if and only if $\mathcal{C} \leq \mathcal{L}$ and $\mathcal{L} \leq \mathcal{C}$. If $\mathcal{L}_1$ and $\mathcal{L}_2$ are two logics then $\mathcal{L}_1 = \mathcal{L}_2$ means that, for all vocabulary τ, every $\mathcal{L}_1(\tau)$-sentence ϕ_1 is equivalent to an $\mathcal{L}_2(\tau)$-sentence ϕ_2 and viceversa. (Here, equivalence of sentences ϕ_1 and ϕ_2 has the following intended meaning: for every finite τ-structure $\mathcal{A}$, $\mathcal{A} \models \phi_1$ if and only if $\mathcal{A} \models \phi_2$, this is commonly denoted as $\models \phi_1 \longleftrightarrow \phi_2$.)

An important notion in Descriptive Complexity is that of a logical reduction.

Definition 2.1 Let $\mathcal{L}$ be some logic, let τ be a vocabulary and $\sigma = \{R_1, \dots, R_r, C_1, \dots, C_s\}$ be another vocabulary, where each R_i is a relation symbol of arity n_i and each C_j is a constant symbol. A set Σ of $\mathcal{L}(\tau)$–formulas of the form

$$\Sigma := \{\phi_1(\overline{x}_1), \dots, \phi_r(\overline{x}_r), \psi_1(\overline{y}_1), \dots, \psi_s(\overline{y}_s)\},$$

where $\overline{x}_i$ and $\overline{y}_j$ are vectors of distinct variables with $|\overline{x}_i| = kn_i$ and $|\overline{y}_j| = k$, for some positive integer k, $1 \leq i \leq r$ and $1 \leq j \leq s$, describes a (τ, σ)*–translation of arity* k, which is a map sending a τ–structure $\mathcal{A}$ into a σ–structure $\mathcal{A}_\Sigma$, such that

- the universe of $\mathcal{A}_\Sigma$ is the set A^k of k-tuples of A;
- for $i = 1, \dots, r$, the relation R_i of σ has the interpretation:

$$R_i^{\mathcal{A}_\Sigma} := \{\overline{a} \in A^{kn_i} : \langle \mathcal{A}, \overline{a} \rangle \models \phi_i(\overline{x}_i)\}$$

- for $j = 1, \dots, s$, the constant C_j of σ has the interpretation:

$$C_j^{\mathcal{A}_\Sigma} := \overline{u}_j$$

  where $\overline{u}_j$ is the unique element in A^k that satisfies ψ_j.

A problem $\Omega_1 \subseteq \text{STRUCT}(\tau)$ $\mathcal{L}$*-reduces to* $\Omega_2 \subseteq \text{STRUCT}(\sigma)$ (denoted $\Omega_1 \leq_{\mathcal{L}} \Omega_2$) if there exists a $k > 0$ and a set Σ of $\mathcal{L}(\tau)$–formulas which defines a (τ, σ)–translation of arity k so that, for all $\mathcal{A} \in \text{STRUCT}(\tau)$,

$$\mathcal{A} \in \Omega_1 \text{ if and only if } \mathcal{A}_\Sigma \in \Omega_2.$$

In particular, if $\mathcal{L}$ in the above definition is FO, then we have a *first order reduction.* If, further, all the formulas in Σ are *projections*, then we have a reduction which is a *first order projection (fop).* (A first order formula ψ, over some vocabulary τ, is a *projection* if it has the form

$$\alpha_0 \vee (\alpha_1 \wedge \beta_1) \vee \dots \vee (\alpha_m \wedge \beta_m)$$

where none of the α_i contains a symbol from τ; for $i \neq j$, α_i and α_j are mutually exclusive; and each β_i is an atomic or negated atomic formula built up from symbols in τ only.)

3 Capturing complexity classes with generalized quantifiers

We can turn a problem into a generalized quantifier and increase the expressive power of first order logic as follows. Let σ be as in the Definition 2.1 and let Ω be a problem over σ. Then the extension of FO with the generalized quantifier Ω, which will be denoted Ω^*[FO], is the smallest set $\mathcal{L}$ of formulas such that: $\mathcal{L}$ contains all first order formulas, $\mathcal{L}$ is closed under all logical connectives and first order quantifiers, and for any finite vocabulary τ, if $\Sigma := \{\phi_1, \ldots, \phi_r, \psi_1, \ldots, \psi_s\}$ is a set of τ–formulas in $\mathcal{L}$ that describes a (τ, σ)–translation of arity k, mapping τ–structure $\mathcal{A}$ into σ–structure $\mathcal{A}_\Sigma$, then

$$\Phi := \Omega[\overline{x}_1, \ldots, \overline{x}_r, \overline{y}_1, \ldots, \overline{y}_s : \phi_1(\overline{x}_1), \ldots, \phi_r(\overline{x}_r), \psi_1(\overline{y}_1), \ldots, \psi_s(\overline{y}_s)]$$

is a new sentence in $\mathcal{L}$, which is interpreted as follows. Given a τ–structure $\mathcal{A}$, $\mathcal{A} \models \Phi$ if, and only if, the (τ, σ)–translation of $\mathcal{A}$ described by Σ, namely, the σ–structure $\mathcal{A}_\Sigma$, is such that $\mathcal{A}_\Sigma \in \Omega$.

Remark 3.1 The formula

$$\Omega[\overline{x}_1, \ldots, \overline{x}_r, y_1, \ldots, y_s : \phi_1(\overline{x}_1), \ldots, \phi_r(\overline{x}_r), y_1 = C_1, \ldots, y_s = C_s]$$

is often abbreviated as

$$\Omega[\overline{x}_1, \ldots, \overline{x}_r : \phi_1(\overline{x}_1), \ldots, \phi_r(\overline{x}_r)](C_1, \ldots, C_s).$$

Some interesting fragments of Ω^*[FO] are:

Ω^n[**FO**] for a positive integer n, is such that at most n nested applications of Ω can appear in a formula.

posΩ^*[**FO**] is the sublogic where no application of Ω is within the scope of a $\neg$ (i.e., Ω has only positive occurrences).

posΩ^n[**FO**] for a positive integer n is the language with all occurrences of Ω positive and at most n nested

Ω^*[$\mathbf{FO}_s$] is the fragment of Ω^*[FO] with the built–in successor and built–in constants 0 and max. In a similar way are defined the fragments $\Omega^n[\mathrm{FO}_s]$, pos$\Omega^*[\mathrm{FO}_s]$ and pos$\Omega^n[\mathrm{FO}_s]$.

Example 3.2 Consider the vocabulary $\tau_2 = \{E, C, D\}$, where E is a binary relation symbol, C and D are constant symbols, and consider the following two problems over τ_2:

$$\begin{aligned} \mathrm{TC} \;:=\; & \{\mathcal{A} \in \mathrm{STRUCT}(\tau_2) : \mathcal{A} \text{ is a directed graph} \\ & \text{and there is a path from vertex } C^{\mathcal{A}} \text{ to vertex } D^{\mathcal{A}}\}. \end{aligned}$$

$$\begin{aligned} \mathrm{DTC} \;:=\; & \{\mathcal{A} \in \mathrm{STRUCT}(\tau_2) : \mathcal{A} \text{ is a directed graph and there is a} \\ & \text{path from } C^{\mathcal{A}} \text{ to } D^{\mathcal{A}} \text{ of all vertices with outdegree 1}\}. \end{aligned}$$

(Recall that the outdegree of a vertex v in a directed graph is the number of edges going out of v.)

