Lattice Embeddings for Abstract Bounded Reducibilities

Wolfgang Merkle

Mathematisches Institut der Universität Heidelberg
Heidelberg, Germany *

Abstract. We give an abstract account of resource bounded reducibilities as exemplified by the polynomial time or logarithmically space bounded reducibilities of Turing, truth-table, and many-one type. We introduce a small set of axioms which are satisfied for most of the specific resource bounded reducibilities which appear in the literature. Some of the axioms are of a more algebraic nature, such as the requirement that the reducibility under consideration is a reflexive relation, while others are formulated in terms of recursion theory and for example are related to delayed computations of arbitrary recursive sets. We discuss basic consequences of these axioms and their relation to previous axiomatic approaches by Mehlhorn [31], Schmidt [41], Mueller [37], and, in a context of relativized Blum measure, by Lynch et al. [26].
As main technical result we show that for every reducibility which satisfies our axioms, every countable distributive lattice can be embedded into every proper interval of the structure induced on the recursive sets. This result extends a corresponding result for polynomial time bounded reducibilities due to Ambos-Spies [1], as well as work by Mehlhorn [31]. Mehlhorn shows from an apparently more restrictive set of assumptions that the recursive sets form a dense structure and claims that in fact every countable partial ordering can be embedded into every proper interval of the recursive sets.

* Universität Heidelberg, Mathematisches Institut, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany; electronic mail: ⟨merkle@math.uni-heidelberg.de⟩ .

Table of Contents

1 Introduction

1.1 Introduction and Overview

Reducibilities such as truth-table reducibility [2] $\leq_{tt}$ or polynomial time bounded Turing reducibility $\leq_T^{\mathcal{P}}$ are effective in the sense that a fact such as $A \leq_T^{\mathcal{P}} B$ implies that A is equal to $\Phi_e(B)$ for some e in ω. Here we denote by Φ_e the partial recursive functional computed by the e-th oracle Turing machine T_e. Such effective reducibilities are usually defined via appropriately constraining Turing reducibility, that is, a set A will be reducible to a set B iff A is equal to $\Phi_e(B)$ for some e where e and B satisfy certain conditions. Standard examples for such conditions are given by restrictions on the way T_e might access its oracle, such as for reducibilities of many-one type, and by bounds on the amount of resources T_e might use, such as for time or space bounded reducibilities. Often we can put the restrictions on the oracle B and the Turing machine performing a reduction into a convenient general form. For example if $\leq_r$ is any of the usual time or space bounded reducibilities, then there is some recursive set E such that for all subsets A and B of ω we have

$$A \leq_r B \quad \text{iff} \quad \exists e \in E \; A = \Phi_e(B) \tag{1}$$

and where in addition E contains only indices of recursive functionals, that is, of partial recursive functionals which are defined for all oracles and number arguments. This feature of time and space bounded reducibilities suggests the following generalization: a binary relation $\leq_r$ on 2^ω is a *bounded reducibility* iff there is a recursive set E such that E contains only indices of total recursive functionals and satisfies (1).

The concept bounded reducibility indeed comprises most of the resource bounded reducibilities which can be found in the literature. Now for several of these specific bounded reducibilities it is known that the respective structures induced on the recursive sets resemble one another w.r.t. properties such as density or embeddability of lattices. These similarities suggest that to some extent non-trivial structural properties of bounded reducibilities might be developed within a generalized or axiomatic approach to bounded reducibilities. However, as we will show in Example 11, even such a basic result as the density of the structure induced on the recursive sets is false for bounded reducibilities in general. This indicates that there is no hope for deriving interesting structural results about bounded reducibilities without adding further conditions or axioms which for example assure nice algebraical properties, such as the existence of least upper bounds for the reducibility under consideration, or which provide,

[2] We introduce the specific reducibilities used as examples by informal descriptions. Full definitions can be found in Sect. 2.7 or in the textbooks by Balcázar et al. [6, 7] and by Odifreddi [38].

intuitively speaking, a minimal amount of computational power that can be used in reducing one set to another. In fact, a major part of the burden in developing an abstract approach to bounded reducibilities consists in the design of a small and intuitively meaningful set of such additional conditions which on the one hand is general enough to be widely applicable, but on the other hand is strong enough to imply interesting consequences.

Our abstract approach to resource bounded reducibilities is based on the the concept standard reducibility as introduced in Sect. 2.6. Archetypal examples of standard reducibilities on 2^ω are polynomial time bounded Turing reducibility $\leq_T^{\mathcal{P}}$ and logarithmic space (logspace) bounded many-one reducibility $\leq_m^{log}$. The definition of standard reducibility excludes bounded reducibilities arising from more restricted models of computations such as Turing reducibility confined to constant space or the reducibility $\leq_1^{\mathcal{P}}$, that is, the variant of polynomial time bounded many-one reducibility where the reduction functions are required to be 1:1.

In Sect. 3, we derive some basic properties of standard reducibilities and we discuss their relations with other abstract approaches to resource bounded reducibilities. The main technical result on standard reducibilities of this paper is shown in Sect. 4: every countable distributive lattice can be embedded into every proper interval of the structure induced on the recursive sets with least or greatest element preserved.

1.2 Related Work

Research on bounded reducibilities usually does not deal with the structure induced on all sets by the reducibility under consideration, but with the substructures induced on the class REC of recursive subsets of ω. Early results obtained for the structures $(\mathrm{REC}, \leq_T^{\mathcal{P}})$ and $(\mathrm{REC}, \leq_m^{\mathcal{P}})$ evolving from polynomial time bounded Turing and many-one reducibility are the density of the structures and the existence of minimal pairs, both due to Ladner [23]. Then Machtey [28] constructed a minimal pair of sets computable in exponential time, and Landweber et al. [25] further improved on this: actually any recursive set strictly above $\emptyset$ bounds a minimal pair. More precisely, in case $\leq_r$ is equal to $\leq_T^{\mathcal{P}}$ or to $\leq_m^{\mathcal{P}}$, we obtain that

- the structure $(\mathrm{REC}, \leq_r)$ is dense, that is, if A and B are recursive sets where $A <_r B$, then there is a recursive set C such that $A <_r C <_r B$,
- the structure $(\mathrm{REC}, \leq_r)$ contains minimal pairs, that is, there are recursive sets A_1 and A_2 where, firstly, $\emptyset <_r A_1$ and $\emptyset <_r A_2$, and, secondly, for all recursive sets C the facts $C \leq_r A_1$ and $C \leq_r A_2$ together imply $C \leq_r \emptyset$; moreover, minimal pairs exist below any recursive set B strictly above $\emptyset$, that is, if $\emptyset <_r B$ holds for a recursive set B, then there are recursive sets A_1 and A_2 as above which are both $\leq_r$ - reducible to B.

Subsequently this results were extended and the presentation of their proofs was substantially improved by several authors including Mehlhorn [29, 31], Chew and Machtey [15], Balcázar and Díaz [5], and Schöning [42, 43]. Then Ambos-Spies [1] showed a general embedding theorem which contains several preceding results as special cases: if $\leq_r$ is equal to $\leq_T^{\mathcal{P}}$, to $\leq_m^{\mathcal{P}}$, or to one of several variants of polynomial time bounded truth table reducibility $\leq_{tt}^{\mathcal{P}}$, then

- every countable distributive lattice can be embedded (as a lattice) into any proper interval of $(\mathrm{REC}, \leq_r)$ with least or greatest element preserved.

From this result we obtain for example Ladner's density result by embedding the three-element total ordering into a given proper interval, and the existence of minimal pairs below a recursive set B which is strictly above $\emptyset$ follows by embedding the four-element Boolean algebra below B with least element preserved.

Already in [23], Ladner pointed out that his results go through for rather general types of time, respectively space, bounded reducibilities. Subsequently also the stronger results mentioned were transferred to bounded reducibilities which are not defined by polynomial time bounds.

- Serna proved in her doctoral dissertation that $(\mathrm{REC}, \leq_{NC_1})$ is dense and contains minimal pairs below any set strictly above $\emptyset$.
- Vollmer [48] extends the results on lattice embeddings due to Ambos-Spies to the relation $\leq_m^{log}$ and to other reducibilities defined in terms of logarithmic space bounds, and he indicates how the proof of the original result can be transferred from polynomial time to logarithmic space.

The facts that several bounded reducibilities bear similar structural properties and, what is more, that the same proof techniques apply in the different cases suggest that there might be an axiomatic framework in which at least for a subclass of all bounded reducibilities common properties can be derived from general conditions or axioms. We conclude this section by a non-exhaustive survey on work in this direction.

First, there is work on structural properties of bounded reducibilities by Basu [8], Mehlhorn [29, 31], Mueller [37], and Schmidt [41], which is carried out within axiomatizations which allow a uniform treatment of a wide class of resource bounded reducibilities. Our generalized approach is closely related to the work of the latter three authors, while Basu's axiom system is designed to be used in connection with reducibilities between functions and is to general if applied to reducibilities between sets. In Sect. 3.7, we give a sketchy survey on the axiomatizations of the other three authors. There we focus on the work of Mehlhorn who has been most influential to our approach via his concept delayed simulation.

Next there are appealing approaches which have apparently less bearing on the work presented here. Lynch, Meyer, and Fischer [26] obtain interesting and rather general results by extending the concept of Blum measure to relativized computations; we briefly discuss their work in Sect. 3.5. Book, Du and Russo, see [10] and the references cited there, extend previous work on polynomial time complexity cores to complexity cores which are defined w.r.t. countable or recursively presentable classes which satisfy properties such as being closed under finite join and under finite variation.

Further there are approaches which are not completely satisfactory in our view, because time and space bounds are treated separately. These approaches, as exemplified by Sect.6 of Ladner [23], usually amount to considering reducibilities defined by classes of time or space bounds which satisfy certain properties such as including functions which grow sufficiently fast. In addition there are approaches such as [27] and [30] where the authors claim that their results hold for a wide class of bounded reducibilities, but fail to give an exhaustive list of the assumptions they work from. For example in [30] it is stated that subrecursive degrees are always half of a minimal pair of recursive sets. However, in the corresponding proof more specific assumptions are used and in particular, as Mehlhorn remarks in [31, p.164], the methods used are not applicable to polynomial time bounded reducibilities (see also Remark 59).

There are recent axiomatic approaches to resource bounded reducibilities in the context of separations by random oracles, Almost classes, and bounded error probabilistic classes by Book, Lutz, and Wagner [11], Book, Vollmer, and Wagner [12, 13], by Merkle and Wang [35], and by Regan and Royer [40]. Recall in this connection that a binary relation on 2^ω is a bounded reducibility iff it can be defined via some *recursive* set of indices of *recursive* functionals. In the papers cited, a more general concept of reducibilities is considered where both effectivity conditions have been dropped. The results obtained for reducibilities of this general type rely heavily on the fact that the concepts involved are formulated in terms of measure theory, and that accordingly some of the proof techniques employed do not involve any form of effectivity.

Finally we give references to further work on standard reducibilities. In [33], a result on polynomial time bounded reducibilities due to Ambos-Spies [2] is extended to standard reducibilities: every recursive set which is $\leq_r$ - hard for some recursively presentable $\leq_r$ - ideal $\mathcal{I}$ is half of an exact pair for $\mathcal{I}$ where the other half is also recursive. In [32], it is shown that for a transitive standard reducibility of 1-tt-type the structure of recursive degrees is a distributive u.s.l. and its two-quantifier theory is decidable. Here the latter result is based on the analysis of similar problems given by Ambos-Spies et al. in [4].

In [32], the concept *bounded relation* is introduced. Formally, a binary relation on 2^ω is bounded iff there is a recursive predicate R where we have

$$A \leq_r B \quad \text{iff} \quad \exists e \in \omega \forall i \in \omega \; R(A, B, e, i) \tag{2}$$

for all sets A and B. Observe that bounded relations are a strict extension of the concept bounded reducibility. Firstly, every bounded reducibility is also a bounded relation because (1) can be rewritten in the form of (2). Secondly, the nonuniform reducibility $\leq_m^{\mathcal{P}/log}$ is a bounded relation but not a bounded reducibility: the relation satisfies (2) for some appropriate predicate $\mathcal{R}$, however, each set in 2^ω has uncountably many predecessors w.r.t. $\leq_m^{\mathcal{P}/log}$, while for bounded reducibilities every set has at most countably many predecessors.

In [32] it is shown that the result on lattice embeddings for standard reducibilities extends from bounded reducibilities to bounded relations, and that in fact both results carry over over to accordingly defined relations on ω^ω by minor adaptations to the proofs. In [34], structural properties of approximation preserving reducibilities between $\mathcal{NP}$ optimization problems such as the relation $\leq_L$ introduced by Papadimitriou and Yannakakis [39] are analyzed in terms of bounded relations on ω^ω.

1.3 A Glimpse Beyond Recursive Sets and Bounded Reducibilities

The axiomatic approach developed in the sequel is meant to capture bounded reducibilities and bounded relations on the recursive sets and functions. One might ask whether the axiomatic approach can be extended, firstly, in order to comprise nonrecursive sets and functions and, secondly, such that we do not only comprise a reasonable subclass of the bounded reducibilities, but also some non-bounded reducibilities such as $\leq_T$ or $\leq_{tt}$.

Regarding an extension to nonrecursive sets, note that Ladner's density result [23] extends to $(2^\omega, \leq_T^{\mathcal{P}})$, as can be shown by an adaptation of the standard *looking back* or *slow diagonalization* method. This adaptation, which is usually attributed to Shinoda, works by inspecting longer and longer initial segments of the nonrecursive oracle, instead of performing a simulating computation w.r.t. initial segments of a recursive oracle as in the standard construction used for recursive sets. Thus the adapted construction can neither be applied to reducibilities, say, of many-one type where the number of accessible places of the oracle is bounded by some constant, nor to honest reducibilities such as the reducibility $\leq_{h-T}^{\mathcal{P}}$, that is, the honest version of polynomial time bounded Turing reducibility where larger and larger initial segments of the oracle cannot be accessed due to the restriction that the length of every oracle query has to be polynomially related to the length of the current number input. In view of the failure of the looking-back technique, one might ask whether the structures induced on all sets in 2^ω by these reducibilities fail to be dense, or more specifically, whether there is a minimal set w.r.t. these reducibilities, that is, whether there is some set B such that every set which is strictly below B w.r.t. the reducibility under consideration is already computable in polynomial time. Observe that by density of the corresponding structures induced on the recursive sets, such a minimal set cannot be recursive. Homer [17, 18, 19] shows that there are no minimal degrees in the case of $\leq_m^{\mathcal{P}}$, while on the other hand the assumption $\mathcal{P} = \mathcal{NP}$ implies that there are minimal sets for $\leq_{h-T}^{\mathcal{P}}$. Improving on the latter result, Ambos-Spies shows in [3] that the assumption $\mathcal{P} = \mathcal{NP}$ in fact implies the existence of a recursively enumerable $\leq_{h-T}^{\mathcal{P}}$ - minimal set. These results show that unless we proof that $\mathcal{NP}$ differs from $\mathcal{P}$, we cannot come up with an axiomatic approach which, firstly, allows to derive the density of the nonrecursive sets and, secondly, comprises the relation $\leq_{h-T}^{\mathcal{P}}$. This indicates that if we want to extend the axiomatic approach in order to obtain results about structural properties of nonrecursive sets such as the density of the structure, then most likely we will end up with a more restrictive set of axioms which for example excludes the reducibility $\leq_{h-T}^{\mathcal{P}}$.

Next we discuss extensions of the axiomatic approach to more powerful reducibilities introduced in recursion theory such as truth-table reducibility $\leq_{tt}$, Turing reducibility $\leq_T$, or the non-effective reducibility $\leq_a$, where $A \leq_a B$ holds for sets A and B iff A is in the arithmetical hierarchy relativized to B. Observe that for these reducibilities every recursive set is reducible to all other set, and hence none of these reducibilities is bounded, because otherwise by simple diagonalization we could construct a recursive set which is not reducible to the empty set w.r.t. the reducibility in question.

In fact, the structure induced on the recursive sets by the bounded reducibility $\leq_T^{\mathcal{P}}$ and the structure induced on the recursive enumerable sets by the non-bounded reducibility $\leq_T$ bear some similarities, for example both are dense and possess minimal pairs, see Soare [47] and Ladner [23], respectively. However, there seems to be little hope for deriving these properties within an axiomatic approach comprising both structures, because the proof techniques employed in the cases of bounded and non-bounded reducibilities are quite different.

An exception from this might be given by truth-table reducibility $\leq_{tt}$, as was pointed out by Shore and Slaman [45, p.282-83]. For truth-table reducibility, while there is no *recursive* set of indices of recursive functionals as is required in the definition of the concept bounded reducibility, there is such a set which is truth-table reducible to $\emptyset''$, that is, to the double jump of the empty set. Therefore some proof techniques and the corresponding results might carry over from bounded reducibilities to the substructure of $(2^\omega, \leq_{tt})$ induced by the sets $\leq_{tt}$ - above $\emptyset''$. If we consider the larger substructure of sets above $\emptyset'$, then at least we are able to apply injury-free diagonalization constructions similar to the ones employed in connection with bounded reducibilities. Thus Shore and Slaman were led to the hypothesis that the two substructures of $(2^\omega, \leq_{tt})$ formed by the sets above $\emptyset''$ and above $\emptyset'$ might to some extent resemble some substructure of $(2^\omega, \leq_T^{\mathcal{P}})$. As evidence in favor of this hypothesis they quote Mohrherr's result, that the substructure of $(2^\omega, \leq_{tt})$ of sets above $\emptyset'$ is dense, see Mohrherr [36] or Odifreddi [38, Sect.VI.5].