As it was shown by Neil Immerman in [9], the positive fragment of the extensions of FO_s with generalized quantifiers corresponding to each one of these problems captures, respectively, the classes nondeterministic logspace (**NL**) and logspace (**L**). Specifically:

$$\mathrm{posTC}^*[\mathrm{FO}_s] = \mathbf{NL} \text{ and } \mathrm{posDTC}^*[\mathrm{FO}_s] = \mathrm{DTC}^*[\mathrm{FO}_s] = \mathbf{L}$$

In general, if $\mathcal{L}$ is any *regular* logic (as defined in [4]), define the extension of $\mathcal{L}$ with the generalized quantifier Ω in a similar manner as for FO, and obtain the logic $\Omega^*[\mathcal{L}]$ and similar fragments. The next proposition is immediate from the definitions, and it just makes explicit the fact that $\mathcal{L}$–reducibility among problems is equivalent to definability in the extension of $\mathcal{L}$ with the generalized quantifier corresponding to the larger problem.

Proposition 3.3 Let $\mathcal{L}$ be a regular logic, σ and τ be two vocabularies, with $\tau = \{R_1, \dots, R_r, C_1, \dots, C_c\}$. Let Ω_1 be a σ–problem and Ω_2 be a τ–problem. Then $\Omega_1 \leq_{\mathcal{L}} \Omega_2$ via a (σ,τ)–translation of arity k if, and only if, $\Omega_1 = MOD(\Phi)$ for some sentence $\Phi \in \mathrm{pos}\Omega_2^1[\mathcal{L}(\sigma)]$ of the form $\Omega_2[\overline{x}_1, \dots, \overline{x}_r, \overline{y}_1, \dots, \overline{y}_c : \phi_1, \dots, \phi_r, \psi_1, \dots, \psi_c]$, where $\{\phi_1, \dots, \phi_r, \psi_1, \dots, \psi_c\} \subset \mathcal{L}(\sigma)$ constitute a (σ,τ)–translation of arity k. ∎

The following important result that links the usual Turing machine based reducibility and the logical reducibility has been proved in [12] (cf. [5, Proposition 10.3.22]).

Proposition 3.4 Let $\Omega_1 \subseteq \mathrm{STRUCT}(\sigma)$ and $\Omega_2 \subseteq \mathrm{STRUCT}(\tau)$ be problems. Then $e_\sigma(\Omega_1)$ is logspace reducible to $e_\tau(\Omega_2)$ if, and only if, $\Omega_1 \leq_{\mathrm{DTC}^1[\mathrm{FO}_s]} \Omega_2$. ∎

As a consequence of the previous propositions and definitions we have the following *sandwich theorem*, which is a key tool for "almost capturing" complexity classes by extensions of FO_s with generalized quantifiers. (This result is a generalization of Corollary 3.1 of [12], and it is not difficult to prove.)

Theorem 3.5 Let $\mathcal{C}$ be a complexity class above **L** and closed under logspace reducibility. Let $\Omega \subseteq \mathrm{STRUCT}(\tau)$ be a problem such that $\mathrm{DTC}^1[\mathrm{FO}_s] \leq \mathrm{pos}\Omega^*[\mathrm{FO}_s]$, and $e_\tau(\Omega)$ is complete in $\mathcal{C}$ via logspace reducibility. Then,

$$\mathrm{pos}\Omega^1[\mathrm{FO}_s] \leq \mathcal{C} \leq \mathrm{pos}\Omega^*[\mathrm{FO}_s]$$

∎

The necessary next step for having an exact description of $\mathcal{C}$ by the logic $\mathrm{pos}\Omega^*[\mathrm{FO}_s]$ is to show that it has a *first order normal form*; that is, every sentence in $\mathrm{pos}\Omega^*[\mathrm{FO}_s]$ is equivalent to one application of the quantifier Ω to a first order sentence ψ. In symbols, that $\mathrm{pos}\Omega^*[\mathrm{FO}_s] = \mathrm{pos}\Omega^1[\mathrm{FO}_s]$ We sometimes get something much better; namely, that the first order sentence ψ is a *quantifier free projection* (qfp), and so it is said that the logic has a *quantifier free projective normal form*.

When there is a normal form then the corresponding problem Ω is complete for the class via either first order or qfp reductions, according to the nature of the normal form. Either of these reducibilities is computationally much weaker than logspace, since the set of all problems definable by first order sentences is properly contained in the class **L**. As an example, the two logics just mentioned, namely, $\mathrm{posTC}^*[\mathrm{FO}_s]$ and $\mathrm{DTC}^*[\mathrm{FO}_s]$, verify a quantifier free projective normal form and, hence, TC and DTC are complete via qfp reductions in their respective classes (see [9] for the details).

4 The WHEX logic

Fix a vocabulary $\tau_2 = \{E, C, D\}$, where E is a binary relation symbol, C and D are constant symbols. A complexity class and a problem that fits in the hypothesis of Theorem 3.5 is **PSPACE** (polynomial space) and the τ_2–problem HEX. An instance of HEX is a graph G with a source s and a sink t, and a yes-instance is an instance (G, s, t) where Player 1 has a winning strategy in the game of Hex on (G, s, t). The game of Hex proceeds as follows. Beginning with Player 1, two players take turns in colouring previously uncoloured vertices of G, apart from s and t, until all vertices are coloured. Player 1 always colours a vertex red and Player 2 always colours a vertex blue. Player 1 wins the game of Hex if in the resulting coloured graph there is a path consisting entirely of red vertices from s to t. That the encoding of this problem as strings of 0's and 1's is complete for **PSPACE** via logspace reducibility was shown in [6]; and that the extension of first-order logic using a uniform sequence of generalized quantifiers corresponding to the problem HEX contains DTC–logic (with respect to arbitrary finite structures) is proved in the following theorem.

Theorem 4.1 $\mathrm{DTC}^1[\mathrm{FO}] \leq \mathrm{posHEX}^*[\mathrm{FO}]$

Proof. Consider the τ_2–sentence $\psi := \mathrm{DTC}[\overline{x}, \overline{y} : \theta(\overline{x}, \overline{y})](\overline{C}, \overline{D})$, where $\overline{x}$ and $\overline{y}$ are k-tuples of distinct variables, for some $k > 0$, $\overline{C}$ and $\overline{D}$ are k–tuples with all entries equal to C and to D respectively, and $\theta \in \mathrm{FO}$. Consider the following formula ϕ with free variables $\overline{x}$, u_1, u_2, $\overline{y}$, v_1 and v_2:

$$\begin{array}{rcl}
\phi & := & (\overline{x} = \overline{C} \wedge u_1 = u_2 = C \wedge \overline{y} = \overline{C} \\
& & \wedge\ ((v_1 = D \wedge v_2 = C) \vee (v_1 = C \wedge v_2 = D))) \\
& \vee & (\overline{x} = \overline{D} \wedge ((u_1 = D \wedge u_2 = C) \vee (u_1 = C \wedge u_2 = D)) \\
& & \wedge\ \overline{y} = \overline{D} \wedge v_1 = D \wedge v_2 = D) \\
& \vee & (u_1 = v_1 \wedge u_2 = v_2 \wedge \theta(\overline{x}, \overline{y})) \\
& \vee & (u_1 = C \wedge u_2 = D \wedge v_1 = D \wedge v_2 = C \wedge \theta(\overline{x}, \overline{y})) \\
& \vee & (u_1 = D \wedge u_2 = C \wedge v_1 = C \wedge v_2 = D \wedge \theta(\overline{x}, \overline{y}))
\end{array}$$

Then

$$\models \psi \ \longleftrightarrow\ \mathrm{HEX}[(\overline{x}, u_1, u_2), (\overline{y}, v_1, v_2) : \phi](\overline{C}, \overline{D}).$$

(Player 1's winning strategy consists of colouring first $(\overline{C}, C, D)$ or $(\overline{C}, D, C)$, and thereafter colour opposite vertex to Player 2's choice; that is, if Player 2 colours $(\overline{x}, u_1, u_2)$ then Player 1 colours $(\overline{x}, u_2, u_1)$.) ∎

In [2] it was shown that the HEX–logic has a quantifier free projective normal form and, hence, captures **PSPACE** applying Theorems 4.1 and 3.5. As an extra bonus we get that the problem HEX is complete for **PSPACE** via quantifier free projections.