We conjecture that Mohrherr's density result can be shown by adapting the standard embedding techniques used in Sect. 4, and that in principle it is possible to show density results in an an axiomatic framework which captures the substructure of $(2^\omega, \leq_{tt})$ of sets above $\emptyset'$, as well as the structures induced on the recursive sets by appropriate bounded reducibilities. However, we refrain from doing so, because we feel that the insights to be gained from such a unified treatment most probably will not make up the additional work necessary.

1.4 Notation

The notation introduced in the following is mostly standard. For notation not explained here or below in the text, see the textbooks of Balcázar et al. [6, 7], Odifreddi [38], and Soare [47]. We denote by $\leq_r$ a binary relation between sets of natural numbers which is meant as a reducibility. In particular, we will define the usual concepts arising in connection with reducibilities such as degrees or lower cones w.r.t. the relation $\leq_r$. In case it is clear from the context which relation is meant, we will suppress reference to the relation $\leq_r$, that is, for example we will use the expression lower cone instead of the more precise, but also more clumsy term lower $\leq_r$ - cone.

Natural Numbers and Binary Strings. We identify the set $\omega = \{0, 1, \ldots\}$ of natural numbers and the set $\Sigma^* = \{\lambda, 0, 1, 00, 01, \ldots\}$ of finite binary strings via the unique order isomorphism which takes the standard ordering on ω to the the length-lexicographical ordering on Σ^*, and we denote both orderings by the symbol $\leq$. We extend the identification in the canonical way to the powerset 2^ω of ω and the powerset of Σ^*. Observe that resource bounded reducibilities are usually defined in terms of Turing machine models where strings over $\{0, 1\}$ are used as inputs and for querying the oracle, and consequently these reducibilities are binary relations on the powerset of Σ^*; by the above identification, we will view such reducibilities as binary relations on 2^ω. We refer to subsets of ω and 2^ω by the terms SETS and CLASSES, respectively. We denote sets by upper case letters $A, B, \ldots$, and classes by upper case calligraphic letters $\mathcal{A}, \mathcal{B}, \ldots$.

Functions and functionals are meant to be total, if not explicitly attributed as being partial. We denote the class of functions from ω to ω by ω^ω. We identify subsets of ω with their characteristic functions and consequently 2^ω becomes a subclass of ω^ω.

Functionals. Unless we explicitly refer to some other domain, say, to ω^ω, functionals are functions from 2^ω to 2^ω. We denote functionals by upper-case Greek letters $\Gamma, \Delta, \ldots$. We identify a functional Γ with a function from $2^\omega \otimes \omega$ to $\{0,1\}$ via the equation

$$\Gamma(X,x) \;=\; (\Gamma(X))(x) \;\; .$$

Partial Recursive Functions and Functionals. We use the notation T_i in order to refer to the i-th Turing machine in the standard enumeration of all Turing machines of some given type, and we assume that it is always understood from the context which type of Turing machine is meant, that is, for example whether we consider Turing machines with or without oracle access. We refer to the partial recursive function or functional computed by Turing machine T_i by

φ_i in the case of $\{0,1\}$-valued Turing machines without oracle access,
ϕ_i in the case of ω-valued Turing machines without oracle access,
Φ_i in the case of $\{0,1\}$-valued oracle Turing machines.

We refer to the class of recursive sets and functions by REC and FREC, respectively, that is

$$\begin{array}{lcl} \text{REC} & = & \{\varphi_e : e \text{ is in } \omega \text{ and } \varphi_e \text{ is total}\} \\ \text{FREC} & = & \{\phi_e : e \text{ is in } \omega \text{ and } \phi_e \text{ is total}\} \end{array} \; .$$

We assume that there is some recursive function s which translates indices w.r.t. the enumeration $\varphi_0, \varphi_1, \ldots$ into indices w.r.t. the enumeration $\phi_0, \phi_1, \ldots$, that is, such that for all e in ω holds $\varphi_e = \phi_{s(e)}$.

Partial Characteristic Functions. By lower-case Greek letters $\alpha, \beta, \gamma, \ldots$ we denote PARTIAL CHARACTERISTIC FUNCTIONS, that is, (total) functions from some subset I of ω to $\{0,1\}$. We denote the DOMAIN of a partial characteristic function α by $dom(\alpha)$; thus for example $dom(\alpha)$ is equal to ω iff α is a set. A partial characteristic function is FINITE iff its domain is finite.

We fix some appropriate effective enumeration $\sigma_0, \sigma_1, \ldots$ of all finite partial characteristic functions where we assume

$$\sigma_0 = \lambda \;\; . \tag{3}$$

We require (3) for technical reasons in connection with the concept of delayed patching introduced in Definition 15. More precisely, this convention ensures, that if we patch the set argument of some functional according to some delayed simulation, then the set argument is not altered for all number arguments where the delayed simulation still "loops on 0". We conjecture that we could drop the convention $\sigma_0 = \lambda$ without loosing any of the subsequent results, however at the cost of additional technicalities in some of the corresponding proofs.

The partial characteristic functions are partially ordered by the relation $\sqsubseteq$, where $\alpha \sqsubseteq \beta$ holds iff the graph of α is contained in the graph of β, that is, iff the domain of α is contained in the domain of β and α agrees there with β.

For a partial characteristic function α and some set I, we denote by $\alpha \mid I$ the RESTRICTION OF $\boldsymbol{\alpha}$ TO $\boldsymbol{I}$, that is, the uniquely determined partial characteristic function $\gamma \sqsubseteq \alpha$ with domain $I \cap dom(\alpha)$. In particular, $A \mid I$ is the partial characteristic function with domain I which agrees there with the set A. We let $\alpha \sqcap \beta$ be equal to $\alpha \mid I$ where I is the maximal subset of $dom(\alpha) \cap dom(\beta)$ on which the partial characteristic functions α and β agree. For partial characteristic functions which are compatible, that is, which agree on the intersection of their domains, $\alpha \sqcup \beta$ is the partial characteristic function with domain $dom(\alpha) \cup dom(\beta)$ which agrees with α and β on their respective domains.

Definition by Cases and Patching. For partial characteristic functions α, β and for some set M, we let

$$\langle \alpha, \beta \rangle^M (x) \ := \ \begin{cases} \alpha(x) & x \in M \\ \beta(x) & x \notin M \end{cases} , \tag{4}$$

that is, $\langle \alpha, \beta \rangle^M$ is the partial characteristic function which agrees with α on $dom(\alpha) \cap M$, with β on $dom(\beta) \cap \overline{M}$, and is undefined otherwise. We denote the partial characteristic function

$$\langle \alpha, \beta \rangle \ := \ \langle \alpha, \beta \rangle^{\omega \setminus dom(\beta)}$$

as $\boldsymbol{\beta}$-PATCH of α. Observe that thus for example for a set A and a partial characteristic function β, $\langle A, \beta \rangle$ is the unique set which agrees with β for all arguments in $dom(\beta)$, and with A, otherwise.

Partial Preorderings and Partial Orderings. A binary relation $\leq$ on some arbitrary set K is

- REFLEXIVE iff $a \leq a$ holds for all a in K,
- TRANSITIVE iff $a \leq b$ and $b \leq c$ together imply $a \leq c$ for all a, b, and c in K,
- SYMMETRIC iff $a \leq b$ implies $b \leq a$ for all a and b in K,
- ANTISYMMETRIC iff for all a and b in K the facts $a \leq b$ and $b \leq a$ together imply that a and b are identical elements of K,
- a PARTIAL PREORDERING (P.P.O.) iff $\leq$ is reflexive and transitive,
- a PARTIAL ORDERING (P.O.) iff $\leq$ is an antisymmetric p.p.o.,
- an EQUIVALENCE RELATION iff $\leq$ is a symmetric p.p.o.

Cones. The LOWER $\boldsymbol{\leq_r}$ - CONE of some set A is the class

$$\leq_r (A) := \{X \subseteq \omega : X \leq_r A\} \ .$$

Likewise, we define the UPPER CONE $\geq_r(A)$ of a set A and other similar subclasses of 2^ω such as $<_r(A)$ and $>_r(A)$.

Equivalence and Interreducibility Two sets A and B are $\leq_r$ - EQUIVALENT, $A \equiv_r B$ for short, iff their upper and lower $\leq_r$ - cones coincide, respectively. Observe, that two sets are $\leq_r$ - equivalent iff, intuitively speaking, they can be substituted for each other *salva veritate* in all contexts involving only the relation $\leq_r$. Two sets A and B are $\leq_r$ - INTERREDUCIBLE, $A =_r B$ for short, iff A is reducible to B and vice versa. Observe, that for a reflexive relation $\leq_r$, every pair of equivalent sets is also interreducible, and likewise, for a transitive relation, interreducibility implies equivalence. As a consequence, interreducibility and equivalence coincide for a p.p.o. $\leq_r$.

The relation $\equiv_r$ is by definition an equivalence relation on 2^ω for arbitrary binary relations $\leq_r$ on 2^ω. We will exploit this fact in Sect. 4 in order to extend the usual concept of $\leq_r$ - degrees to relations $\leq_r$ which are not necessarily transitive.

Joins. We define inductively the join $\oplus(A_0, \ldots, A_n)$ of sets $A_1, \ldots, A_n$:

- in case $n = 0$ by $\oplus(A_0) := A_0$,
- in case $n = 1$ by

$$\oplus(A_0, A_1)(x) := \begin{cases} 0 & \text{if } x = \lambda \\ A_0(y) & \text{if } x = 0y \\ A_1(y) & \text{if } x = 1y \end{cases} ,$$

- and in case $n \geq 2$ by $\oplus(A_0, \ldots, A_n) := \oplus(A_0, \oplus(A_1, \ldots, A_n))$.

Occasionally, we write join expressions in infix notation, that is, for example we write $\oplus(A_0, A_1)$ as $A_0 \oplus A_1$. The join $\bigoplus_{i \in \omega} A_i$ of countably many sets $A_0, A_1, \ldots$ is the least subset of ω with respect to set theoretical inclusion such that for all n in ω there is some X where we have

$$\bigoplus_{i \in \omega} A_i \;=\; \oplus(A_0, \ldots, A_n, X) \;,$$

that is, the unique set which satisfies the latter condition and does not contain strings of the form 1^n with n in ω. Observe that for all n in ω, for all sets $A_0, \ldots, A_n$, X, and for all $j \leq n$ and all x in ω we have

$$x \text{ is in } A_j \quad \text{iff} \quad 1^j 0x \text{ is in } \oplus(A_0, \ldots, A_n, X) \;,$$

and consequently, for all j and x in ω, and all sets $A_0, A_1, \ldots$ we have

$$x \text{ is in } A_j \quad \text{iff} \quad 1^j 0x \text{ is in } \bigoplus_{i \in \omega} A_i \;.$$

We refer by *Left* and *Right* to the uniquely determined functions from 2^ω to 2^ω where we have for all sets A

$$\mathit{Left}(A) \oplus \mathit{Right}(A) \;=\; A \setminus \{\lambda\} \;.$$

Upper Bounds and Least Upper Bounds. For a binary relation $\leq$ on some arbitrary set K, an element u of K is

- an UPPER BOUND w.r.t. $\leq$ for (the elements in) some subset L of K iff $a \leq u$ holds for all a in L;
- a LEAST UPPER BOUND (l.u.b.) w.r.t. $\leq$ for (the elements in) some subset L of K iff, firstly, u is an upper bound for L and, secondly, $u \leq v$ holds for all upper bounds v in K of the set L.

Observe that two l.u.b.'s for some given set of elements are always $\leq$ - interreducible, and thus, if the relation $\leq$ is antisymmetric, then the l.u.b. of any given set is uniquely determined, if it exists at all.

Upper Semi-Lattices and Lattices. A p.o. $(U, \leq)$ is an upper semi-lattice (u.s.l.) iff for every pair a and b of elements in U there exists a l.u.b. of a and b in U. We denote the uniquely determined l.u.b. of two elements a and b of some u.s.l. by $a \vee b$. An u.s.l. $(U, \leq)$ is distributive iff for all elements a, b, c of U where $c \leq a \vee b$ holds there are $a_0 \leq a$ and $b_0 \leq b$ such that c is equal to $a_0 \vee b_0$.

An u.s.l. is a lattice iff for every pair of elements there exist a greatest lower bound (g.l.b.) in U, that is, there is some c in K where $c \leq a$ and $c \leq b$ holds and in addition for all x in U the facts $x \leq a$ and $x \leq b$ together imply $x \leq c$. We denote the uniquely determined g.l.b. of two elements a and b of some lattice by $a \wedge b$. A lattice L is distributive iff for all a, b, and c in L we have

$$a \wedge (b \vee c) \;=\; (a \wedge b) \vee (a \wedge c) \;.$$

Note that a lattice is distributive as a lattice iff it is distributive as an u.s.l., for a proof see Odifreddi [38, Sect. VI.1].

Closure under Finite Variation. A set A is a FINITE VARIATION of some set B, $A =^* B$ for short, iff A and B differ at most at finitely many places. A subclass $\mathcal{A}$ of 2^ω is CLOSED UNDER FINITE VARIATION (C.F.V.) iff for every set A in $\mathcal{A}$, all finite variations of A are in $\mathcal{A}$, too. A binary relation $\leq_r$ on 2^ω is

- DOWNWARDS C.F.V. iff $\leq_r (A)$ is c.f.v. for all sets A,
- UPWARDS C.F.V. iff $\geq_r(A)$ is c.f.v. for all sets A,
- C.F.V. iff $\leq_r (A)$ and $\geq_r(A)$ are both c.f.v. for all sets A.

Observe that for a p.p.o. $\leq_r$ all three variants of closure under finite variation are pairwise equivalent: for reflexive $\leq_r$, each of these conditions implies that finite variations are interreducible, while for a transitive relation $\leq_r$ in turn interreducibility implies equivalence. Therefore, if $\leq_r$ is a p.p.o., then each of the three conditions implies that finite variations are $\leq_r$ - equivalent, from which closure under finite variation then is immediate.

Pairing Function. We let $\langle .,. \rangle : \omega^2 \to \omega$ be the standard effective and effectively invertible pairing function from $\omega \otimes \omega$ onto ω, see Soare [47] for a definition and Smoryński [46] for further discussion. We denote by $\pi_0(.)$ and $\pi_1(.)$ the corresponding effective decoding functions, that is, we have for all x and y in ω

$$\pi_0(\langle x, y \rangle) = x \qquad \text{and} \qquad \pi_1(\langle x, y \rangle) = y \enspace .$$

Recursively Presentable Classes. A set E is an EFFECTIVE REPRESENTATION of some subclass $\mathcal{A}$ of 2^ω iff E is recursive and $\mathcal{A}$ is equal to $\{\varphi_e : e \in E\}$. A subclass $\mathcal{A}$ of 2^ω is RECURSIVELY PRESENTABLE iff there is some effective representation for $\mathcal{A}$. Observe that if a set E is a recursive representation for some subclass of 2^ω, then φ_e is total for all e in E. Observe further that a nonempty class is recursively presentable iff it coincides with the "rows" of some recursive set.

1.5 Some Facts from Topology.

In this section, we review some topological properties of 2^ω and ω^ω. The standard topology on 2^ω is given by the basic open sets

$$\mathcal{O}_\sigma := \{X \in 2^\omega : \sigma \sqsubseteq X\} \tag{5}$$

where σ ranges over all finite partial characteristic functions. The standard topology on ω^ω is defined likewise, where the basic open sets correspond to functions from finite subsets of ω to ω. We denote the topological spaces thus defined as CANTOR SPACE in the case of 2^ω, and as BAIRE SPACE in the case of ω^ω. Occasionally, we will refer to a topological space by the name of its underlying set, that is, for example we will refer to the topological space 2^ω. For alternate definitions and basic properties of Cantor and Baire space, we refer the reader to Sect.V.3 of Odifreddi's textbook [38]. Observe that for open sets of the form $\mathcal{O}_\sigma$ as we have used to define Cantor and Baire space, in case the domain of σ_0 is contained in the domain of σ_1 then either $\mathcal{O}_{\sigma_1}$ is a subset of $\mathcal{O}_{\sigma_0}$ or both open sets are disjoint.

An OPEN COVER for a topological space X is a collection of open subsets of X such that the union of the collection contains X. A topological space is COMPACT iff every open cover contains a finite collection of open sets which is again an open cover. Fact 1 shows that Cantor and Baire space differ w.r.t. compactness.

Fact 1. *Cantor space is compact, while Baire space is not.*

For ease of reference, we give a proof of this fact, which for example can also be found in Odifreddi's book. In the proof for compactness of 2^ω, we will use Fact 2.

Fact 2 – König's Lemma. *A finitely branching infinite tree has an infinite branch.*

Proof of König's lemma. Given a finitely branching infinite tree T, we define inductively increasingly long initial segments α_i of some infinite branch B on T. We let α_0 be the totally undefined partial characteristic function, and given α_i we let α_{i+1} be some proper extension of α_i on T such that the subtree of T below α_{i+1} is infinite. Such an extension always exists, because T has finite branching and because by our construction the subtree below α_i is infinite.

Proof of Fact 1. Baire space is not compact, because the functions $f_0, f_1, \ldots$ where the function f_i is defined at place 0 only and has the value i there form an open cover of Baire space which obviously does not contain a finite open cover.

In order to show that Cantor space is compact, we have to show that every open cover contains a finite open cover. Here it is sufficient to consider open covers which contain only basic open sets, because in the general case we can replace each open set by at most countably many basic open sets. Now, given an open cover $\{\mathcal{O}_{\alpha_0}, \mathcal{O}_{\alpha_1}, \ldots\}$, of basic open sets, we let

$$I \;:=\; \{i \in \omega : \mathcal{O}_{\alpha_i} \not\subseteq \bigcup_{j<i} \mathcal{O}_{\alpha_j}\} \;.$$

Then $\{\mathcal{O}_{\alpha_i} : i \in I\}$ is again an open cover, because an easy induction argument shows that for all n in ω we have

$$\bigcup_{\{i\in\omega : i \leq n\}} \mathcal{O}_{\alpha_i} \;=\; \bigcup_{\{i\in\omega : i\leq n \text{ and } i\in I\}} \mathcal{O}_{\alpha_i} \;.$$

Thus we are done, once we have shown that I is finite. Assuming that I is infinite, let T be the closure under initial segments of $\{\alpha_i : i \in I\}$. Then T is obviously a subtree of 2^ω, and is infinite, because by definition of I, α_i differs from α_j whenever i and j are distinct elements of I. By König's lemma, there is some infinite path $B \subseteq \omega$ on T, and by definition of open cover, there is some k in ω where $\alpha_k \sqsubseteq B$. But by definition of I, $\alpha_k \sqsubseteq \alpha_l$ implies that l is not in I for all $l > k$, that is, by definition of T there are at most finitely many extensions of α_k contained in T, which contradicts the fact that B extends α_k and is on T. □

A functional Γ is CONTINUOUS iff for all sets A in 2^ω and all x in ω, there is some finite set $I(A,x) \subseteq \omega$, such that for all sets B in 2^ω we have

$$A \mid I(A,x) = B \mid I(A,x) \;\Rightarrow\; \Gamma(A,x) = \Gamma(B,x) \;. \tag{6}$$

The point of this definition is that for a continuous functional the value $\Gamma(A,x)$ is determined by x and the restriction of the set argument A to some finite subset of ω. Fact 3 shows that there is such a finite set which works for all oracles. The proof of Fact 3 relies on compactness of Cantor space, and indeed a corresponding statement for Baire space is false, as is witnessed by the functional Γ on ω^ω where $\Gamma(a, x)$ is equal to $a(a(x))$. Note in connection with Fact 3 that among all the sets $I(x)$ which satisfy (7) there is a least one w.r.t. set theoretical inclusion, as we will show in Sect. 3.1.

Fact 3. *A functional Γ on 2^ω is continuous iff for all x in ω, there is some finite set $I(x) \subseteq \omega$ such that for all sets X and Y in 2^ω we have*

$$X \mid I(x) = Y \mid I(x) \ \Rightarrow\ \Gamma(X, x) = \Gamma(Y, x)\ . \tag{7}$$

Proof. The backwards direction of Fact 3 is immediate from the definition of continuity. In order to show the remaining implication, we let Γ be continuous and we let x be in ω. Then for every set A there is some finite set $I(A, x)$ such that (6) is satisfied. We let $\sigma(A)$ be the restriction of A to $I(A, x)$ and hence $\mathcal{O}_{\sigma(A)}$ is open and contains A. Thus all the sets $\mathcal{O}_{\sigma(A)}$ with A in 2^ω together form an open cover for Cantor space, which, by compactness, contains a finite subcover $\{\mathcal{O}_{\sigma(A_0)}, \ldots, \mathcal{O}_{\sigma(A_n)}\}$ for some n in ω. By joining all the sets $I(A_j, x)$ where $j \leq n$, we obtain a finite set $I(x)$. The set $I(x)$ satisfies (7), because every set A in 2^ω is contained in some open set $\mathcal{O}_{\sigma(A_k)}$ in the finite subcover, and thus the value of $\Gamma(A, x)$ is determined by the restriction of A to $I(A_k, x) \subseteq I(x)$. □

The proof of Fact 3 is a straightforward adaptation of a result due to Trakhtenbrot and Nerode (see Odifreddi [38, Proposition III.3.2.]) according to which the use of a total oracle Turing machine at some place x is contained in some finite set which depends only on x, but not on the oracle. An equivalent formulation of their result is that recursive functionals coincide with effective reductions of truth-table type. Likewise, Fact 3 alternatively can be expressed by saying that a functional is continuous iff its outcome at some place x is given by a truth-table condition on its set argument, where a TRUTH-TABLE CONDITION (TT-CONDITION) is a subclass of 2^ω which is the union of finitely many basic open sets. Before we give a precise formulation of this observation in Fact 5, we state an alternate characterization of tt-conditions.

Fact 4. *A subclass of 2^ω is a tt-condition iff it is closed and open.*

Proof. A tt-condition is obviously open, and it is not hard to see that the complement of a tt-condition in 2^ω is again a tt-condition, that is, a tt-condition is also closed due to being the complement of an open set. In order to show the reverse direction, assume that some subclass $\mathcal{P}$ of 2^ω is closed and open, that is, $\mathcal{P}$ and its complement are open. But then there are two countable classes of basic open sets, the union of which is $\mathcal{P}$ and the complement of $\mathcal{P}$, respectively. The two classes together form an open cover, which by compactness contains a finite open cover. This finishes the proof, because each element of the finite open cover is either a subset of or disjoint from $\mathcal{P}$, that is, $\mathcal{P}$ must be the union of some of the elements in the finite open cover. □

We conclude this section by stating some equivalent characterizations of continuous functionals on 2^ω. In particular, we show that a functional is continuous in the sense of the definition given above iff it is a continuous mapping from Cantor space into itself in the usual topological sense.

Fact 5. *For a functional Γ on 2^ω the following conditions are equivalent.*

1. *Γ is continuous.*
2. *The class $\{X \in 2^\omega : \Gamma(X, x) = 0\}$ is a tt-condition for all x in ω.*
3. *The inverse image $\Gamma^{-1}(\mathcal{O})$ of every basic open set $\mathcal{O}$ is open.*
4. *The inverse image $\Gamma^{-1}(\mathcal{O})$ of every open set $\mathcal{O}$ is open.*

Proof. The first condition implies the second by Fact 3, and the second condition implies the first, because membership of some set in a tt-condition obviously depends only on a finite part of the set. Further, given the second condition, it is easy to see that the inverse image of a basic open set under Γ is a finite intersection of tt-conditions and complements of tt-conditions, that is, according to Fact 4, a finite intersection of open sets, which is then open by definition. The third condition implies the fourth, because every open set is the union of at most countably many basic open set, and consequently the inverse image of an open set is the union of at most countably many inverse images of basic open sets, where the latter are all open by assumption. Finally, if the inverse image under Γ of every open set is open, then for all x in ω the set $\{X \in 2^\omega : \Gamma(X, x) = 0\}$ and its complement $\{X \in 2^\omega : \Gamma(X, x) = 1\}$ are both open, and consequently the former is a tt-condition by Fact 4, which then implies the second condition. □

2 Standard Reducibilities

2.1 Faithful Relations

Our generalized approach to resource bounded reducibilities is based on the concept standard reducibility introduced in Definition 20. In order to be able to mimic the usual proof techniques employed in connection with resource bounded reducibilities, we require that standard reducibilities are bounded reducibilities which, firstly, are sort of "faithful" to the information contained in their set arguments and, secondly, intuitively speaking, can use a non-trivial amount of computational power in reducing one set to another. In the remainder of this section, we introduce notation related to our concept of faithfulness, and in the following sections we develop concepts related to computational power.

In the case of a transitive relation $\leq_r$, the concept faithfulness is equivalent to the natural conditions that $\leq_r$ is reflexive, that $\emptyset$ and ω are $\leq_r$ - reducible to all sets, and that the join of two sets is a l.u.b. for them (see Proposition 8). However, as we want to comprise reducibilities which are not necessarily transitive, we use a more involved definition of faithfulness stated in Definition 7. Note in this connection that indeed many results shown in the following can be obtained for both, transitive and non-transitive standard reducibilities. Observe further that in the literature there are non-transitive relations such as $\leq_{k-tt}$ for any fixed $k \geq 2$ or the relation $\leq^{\mathcal{NP}}$, see Example 21 and [6], respectively, and that these relations are usually referred to as reducibilities, which indicates that transitivity does not seem to be an essential ingredient of the usual understanding of the concept reducibility.

Definition 6. Let $\leq_r$ be a binary relation on 2^ω, let A and B be sets, and let $\mathcal{A}$ be a subclass of 2^ω.

1. The relation $\leq_r$ is LOCALLY TRANSITIVE at A and B iff A is not $\leq_r$ - reducible to B or we have for all sets X and Y

$$X \leq_r A \quad \text{implies} \quad X \leq_r B \ , \tag{8}$$

$$B \leq_r Y \quad \text{implies} \quad A \leq_r Y \ . \tag{9}$$

2. A set U is a LOCALLY TRANSITIVE UPPER BOUND for (the elements of) $\mathcal{A}$ iff U is an upper bound for $\mathcal{A}$ and for all sets A in $\mathcal{A}$ the relation $\leq_r$ is locally transitive at A and U. Likewise, U is a LOCALLY TRANSITIVE L.U.B. for $\mathcal{A}$ iff U is a l.u.b. for $\mathcal{A}$ and for all sets A in $\mathcal{A}$ the relation $\leq_r$ is locally transitive at A and U.

Observe that a transitive relation is locally transitive at every pair of sets, and hence for a transitive relation the standard and the locally transitive variants of upper bounds and l.u.b.'s coincide, respectively.

Definition 7. A binary relation $\leq_r$ on 2^ω is FAITHFUL iff for all sets A, B and X, firstly, the set $A \oplus B$ is a locally transitive l.u.b. for A and B in $(2^\omega, \leq_r)$ and, secondly,

$$X \leq_r A \oplus B \text{ implies } X \leq_r B \oplus A \ , \tag{10}$$

$$X \leq_r A \oplus \emptyset \text{ implies } X \leq_r A \ , \tag{11}$$

$$X \leq_r A \oplus \omega \text{ implies } X \leq_r A \ . \tag{12}$$

Observe that most resource bounded reducibilities which can be found in the literature are faithful, and that in particular there are non-transitive faithful reducibilities such as $\leq_{k-tt}$ for every fixed $k \geq 2$, see Example 23. Faithful relations are reflexive and for them $\emptyset$ and ω are reducible to all other sets, as we show in Proposition 8.

Proposition 8. *Let $\leq_r$ be a faithful relation on 2^ω.*

- *For all sets A, the sets $\emptyset$, ω, and A are $\leq_r$ - reducible to A.*
- *For all sets A and B, the set $A \oplus B$ is a l.u.b. for A and B in $(2^\omega, \leq_r)$.*

Conversely, if the relation $\leq_r$ is transitive and satisfies both of these conditions, then $\leq_r$ is faithful.

Proof. For a faithful relation $\leq_r$ the second condition is immediate, so it remains to show the first. Now, given some set A, then A and $\emptyset$ are both $\leq_r$ - reducible to $A \oplus \emptyset$, and hence they are also $\leq_r$ - reducible to A by (11). Likewise, we obtain $\omega \leq_r A$ from (12). Conversely, assume that the relation $\leq_r$ is transitive and satisfies the two conditions from the proposition. By transitivity and the second condition it is immediate that the join operator provides locally transitive l.u.b.'s for every pair of sets. Further, we obtain (10), (11), and (12) by transitivity of $\leq_r$ and because by assumption on the join operator we have

$$A \oplus B \leq_r B \oplus A, \qquad A \oplus \emptyset \leq_r A, \qquad \text{and} \qquad A \oplus \omega \leq_r A \ . \qquad \square$$

For further use, we state some easy properties of faithful relations. Recall from the introduction that two sets A and B are $\leq_r$ - equivalent, written $A \equiv_r B$, iff their lower and upper cones, respectively, are identical.

Proposition 9. *Let $\leq_r$ be a faithful relation on 2^ω and let A and B be sets.*

- $\geq_r(A \oplus B) \ = \ \geq_r(A) \cap \geq_r(B)$
- $A \oplus B \equiv_r B \oplus A$
- $A \equiv_r A \oplus \emptyset \equiv_r A \oplus \omega$

Proof. For a faithful relation $\leq_r$, the join of two sets is a locally transitive l.u.b. for the sets joined. Thus, concerning the first assertion, the inclusion from left to right is immediate from the definition of locally transitive upper bound, and the reverse inclusion follows by definition of l.u.b. Using the first assertion, we then infer that the two sets denoted by the join expressions on both sides of the second assertion have identical upper cones, while for the lower cones this is immediate from (10). Concerning the third assertion, it is sufficient to show that the lower, respectively upper cones, of A, $A \oplus \emptyset$ and $A \oplus \omega$ are identical; we show this for the first two sets and leave the almost identical considerations for the third set to the interested reader. By the first assertion, the upper cone of $A \oplus \emptyset$ is equal to the intersection of the upper cones of A and $\emptyset$, and hence is equal to the upper cone of A. On the other hand, the lower cone of $A \oplus \emptyset$ is contained in $\leq_r (A)$ by (11), and the reverse containment holds because $A \oplus \emptyset$ is a locally transitive upper bound for A. □

2.2 Bounded and Generalized Reducibilities

Recall from the introduction that functionals are functions from 2^ω to 2^ω, which we view equivalently as functions from $2^\omega \otimes \omega$ to the set $\{0, 1\}$ via the equation

$$\Gamma(A, x) \ := \ (\Gamma(A))(x) \ .$$

Recall further that a functional Γ is continuous iff for all x in ω the value $\Gamma(A, x)$ is determined by some finite part of A.

Definition 10. Let $\leq_r$ be a binary relation on 2^ω.

- We call a functional Γ a REDUCTION iff it is continuous.
- A reduction Γ is a reduction w.r.t. $\leq_r$ or an $\boldsymbol{\leq_r}$ - REDUCTION, for short, iff for all sets B we have
$$\Gamma(B) \leq_r B \ .$$
We say, a fact $A \leq_r B$ is WITNESSED by the functional Γ, or $A \leq_r B$ VIA $\boldsymbol{\Gamma}$, for short, iff Γ is a $\leq_r$ - reduction where A is equal to $\Gamma(B)$.
- A set $\mathcal{R}$ of functionals is a REDUCTION COVER for $\leq_r$ iff $\mathcal{R}$ is a countable set of $\leq_r$ - reductions such that when $A \leq_r B$ holds for sets A and B, then this fact is witnessed by some functional Γ in $\mathcal{R}$.
- A set E is a RECURSIVE REPRESENTATION for some class $\mathcal{R}$ of functionals iff, firstly, E is recursive and, secondly, $\mathcal{R}$ is equal to $\{\Phi_e : e \in E\}$. We call a reduction cover EFFECTIVE iff it is has some recursive representation.
- The relation $\leq_r$ is a GENERALIZED REDUCIBILITY (on 2^ω) iff there is some reduction cover for $\leq_r$. The relation $\leq_r$ is a BOUNDED REDUCIBILITY (on 2^ω) iff there is some effective reduction cover for $\leq_r$.

The term bounded reducibility is due to Book et al. [11], but the concept has been used before by several authors. Concerning the definition of the notion effective reduction cover, recall that Φ_e is the partial functional computed by the e-th oracle Turing machine and observe that the definition of reduction cover implies that each partial recursive functional Φ_e in an effective reduction cover is in fact total. Given some set E which is a recursive representation for some class $\mathcal{R}$ of functionals, for the sake of convenience we occasionally identify $\mathcal{R}$ with the sequence $\Delta_0, \Delta_1, \ldots$ such that Δ_i is equal to $\Phi_{e(i)}$ where $e(i)$ is the i-th element in E.

An example of a generalized reducibility which is not a bounded reducibility is given by truth-table reducibility $\leq_{tt}$: the set of all recursive functionals provides a reduction cover for $\leq_{tt}$ according to the effectivized version of Fact 3, but there cannot be an effective reduction cover for $\leq_{tt}$, because otherwise, the lower cone of any given recursive set B would be recursively presentable, and hence by a simple diagonalization argument we would obtain a recursive set A which is not $\leq_{tt}$ - reducible to B, thus contradicting the fact that every recursive sets is $\leq_{tt}$ - reducible to all other sets.

Example 11 shows that for a bounded reducibility in general, even if it satisfies additional properties such as being faithful or being c.f.v., we cannot show the density of the structure induced on the recursive sets. As we have mentioned in Sect. 1.2, this result holds for several bounded reducibilities. A look at the corresponding proofs gives some indication why the assumptions listed above are not sufficient for deriving density: the employed proof techniques hinge on the fact that the reducibilities in question are defined via oracle Turing machines which are able to perform resource bounded, oracle-independent subcomputations where the outcome of such subcomputations is for example used to overwrite the oracle. In order to be able to mimic the corresponding proof techniques within the abstract framework, we will develop an abstract account of this type of oracle patching in the following sections.