I would like to remark that similar results hold for the version of the Hex game where players colour *edges* (as opposed to vertices) of a *directed* graph. The version of the Hex game where players colour edges of an undirected graph is known as the *Shannon switching game* for which polynomial time algorithms are known (see, for example, [15]).

The main concern of this paper is the following variation of the problem HEX. An instance is a graph G with a source s and a sink t, and a yes–instance is an instance where Player 1 has a winning strategy in a game which proceeds as the game of Hex but with the following restriction:

> Players can not colour an arbitrary vertex but must proceed as follows: Player 1 begins the game and he must do it by colouring red a vertex adjacent to the source. From this move and on, Player 2 must colour blue an uncoloured vertex adjacent to the vertex last coloured red (i.e. coloured by Player 1), and Player 1 replies by colouring red an uncoloured vertex adjacent to the vertex that he coloured red last.

Thus, Player 1 tries to build a path in a step–by–step fashion and linked to the source, whilst Player 2 tries continually to block Player 1's construction. As in the Hex game, Player 1 wins if he reaches t with a path of red vertices only from s, with s and t uncoloured and possibly some other vertices at the end of the game.

The above restriction placed upon the game of Hex makes the construction process of a path local and reduces the space of search for each player (in fact, it makes it linear); for these reasons I named this game *Weak Hex* (abbreviated Whex) and the corresponding decision problem WHEX. However I shall prove below that WHEX is similar to HEX with regards to computational complexity, when both are treated as decision problems, and, on the other hand, they also coincide in expressive power when treated as generalized quantifiers. So, in those two aspects, WHEX is no weaker than HEX.

WHEX is encoded as a class of τ_2-structures as follows:

$$\begin{aligned}\text{WHEX} = \{\mathcal{A} = \langle A, E^{\mathcal{A}}, C^{\mathcal{A}}, D^{\mathcal{A}}\rangle \in \text{STRUCT}(\tau_2) :\ &\text{Player 1 has}\\ &\text{a winning strategy for the game of Whex played on } \mathcal{A}\ \}\end{aligned}$$

It is worth observing that, as classes of structures, WHEX $\neq$ HEX: There are graphs where Player 1 wins the game of Whex but can loose the game of Hex (for example, a graph as constructed in Theorem 4.3 below, corresponding to a satisfiable sentence), and viceversa (for example, the complete bipartite graph $K_{n,n}$, with $n \geq 4$ and where the top two vertices are joined to a vertex s and the bottom two vertices are joined to a vertex t, is a yes–instance of HEX but not of WHEX). Furthermore, there are graphs where Player 1 wins both the Hex and the Whex games, as, for example, a ladder with diagonal rungs with the source joined at one end and the sink joined at the other end. One such a graph is constructed in the proof of Theorem 4.1, and, so, we have

Theorem 4.2 $\text{DTC}^1[\text{FO}] \leq \text{posWHEX}^*[\text{FO}]$ ∎

4.1 WHEX is **PSPACE***-complete*

QSAT, or the problem of determining if a quantified boolean formula in conjunctive normal form is true, is the classical example of a **PSPACE**–complete problem [7] and can be regarded as a game as follows (see [11] also): given

$$\Phi := \exists x_1 \forall x_2 \ldots Q_{n-1}x_{n-1}Q_n x_n\, \phi$$

where ϕ is a boolean formula in conjunctive normal form involving the variables x_1, $\ldots$, x_n, and the quantifiers $Q_i \in \{\forall, \exists\}$ alternate starting with $\exists$ [1]. We have two players, Player 1 and Player 2, who take turns assigning a value of 0 (false) or 1 (true) to each variable, beginning with x_1 and with Player 1 making the first move. So, Player 1 assigns truth values to the variables existentially quantified, whilst Player 2 assigns truth values to the variables universally quantified; for those reasons Player 1 is also known as the $\exists$ player and Player 2 as the $\forall$ player. This game is named Qsat (played on Φ), and one say that Player 1 (or $\exists$) wins the game of Qsat on Φ if, and only if, after the n-th move ϕ is true.

To show that WHEX is **PSPACE**–complete I will describe below how to construct, using logarithmic space, an instance, (G, s, t), of WHEX from an instance, Φ, of QSAT.

Theorem 4.3 QSAT $\leq_{\log}$ WHEX.

Proof. Let $\Phi := \exists x_1 \forall x_2 \ldots Q_n x_n (C_1 \wedge C_2 \wedge \ldots \wedge C_m)$, where each clause C_i is a conjunction of literals. Define the graph $G_\Phi = (V, E)$ as follows:

$$\begin{aligned}
V &= \{s, t, y\} \cup \{x_i, \overline{x}_i, u_i, \overline{u}_i, v_i : 1 \leq i \leq n\} \\
&\cup \{w_{2i} : 1 \leq i \leq \lceil n/2 \rceil\} \cup \{c_i, z_i : 1 \leq i \leq m\}
\end{aligned}$$

$$\begin{aligned}
E &= \{(s, x_1), (s, \overline{x}_1), (v_n, c_1), (v_n, z_1), (z_m, t), (y, t)\} \\
&\cup \{(x_i, u_i), (\overline{x}_i, \overline{u}_i), (u_i, t), (\overline{u}_i, t), (x_i, v_i), (\overline{x}_i, v_i) : 1 \leq i \leq n\} \\
&\cup \{(v_i, x_{i+1}), (v_i, \overline{x}_{i+1}) : 1 \leq i \leq n-1\} \\
&\cup \{(v_{2i}, w_{2i}), (w_{2i}, t) : 1 \leq i \leq \lceil n/2 \rceil\} \\
&\cup \{(z_i, z_{i+1}), (z_i, c_{i+1}) : 1 \leq i \leq m-1\} \\
&\cup \{(c_i, y) : 1 \leq i \leq m\} \\
&\cup \{(c_i, u_j) : \text{ the literal } \neg x_j \text{ is in clause } C_i\} \\
&\cup \{(c_i, \overline{u}_j) : \text{ the literal } x_j \text{ is in clause } C_i\}
\end{aligned}$$

(Figure 1 illustrates the graph G_Φ for

$$\Phi := \exists x_1 \forall x_2 \exists x_3 [(\neg x_1 \vee \neg x_2) \wedge (x_2 \vee x_3 \vee \neg x_1) \wedge (x_2 \vee \neg x_3)] \;)$$