Example 11. *Given some set A and binary strings v and w, we let*

$$vA := \{vx : x \in A\} \quad \text{and} \quad A^{<w>} := \{x : wx \in A\}$$

where vx denotes the concatenation of v and x. Further, we let a set A be $\leq_\beta$ - reducible to some set B iff for some n in ω there is a mapping

$$r : \{0,1\}^n \to \{0,1\}^* \tag{13}$$

such that for all strings w of length n

$$A^{<w>} \text{ is a finite variation of } \emptyset,\ \omega,\ B^{<r(w)>},\ \text{or } \overline{B^{<r(w)>}}\ . \tag{14}$$

The reducibility $\leq_\beta$ is bounded, as is witnessed by the effective reduction cover $\mathcal{R}$ where each reduction in $\mathcal{R}$ corresponds in the obvious way to *a number n and some function r as in (13),*

a function which assigns to each w of length n one of the four possibilities in (14),

and some finite partial characteristic function σ which takes care of the fact that (14) is formulated in terms of equality up to finite variation and not in terms of strict equality.

Observe that each such $\leq_\beta$ - reduction is a rather restricted one-question truth-table reduction, where for each string w of length n and for almost all number arguments wx which extend w the same one-place evaluation function is applied to the answer received on querying the oracle at place $r(w)x$.

Obviously, the relation $\leq_\beta$ is c.f.v. and the sets $\emptyset$ and ω are $\leq_\beta$ - reducible to all other sets. Furthermore, the relation $\leq_\beta$ is transitive and the join of two sets is a l.u.b. for them, and consequently, by Proposition 8, the relation $\leq_\beta$ is faithful. We omit the routine proofs of this assertions, however note in connection with transitivity that if facts $A \leq_\beta B$ and $B \leq_\beta C$ are witnessed by $\leq_\beta$ - reductions in $\mathcal{R}$ which depend on the first n_0 and n_1 places of the number argument, respectively, then A is $\leq_\beta$ - reducible to C via some reduction in $\mathcal{R}$ which depends on the first $n_0 + n_1$ places. Next, we show that the set

$$T := \{1^k : k \in \omega\}$$

is minimal in $(2^\omega, \leq_\beta)$, that is, T is not a least set and every set D which is $\leq_\beta$ - reducible to T either is a least set or we have $T \leq_\beta D$. Thus, in particular, $(REC, \leq_\beta)$ is not dense.

For a proof of minimality of the set T, observe first that by definition of $\leq_\beta$, given some set A which is $\leq_\beta$ - reducible to $\emptyset$ via some reduction in $\mathcal{R}$ which depends on the first n places of its number argument, then A contains either almost all or only finitely many strings of the form $1^n y$. As a consequence, T is not $\leq_\beta$ - reducible to $\emptyset$.

Next, assuming $A \leq_\beta T$ for some set A, this fact is witnessed by some reduction Γ in $\mathcal{R}$ such that the corresponding function r as in (13) depends on the first n places of the number input for some fixed n in ω. Now, if there is some string w of length n where $r(w)$ is in T and the evaluation function is non-constant, then by choice of T, $A^{<w>}$ is some finite variation of either T or $\overline{T}$, depending on the evaluation function being identity or negation. So in this case, we obtain $T \leq_\beta A$. On the other hand, if for all strings w of length n the evaluation function is constant or $r(w)$ is not in T, then Γ witnesses that A is $\leq_\beta$ - reducible to $\emptyset$. As a consequence, for every set A which is $\leq_\beta$ - reducible to T, we either have $T \leq_\beta A$ or $A \leq_\beta \emptyset$. Intuitively speaking, a reduction in $\mathcal{R}$ either extracts all information contained in T or it does not extract any information at all.

2.3 Delayed Simulations

In the current and the following section, we develop an abstract account of the ability of resource bounded oracle Turing machines to overwrite or "patch" their oracle according to the results of resource bounded subcomputations. This account is based on the concept *delayed patching*, where we mean by patching to replace the set argument B of some reduction with some patched version $\langle B, \sigma \rangle$ before evaluating the reduction, and the attribute delayed corresponds to the fact that the patching is done w.r.t. to arbitrary effective enumerations $\alpha_0, \alpha_1, \ldots$ of finite partial characteristic functions where however in general the i-th partial characteristic function will not be used while computing the value $\Gamma(B, i)$, but delayed, that is, for number arguments larger than i. Note again that this kind of delayed access to effectively given information is common in connection with resource bounded oracle Turing machines: given an effective enumeration $\alpha_0, \alpha_1, \ldots$ as above, a resource bounded oracle Turing machine can eventually compute and access α_i for arbitrarily large values of i, however, intuitively speaking, the Turing machine has to wait until its number input and hence its resource bounds become large enough. The ability to perform such delayed computations is modeled by the concepts delayed simulation and simulation class introduced in Definition 12.

Definition 12. Let h, s, and l be (not necessarily recursive) functions from ω to ω.

- The function s is MANY-ONE REDUCIBLE to h via l iff we have for all x in ω

$$s(x) = h(l(x)) \ .$$

- The function s is a DELAYED SIMULATION of the function h iff s is many-one reducible to h via some nondecreasing function l with range ω.
- A subclass $\mathcal{F}$ of ω^ω is a FUNCTIONAL SIMULATION CLASS iff there is a recursive function sim from ω to ω where for all e in ω
  - $\phi_{sim(e)}$ is a function in $\mathcal{F}$,
  - in case ϕ_e is total and $\phi_e(0)$ is equal to 0, then $\phi_{sim(e)}$ is a delayed simulation of ϕ_e.
- A class $\mathcal{S} \subseteq 2^\omega$ is a SIMULATION CLASS iff there is some recursive function sim such that $\varphi_{sim(e)}$ is a set in $\mathcal{S}$ for all e in ω, and in addition $\varphi_{sim(e)}$ is a delayed simulation of φ_e whenever φ_e is a set which does not contain 0. (That is, $\mathcal{S}$ is a simulation class if it satisfies the definition of functional simulation class with ω^ω and $\phi_0, \phi_1, \ldots$ replaced by 2^ω and $\varphi_0, \varphi_1, \ldots$.)

The concept delayed simulation was introduced by Mehlhorn in order to formulate one of his axioms, see [31], Axiom 6; we discuss the relations of his axiomatic approach to our concept standard reducibility in Sects. 3.7. Further, we will argue in Sect. 3.7 that the concepts introduced in Definition 12 are robust under various changes to their definition.

Observe that in the definition of the concepts functional simulation class and simulation class it is reasonable to require that the function *sim* yields delayed simulations only in case the functions ϕ_e and φ_e, respectively, are equal to 0 at place 0. Remark 13 shows for the case of simulation classes that there cannot be a recursive function *sim* where $\varphi_{sim(e)}$ is total for all e in ω, and is a delayed simulation of φ_e for *all* sets φ_e. In this connection, see also Proposition 48 and Remark 52.

Remark 13. *Recall from recursion theory that the sets A_0 and A_1 defined by*

$$A_i := \{e \in \omega : \varphi_e(e) = i\} \qquad\qquad i = 0, 1 \ ,$$

are recursively inseparable, that is, there is no recursive set D such that A_0 is contained in D and A_1 is contained in $\overline{D}$. Indeed this is easy to see, because any such set D must differ from every recursive set φ_e at place e. Now, by the smn-theorem, there is a recursive function g such that for all e,x, and y in ω we have

$$\varphi_{g(e,x)}(y) = \varphi_e(x) \ .$$

But if we assume that there is a recursive function sim such that $\varphi_{sim(e)}$ is a set for all e in ω, and is a delayed simulation of φ_e for all *sets φ_e, then we obtain a contradiction, because the set*

$$D := \{e \in \omega : \varphi_{sim(g(e,e))}(0) = 0\}$$

is recursive and separates A_0 from A_1 by definition of g.

Example 14 shows that the class of functions computable in polynomial time is a functional simulation class.

Example 14. *For every e in ω, there is some Turing machine $T_{sim(e)}$ which operates as follows:*

> *On any given input of length n, $T_{sim(e)}$ simulates the Turing machine T_e in order to compute the values $\phi_e(0), \phi_e(1), \ldots$. The simulating machine $T_{sim(e)}$ runs for a total of n steps and then outputs the last successfully computed value or, if even $\phi_e(0)$ could not be computed, the value 0.*

By definition of $sim(e)$, it should be clear that

> *we can choose the function sim to be recursive,*
> *for every e in ω, the Turing machine $T_{sim(e)}$ is total and runs in polynomial time, and*

$\phi_{sim(e)}$ is a delayed simulation of ϕ_e, whenever ϕ_e is total and has the value 0 at place 0.

In connection with the last property, we can safely assume that the simulation of T_e for a single number argument requires at least one step, and that consequently the simulating computation cannot skip function values of φ_e in the sense that for example the function ϕ_e has value 1 at some place x and has value 0 everywhere else, but $\phi_{sim(e)}$ is the constant function with value 0. Note that the last remark becomes oblivious by Proposition 50, which states that the concept functional simulation class is robust under changing its definition by replacing delayed simulations with weak delayed simulations, a variant of delayed simulations for which it is allowed to skip values of the simulated function.

2.4 Delayed Patching

Recall from the introduction that $\sigma_0, \sigma_1, \ldots$ is an appropriate effective enumeration of all partial characteristic functions where by convention σ_0 is equal to the empty string. Recall further that for a set A and a partial characteristic function α, the set $\langle A, \alpha \rangle$ is denoted as α-patch of A and is equal to α on $dom(\alpha)$, and is equal to A, otherwise.

Definition 15. 1. For a functional Γ and a function g from ω to ω, the functional $\Gamma \otimes g$ is denoted as ***g*-PATCH OF *Γ*** and is defined by

$$(\Gamma \otimes g)(A, x) := \Gamma(\langle A, \sigma_{g(x)} \rangle, x) \ .$$

2. For a class $\mathcal{R}$ of functionals and a subclass $\mathcal{F}$ of ω^ω the **CLASS OF $\mathcal{F}$-PATCHES OF $\mathcal{R}$** is

$$\mathcal{R} \otimes \mathcal{F} := \{\Gamma \otimes g : \Gamma \in \mathcal{R} \text{ and } g \in \mathcal{F}\} \ .$$

3. A class $\mathcal{R}$ of functionals is closed under DELAYED PATCHING iff there is a functional simulation class $\mathcal{F}$ where $\mathcal{R} \otimes \mathcal{F}$ is contained in $\mathcal{R}$.

Example 16 shows that the standard enumeration of polynomial time bounded Turing machines yields a reduction cover for the reducibility $\leq_T^{\mathcal{P}}$, which is closed under delayed patching.

Example 16. *We fix an effective enumeration of polynomial time bounded oracle Turing machines where the $\langle e, j \rangle$-th machine in the enumeration corresponds to the e-th Turing machine in the standard enumeration of Turing machines restricted to the time bound as given by the j-th polynomial in some appropriate enumeration of all polynomials. Then the class $\mathcal{R}_T^{\mathcal{P}}$ of all functionals computed by some oracle Turing machine in this enumeration is an effective reduction cover for $\leq_T^{\mathcal{P}}$. Now, firstly, adding polynomial time bounded subcomputations to an polynomial time bounded oracle Turing machine will not put us outside of* *the class* $\mathcal{R}_T^{\mathcal{P}}$ *and, secondly, the class* $\mathcal{FP}$ *of polynomial time computable functions is a functional simulation class according to Example 14. However, this does not imply directly that the standard reduction cover* $\mathcal{R}_T^{\mathcal{P}}$ *is closed under delayed patching, because given some function* g *which is computable in polynomial time, the corresponding function values might be so large that for more reasonable effective enumerations of the finite partial characteristic functions it it will be impossible to access the values of the partial characteristic function* $\sigma_{g(x)}$ *in time polynomially bounded in the length of* x. *Thus if we patch a functional* Γ *in* $\mathcal{R}_T^{\mathcal{P}}$ *by some function* g *computable in polynomial time, then in general the new functional* $\Gamma \otimes g$ *will not be computable in polynomial time unless we choose an effective enumeration* $\sigma_0, \sigma_1, \ldots$ *of the finite partial characteristic functions where the coding is highly redundant.*

Our salvation here is Proposition 51 below, which states that for a functional simulation class $\mathcal{F}$ *and for some arbitrary nondecreasing and unbounded recursive function* b *from* ω *to* ω, *the class*

$$\{g \in \mathcal{F} : g(i) \leq b(i) \text{ for all } i \in \omega\}$$

is again a functional simulation class. So we let b *be a nondecreasing and unbounded recursive function which grows so slowly that for all* x *and for all* $j \leq b(x)$, *we can in time* $|x|$ *answer all relevant questions about the domain and the values of* σ_j. *Then by Example 14 and the previous discussion the class*

$$\mathcal{FP}_{\leq b} := \{g \in \mathcal{FP} : g(i) \leq b(i) \text{ for all } i \in \omega\}$$

is a functional simulation class. Now, by our choice of b *the reduction cover* $\mathcal{R}_T^{\mathcal{P}}$ *is closed under patching with functions in* $\mathcal{FP}_{\leq b}$. *More precisely, for a functional* Γ *computed by some polynomial time bounded oracle Turing machine* T, *and for some function* g *in* $\mathcal{FP}_{\leq b}$, *there is some oracle Turing machine* T' *which computes* $\Gamma \otimes g$:

- T' *works essentially like* T, *but every query state of* T *is replaced by a sequence of new states corresponding to a subcomputation which takes care of the* g*-patching. In the subcomputation, firstly, the value of* $g(x)$ *for the current number input* x *is computed and, secondly, it is checked whether the value* y *written currently on the oracle tape is in the domain of* $\sigma_{g(x)}$. *This check can be done in polynomial time by choice of* g *in* $\mathcal{FP}_c$ *and because we have chosen* c *sufficiently small. After this subcomputation, the execution of* T' *resumes as follows: in case* y *is not in the domain of* $\sigma_{g(x)}$, *then the next state is determined in exactly the same way as for* T *by the value of the oracle at place* y; *otherwise, the value* $\sigma_{g(x)}(y)$ *is used in place of the actual oracle value in order to determine the next state.*

In connection with the description of the oracle Turing machine T', firstly, recall that once an oracle Turing machine enters a query state, the next state is determined by the value of the oracle for the number currently written on the oracle tape. Secondly, observe that for Turing machine models with write-only access to the oracle tape, we have to keep a copy of the current oracle question on some work tape in order to be able to check whether the current query is in the domain of the patch we want to apply.

In the following, we will consider bounded reducibilities which have *some* reduction cover which is closed under delayed patching. Example 17 shows that closure under delayed patching is better not defined with respect to the relation $\leq_r$ itself, that is, for example by requiring that the class of all $\leq_r$ - reductions is closed under delayed patching. The reason for this is that even in the case of the well-behaved reducibility $\leq_T^{\mathcal{P}}$, there are "strange" $\leq_T^{\mathcal{P}}$-reductions which cannot be patched by the functions in any simulation class without leaving the class of $\leq_T^{\mathcal{P}}$ - reductions.

Example 17. *We let $P_0, P_1, \dots$ be an effective enumeration of the class $\mathcal{P}$ of sets computable in polynomial time and we let the functional Γ be defined by*

$$\Gamma(X,x) := \begin{cases} P_{\min(X \cap \{0,\dots,x\})}(x) & \text{in case } X \cap \{0,\dots,x\} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} .$$

Observe that the functional Γ is continuous. In fact, Γ is an $\leq_T^{\mathcal{P}}$ - reduction, because $\Gamma(X)$ is equal to $\emptyset$ in case X is equal to $\emptyset$, and is equal to some finite variation of $P_{\min X}(X)$, otherwise, that is, $\Gamma(X)$ is $\leq_T^{\mathcal{P}}$ - reducible to X for all sets X.

We choose some recursive function f such that $\sigma_{f(i)}$ is the finite partial characteristic function with domain $\{0,\dots,i\}$ which yields 1 at place i, and yields 0 at all other places of its domain. We show that for every delayed simulation s of f, the functional $\Gamma \otimes s$ cannot be a $\leq_T^{\mathcal{P}}$ - reduction, because the set

$$A \;:=\; (\Gamma \otimes s)(\emptyset)$$

is not $\leq_T^{\mathcal{P}}$ - reducible to $\emptyset$, or equivalently, is not in $\mathcal{P}$. If we assume that set A is in $\mathcal{P}$, then so is $\overline{A}$, that is, $\overline{A}$ is equal to P_j for some j in ω. But the delayed simulation s of f has to attain the value $f(j)$ at some place $z \geq j$, from which we obtain the contradiction

$$A(z) = \Gamma \otimes s(\emptyset, z) = \Gamma(\langle \emptyset, \sigma_{s(z)} \rangle, z) = \Gamma(\langle \emptyset, \sigma_{f(j)} \rangle, z) = P_j(z) = \overline{A}(z) \;,$$

where the equations hold, by definition of A, $\Gamma \otimes s$, z, Γ, and j, respectively.

2.5 Definition by Oracle-Dependent Cases

Recall from the introductory section on topology that a tt-condition $\mathcal{T}$ is a subclass of 2^ω where there is some finite set I such that for all sets A membership in $\mathcal{T}$ depends only on the restriction of A to I.

Definition 18.
- Let $\mathcal{T}$ be a tt-condition, and let Γ_0 and Γ_1 be functionals. The $\boldsymbol{\mathcal{T}}$ - MIX of Γ_0 and Γ_1 is the functional $\langle \Gamma_0, \Gamma_1 \rangle^{\mathcal{T}}$ defined by

$$\langle \Gamma_0, \Gamma_1 \rangle^{\mathcal{T}} (A) := \begin{cases} \Gamma_0(A) & \text{if } A \in \mathcal{T} \\ \Gamma_1(A) & \text{if } A \notin \mathcal{T} \end{cases} \tag{15}$$

- For a tt-condition $\mathcal{T}$, we denote the transition from two functionals to their $\mathcal{T}$-mix as DEFINITION BY ORACLE-DEPENDENT CASES w.r.t. $\mathcal{T}$.
- A class $\mathcal{R}$ of reductions is CLOSED UNDER DEFINITION BY ORACLE-DEPENDENT CASES iff for every tt-condition $\mathcal{T}$ the $\mathcal{T}$-mix of two reductions in $\mathcal{R}$ is again in $\mathcal{R}$.

Remark 19. *Given a binary relation $\leq_r$ on 2^ω, the class of $\leq_r$ - reductions is closed under definition by oracle-dependent cases, because for each tt-condition $\mathcal{T}$, the $\mathcal{T}$-mix $\langle \Gamma_0, \Gamma_1 \rangle^{\mathcal{T}}$ of two $\leq_r$ - reductions is obviously continuous and maps an arbitrary set B to $\Gamma_0(B)$ or to $\Gamma_1(B)$, that is, to a set which by assumption is reducible to B. It is not hard to see that the closure under definition by oracle-dependent cases of some effective reduction cover for some bounded reducibility $\leq_r$ is again an effective reduction cover for $\leq_r$, and consequently, a bounded reducibility has always an effective reduction cover which is closed under definition by oracle-dependent cases.*

In Definition 20, we will introduce standard reducibilities, a subclass of bounded reducibilities which in particular are required to have an effective reduction cover which is both, closed under delayed patching and closed under definition by oracle-dependent cases. By Remark 19, the latter condition alone is vacuously satisfied for every bounded reducibility. However, there are bounded reducibilities for which there is no effective reduction cover which satisfies *both* closure conditions. For example if there were such a reduction cover for the reducibility $\leq_\beta$ in Example 11, then, contrary to the existence of minimal sets for the relation $\leq_\beta$, we would obtain by Theorem 68 that the structure $(\text{REC}, \leq_\beta)$ is dense. In fact, in case some bounded reducibility has a reduction cover which satisfies both closure conditions, we are able to mimic switching between reductions according to oracle-independent subcomputations, as is done with the *gap language technique*: if we first put two reductions together via a definition by cases w.r.t. some tt-condition, then afterwards, by finitely patching the set argument of the new compound reduction at all the places the selecting tt-condition depends on, we can alternate between the two initial reductions; for details of this construction, see the proof of Proposition 40.

2.6 Standard Reducibilities

Our axiomatic approach to bounded reducibilities is based on the concept standard reducibility as introduced in Definition 20.

Definition 20. A binary relation $\leq_r$ on 2^ω is a STANDARD REDUCIBILITY (on 2^ω) iff

- the relation $\leq_r$ is faithful,
- there is some effective reduction cover for $\leq_r$ which is closed under delayed patching and under definition by oracle-dependent cases.

We will show in Sect. 3.3 that standard reducibilities are always c.f.v. An example for a standard reducibility is given by the relation $\leq_T^{\mathcal{P}}$. It is easy to see that $\leq_T^{\mathcal{P}}$ is faithful, and the standard reduction cover of $\leq_T^{\mathcal{P}}$ is closed under delayed patching according to Example 16, and is obviously also closed under definition by oracle-dependent cases. Examples 21 through 24 in the next section show that the relations $\leq_{h-T}^{\mathcal{P}}$, $\leq_m^{log}$, and $\leq_{NC_1}$, as well as the non-transitive relations $\leq_{k-tt}^{\mathcal{P}}$ for any fixed $k \geq 2$ are standard reducibilities, too.

The concepts we have used in formulating the definition of standard reducibility, such as bounded reducibility or faithfulness, extend canonically to binary relations on ω^ω, however we have to apply the following minor adjustments.

- Effective reduction covers are defined w.r.t. to the standard enumeration of partial recursive functionals from ω^ω to ω^ω.
- In the definition of faithfulness, we consider arbitrary constant functions instead of just $\emptyset$ and ω, that is, for example we require that for all functions f and for all constant functions g, the lower cone of $f \oplus g$ is contained in the lower cone of f.
- During defining a concept of closure under definition by oracle-dependent cases, we consider only tt-conditions which are unions of finitely many basic open sets or are complements of such unions. Observe that if we define a tt-condition on ω^ω to be a subclass of ω^ω where membership in this class depends only on the function values at a fixed finite set of places, then for example the class of all functions g where $g(0)$ is even is a tt-condition, but unlike in the case of 2^ω this tt-condition is neither the union of finitely many basic open sets nor is it the complement of such a union.
- In the definition of delayed patching the enumeration of finite partial characteristic functions is replaced by an appropriate effective enumeration of all partial functions from ω to ω with finite domain.

Using these adapted concepts, we introduce the concept STANDARD REDUCIBILITY ON $\boldsymbol{\omega^\omega}$ by literally the same formulation as in Definition 20. In Example 26, we show that the reducibility on ω^ω introduced by Mehlhorn [31] via his class of basic feasible functionals is indeed a standard reducibility.

Our results about lattice embeddings and exact pairs for standard reducibilities on 2^ω extend by almost identical proofs to standard reducibilities on ω^ω. The minor adjustments necessary are due to the fact that 2^ω is compact, while ω^ω is not. In particular, the concept generalized use introduced in Sect. 3.1 is available for reducibilities on ω^ω only if relativized to some compact subspace of ω^ω, see Remark 32.

We assume that it is possible to develop many properties shared by standard reducibilities on 2^ω and on ω^ω within some unified framework where the domain of the binary relation $\leq_r$ under consideration is required to satisfy certain properties, say, is downwards closed under $\leq_r$ and can be written as the countable product of some appropriate subset of ω. However, we feel that the benefits to be gained from such a unified approach probably will not make up for the additional work necessary. For this reason, we state the results for standard reducibilities on 2^ω and ω^ω separately. We focus on the case 2^ω, while in the case of ω^ω we just indicate the necessary adjustments to the proofs.

2.7 Some Examples of Standard Reducibilities

From the Examples 14 and 16 it is immediate that the relation $\leq_T^{\mathcal{P}}$ is a standard reducibility where this fact is witnessed by the canonical reduction cover of this relation. For the reducibilities considered in Examples 21, 22, and 23, the canonical reduction covers are closed under delayed patching, but, due to restrictions on the oracle access, not under definition by oracle-dependent cases. However, by closing the corresponding reduction covers under definition by oracle-dependent cases, we see that these reductions are indeed standard reducibilities. We show this in detail in Example 21 for the reduction $\leq_{h-T}^{\mathcal{P}}$, and sketch the similar proofs for the other reducibilities.

Example 21. *We show that the relation $\leq_{h-T}^{\mathcal{P}}$ introduced by Homer [17, 18, 19] is a standard reducibility. A set A is $\leq_{h-T}^{\mathcal{P}}$ - reducible to some set B iff there is a polynomial time bounded oracle Turing machine T and some polynomial p where, firstly, T reduces A to B and, secondly, for all x and y in ω where T queries place y of its oracle on inputs B and x, we have*

$$|x| \leq p(|y|) \ . \tag{16}$$

Homer observes that the relation $\leq_{h-T}^{\mathcal{P}}$ is reflexive and transitive, and by Proposition 8 it is then immediate that $\leq_{h-T}^{\mathcal{P}}$ is faithful. The relation $\leq_{h-T}^{\mathcal{P}}$ is a bounded reducibility, as is witnessed by the the effective reduction cover $\mathcal{R}$ from [19] where each reduction Δ in $\mathcal{R}$ corresponds to a pair of some polynomial time *bounded oracle Turing machine T and some polynomial p where T, while working on some number argument x, queries place y only after checking that (16) is satisfied, and outputs immediately 0 in case it encounters some query y where this check fails.*

Like in the case of the reducibility $\leq_T^{\mathcal{P}}$ considered in Example 16, we infer that $\mathcal{R}$ is closed under delayed patching. We choose some witnessing subclass $\mathcal{F}$ of the class of polynomial time computable functions, and by Proposition 51, we assume that the functions in $\mathcal{F}$ are bounded by some function which increases so slowly that the remainder of this proof goes through.

The reduction cover $\mathcal{R}$ is not closed under definition by oracle-dependent cases, because given some tt-condition, intuitively speaking, a reduction in $\mathcal{R}$ cannot access the places of the oracle the tt-condition depends on for all sufficiently large number inputs. In order to obtain a reduction cover of $\leq_{h-T}^{\mathcal{P}}$ which is closed under definition by oracle-dependent cases and under delayed patching, we let $\mathcal{R}'$ be the closure of $\mathcal{R}$ under definition by oracle-dependent cases. Observe that we obtain the reductions in $\mathcal{R}'$ from the reductions in $\mathcal{R}$ via iterated definitions by oracle-dependent cases, and thus for each reduction Γ in $\mathcal{R}'$ there is some finite set I and reductions $\Delta_1, \ldots, \Delta_k$ in $\mathcal{R}$ where k is equal to $2^{|I|}$ such that for all sets B and all x in ω we have

$$\Gamma(B, x) \;=\; \begin{cases} \Delta_1(B, x) & \text{in case } B \mid I = \alpha_0 \\ \ldots & \ldots \\ \Delta_k(B, x) & \text{in case } B \mid I = \alpha_k \end{cases} \quad . \tag{17}$$

Here α_i is the i-th partial characteristic function with domain I. As a consequence, the functionals in $\mathcal{R}'$ are all $\leq_{h-T}^{\mathcal{P}}$ - reductions, because for every given set argument, they agree with some reduction in $\mathcal{R}$. The reduction cover $\mathcal{R}'$ is effective and by definition closed under definition by oracle-dependent cases, and so it remains to show that $\mathcal{R}'$ is closed under delayed patching. In order to do so, we show that $\mathcal{R}'$ is closed under patching with functions in the functional simulation class $\mathcal{F}$ introduced above, that is, for some functional Γ as in (17) and for some function g in $\mathcal{F}$, the reduction $\Gamma \otimes g$ is in $\mathcal{R}'$. For every partial characteristic function β with domain I, we let

$$d(\beta, x) := \min_{i \in \{1,\ldots,k\}} [\; \alpha_i = \langle \beta, \sigma_{g(x)} \rangle \;] \ . \tag{18}$$

Observe that for every set B where the restriction of B to I is equal to β, the value $d(\beta, x)$ corresponds to the case selected in (17) while evaluating $\Gamma \otimes g(B, x)$. With β and the α_i fixed, the computation of $d(\beta, .)$ amounts to a look-up in some finite table plus the computation of the restriction of $\sigma_{g(x)}$ to the domain of β. Now, because we have chosen the functions in $\mathcal{F}$ to increase very slowly, we can assume that the computation of $d(\beta, .)$ can be done so fast that the functionals

Δ^β defined by

$$\Delta^\beta(X,x) \;=\; \begin{cases} \Delta_1(X,x) & \textit{in case } d(\beta,x)=1 \\ \dots & \dots \\ \Delta_k(X,x) & \textit{in case } d(\beta,x)=k \end{cases}$$

are in $\mathcal{R}$. Then, given some set B where the restriction of B to I is equal to β, the g-patch of Γ and of Δ^β, respectively, agree on B, because we have for all x in ω

$$\begin{aligned}[\Gamma \otimes g](B,x) \;&=\; \Gamma(\langle B, \sigma_{g(x)}\rangle, x) \;&=\; \Delta_{d(\beta,x)}(\langle B, \sigma_{g(x)}\rangle, x) \\ &=\; \Delta^\beta(\langle B, \sigma_{g(x)}\rangle, x) \;&=\; [\Delta^\beta \otimes g](B,x)\end{aligned} \quad ,$$

where the equations hold, respectively, by definition of patching, by definition of $d(\beta,.)$, by definition of Δ, and finally, again by definition of patching. But then we have for all sets B and for all x in ω

$$[\Gamma \otimes g](B,x) \;=\; \begin{cases} [\Delta^{\alpha_0} \otimes g](B,x) & \textit{in case } \alpha_0 \sqsubseteq B \\ \dots & \dots \\ [\Delta^{\alpha_k} \otimes g](B,x) & \textit{in case } \alpha_k \sqsubseteq B \end{cases} \quad , \tag{19}$$

where the functionals Δ^{α_i}, and hence also the functionals $\Delta^{\alpha_i} \otimes g$ are in $\mathcal{R}$, and thus the functional on the right-hand side of (19), and hence also $\Gamma \otimes g$, is in $\mathcal{R}'$, because we have chosen $\mathcal{R}'$ as the closure of $\mathcal{R}$ under definition by oracle-dependent cases.

Example 22. *We let $\mathcal{FL}$ be the class of functions from ω to ω which are computable in logarithmic space. More precisely, we require that for each f in $\mathcal{FL}$, there is some constant c in ω and some Turing machine which computes $f(x)$ in space $c \log |x|$. Here we use a Turing machine model where we can move in both directions on the input tape, but only from left to right on the output tape. By the latter condition, the class $\mathcal{FL}$ is closed under composition, because given two Turing machines as above, we can feed the output of the first to the input of the second; for details of this construction see for example the textbook of Hopcroft and Ullman [20]. Observe that we cannot just compute the two functions one after the other, because the output of the first might be too large to be stored within logarithmic space.*

We let A be $\leq_m^{log}$ - reducible to some set B iff A is many-one reducible to B via some function in $\mathcal{FL}$. As usual in connection with many-one reducibilities, here we assume that the sets $\emptyset$ and ω are reducible to all other sets. The relation $\leq_m^{log}$ is reflexive and, by closure of $\mathcal{FL}$ under composition, also transitive. Using Proposition 8, we easily obtain that $\leq_m^{log}$ is faithful.

We let $\mathcal{R}$ be the effective reduction cover for $\leq_m^{log}$ where the reductions in $\mathcal{R}$ correspond in the natural way to the functions in $\mathcal{FL}$. Schmidt [41] implicitly shows that the class $\mathcal{FL}$ is a functional simulation class, and thus according to Proposition 51, we can choose some appropriate subclass $\mathcal{F}$ of $\mathcal{FL}$ which is *again a simulation class and where the functions in $\mathcal{F}$ increase so slowly that $\mathcal{R}$ is closed under patching with the functions in $\mathcal{F}$. The reduction cover $\mathcal{R}$ is obviously not closed under definition by oracle-dependent cases, however, we infer by an argument which is basically the same as the one given in Example 21 that the closure of $\mathcal{R}$ under definition by oracle-dependent cases is not only closed under definition by oracle-dependent cases, but also under delayed patching, that is, the relation $\leq_m^{log}$ is indeed a standard reducibility.*

Example 23. *We fix some k in ω and consider the polynomial time bounded k-question truth-table reducibility $\leq_{k-tt}^{\mathcal{P}}$. The reducibility $\leq_{k-tt}^{\mathcal{P}}$ is a standard reducibility, as follows by basically the same proof as for the relation $\leq_{h-T}^{\mathcal{P}}$ in Example 21. In case $k \geq 2$, the relation $\leq_{k-tt}^{\mathcal{P}}$ is not transitive. Observe that a bounded reducibility $\leq_r$ is transitive iff the class of $\leq_r$ - reductions is closed under composition. Now, if we define for all l in ω the functional Γ_l by*

$$\Gamma_l(B,x) \; := \; \begin{cases} 1 & \text{if } B \cap \{x,\ldots,x+l-1\} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} ,$$

then Γ_k is a $\leq_{k-tt}^{\mathcal{P}}$ - reduction for all k in ω, but the composition of Γ_k with itself is equal to Γ_{2k-1} and hence is not a $\leq_{k-tt}^{\mathcal{P}}$ - reduction for $k \geq 2$, as can be shown by the standard methods for separating reducibilities due to Ladner et al.[24].