First note that the moves of Player 1 and Player 2 on the graph G_Φ corresponds to moves by $\exists$ and $\forall$ on Φ. The literal $\overline{x}_i$ stands for the negation of x_i; hence, Player 1 colouring x_{2i-1} or $\overline{x}_{2i-1}$ red corresponds to $\exists$ deciding to give value 1 or 0 to x_{2i-1}. After Player 1's colouring, Player 2 is forced to colour vertex u_{2i-1} or $\overline{u}_{2i-1}$; otherwise Player 1 reaches t in his next move. After the forced colouring of u_{2i-1} or $\overline{u}_{2i-1}$ by Player 2, Player 1's only possible choice is v_{2i-1}. Then it is Player 2's turn to decide whether to colour blue vertex x_{2i} or $\overline{x}_{2i}$, which corresponds to $\forall$ making the decision of assigning value 1 or 0 to x_{2i}. According to the selection of Player 2, Player 1 must continue with colouring the opposite vertex and, again, Player 2 has a forced

[1] This is no loss of generality since we can always add clauses of the form $x \vee \neg x$ without altering the truth value of ϕ

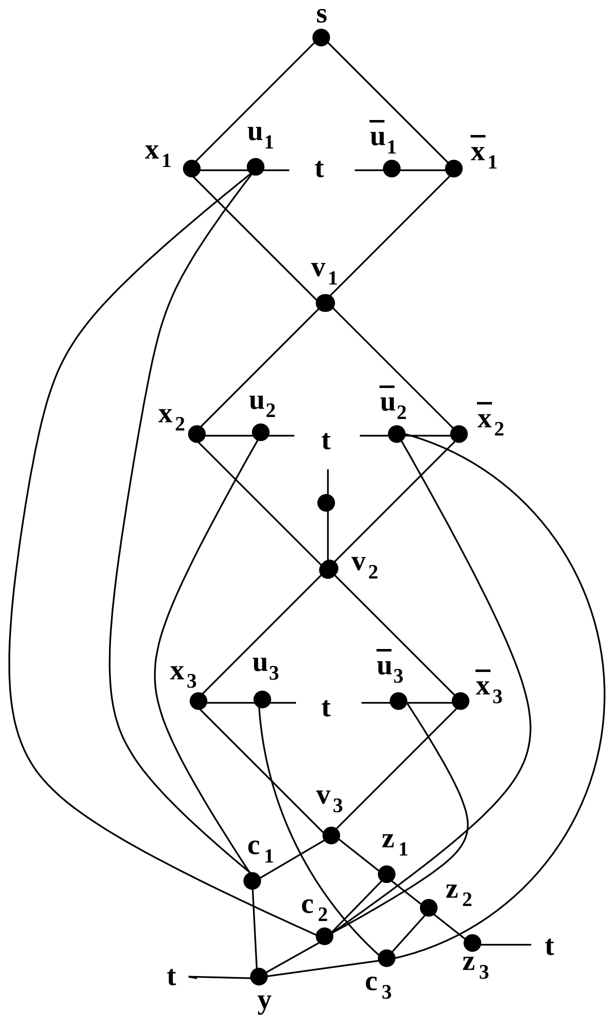

FIG. 1. G_Φ for $\Phi := \exists x_1 \forall x_2 \exists x_3[(\neg x_1 \vee \neg x_2) \wedge (x_2 \vee x_3 \vee \neg x_1) \wedge (x_2 \vee \neg x_3)]$.

response, which obliges Player 1 to colour v_{2i}. From there, Player 1 either wins or, once again, must choose between x_{2i+1} or $\overline{x}_{2i+1}$, and so on.

Now, suppose $\exists$ has a winning strategy in the game of Qsat on Φ. Then Player 1 has the following winning strategy in the game of Whex on G_Φ: if $\exists$ gives value 1 (respectively, 0) to variable x_{2i-1} then Player 1 colours vertex x_{2i-1} (resp. $\overline{x}_{2i-1}$) red. If, when Player 1 reaches vertex v_n, Player 2 subsequently colours all vertices c_i blue, then Player 1 will be left with z_m to colour and wins; otherwise, Player 1 will colour some c_i red. Since all clauses are satisfiable, some literal in C_i is true; therefore there is an edge from c_i to some unvisited vertex u_j or $\overline{u}_j$, besides the edge to y, so Player 1 colours the one left free by Player 2 and wins.

For the converse, suppose $\forall$ has a winning strategy in the game of Qsat on Φ. Then

Player 2 wins the game of Whex in the following way: if $\forall$ gives value 1 (resp. 0) to variable x_{2i}, then Player 2 colours vertex $\overline{x}_{2i}$ (resp. x_{2i}) blue; this forces Player 1 to colour x_{2i} (resp. $\overline{x}_{2i}$) red, and Player 2 to colour u_{2i} (resp. $\overline{u}_{2i}$) blue. When Player 1 reaches v_n, there is one clause C_i that is false, and, therefore, Player 2 forces Player 1 to colour vertex c_i by colouring vertex z_i blue. Then Player 2 colours y, thus succeeding in blocking Player 1, since all other edges lead to vertices u_j already coloured by Player 2.

Finally it is easy to see that the construction of G_Φ from Φ can be done deterministically using logarithmic space. ∎

Next it will be shown that the logic posWHEX*[FO$_s$] has a quantifier–free projective normal form.

Theorem 4.4 Let τ be some vocabulary. Every sentence $\phi \in$ posWHEX*[FO$_s(\tau)$] is equivalent to a sentence of the form WHEX$[\overline{x}, \overline{y} : \psi](\overline{0}, \overline{max})$, where $\psi \in$ FO$_s(\tau)$, ψ is quantifier–free projective and over the distinct k-tuples of variables $\overline{x}$ and $\overline{y}$, for some $k \geq 1$, and where $\overline{0}$ (resp. $\overline{max}$) is the constant symbol 0 (resp. max) repeated k-times.

Proof. (Sketch) The proof runs along the same lines as the proof of the normal form for posHEX*[FO$_s$] in [2], which in turn is inspired by the proof of [9, Theorem 3.3]. I proceed by induction on the complexity of the sentence ϕ, having to consider essentially five cases; the first one being ϕ an atomic or negated atomic sentence which is trivially equivalent to WHEX$[(x_1, y_1), (x_2, y_2) : \phi]((0, 0), (max, max))$ with x_1, x_2, y_1, and y_2 new variables not occurring in ϕ.

For the remaining cases I adapted the constructions done in [9, Theorem 3.3], so as to fit in with the combinatorics of the Whex game, by incorporating the following gadget. Given an undirected graph $G = (V, E)$ with source s and sink t in V, built a new graph $X(G)$ consisting of a copy of G with two sources, s_1 and s_2, in place of s, and two sinks, t_1 and t_2, in place of t. Draw edges between s_1 (respectively s_2) and each vertex that forms an edge with s in G, and, similarly, draw edges between t_1 (respectively t_2) and each vertex that forms an edge with t in G. Add a new vertex w, distinct from all the vertices in G, and edges between w and s_1, s_2, t_1 and t_2, respectively. The gadget $X(G)$ can be seen in Figure 2.