Example 24. *Wilson [49, 50], considers relativized circuits, that is, circuits which besides the usual logical gates can contain oracle gates. Each oracle gate has a fixed number of inputs and a single output, and on some input x of the required length, the oracle gate outputs 1 iff x is in the given oracle. Wilson introduces for each fixed k in ω the binary relation $\leq_{NC_k}$ on 2^ω where we have $A \leq_{NC_k} B$ iff there is some constant c in ω and some function g in $\mathcal{FL}$ such that, given some appropriate encoding scheme for circuits, for all n in ω, the value $g(1^n)$ is the code of some relativized circuit of size $n^c + c$ and depth $c \log^k n$ which on oracle B computes the restriction of A to strings of length n.*

We fix $k \geq 1$, and we let $\mathcal{R}_k$ be the effective reduction cover for $\leq_{NC_k}$ where the reductions in $\mathcal{R}_k$ correspond to pairs of some constant c in ω and some function g in $\mathcal{FL}$. The reduction cover $\mathcal{R}_k$ is closed under definition by oracle-dependent cases, because given some tt-condition $\mathcal{T}$ and two reductions in $\mathcal{R}_k$, within the given resource bounds, we can construct some relativized circuit which checks membership of its oracle in $\mathcal{T}$, and which then in turns can be used as a "subcircuit" in a relativized circuit which computes the $\mathcal{T}$-mix of the two reductions under consideration. The reduction cover $\mathcal{R}_k$ is also closed under delayed patching, because according to Proposition 51, we can choose some appropriate subclass $\mathcal{F}$ of the simulation class $\mathcal{FL}$ which is again a simulation class and where the functions in $\mathcal{F}$ increase so slowly that for every f in $\mathcal{F}$ we can patch the oracle by hard-wiring the partial characteristic function $\sigma_{f(n)}$ into *the relativized circuit which corresponds to length n. We leave the details of this construction, as well as the proof of faithfulness to the reader.*

Next, we consider a bounded reducibility on 2^ω which, intuitively speaking, has sufficient computation power in order to qualify for the concept standard reducibility, but which fails to be a standard reducibility due to having severely restricted oracle access.

Example 25. *The relation $\leq_1^{\mathcal{P}}$ is the variant of polynomial time bounded many-one reducibility where the reduction functions are required to be one-one. The relation $\leq_1^{\mathcal{P}}$ is a bounded reducibility, but not a standard reducibility. In particular, there are sets A such that $A \oplus A$ is not $\leq_1^{\mathcal{P}}$ - reducible to A, that is, the join operator does not provide l.u.b.'s for the structure $(2^\omega, \leq_1^{\mathcal{P}})$ and consequently, the relation $\leq_1^{\mathcal{P}}$ is not faithful. In order to show this, we consider the the set*

$$A := \{0^{g(n)} : n \in \omega\}$$

where g is a function which grows fast enough, say $g(n) = 2^{2^n}$. For a proof by contradiction, we assume that there is some 1:1 reduction function r computable in polynomial time which witnesses that $A \oplus A$ is reducible to A. If we choose n sufficiently large, by assumption on g and because r is computable in polynomial time, r cannot map any string of length less or equal to $1 + g(n)$ to a string of length greater than $g(n+1)$. Hence r maps the $2n+2$ strings in $(A \oplus A)^{\leq[1+g(n)]}$ to the $n+1$ strings in $A^{\leq g(n+1)}$, which contradicts r being 1:1.

We conclude this section by a sketch of the most widely known reducibility on ω^ω, which is due to Mehlhorn and which we denote by $\leq_T^{\mathcal{FP}}$. We introduce this relation by an alternative characterization due to Kapron and Cook, for discussion and further references see the recent survey by Clote [16].

Example 26. *Mehlhorn [31] introduces the concept basic feasible functional and then defines the relation $\leq_T^{\mathcal{FP}}$ on ω^ω as the reducibility which has the class of basic feasible functionals as a reduction cover. Kapron and Cook [21, 22], introduce basic poly-time functionals and show that the class of all these functionals is a reduction cover for $\leq_T^{\mathcal{FP}}$, too. We sketch their concept of basic poly-time functional where for the sake of simplicity and in accordance with our conception of reducibility we restrict our account to the case of a single number and function argument. We leave it to the reader to show that the reducibility $\leq_T^{\mathcal{FP}}$, which has the class of basic poly-time functionals as a reduction cover, is in fact a standard reducibility. Kapron and Cook use oracle Turing machines which have a designated oracle-query tape and a designated oracle-answer tape where on entering some special query state with string x written on the oracle-query tape, the oracle-query tape is erased, and the value of the function oracle at place x is written in binary on the oracle-answer tape. For each such oracle call, the* *length of the binary expansion of the returned function value is charged against the running time of the oracle Turing machine.*

In order to obtain appropriate bounds on the running time, Kapron and Cook consider second-order polynomials. The class of second-order polynomials over a single function variable L and a single numeric variable k is defined inductively by requiring that the class contains k and all c in ω, and if P and Q are second-order polynomials in this class, then so are $P+Q$, $P \cdot Q$, and $L(P)$. Second-order polynomials are evaluated in the obvious way, for example given the second-order polynomial

$$P \ := \ L(L(k)) \ + \ 2 \cdot L(k+1) \ + \ 4 \cdot k$$

and the function f which maps x to x^2, then $P(f, 2)$ evaluates to 42. Then, a functional Γ on ω^ω is a basic poly-time functional if there is some oracle Turing machine T and some second-order polynomial P such that for all functions f in ω^ω and for all x in ω, T computes $\Gamma(f, x)$ in time $P(|f|, |x|)$ where $|x|$ is the usual length of the string x and $|f|$, the length of f, is defined by

$$|f|(n) := \max_{|y| \leq n} |f(y)| \ .$$

3 Basic Concepts Related to Standard Reducibilities

3.1 Generalized Use

In subsequent proofs, we use the concept generalized use introduced in Definition 28 as a substitute for the usual concept of use of an oracle Turing machine.

Proposition 27. *Let Γ be a continuous functional and let x be in ω.*

1. *Among all sets I which satisfy*

$$A \mid I \;=\; B \mid I \quad \textit{implies} \quad \Gamma(A,x) = \Gamma(B,x) \quad \textit{for all sets } A \textit{ and } B \ , \tag{20}$$

   there is a least one w.r.t. set theoretical inclusion.
   (Observe that this least set is uniquely determined and, by continuity of Γ and Fact 3, is actually finite.)

2. *Given some recursive functional Φ_e, we can effectively obtain the least set I which satisfies (20) where Γ is replaced with Φ_e. The procedure which computes I is uniform in e and x, however it might fail to terminate in case Φ_e is undefined for some oracle.*

Definition 28. For a continuous functional Γ and for x in ω, we denote the least set I which satisfies (20) as GENERALIZED USE $u(\Gamma, x)$ of Γ at place x.

Proof of Proposition 27. In Sect. 1.5 on topology we already mentioned that due to compactness of 2^ω, for a continuous functional Γ and for all x in ω there is always some finite set I which satisfies (20). Thus, concerning the first assertion we are done if we can show that the class of sets I which satisfy (20) is closed under intersection. So we let I_0 and I_1 be two sets in this class and we assume that the sets A and B agree on $I_0 \cap I_1$. Then there is some set C which agrees with A on I_0 and with B on I_1. Now on the one hand, the value $\Gamma(C, x)$ is equal to $\Gamma(A, x)$ by assumption on I_0, and on the other hand $\Gamma(C, x)$ is equal to $\Gamma(B, x)$ by assumption on I_1, that is, $\Gamma(A, x)$ is equal to $\Gamma(B, x)$.

For a proof of the second assertion in Proposition 27, observe that for a partial recursive functional Φ_e where $\Phi_e(A, x)$ is defined for all sets A, the proof of Fact 3 can be effectivized in the sense that we can obtain effectively in e and x some finite, but not necessarily minimal set I which satisfies (20). In order to obtain the least set which satisfies (20), it is then sufficient to search through all subsets of I. □

Proposition 29 provides an alternate characterization of the concept generalized use.

Proposition 29. *Let Γ be some functional, and let x and z be in ω. Then z is in $u(\Gamma, x)$ iff there is some set C where we have*

$$\Gamma(C \cup \{z\}, x) \;\neq\; \Gamma(C \setminus \{z\}, x) \ . \tag{21}$$

Proof. First, assume that z satisfies (21) for some set C, but z is not in $u(\Gamma, x)$. Then the sets $C \cup \{z\}$ and $C \setminus \{z\}$ witness that the set $I := u(\Gamma, x)$ does not satisfy (20), thus contradicting the definition of generalized use. Next, we assume that (21) is false for all sets C, but z is in $u(\Gamma, x)$. In this case the set

$$I \;:=\; u(\Gamma, x) \setminus \{z\}$$

also satisfies (20), thus contradicting the minimality condition in the definition of generalized use. In fact, if I does not satisfy (20), there are sets A and B which agree on I, and where $\Gamma(A, x)$ differs from $\Gamma(B, x)$. But then we obtain

$$\Gamma(A, x) = \Gamma(A \cup \{z\}, x) \qquad \text{and} \qquad \Gamma(B, x) = \Gamma(B \cup \{z\}, x) \;\; ,$$

because (21) is false for all sets. So, $A \cup \{z\}$ and $B \cup \{z\}$ agree on $u(\Gamma, x)$, but for these sets Γ yields different values at x, thus contradicting the definition of generalized use. □

Remark 30. *We have chosen the term generalized use because the concept is similar to the usual notion of use of an oracle Turing machine, but is more general and is also applicable to nonrecursive functionals. However, for the functional which is computed by some total oracle Turing machine, the generalized use at some place x might be strictly contained in the use of the corresponding oracle Turing machine, because an oracle Turing machine in general asks superfluous queries. On the other hand observe that there is some recursive function u such that for every recursive functional Φ_e we have for all x*

- *if $\Phi_e(., x)$ is total, then so is $\Phi_{u(e)}(., x)$, and in this case $\Phi_{u(e)}(A, x)$ is equal to $\Phi_e(A, x)$ for all sets A,*
- *if $\Phi_e(., x)$ is total, then the oracle Turing machine $T_{u(e)}$ which computes $\Phi_{u(e)}$ on number input x queries for all oracles exactly the places in the generalized use $u(\Phi_e, x)$ of Φ_e at x,*
- *if $\Phi_e(X, x)$ is undefined for some set X, then $\Phi_{u(e)}(A, x)$ is undefined for all sets A.*

For example we can choose the function u such that the oracle Turing machine $T_{u(e)}$ on number input x, intuitively speaking, first tries to compute $u(\Phi_e, x)$, then queries its oracle in order to obtain the restriction σ of the oracle to $u(\Phi_e, x)$, and finally simulates T_e on number input x and oracle $\langle \emptyset, \sigma \rangle$. Observe that by Example 31, if T_e works in polynomial time, then unless $\mathcal{P}$ equals $\mathcal{NP}$, in general we cannot choose $T_{u(e)}$ to work in polynomial time, too.

Example 31 shows that given some arbitrary, but fixed functional Γ which is computed by a polynomial time bounded oracle Turing machine T, in general the following two problems are both $\mathcal{NP}$-complete: to decide for a given pair

(x, z) of natural numbers, firstly, whether z is in $u(\Gamma, x)$ and, secondly, whether for some oracle the oracle Turing machine T queries place z of the oracle while working on number argument x.

Example 31. *Let Γ_e be the functional computed by the e-th machine in some appropriate enumeration of all polynomial time bounded oracle Turing machines. Consider the sets*

$$G_e := \{\langle x, z\rangle : z \in u(\Gamma_e, x)\} \ ,$$
$$Q_e := \{\langle x, z\rangle : \text{ there is some set } B \text{ where } T_e \text{ queries } z \text{ on input } B \text{ and } x\} \ .$$

In connection with simulations of an oracle Turing machine T by some nondeterministic Turing machine, we refer by the expression nondeterministically guessing an oracle *to the following technique.*

- *Given some number input x and a polynomial p which bounds the running time, and hence also bounds the number of oracle queries of T, we perform $p(|x|)$ nondeterministic choices. We identify these choices in the canonical way with values in $\{0, 1\}$, and record them on some work tape. Then, during the simulation of T, the outcome of the i-th choice is substituted for the answer to the i-th* new *oracle question. Note that it is necessary to keep track of all the oracle queries asked during the simulation, in order to be able to decide for a query whether it is new or has been asked before, because in the latter case the same answer as before has to be assigned to the query.*

We show first that the sets G_e and Q_e are in $\mathcal{NP}$ for each fixed e in ω. In order to decide in nondeterministic polynomial time whether $\langle x, z\rangle$ is in G_e, respectively in Q_e, we guess nondeterministically an oracle B, and we accept on the path which corresponds to B iff by simulating T_e we obtain that

- *$\Gamma_e(B \cup \{z\}, x)$ differs from $\Gamma_e(B \setminus \{z\}, x)$ (in order to decide G_e),*
- *the query z is asked on inputs x and B (in order to decide Q_e).*

Next, we construct a polynomial time bounded oracle Turing machine T_e where the sets G_e and Q_e are $\mathcal{NP}$-complete. We show the completeness of these sets by defining $\leq_m^{\mathcal{P}}$ - reductions from the standard $\mathcal{NP}$-complete set Sat, where x is in Sat iff the x-th propositional formula in conjunctive normal form has some satisfying assignment.

- *On inputs x and B, the oracle Turing machine T_e checks whether the values $B(\langle x, 1\rangle), B(\langle x, 2\rangle), \ldots$ describe a satisfying assignment for the x-th propositional formula in conjunctive normal form.*
  - *If the answer is yes, then T_e outputs $B(\langle x, 0\rangle)$.*
  - *If the answer is no, then T_e outputs 0,* without *querying $\langle x, 0\rangle$.*

We leave it to the reader to check that the function defined by $g(x){:=}\langle x, \langle x, 0\rangle\rangle$ *witnesses* $Sat \leq^{\mathcal{P}}_m G_e$ *and* $Sat \leq^{\mathcal{P}}_m Q_e$.

We conclude this section by introducing an extension of the concept generalized use which we will apply in connection with standard reducibilities on ω^ω.

Remark 32 – Generalized Use for Reducibilities on ω^ω . *The proof of Proposition 27 and the subsequent definition of the concept generalized use basically depend only on the fact that* 2^ω *can be written as the countable product of uniformly recursive finite sets, that is,* 2^ω *is equal to* $\otimes_{i\in\omega} E_i$ *where all sets* E_i *are equal to* $\{0,1\}$. *Accordingly, a subclass* $\mathcal{C}$ *of* ω^ω *is* EFFECTIVELY COMPACT *iff we have*

$$\mathcal{C} \;=\; \bigotimes_{i\in\omega} E_i$$

where the nonempty and finite sets E_i *are uniformly recursive. Now, given a recursive functional* Γ *on* ω^ω *and some effectively compact subclass* $\mathcal{C}$ *of* ω^ω, *we let* $u^{\mathcal{C}}(\Gamma, x)$, *the* GENERALIZED USE *of* Γ *at* x W.R.T. $\boldsymbol{\mathcal{C}}$, *be the least set* I *which satisfies*

$$f \mid I \;=\; g \mid I \quad \text{ implies } \quad \Gamma(f,x) = \Gamma(g,x) \quad \text{for all } f, g \text{ in } \mathcal{C} \;.$$

As in the special case where $\mathcal{C}$ *is equal to* 2^ω, *the set* $u^{\mathcal{C}}(\Gamma, x)$ *is finite and uniquely determined. Furthermore, we can obtain the generalized use of some recursive functional* Φ_e *uniformly effective in* x, e *and an index of some recursive representation of* $\mathcal{C}$. *Observe that while we introduce this relativization of the concept generalized use to some effectively compact class only for recursive functionals, the relativization extends in the canonical way to arbitrary continuous functionals on* ω^ω.

3.2 Closure under Finite Variation

We show in this section that standard reducibilities are c.f.v. In the proof of the corresponding Proposition 35, we use Lemmas 33 and 34, which we state separately for further use.

Lemma 33. *Let the relation* $\leq_r$ *be faithful. If for all sets* B *there is some nonempty and finite set* M *such that for all sets* A *which are reducible to* B *we have*

$$A \setminus M \leq_r B \qquad \text{and} \qquad A \cup M \leq_r B \;,$$

then the relation $\leq_r$ *is downwards c.f.v.*

Proof of Lemma 33. Assuming that for each set B there is some set M as in the lemma, we show a series of claims which then imply that $\leq_r$ is downwards c.f.v. In connection with the first claim, recall that by zA we refer to the set $\{zw : w \in A\}$ where zw denotes the concatenation of the strings z and w.

Claim 1. For all z in ω and all sets A and B we have $A \leq_r B$ iff $zA \leq_r B$.

Proof of Claim 1. We show Claim 1 by induction on the length of z. In case z is the empty string, the assertion is vacuously satisfied. In case z differs from the empty string, we first assume that z is equal to $1y$ for some string y, and in this case we have

$$zA \;=\; \emptyset \oplus (yA) \;\; .$$

Thus, by faithfulness of $\leq_r$, the upper cone of zA coincides with the intersection of the upper cones of $\emptyset$ and yA, and we obtain

$$\geq_r(zA) \;=\; \geq_r(\emptyset) \cap \geq_r(yA) \;=\; \geq_r(yA) \;=\; \geq_r(A) \;\; ,$$

where the equations, from left to right, hold by the preceding remark, because the upper cone of the empty set contains all sets, and by the induction hypothesis for y. So we are done with the induction step in case z has the form $1y$. In case z has the form $0y$, we proceed likewise while exploiting the fact that zA can be written as $(yA) \oplus \emptyset$.

Claim 2. $A \leq_r B$ implies $A \setminus \{\lambda\} \leq_r B$ for all sets A and B.