The idea behind $X(G)$ is to guarantee a winning strategy for Player 1 in the game of Whex played on all the set of vertices V, provided that Player 1 has a winning strategy in the game of Whex played on the set of vertices $V - \{s, t\}$. Consider the game of Whex on $X(G)$ as the usual game of Whex on a graph with distinct source and sink, except that Player 1 must begin by colouring one of the two sources, the players continue as in the usual game of Whex from the previous coloured source, and Player 1 wins if he reaches (and colours) one of the two sinks. Suppose Player 1 has a winning strategy in the game of Whex played on G; then, Player 1 wins in the game of Whex played on (all the vertices of) $X(G)$ by first taking one of the sources, and, afterwards, if Player 2 doesn't take w, Player 1 takes it and wins in his next move; otherwise, Player 1 uses his winning strategy for G, which applies *since he is the first one to take a vertex of* $V - \{s, t\}$; he will then reach one of the sinks and win. Conversely, suppose Player 2 has a winning strategy in the game of Whex played on G. Then Player 2 wins the game of Whex played on $X(G)$ by taking, in his first

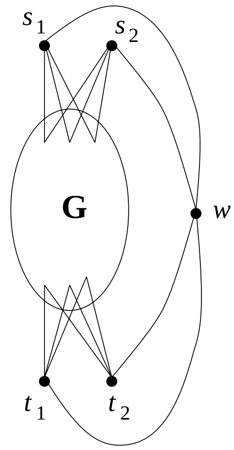

FIG. 2. The gadget $X(G)$.

move, vertex w, and then applying his winning strategy for G, which is possible since Player 1 is the first one to move in $V - \{s, t\}$.

Now, let me explain how to use $X(G)$ to obtain the formulas in normal form. Say we want to eliminate the existential quantifier in $\phi := \exists z \mathrm{WHEX}[\overline{x}, \overline{y} : \theta(\overline{x}, \overline{y}, z)](\overline{0}, \overline{max})$ where θ is, by inductive hypothesis, a quantifier free projection and $|\overline{x}| = |\overline{y}| = k$. Proceed to construct, from a given finite τ_2-structure (a graph) $\mathcal{A} = \langle\{0, 1, \dots, n-1\}, E^{\mathcal{A}}, C^{\mathcal{A}}, D^{\mathcal{A}}\rangle$, the graphs $X(\mathcal{A}_z)$ for each $z \in \{0, 1, \dots, n-1\}$, where each $\mathcal{A}_z$ has universe $|\mathcal{A}|^k$ and edge relation determined by $\theta(\overline{x}, \overline{y}, z)$. Then add two new vertices $\mathbf{s}$ and $\mathbf{t}$ and, for each $z \in \{0, \dots, n-1\}$, join the sources in $X(\mathcal{A}_z)$ to $\mathbf{s}$, and the sinks to $\mathbf{t}$. In the new graph thus obtained, named G_ϕ, $\mathbf{s}$ is codified as a $k+4$ tuple of 0's and $\mathbf{t}$ as a $k+4$ tuple of max's; the vertices in each $X(\mathcal{A}_z)$ are codified as follows: the sources as $(\overline{0}, max, 0, max, z)$ and $(\overline{0}, 0, 0, max, z)$; the sinks as $(\overline{max}, max, 0, max, z)$ and $(\overline{max}, 0, 0, max, z)$; the w as $(\overline{0}, 0, max, 0, z)$, and the other vertices as $(\overline{x}, max, max, 0, z)$ for $\overline{x} \notin \{\overline{0}, \overline{max}\}$. It is not difficult to describe G_ϕ with a quantifier free projection $\psi(\overline{x}, u_1, u_2, u_3, u_4, \overline{y}, v_1, v_2, v_3, v_3)$, and to show that Player 1 has a winning strategy in the game of Whex played on G_ϕ if and only if, for some $z \in \{0, \dots, n-1\}$, Player 1 has a winning strategy in the game of Whex played on $X(\mathcal{A}_z)$.

To eliminate the universal quantifier place the gadgets in series, and do a similar codification as in the existential case. It is here where the ordering relation is needed to write the appropriate quantifier free formula. The nested case is resolved in a similar manner by incorporating the $X(G)$ gadget to the construction of the nested case for TC in [9, Theorem 3.3], and following that proof the reader can write the appropriate formulas for our present problem. This completes the proof of the theorem. ∎

Corollary 4.5 PSPACE = $\mathrm{WHEX}^*[\mathrm{FO}_s] = \mathrm{posWHEX}^*[\mathrm{FO}_s] = \mathrm{posWHEX}^1[\mathrm{FO}_s]$ and WHEX is complete for **PSPACE** via quantifier free projections.

PROOF. Put together Theorems 4.3, 4.2, 3.5 and 4.4. ∎

I would like to remark that the variations of the game of Whex where: 1) players colour vertices of an *undirected* graph; 2) players colour *edges* of a directed or undirected graph, remain complete for **PSPACE** via quantifier free projections. To see the second point, observe that colouring an edge $\{u, v\}$ adjacent to an edge $\{w, u\}$ coloured in a preceding move has the same effect of colouring the vertex v adjacent to the previously coloured vertex u; hence, all results for the Whex game on vertices apply with minor fixes to the game of Whex on edges. This contrast with Hex, where playing on edges does make a difference in the computational complexity of the problem, as it was remarked before.

Also, it should be noted that the version of the game of Whex where Player 2 begins the game (by colouring blue a vertex adjacent to the source), and where we ask the same query as before, namely, does Player 1 has a winning strategy?, yields a decision problem, which it will be distinguished by WHEX$'$, with the same computational and logical characteristics than our original WHEX (where Player 1 is the first one to play). The proofs of these facts amounts to slight modifications of the proofs given for the corresponding facts for WHEX. For example, to show WHEX$'$ is complete for **PSPACE**, view QSAT as the same alternating game among $\exists$ and $\forall$, but with $\forall$ beginning the game instead of $\exists$, and the board being formulas with alternating quantifiers beginning with $\forall$. Then the construction in Theorem 4.3 goes through and we have QSAT $\leq_{\log}$ WHEX$'$. We shall find WHEX$'$ more convenient to use for the application of our games given in the next section.

5 The complexity of a user–aided proof system

Let $\tau_3 = \{R, C, D\}$ be a vocabulary with a relation symbol R of arity 3 and two constant symbols C and D. A τ_3–structure $\mathcal{A}$ of size n can be regarded as a *path system*, that is, a set A of n vertices, a relation $R^{\mathcal{A}} \subseteq A \times A \times A$, a source $C^{\mathcal{A}} \in A$ and a sink $D^{\mathcal{A}} \in A$, where in order to reach a vertex z, there must be two other reachable vertices x and y so that $(x, y, z) \in R^{\mathcal{A}}$. This view of structures over τ_3 is suitable for encoding the problem *Path System Accessibility* (see [7]), whose instances are path systems and yes–instances are instances where the sink is accessible from the source, a vertex z being accessible if it is the source of the path system or if $R(x, y, z)$ holds for some accessible vertices x and y. The resulting class of τ_3-structures corresponding to Path System Accessibility was denoted PS by Iain Stewart in [13], considered by him as a generalized quantifier and added to first order logic to obtain the extension PS*[FO$_s$], path system logic, which satisfies a projective normal form and which captures the class of polynomial time computable problems, **P**.

Theorem 5.1 ([13]) PS*[FO$_s$] = posPS*[FO$_s$] = posPS^1[FO$_s$] = **P** and the normal form for posPS*[FO$_s$] is quantifier free projective. Therefore, the problem PS is complete for **P** via quantifier free projections. ∎

Alternatively, we can regard a τ_3–structure $\mathcal{A}$ of size n as a *proof system*, by interpreting the elements of $\mathcal{A}$ as statements and the 3–ary relation $R^{\mathcal{A}}$ as a rule of inference, say, for example, one like resolution, which when applied to statements x and y yield the statement z, whenever $(x, y, z) \in R^{\mathcal{A}}$. This is the way I like to think of τ_3–structures here, and, under this view, the problem PS is the set of proof systems $\mathcal{A}$ where the statement $D^{\mathcal{A}}$ is *provable* from the statement $C^{\mathcal{A}}$, where *provable* is synonymous with *accessible* and defined likewise.