Proof of Claim 2. Given some set B, we choose some nonempty finite set M as in the assumption of the lemma, and we let z be the maximal element in M. Obviously, the least element of zA is greater or equal to z, and hence we obtain

$$(zA) \setminus M \;=\; (zA) \setminus \{z\} \;=\; z(A \setminus \{\lambda\}) \;\; . \tag{22}$$

Now, in case A is reducible to B, then so is zA by Claim 1, and further, by choice of M, also $(zA) \setminus M$ is reducible to B. Thus also the set on the right of (22) is reducible to B, and applying Claim 1 again, we infer $A \setminus \{\lambda\} \leq_r B$.

Claim 3. $A \leq_r B$ implies $Left(A) \leq_r B$ and $Right(A) \leq_r B$ for all sets A and B.

Proof of Claim 3. Assuming $A \leq_r B$ we infer

$$Left(A) \oplus Right(A) \;=\; A \setminus \{\lambda\} \;\leq_r\; B \;\; ,$$

where the relations hold by definition of the functionals $Left$ and $Right$ and by Claim 2, respectively. Now we are done, because for a faithful relation $\leq_r$ the join of two sets is a locally transitive upper bound for them, and hence $Left(A)$ and $Right(A)$ are both reducible to B.

Claim 4. $A \leq_r B$ implies $A \cup \{\lambda\} \leq_r B$ for all sets A and B.

Proof of Claim 4. Given some set B, we show by induction on the length of the maximal element in M that if we have

$$A \leq_r B \quad \text{implies} \quad A \cup M \leq_r B \quad \text{for all sets} \quad A \ , \tag{23}$$

for some nonempty and finite set M, then (23) holds also with M replaced by $\{\lambda\}$. Claim 4 then is immediate, because by the assumption of the Lemma there is some nonempty and finite set M which satisfies (23). Assuming that the nonempty and finite set M satisfies (23), we let z be the maximal element in M. The assertion is immediate if z is the empty string, so we assume that z is equal to $0y$ for some y, and we leave the similar symmetric case $z = 1y$ to the interested reader. Then for all sets A we have

$$A \cup Left(M) \ = \ Left(A \oplus \emptyset) \cup Left(M) \ = \ Left((A \oplus \emptyset) \cup M) \ , \tag{24}$$

where the equations hold, because A coincides with $Left(A \oplus \emptyset)$, and because the functional $Left$ distributes over the union operator. Now, if some set A is reducible to B, then so is the set on the right of (24), due to assumption on M, and because lower $\leq_r$ - cones are closed under $\oplus$ and the functional $Left$ due to faithfulness and Claim 3, respectively. As a consequence, the set $Left(M)$ satisfies (23), and hence we are done by the induction hypothesis, because the maximal element y of $Left(M)$ is strictly shorter than the maximal element $0y$ of M.

Claim 5. $A \leq_r B$ implies $A \setminus \{z\} \leq_r B$ and $A \cup \{z\} \leq_r B$ for all sets A and B and for all z in ω.

Proof of Claim 5. We show Claim 5 by induction on the length of z. In case z is the empty string, the assertion is immediate from Claims 2 and 4. So, again leaving the similar proof of the case $z = 1y$ to the reader, we assume that z is equal to $0y$ for some y in ω. Observe that by definition of the join operator we have for all sets A

$$(A \setminus \{z\}) \setminus \{\lambda\} \ = \ (Left(A) \setminus \{y\}) \oplus Right(A) \tag{25}$$

$$(A \cup \{z\}) \setminus \{\lambda\} \ = \ (Left(A) \cup \{y\}) \oplus Right(A) \ . \tag{26}$$

Assuming $A \leq_r B$, we obtain by Claim 3 that $Left(A)$ and $Right(A)$ are both reducible to B, and further, by the induction hypothesis for y, the same holds for $Left(A) \setminus \{y\}$ and $Left(A) \cup \{y\}$. Now, by faithfulness of $\leq_r$, the lower cone of B is closed under join, and hence the sets on the right-hand side of (25) and (26) are reducible to B. But then so are the sets on the left-hand side, and, by Claim 4, also the sets $A \setminus \{z\}$ and $A \cup \{z\}$.

Lemma 33 now is immediate by Claim 5, because given some set A in the lower cone of some set B, changing A at finitely many arbitrarily chosen places will not put us outside the lower cone of B. □

Next, we state another technical lemma which we will use in the proof of Proposition 35.

Lemma 34. *Let $\mathcal{R}$ be a reduction cover for some binary relation $\le_r$ on 2^ω where $\mathcal{R}$ is closed under definitions by oracle-dependent cases. Let $A_0, \dots, A_n$ and $B_0, \dots, B_n$ be sets where the sets B_i are pairwise distinct and where for all $i \le n$ the set A_i is $\le_r$ - reducible to B_i. Then there is a functional Γ in $\mathcal{R}$ which simultaneously witnesses these relations, that is, for all $i \le n$ the set A_i is equal to $\Gamma(B_i)$.*

Proof of Lemma 34. We show the lemma by induction on n. The case $n = 0$ is immediate by the definition of reduction cover, so we assume $n > 0$. By assumption, for all $i < n$ the sets B_i and B_n differ at some place z_i, and consequently, the restriction of B_n to the set $\{z_i : i < n\}$ defines a tt-condition $\mathcal{T}$ which contains B_n, but none of the sets B_i where $i < n$. Now by assumption on B_n there is a functional Γ_0 in $\mathcal{R}$ where A_n is equal to $\Gamma_0(B_n)$, and by the induction hypothesis, there is a functional Γ_1 in $\mathcal{R}$ where A_i is equal to $\Gamma_1(B_i)$ for all $i < n$. By choice of Γ_0, Γ_1 and $\mathcal{T}$ it is then immediate that A_i is reducible to B_i for all $i \le n$ via the functional $\langle \Gamma_0, \Gamma_1 \rangle^{\mathcal{T}}$, which is in $\mathcal{R}$ by closure of $\mathcal{R}$ under definition by oracle-dependent cases. □

Proposition 35. *Standard reducibilities are c.f.v.*

Proof of Proposition 35. Given some standard reducibility $\le_r$, by definition of standard reducibility, we choose some effective reduction cover $\mathcal{R}$ for $\le_r$ which is closed under definition by oracle-dependent cases and under delayed patching. Here we assume that the latter closure property is witnessed by some functional simulation class $\mathcal{F}$, that is, $\mathcal{R} \otimes \mathcal{F}$ is contained in $\mathcal{R}$.

Claim 1. For each set B, there is some nonempty finite set M, such that for all sets A which are reducible to B we have

$$A \setminus M \le_r B \quad \text{and} \quad A \cup M \le_r B.$$

Proof of Claim 1. Recall that $\sigma_0, \sigma_1, \dots$ is an effective enumeration of all finite partial characteristic functions where σ_0 is equal to λ. We choose some i in ω where $\sigma_i \not\sqsubseteq B$, and we let f be the function from ω to ω which yields i at place 1, and yields 0 at all other places. We choose some delayed simulation s of f in $\mathcal{F}$ and consider the set

$$M := \{z \in \omega : s(z) = i\} \;=\; \{z \in \omega : \sigma_{s(z)} \not\sqsubseteq B\} \;, \tag{27}$$

where the second equation in (27) holds because the range of s contains exactly the values 0 and i. The set M is nonempty and finite, because $s(z)$ must attain the value $f(1) = i$ for some z, and must yield 0 for almost all z.

Now, by choice of i, the sets B and $\langle B, \sigma_i \rangle$ are different, and consequently, by Lemma 34 and because the empty set is reducible to all sets, there is a functional Γ in $\mathcal{R}$ where

$$A = \Gamma(B) \quad \text{and} \quad \emptyset = \Gamma(\langle B, \sigma_i \rangle) \ .$$

By choice of M and Γ, we obtain

$$[\Gamma \otimes s](B, z) \ = \begin{cases} 0 & \text{in case } z \text{ is in } M \\ A(z) & \text{otherwise} \end{cases} ,$$

that is, the functional $\Gamma \otimes s$ witnesses that $A \setminus M$ is reducible to B. By a modification of this argument where we choose the functional Γ such that it witnesses simultaneously

$$A = \Gamma(B) \quad \text{and} \quad \omega = \Gamma(\langle B, \sigma_i \rangle) \ ,$$

we obtain that $A \cup M$ is reducible to B, too.

Now, by Claim 1 and Lemma 33, the relation $\leq_r$ is downwards c.f.v, and it remains to show that $\leq_r$ is upwards c.f.v, too. Assuming $A \leq_r B$ and $B =^* C$, we let I be the finite set of arguments on which B and C differ, and we fix some j in ω where σ_j is equal to $B \mid I$, that is, B is equal to $\langle C, \sigma_j \rangle$. We let the function f from ω to ω be defined by $f(0) = 0$ and by $f(x) = j$ for all $x > 0$, and we let s be some delayed simulation of f in $\mathcal{F}$. We choose some functional Γ in $\mathcal{R}$ which witnesses $A \leq_r B$. Then the s-patch $\Gamma \otimes s$ of Γ is again in $\mathcal{R}$, and hence the set $\Gamma \otimes s(C)$ is reducible to C. But by choice of s, the set $\langle C, \sigma_{s(x)} \rangle$ is equal to B for all but finitely many x, and consequently, $\Gamma \otimes s(C)$ is a finite variation of $\Gamma(B) = A$. Consequently, because $\leq_r$ is downwards *c.f.v.*, A is reducible to C, that is, $\leq_r$ is upwards c.f.v., too. □

3.3 The Class of Admissible Cases

Recall from the introduction that a functional can alternatively be described as a unary function from 2^ω to 2^ω, or as a binary function from $2^\omega \otimes \omega$ to $\{0, 1\}$. The latter characterization suggests two ways, how we might combine two functionals Γ_0 and Γ_1 into a new functional Γ via a definition by cases.

1. In Definition 18 we have already introduced DEFINITION BY ORACLE-DEPENDENT CASES: given functionals Γ_0 and Γ_1, and some tt-condition $\mathcal{T}$, we let $\langle \Gamma_0, \Gamma_1 \rangle^{\mathcal{T}}(A)$ be equal to $\Gamma_0(A)$ or be equal to $\Gamma_1(A)$, depending on whether A is in $\mathcal{T}$ or is not.
2. Now, given some set M, we let

$$\Gamma(X) \ := \ \langle \Gamma_0(X), \Gamma_1(X) \rangle^M \ , \tag{28}$$

   that is, $\Gamma(X, x)$ is equal to $\Gamma_0(X, x)$ if x is in M, and is equal to $\Gamma_1(X, x)$, otherwise. We denote this type of combination as DEFINITION BY NUMBER-DEPENDENT CASES.

Proposition 37 shows that for a set M it does not matter whether we require, firstly, that the class of all $\leq_r$ - reductions is closed under definition by number-dependent cases w.r.t. M or, secondly, that all lower $\leq_r$ - cones are closed under definition by cases w.r.t. M. In order to formulate this and further result, we introduce some notation.

Definition 36. Let $\leq_r$ be a binary relation on 2^ω.

1. The CLASS OF ADMISSIBLE CASES of $\leq_r$ is

$$\mathcal{M}_r \ := \ \{M \subseteq \omega : \ \text{for all sets } A, B, X \ [A \leq_r X \text{ and } B \leq_r X \text{ implies } \langle A, B\rangle^M \leq_r X]\} \ .$$

2. The CLASS OF LEAST SETS of $\leq_r$ is

$$\mathcal{L}_r \ := \ \{A \subseteq \omega : A \leq_r B \text{ for all sets } B\} \ .$$

Proposition 37. *Let $\leq_r$ be a binary relation on 2^ω and let M be some set. Then the following conditions are equivalent.*

- *M is in $\mathcal{M}_r$.*
- *The functional Γ defined by*

$$\Gamma(X) := \langle \Gamma_0(X), \Gamma_1(X)\rangle^M \tag{29}$$

  is an $\leq_r$ - reduction for all $\leq_r$ - reductions Γ_0 and Γ_1.
- *The functional Γ defined by (29) is an $\leq_r$ - reduction for all $\leq_r$ - reductions Γ_0 and Γ_1 in some fixed reduction cover for $\leq_r$.*

Proof. The second assertion obviously implies the third, so it remains to show that the first assertion implies the second, and that the third implies the first. In order to prove the former implication, let M be in $\mathcal{M}_r$, and let Γ_0 and Γ_1 be arbitrary $\leq_r$ - reductions. By assumption, for all sets X both of $\Gamma_0(X)$ and $\Gamma_1(X)$ are $\leq_r$ - reducible to X, and then so is the set $\Gamma(X)$ as defined in (29), by definition of Γ and by assumption on M; consequently, as the set X was chosen arbitrarily, Γ is an $\leq_r$ - reduction. Next, let $\mathcal{R}$ be some reduction cover for $\leq_r$ and let the M be a set where Γ as defined in (29) is an $\leq_r$ - reduction for all functionals Γ_0 and Γ_1 in $\mathcal{R}$. Now, given sets A and B which are both reducible to some set C, we choose functionals Γ_0 and Γ_1 in $\mathcal{R}$ which witness these facts, respectively. Then for the functional Γ as defined in (29), the set $\Gamma(C)$ is equal to $\langle A, B\rangle^M$, and the latter set is reducible to C by assumption on Γ, that is, M is in $\mathcal{M}_r$. □

For the bounded reducibility $\leq_T^{\mathcal{P}}$, it is easy to see that the class of least sets $\mathcal{L}_T^{\mathcal{P}}$ and the class of admissible cases $\mathcal{M}_T^{\mathcal{P}}$ are both equal to $\mathcal{P}$, and therefore in case of the relation $\leq_T^{\mathcal{P}}$, in particular the class of admissible cases is closed under the set theoretical operations union, intersection and complement. Proposition 38 shows that the latter holds for all standard reducibilities and that $\mathcal{M}_r$ is always included in $\mathcal{L}_r$. Example 39 shows that in general this last inclusion is proper.

Proposition 38. *Let $\leq_r$ be a faithful relation on 2^ω.*

1. *$\mathcal{M}_r$ is a subset of $\mathcal{L}_r$.*
2. *$\mathcal{M}_r$ contains $\emptyset$ and is closed under the set theoretical operations union, intersection, and complement. Equivalently, $(\mathcal{M}_r, \subseteq)$ is a subalgebra of $(2^\omega, \subseteq)$.*
3. *If $\leq_r$ is downwards c.f.v., then $\mathcal{M}_r$ is closed under the join operator.*

Proof. For a proof of the first assertion, observe that for a faithful relation $\leq_r$ the sets $\emptyset$ and ω are reducible to all other sets, and thus by definition of $\mathcal{M}_r$ we have for all sets A and for all sets M in $\mathcal{M}_r$

$$M = \langle \omega, \emptyset \rangle^M \leq_r A \quad .$$

The second assertion follows from the definition of $\mathcal{M}_r$ and because we have for all sets A, B, M, M_0, and M_1

$$\langle A, B \rangle^\emptyset = B, \quad \langle A, B \rangle^{\overline{M}} = \langle B, A \rangle^M, \quad \langle A, B \rangle^{M_0 \cap M_1} = \langle \langle A, B \rangle^{M_0}, B \rangle^{M_1} \quad ,$$

where the equations, from left to right, show that $\mathcal{M}_r$ contains $\emptyset$ and is closed under complement and intersection; closure under union then follows by the De Morgan formula.