Considering τ_3–structures as proof systems, one then wonder about the complexity of proving theorems in a proof system where the user can help the system by adding "lemmas" along a proof, which are not necessarily provable from the initial statement $C^{\mathcal{A}}$. The intended meaning of this last statement is formalize in the following definition.

Definition 5.2 Let $\mathcal{A} = \langle A, R^{\mathcal{A}}, C^{\mathcal{A}}, D^{\mathcal{A}} \rangle$ be a proof system. A statement z in $\mathcal{A}$ is *partially provable* (*p.p*) if it is $C^{\mathcal{A}}$ or it is obtained from some partially provable statement x and an arbitrary, non partially provable statement y, by applying the rule R to x and y (i.e. $(x, y, z) \in R^{\mathcal{A}}$).

A statement w is *provable from u by a proof of partially provable statements* if, and only if, there is a sequence of partially provable statements $c_1, c_2, \ldots, c_n$, with $c_n = w$, which witnesses the following set of expressions:

$$\forall b_1 \exists c_1 : (b_1 = w \text{ or } b_1 \text{ is not p.p.}) \text{ and } (u, b_1, c_1) \in R \tag{5.1}$$

and, for $1 \leq i < n$,

$$\forall b_{i+1} \exists c_{i+1} : (b_{i+1} = w \text{ or } b_{i+1} \text{ is not p.p.}) \text{ and } (c_i, b_{i+1}, c_{i+1}) \in R. \tag{5.2}$$

Define the problem Partial Proof System (PPS) as the following class of τ_3–structures:

$$\begin{aligned} \text{PPS} \quad := \quad & \{\mathcal{A} \in \text{STRUCT}(\tau_3) : \mathcal{A} \text{ is a proof system, where } D^{\mathcal{A}} \\ & \text{is provable from } C^{\mathcal{A}} \text{ by a proof of p.p. statements}\} \end{aligned}$$

It is not difficult to see that PPS is in **PSPACE**: at the i–th step of a proof we need to have in store the last statement c_i partially provable, nondeterministically generate and store a statement b_{i+1}, and store a c_{i+1} for which $(c_i, b_i, c_{i+1}) \in R^{\mathcal{A}}$. Furthermore, we have

Theorem 5.3 WHEX$'$ $\leq_{\log}$ PPS.

Proof. Given an undirected graph $G = \langle V, E, u, w \rangle$ with source u and sink w, define the proof system $\mathcal{A}_G = \langle A, R, s, t \rangle$, with $A = V$, $s = u$, $w = t$ and

$$\begin{aligned} R \quad = \quad & \{(a, b, c) : E(a, c) \wedge E(a, b) \wedge b \neq c\} \\ & \cup \{(a, w, w) : E(a, w)\} \end{aligned}$$

Suppose $G \in$WHEX. Then, for whatever vertex b_1 Player 2 starts off the game of Whex on G, Player 1 can respond with c_1 such that $E(u, b_1)$ and $E(u, c_1)$ holds (and c_1 is on a path to w). Successively, in the i–th move, for whatever vertex b_i Player 2 selects, Player 1 can respond with a c_i such that $E(c_{i-1}, b_i)$ and $E(c_{i-1}, c_i)$ holds (and c_i is on a path to w), and so on, until Player 1 reaches a vertex c_n such that $E(c_n, w)$ holds. Then the sequence $c_1, c_2, \ldots, c_n, w$, is made of partially provable statements in $\mathcal{A}_G$, satisfying conditions (5.1) and (5.2). Hence, it constitutes a proof of t from s and, hence, $\mathcal{A}_G \in$PPS.

Conversely, if we assume $\mathcal{A}_G \in$PPS, then there is a proof of t from s by partially provable statements, which satisfies conditions (5.1) and (5.2). These conditions describe a winning strategy for Player 1 in the version of the game of Whex played on G where Player 2 makes the first move. ∎

We have that the problem PPS is complete for **PSPACE** via log–space reducibility. It is argue next why PPS is also complete via quantifier free projections. First, consider PPS as a generalized quantifier and add it to first order logic (with built–in successor), and form the logic $\text{PPS}^*[\text{FO}_s]$. Then observe that the problem DTC is expressible in $\text{posPPS}^*[\text{FO}_s]$: the intuitive idea is that for a given graph G, with distinguished vertices s and t, a path $\langle s, c_1, c_2, \ldots, c_n, t\rangle$ from s to t can be seen as a proof of t from s by the following sequence of applications of the rule R:

$$(s,t,c_1),(c_1,t,c_2),\ldots,(c_i,t,c_{i+1}),\ldots,(c_n,t,t)$$

Next, to show that $\text{posPPS}^*[\text{FO}_s] = \text{posPPS}^1[\text{FO}_s]$ proceed almost identically as in Stewart's proof of the normal form for $\text{PS}^*[\text{FO}_s]$ (Theorem 4.2 of [13]). For example, to eliminate the existential quantifier in the sentence

$$\exists w \text{PPS}[x,y,z:\psi(x,y,z,w)](C,D)$$

consider, for a structure $\mathcal{A}$ of size n, a disjoint union of the n proof systems described by $\psi^{\mathcal{A}}(x,y,z,i)$, for $i=0,1,\ldots,n-1$, with a common initial statement and a common final statement. We then have a result analogous to Theorem 5.1.

Theorem 5.4 $\text{PPS}^*[\text{FO}_s] = \text{posPPS}^*[\text{FO}_s] = \text{posPPS}^1[\text{FO}_s] =$ **PSPACE** and the normal form for $\text{posPPS}^*[\text{FO}_s]$ is quantifier free projective. Therefore, the problem PPS is complete for **PSPACE** via quantifier free projections (with the successor relation). ∎

(An alternative way of proving the completeness of PPS via qfp's follows from observing that the reduction of WHEX′ to PPS in Theorem 5.3 can be described by quantifier free projective formulas. On the other hand, WHEX′ can be shown complete via qfp's by same arguments employed for the completeness of WHEX, and then use that quantifier free projections are transitive [10].)

The above theorem, together with Theorem 5.1, have as consequence a characterization of the **P** versus **PSPACE** problem. **P** $=$ **PSPACE** if and only if PPS is first order reducible to PS. This tells us, informally, that in order for the class **PSPACE** be equal to **P**, it is sufficient to show that any proof system that uses auxiliary statements in its proofs, can be effectively simulated by a proof system where all proofs are constituted by provable statements from the initial one $C^{\mathcal{A}}$.

One more observation about the problem PPS is that it is definable in *partial fixed point logic*, or PFP[PO]. This is the closure of first order logic with the partial fixed point operator PFP and all boolean operations. PFP applies to formulas of the form $\varphi(x_1,\ldots,x_k,R)$, where R is a relational variable of arity k, to make up the formula $\text{PFP}[\overline{x},R:\varphi(\overline{x},R)](t_1,\ldots,t_k)$ whose interpretation is as follows: for an appropriate structure $\mathcal{A}$, $\mathcal{A}\models \text{PFP}[\overline{x},R:\varphi(\overline{x},R)](t_1,\ldots,t_k)$ if and only if $(t_1,\ldots,t_k)^{\mathcal{A}}$ is in the fixed point (if it exists) of the sequence of sets

$$\varphi^0_{\mathcal{A}} := \{(a_1,\ldots,a_k)\in A^k : \mathcal{A}\models\varphi(a_1,\ldots,a_k,\emptyset)\}$$

and for $i>0$

$$\varphi^i_{\mathcal{A}} := \{(a_1,\ldots,a_k)\in A^k : \mathcal{A}\models\varphi(a_1,\ldots,a_k,\varphi^{i+1}_{\mathcal{A}})\}$$

(For further details see [5].)

Now, it is not difficult to see that PPS is definable in PFP[PO] by the sentence

$$\exists w(\mathrm{PFP}[z, X : z = C \quad \vee \quad \exists x \forall y(X(x) \wedge (\neg X(y) \vee y = D) \wedge R(x, y, z))](w) \wedge w = D) \tag{5.3}$$

(essentially we put in the set X the partially provable statements, beginning with C and until we reach D).

If we consider PFP[PO$_s$], namely, PFP over first order logic with the built–in successor, then using the facts just proved, that PPS is complete for **PSPACE** via qfp's with successor, and that PPS is definable in PFP[PO] (hence in PFP[PO$_s$]), we obtain that every problem computable by Turing machines with a polynomial space bound is definable in PFP[PO$_s$]. Conversely, the set of finite structures that satisfy a given sentence in PFP[PO$_s$] can be decided by a polynomial space bounded algorithm (see [5]). Thus, we have another proof of Abitebuol and Vianu's logical characterization of **PSPACE**:

Theorem 5.5 ([1]) PSPACE = PFP[PO$_s$]. ∎

6 WHEX on graph with outdegree ≤ 3.

Let WHEX3 be the set of digraphs of outdegree at most 3 which are yes–instances of WHEX. I shall reduce the problem QSAT to WHEX3 via a quantifier free projection. To do that view QSAT as a problem over the vocabulary $\sigma = \{P, N, U\}$, where P and N are binary relation symbols and U is unary; then, an instance of QSAT is a finite σ–structure $\mathcal{A} = \langle A, P, N, U\rangle$, where $P(i, j)$ holds in $\mathcal{A}$ if and only if variable j occurs positive in clause i, $N(i, j)$ holds in $\mathcal{A}$ if and only if variable j occurs negative in clause i, and $U(i)$ holds in $\mathcal{A}$ if and only if variable i is universally quantified. A yes–instance of QSAT is an instance where Player 1 has a winning strategy in the game of Qsat described previously.

Theorem 6.1 QSAT $\leq_{qfp}$ WHEX3.

Proof. The strategy is as follows: first, a log–space reduction from QSAT to WHEX3 is given, which is essentially the reduction of QSAT to WHEX given in Theorem 4.3 with suitable modifications meant to keep the outdegree below 3. Then it is indicated how to codify the reduction and express it with a quantifier free first order projection.

Let $\Phi := \exists x_1 \forall x_2 \cdots Q_n x_n(C_1 \wedge C_2 \wedge \ldots \wedge C_k)$, where each clause C_i is a conjunction of literals. Observe that n is the number of variables in Φ, and k is the number of clauses. Define the graph $G_\Phi = \langle V, E\rangle$ as follows:

$$\begin{aligned} V &= \{s, t, y\} \cup \{x_i, \overline{x}_i, u_i, \overline{u}_i, v_i : 1 \leq i \leq n\} \\ &\cup \{w_{2i} : 1 \leq i \leq \lceil n/2 \rceil\} \cup \{c_i, z_i : 1 \leq i \leq k\} \\ &\cup \{p_i^j, n_i^j, y_i^j : 1 \leq j \leq n, 1 \leq i \leq k\}. \end{aligned}$$

$$\begin{aligned} E &= \{(s, x_1), (s, \overline{x}_1), (v_n, c_1), (v_n, z_1), (z_k, t), (y, t)\} \\ &\cup \{(x_i, u_i), (\overline{x}_i, \overline{u}_i), (u_i, t), (\overline{u}_i, t), (x_i, v_i), (\overline{x}_i, v_i) : 1 \leq i \leq n\} \\ &\cup \{(v_i, x_{i+1}), (v_i, \overline{x}_{i+1}) : 1 \leq i \leq n-1\} \end{aligned}$$

$$
\begin{aligned}
\cup \quad & \{(v_{2i}, w_{2i}), (w_{2i}, t) : 1 \leq i \leq \lceil n/2 \rceil\} \\
\cup \quad & \{(z_i, z_{i+1}), (z_i, c_{i+1}) : 1 \leq i \leq k-1\} \\
\cup \quad & \{(c_i, y), (c_i, p_i^1), (c_i, n_i^1) : 1 \leq i \leq k\} \\
\cup \quad & \{(p_i^j, y_i^j), (n_i^j, y_i^j), (y_i^j, t) : 1 \leq i \leq k, 1 \leq j \leq n\} \\
\cup \quad & \{(p_i^j, p_i^{j+1}), (n_i^j, n_i^{j+1}) : 1 \leq i \leq k, 1 \leq j \leq n-1\} \\
\cup \quad & \{(p_i^j, \ddot{u}_j) : \text{the literal } \ x_j \ \text{ is in clause } \ C_i\} \\
\cup \quad & \{(n_i^j, u_j) : \text{the literal } \ \neg x_j \ \text{ is in clause } \ C_i\}.
\end{aligned}
$$

G_Φ is a graph with outdegree at most 3, and it is not difficult to see that Player 1 has a winning strategy in the Qsat game played on Φ if and only if Player 1 has a winning strategy in the Whex game played on (G_Φ, s, t). In order to write the above reduction as a quantifier free formula, begin by codifying the 14 types of vertices as the following sextuples (max is abbreviated as m): $s = (0,0,0,0,0,0)$, $t = (m,m,m,m,m,m)$, $y = (m,m,m,0,0,0)$, $x_i = (i,0,m,m,m,m)$, $\overline{x}_i = (i,m,0,m,m,m)$, $u_i = (i, 0, 0, m, m, m)$, $\overline{u}_i = (i,m,0,0,m,m)$, $v_i = (i,0,m,0,m,m)$, $w_i = (i,m,m,0,m,m)$, $c_i = (i, m, m, m, 0, m)$, $z_i = (i,m,m,0,0,m)$, $p_i^j = (i,j,0,m,0,m)$, $n_i^j = (i,j,0,m,0,0)$, and $y_i^j = (i, j, 0, 0, m, 0)$. Then I proceed to define the edge relation with a formula in the variables $x_1, x_2, \ldots, x_6, y_1, y_2, \ldots, y_6$, consisting of disjunctions of clauses, one for each of the type of edges described above, and most of which can be easily seen to be a quantifier free conjunction. Therefore, I just write down the (possibly) not so trivial quantifier free formulas that describe some of the edges. Take for example an edge of the form (v_{2i}, w_{2i}). This is an edge corresponding to variable i universally quantified, and so is equivalent to have $(v_i, w_i) \longleftrightarrow U(i)$ holding in G_Φ; hence, we can express this edge with the formula:

$$(\neg\phi_{vw} \wedge \neg U(x_1)) \vee (\phi_{vw} \wedge U(x_1))$$

where

$$
\begin{aligned}
\phi_{vw} \quad := \quad & x_1 = y_1 \wedge x_2 = x_4 = y_4 = 0 \\
\wedge \quad & x_3 = x_5 = x_6 = y_2 = y_3 = y_5 = y_6 = m
\end{aligned}
$$

Edges such as $(p_i^j, \overline{u}_j)$ are given by the formula

$$
\begin{aligned}
& x_2 = y_1 \wedge x_3 = x_5 = y_3 = y_4 = 0 \\
\wedge \quad & P(x_1, x_2) \wedge x_4 = x_6 = y_2 = y_5 = y_6 = m
\end{aligned}
$$

Edges such as (n_i^j, u_j) are given by the formula

$$
\begin{aligned}
& x_2 = y_1 \wedge x_3 = x_5 = x_6 = y_2 = y_3 = 0 \\
\wedge \quad & N(x_1, x_2) \wedge x_4 = y_4 = y_5 = y_6 = m
\end{aligned}
$$

And for edges such as (v_i, x_{i+1}), (z_i, z_{i+1}), or (p_i^j, p_i^{j+1}) we use the (built–in) successor relation to write up appropriate quantifier free conjunctions. This ends the proof of the theorem. ∎

Corollary 6.2 WHEX3 is complete for **PSPACE** via first order reductions, and

$$\text{posWHEX3}^1[\text{FO}_s] = \text{WHEX3}^1[\text{FO}_s] = \textbf{PSPACE}.$$

∎

The problem WHEX2 whose instances are digraphs of outdegree at most 2, and yes–instances are instances where Player 1 has a winning strategy for the game of Whex, can be solved in polynomial time. This is immediate, since there is at most one alternative for each player's next move, and so all the moves are forced.

7 WHEX on unordered structures

In this section a comment is made on the expressive capabilities of WHEX, as a generalized quantifier, with respect to properties of arbitrary (but finite) structures. If we do not include the built–in successor relation, so that our input structures can be unordered as well as ordered, then the logic WHEX*[FO] has a 0–1 law, since WHEX is a problem closed under extensions and, thus, satisfies the conditions for logics with generalized quantifiers to have asymptotic probability equal to either 0 or 1, as established in [3] for graph problems and later generalized to any problem in [14]. Therefore, WHEX*[FO] does not capture **PSPACE**.

Also, just as it was shown in [2] that HEX*[FO] does not have a normal form using suitable Ehrenfeucht–Fraissé type of games, the same can be shown for the logic WHEX*[FO]:

Theorem 7.1 There are problems definable in WHEX*[FO] which can not be defined by a sentence of WHEX^1[FO] in which the operator WHEX does not appear within the scope of the quantifier $\forall$.

PROOF. The proof is the same as the proof of [2, Proposition 4]. Incidentally, the structures $\mathcal{S}_m$ and $\mathcal{T}_m$ in the proof of that proposition are also in WHEX and not in WHEX, respectively. ∎

8 Final Remarks

WHEX illustrates the necessity of developing tools for sharpening the classification of computational problems. It is a problem based on a game where, intuitively, should be easier than the game of Hex to design winning strategies. However I have shown how similar WHEX and HEX are with respect to their computational complexity and logic expressive power. Nonetheless, I believe some distinctive features among these two problems can be obtained by further exploiting their logical characteristics (besides the few structural differences pointed out through this paper). For example, after Theorem 5.4, it was remarked that WHEX$'$, and hence WHEX, can be reduced to PPS via a quantifier free projection; sustituting in formula (5.3) the relation R by its description in the vocabulary $\tau_2 = \{E, C, D\}$, as suggested in the proof of Theorem 5.3, we get a definition of WHEX in the logic PFP[PO] by a formula structurally simple: it contains a second order variable of arity 1 and the quantifiers (PFP and the first order quantifiers) are relativized by atomic formulas. This fit the pattern of formulas in a *guarded fixed point logic* (see [8]), therefore suggesting a finer subclass of **PSPACE** problems (where WHEX and PPS belongs) as those problems definable in some fragment of the logic PFP[PO], possibly a "guarded partial fixed point" logic. I will leave this characterization as an open problem.

Finally, I will end with the following application of WHEX to a problem of communication in networks, suggested to me by Iain Stewart. Given a network with a source

s, a sink t and a fixed positive integer λ, consider each edge between two nodes i and j as having a valuation t_{ij}, which might represent the time that a message takes to go from i to j. We consider also that, for each node i, up to λ nodes might fail in processing the information per unit of time t_{ij} among i and any other adjacent node j. We wish then to know if a message can be send from s to t, and in the afirmative, which is the strategy for transmission that gives the less time possible. Observe that WHEX is a particular instance of this problem: take $\lambda = t_{ij} = 1$; hence, this problem is **PSPACE**–hard.

Acknowledgement: I came around to the definition of WHEX, its logical and computational properties, through various conversations with Iain Stewart. I am grateful to him for all his help. Thanks also to EPSRC for a Visiting Fellowship (Grant GR/M 91006) that made possible one of my visits to Stewart's research group at Leicester. I am also grateful to the referees for their many valuable remarks and comments.

References

[1] S. Abiteboul and V. Vianu. Fixpoint extensions of first–order logic and Datalog–like languages, in: Proc. 4th IEEE Ann. Symp. on Logic in Comp. Sci. (1989) 71–79.

[2] A.A. Arratia-Quesada and I.A. Stewart. Generalized Hex and logical characterizations of polynomial space. Inf. Procs. Letters 63 (1997) 147–152.

[3] A. Dawar and E. Grädel. Generalized quantifiers and 0–1 laws, in: Proc. 10th IEEE Ann. Symp. on Logic in Comp. Sci. (1995) 54–64.

[4] H.D. Ebbinghaus. Extended logics: the general framework, in: Barwise and Feferman, eds., Model Theoretic Logics (Springer–Verlag, 1985).

[5] H.D. Ebbinghaus and J. Flum. Finite Model Theory (Springer-Verlag, 1995).

[6] S. Even and R.E. Tarjan. A combinatorial problem which is complete in polynomial space. J. Assoc. Comput. Mach. 23 (1976) 710–719.

[7] M. R. Garey and D. S. Johnson. Computers and Intractability - A Guide to the Theory of NP-Completeness (W. H. Freeman and Co., 1979).

[8] E. Grädel and I. Walukiewicz. Guarded fixed point logic, in: Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS '99, Trento, pp. 45-54, 1999.

[9] N. Immerman. Languages that capture complexity classes. SIAM J. of Computing 16, 4 (1987) 760–778.

[10] N. Immerman. Descriptive Complexity (Springer, 1998).

[11] T.J. Schaefer. On the complexity of some two-person perfect information games. J. Comp. Syst. Sci. 16 (1978) 185–225.

[12] I.A. Stewart. Comparing the expressibility of languages formed using NP–complete operators. J. Logic Computat. 1 (1991) 305-330.

[13] I.A. Stewart. Logical description of monotone NP problems. J. Logic Computat. 4 (1994) 305-330.

[14] I.A. Stewart. Program schemes, arrays, Lindström quantifiers and zero-one laws, in: J. Flum and M. Rodriguez-Artalejo, eds., Proc. 8th Ann. Conference of the European Association for Computer Science Logic (LNCS Vol. 1683, Springer-Verlag, Berlin, 1999).

[15] L.J. White. An efficient algorithm for the Shannon switching game. Ann. New York Acad. Sci. 175 (1970) 411–412.

Received 21 of March, 2002