In order to show the third assertion, observe that by definition of the operators involved, we have for all sets A and B and for all sets M_0 and M_1 in $\mathcal{M}_r$

$$\langle A, B \rangle^{M_0 \oplus M_1} \;=^*\; \mathit{Left}(\langle A \oplus \emptyset, B \oplus \emptyset \rangle^{M_0}) \oplus \mathit{Right}(\langle \emptyset \oplus A, \emptyset \oplus B \rangle^{M_1}) \quad . \tag{30}$$

Here Left and Right are implicitly defined by the equation

$$X \setminus \{\lambda\} \;=\; \mathit{Left}(X) \oplus \mathit{Right}(X) \quad .$$

Observe that in general equality does not hold in (30), because due to the definition of the join operator the empty string is in the set on the left-hand side of (30) iff it is in B, but it is never contained in the set on the right-hand side. Now, if A and B are in the lower cone of some set C, then so is the set on the right-hand side of (30), because for a faithful relation $\leq_r$ the empty set is in the lower cone of every set, and because lower cones are closed under join, under definition by cases with sets in $\mathcal{M}_r$ and, in case $\leq_r$ is downwards c.f.v., also under the operations Left and Right. Then, again by $\leq_r$ being downwards c.f.v., the set on the left-hand side of (30) is in the lower cone of C, and consequently, as A and B are chosen arbitrarily, $M_0 \oplus M_1$ is in $\mathcal{M}_r$. □

Example 39. *We call a set* NON-SELFDUAL *iff it is not* $\leq_m^{\mathcal{P}}$ *- reducible to its complement. Ladner et al. [24] have shown that there are recursive non-selfdual sets. We choose such a set* N *and we let*

$$A \leq_\nu B \;:\Leftrightarrow\; A \leq_m^{\mathcal{P}} (B \oplus \overline{N}) \;, \tag{31}$$

where we denote by $\overline{N}$ *the complement of* N*. We leave it to the interested reader to check that the relation* $\leq_\nu$ *is faithful and that the class of least sets* $\mathcal{L}_\nu$ *is exactly the lower* $\leq_m^{\mathcal{P}}$ *- cone of* $\overline{N}$*. Thus in particular,* $\mathcal{L}_\nu$ *is not closed under complement, because it contains* $\overline{N}$*, but cannot contain the set* N*, due to* N *being non-selfdual. Consequently, by Proposition 38,* $\mathcal{M}_\nu$ *is strictly contained in* $\mathcal{L}_\nu$*.*

Proposition 40. *For a standard reducibility* $\leq_r$*, the class* $\mathcal{M}_r$ *is a simulation class.*

Proof. We choose some effective reduction cover $\mathcal{R}$ for the standard reducibility $\leq_r$ which is closed under definition by oracle-dependent cases and under delayed patching, where the latter closure property is witnessed by some functional simulation class $\mathcal{F}$, that is, $\mathcal{R} \otimes \mathcal{F}$ is contained in $\mathcal{R}$. Further, we let i and j be different from 0 and such that the finite partial characteristic functions σ_i and σ_j are incompatible, that is, there is some y in ω where σ_i and σ_j are both defined and disagree. Given a recursive set L, we let

$$f_L(x) := \begin{cases} 0 & \text{if } x = 0 \\ i & \text{if } x \neq 0 \text{ and } L(x) = 0 \\ j & \text{otherwise} \end{cases} \;,$$

that is, basically we replace in the characteristic function of L all values 0 with i, and all values 1 with j. Here we let $f_L(0)$ be equal to 0 in order to ensure that we can effectively obtain a delayed simulation s_L of f_L in $\mathcal{F}$. We let

$$M_L \;=\; \{y \in \omega : s_L(y) = j\} \;.$$

We leave it to the reader to check that, firstly, the set M_L thus defined is a delayed simulation of L for all sets L where $L(0) = 0$ holds and that, secondly, the definition of M_L can be made effective in the sense that there is a recursive function sim where for all sets $L = \varphi_e$ the set M_L is equal to $\varphi_{sim(e)}$. Then, in order to show that $\mathcal{M}_r$ is a simulation class, it is sufficient to show that for every recursive set L, the set M_L is in fact in $\mathcal{M}_r$. We assume that A_0, A_1, and B are sets where A_0 and A_1 are both $\leq_r$ - reducible to B. The standard reducibility $\leq_r$ is c.f.v. according to Proposition 35, and consequently A_0 and A_1 are reducible to $\langle B, \sigma_i \rangle$ and $\langle B, \sigma_j \rangle$, respectively. These two facts are simultaneously witnessed by some functional Γ in $\mathcal{R}$, according to Lemma 34 and because the sets $\langle B, \sigma_i \rangle$ and $\langle B, \sigma_j \rangle$ are distinct by choice of i and j. Then, for the functional

$$\Gamma := \langle \Gamma_0, \Gamma_1 \rangle^{\{X \subseteq \omega : \sigma_i \sqsubseteq X\}} \;,$$

we have

$$\langle A_0, A_1\rangle^{M_L} \;=\; \langle \Gamma_0(\langle B, \sigma_i\rangle), \Gamma_1(\langle B, \sigma_j\rangle)\rangle^{M_L} \;=^*\; [\Gamma \otimes s_L](B) \;\leq_r\; B \;, \qquad (32)$$

and hence, by $\leq_r$ being c.f.v. the set $\langle A_0, A_1\rangle^{M_L}$ is reducible to B, and further, the set M_L is in $\mathcal{M}_r$. The relations in (32) hold from left to right, respectively, by choice of Γ_0 and Γ_1, by definition of Γ, and because $\Gamma \otimes s_L$ is an $\leq_r$ - reduction due to the closure properties of $\mathcal{R}$. Observe that in general in (32) we cannot replace equality up to finite variation by equality, because the delayed simulation s_L of f_L will yield the value $f_L(0) = 0$ on a finite initial segment of the natural numbers. □

3.4 Simulation of Recursive Oracles

Based on the fact that standard reducibilities possess reduction covers which are closed under delayed patching, we have shown in the last section that for a standard reducibility $\leq_r$ the class $\mathcal{M}_r$ of admissible cases is a simulation class. In this section, we consider a second important consequence of closure under delayed patching: any reduction to some recursive set is witnessed by some reduction which on increasing number inputs ignores larger and larger initial parts of its set argument. This fact amounts to some special form of delayed patching, where we do not patch according to some arbitrary effective sequence of finite partial characteristic functions, but with increasingly long initial segments of the course of value of some *fixed* recursive set.

Definition 41. 1. A function $m : \omega^2 \to \omega$ is a MODULUS OF ORACLE SIMULATION iff, firstly, m is recursive and, secondly, we have for all e in ω

- $\sigma_{m(e,x)} \sqsubseteq \sigma_{m(e,x+1)}$ for all x in ω,
- $\bigsqcup_{x\in\omega} \sigma_{m(e,x)} = \varphi_e$

2. A reduction cover is CLOSED UNDER ORACLE SIMULATION iff there is a modulus of oracle simulation m such that $\mathcal{R}$ is closed under patching with the functions $m(e,.)$ from ω to ω, that is, for all Γ in $\mathcal{R}$ and all e in ω, the functional $\Gamma \otimes m(e,.)$ is in $\mathcal{R}$.

Example 42. *For all e and x in ω we let*

$$I(e,x) \;:=\; \{i \in \omega : i < x \;\&\; \varphi_{e,|x|}(i)\downarrow\}$$
$$\alpha(e,x) \;:=\; \varphi_e \mid I(e,x) \;\;;$$

and we let $m(e,x)$ be the least natural number where $\sigma_{m(e,x)}$ is equal to $\alpha(e,x)$. Observe that the definition of the set $I(e,x)$ is formulated in terms of the approximation $\varphi_{e,s}(x)$ and not in terms of $\varphi_e(x)$, and that hence we can obtain the set $I(e,x)$ effectively in e and x. As a consequence the function m is recursive and is indeed a modulus of oracle simulation.

Proposition 43. *1. If $\mathcal{F}$ is a functional simulation class, then there is some modulus of oracle simulation m where for all e in ω the function*

$$m(e,.) : \omega \to \omega$$

is in $\mathcal{F}$.

2. If a reduction cover is closed under delayed patching, then it is also closed under oracle simulation.

3. Every standard reducibility has an effective reduction cover which is closed under oracle simulation.

Proof. From the first assertion, the second assertion follows by definition of the two closure conditions involved. The third assertion then is immediate, because by the second assertion any effective reduction cover which witnesses that some standard reducibility is closed under delayed patching is also closed under oracle simulation. So it remains to prove the first assertion. Given some functional simulation class $\mathcal{F}$ we choose some witnessing function sim, and we let m be the modulus of oracle simulation defined in Example 42. Observe that for all e, we have $m(e, 0) = 0$ because on the one hand by convention σ_0 is the empty partial characteristic function, while on the other hand by definition of m the value $m(e, 0)$ is the least index of the empty partial characteristic function. By the *smn*-theorem, there is a recursive function g where $m(e, x) = \phi_{g(e)}(x)$ holds for all e and x in ω. Then the function m' defined by

$$m'(e, x) = \phi_{sim(g(e))}(x)$$

is recursive, and m' is in fact a modulus of oracle simulation: for each e in ω, the function $m'(e,.)$ is a delayed simulation of $m(e,.)$, that is, intuitively speaking, m' yields a simulation of φ_e which is basically the same, but which is "slower" than the one provided by m. □

In the proof of Proposition 43 we have used that for the modulus of oracle simulation from Example 42 we have $m(e, 0) = 0$ for all natural numbers e. In fact, given an arbitrary modulus of oracle simulation m_0 then by our convention that σ_0 is the empty partial characteristic function λ, we obtain again a modulus of oracle simulation if for all e, we change the value $m_0(e, 0)$ into 0.

We conclude this section by Lemma 44, which shows in particular that for every modulus of oracle simulation m and for every recursive set, we can choose some index i for this set such that the domains of the partial characteristic functions $\sigma_{m(i,0)}, \sigma_{m(i,1)}, \ldots$ are all initial segments of the natural numbers. The intuition for the existence of such an index is that computing $\varphi_i(x + 1)$ is so much harder than computing $\varphi_i(x)$ that our fixed resource bounded simulation scheme which corresponds to m obtains $\varphi_i(x)$ before it obtains the values of φ_i for any argument larger than x.

Lemma 44. *Let m be a modulus of oracle simulation. Then there is some recursive function d such that for all e in ω we have*

- *for all x in ω, the domain of $\sigma_{m(d(e),x)}$ has the form $\{0,\dots,n\}$,*
- *in case φ_e is total, $\varphi_{d(e)}$ is equal to φ_e,*
- *in case φ_e is not total and z is the least number where $\varphi_e(z)$ is undefined, the domain of $\varphi_{d(e)}$ is $\{x \in \omega : x < z\}$ and $\varphi_{d(e)}$ agrees there with φ_e.*

A proof of Lemma 44 can be found in [33] and [32].

3.5 Relations to Blum Measure

The concept of a modulus of oracle simulation is related to Blum measures as introduced by Blum [9]. Recall that a Blum measure with respect to the standard enumeration $\varphi_0, \varphi_1, \dots$ of the partial recursive functions is some partial function $\Psi : \omega^2 \to \omega$ such that, firstly, the domain of $\Psi(e,.)$ is equal to the domain of φ_e for all e in ω and, secondly, the question "$\Psi(e,x) = z$?" can be decided effectively in e, x, and z.

Now, for a modulus of oracle simulation m, by definition for all e and z in ω, φ_e is defined at z iff there is some x in ω such that z is in $dom(\sigma_{m(e,x)})$. Thus for every fixed e in ω, the domain of the partial function $\Psi_m(e,.)$ defined by

$$\Psi_m(e,z) := \mu x[z \in dom(\sigma_{m(e,x)})]$$

is equal to the domain of φ_e. Further, we can decide effectively, whether any given z in ω is an element of $dom(\sigma_{m(e,x)})$, and consequently, the function Ψ_m is a Blum measure.

Conversely, given a Blum measure Ψ and a non-decreasing, unbounded recursive function f from ω to ω, there is for example a recursive modulus of oracle simulation m such that we have

$$dom(\sigma_{m(e,x)}) \;=\; \{y \le x : \Psi(e,y) \le f(x)\} \ .$$

Intuitively speaking, $dom(\sigma_{m(e,x)})$ contains all potential oracle queries $y \le x$ such that $\varphi_e(y)$ can be computed within the Ψ - bound $f(x)$.

Lynch, Meyer, and Fischer [26] extend the concept Blum measure to relativized computations.

Definition 45. [Lynch, Meyer, and Fischer]

- A sequence $\Psi_0, \Psi_1, \dots$ of partial functionals

  $$\Psi_e : 2^\omega \otimes \omega \to \omega$$

  is a RELATIVE COMPLEXITY MEASURE (w.r.t. the standard enumeration $\Phi_0, \Phi_1, \dots$ of partial recursive functionals) iff for all sets A and all e, x, and y we have

- $\Psi_e(A, x)$ is defined iff $\Phi_e(A, x)$ is defined,
- we can decide uniformly effectively in e, x, y, and A whether $\Psi_e(A, x)$ is equal to y.

- Given some relative complexity measure Ψ and some function b from ω to ω, we let $A \leq_b^\Psi B$ iff there is some e in ω where A is equal to $\Phi_e(B)$ and where for almost all x we have $\Psi_e(B, x) \leq b(x)$. Further, for some subclass $\mathcal{B}$ of ω^ω, we let $A \leq_\mathcal{B}^\Psi B$ iff there is some b in $\mathcal{B}$ where $A \leq_b^\Psi B$.
- We call a binary relation on 2^ω COMPLEXITY DETERMINED if it can be written in the form $\leq_\mathcal{B}^\Psi$ where Ψ is some relative complexity measure and $\mathcal{B}$ is some subclass of ω^ω.

Lynch et al. [26, Footnote 4, p. 257] remark in a context of resource bounded reducibilities that

> "..., many-one reducibilities and one-one reducibilities are not determined by a complexity restriction in any complexity measure. ...".

While this remark is apparently true if restricted to more natural relative complexity measures such as time or space, Proposition 46 shows that if stated in full generality, their claim is false. The proposition shows that bounded reducibilities can be equivalently characterized in terms of relative complexity measure. While we consider an account in terms of reduction covers more appropriate for the work presented here, it might be reasonable to use relative complexity measures in cases where we want to consider, say, in greater detail than possible by using the concept delayed patching, relations between complexity determined reducibilities and subcomputations which are bounded w.r.t. some non-relativized Blum measure.

Proposition 46. *For a binary relation $\leq_r$ on 2^ω which is reflexive and c.f.v., the following conditions are equivalent.*

1. *The relation $\leq_r$ is a bounded reducibility.*
2. *There is some relative complexity measure and some recursive function b from ω to ω such that $\leq_r$ coincides with $\leq_b^\Psi$.*
3. *There is some relative complexity measure and some recursively presentable subclass $\mathcal{B}$ of ω^ω such that $\leq_r$ coincides with $\leq_\mathcal{B}^\Psi$.*

Proof. In order to show that the first assertion implies the second, we choose some recursive presentation E for some effective reduction cover of $\leq_r$, and we let e be the recursive function where $e(j)$ is the j-th element of E. Further, we let Ψ be some arbitrary relative complexity measure, say, the measure which corresponds to the usual concept of running time. By choice of E, we have that $\Phi_{e(j)}$ is total for all j in ω, and hence the same holds for $\Psi_{e(j)}$ by definition of relative complexity measure. Thus $\Psi_{e(j)}$ is actually recursive, because given arguments A and x we can successively check whether $\Psi_{e(j)}(A,x)$ is equal to $0, 1, \ldots$, until we find the corresponding function value, which exists by totality of $\Psi_{e(j)}$. Then also the function b defined by

$$b(x) := \max_{j \leq x} \max_{A \subseteq \omega} \ \Psi_{e(j)}(A,x)$$

is recursive, because the maximization over all sets A can be replaced by a maximization over all partial characteristic functions with domain equal to the generalized use $u(\Psi_{e(j)}, x)$, which in turn is recursive in j and x by recursiveness of e and the $\Psi_{e(j)}$. We leave it to the reader to check that

$$\Psi'_e(A,x) := \begin{cases} \Psi_e(A,x) & \text{in case } e \text{ is in } E \\ \Psi_e(A,x) + b(x) + 1 & \text{otherwise} \end{cases}$$

is a relative complexity measure where $\leq_r$ coincides with $\leq_b^{\Psi'}$.

The second assertion in Proposition 46 obviously implies the third, so we are done if we show that the third assertion implies the first. In order to do so , we let $\leq_r$ be complexity determined via some relative complexity measure Ψ and some subclass $\mathcal{B}$ of ω^ω where $\mathcal{B}$ is equal to $\{\phi_{e(j)} : j \in \omega\}$ for some recursive function e. We construct an effective reduction cover $\{\Gamma_{(e,i,j)} : e, i, j \in \omega\}$ for $\leq_r$ where each functional $\Gamma_{(e,i,j)}$ in the reduction cover corresponds to a triple of some oracle Turing machine T_e, some finite partial characteristic function σ_i, and some function $\phi_{e(j)}$ in $\mathcal{B}$ via the definition

$$\Gamma_{(e,i,j)}(A,x) := \begin{cases} \sigma_i(x) & \text{if } x \in dom(\sigma_i) \\ \Phi_e(A,x) & \text{if } x \notin dom(\sigma_i) \text{ and for all } y \leq x \\ & \qquad [y \notin dom(\sigma_i) \ \Rightarrow\ \Psi_e(A,y) \leq \phi_{e(j)}(y)] \\ A(x) & \text{otherwise} \end{cases} .$$

Observe that the $\Gamma_{(e,i,j)}$ are uniformly recursive in e, i, and j, and that in fact, each functional $\Gamma_{(e,i,j)}$ is an $\leq_r$ - reduction, because it maps each set A to a finite variation of A or of $\Phi_e(A)$, where in the latter case the computation of Φ_e on oracle A obeys on almost all number arguments the Ψ-bound $\phi_{e(j)}$. Moreover, all functionals $\Gamma_{(e,i,j)}$ together form an effective reduction cover for $\leq_{\mathcal{B}}^{\Psi}$, because given a set A which is $\leq_{\mathcal{B}}^{\Psi}$ - reducible to some set B, this fact is witnessed by some partial recursive functional Φ_e which on oracle B computes the set A and where for some j in ω and for almost all number arguments x the functional Φ_e obeys, for the given oracle B, the Ψ-bound $\Phi_{e(j)}$. But then there is some i in ω such that A is equal to $\Gamma_{(e,i,j)}(B)$. □

3.6 Properties of Simulation Classes

We show in this section that the concept simulation class is robust under several changes to its definition. While a similar remark holds for functional simulation classes, we leave the statement of the corresponding results to the reader. We start by extending the concept delayed simulation from total to partial characteristic functions, then showing in Proposition 48 that in a simulation class we can effectively find delayed simulations not only for all recursive sets, but for all partial recursive functions from ω to $\{0,1\}$.

Definition 47. A set S is a *delayed simulation* of some partial characteristic function α iff there is some nondecreasing function l from ω to ω where for all x in ω we have $S(x) = \alpha(l(x))$ and where the range of l is

$$\{z \in \omega : \alpha(y) \text{ is defined for all } y \leq z\} \ .$$

Proposition 48. *A subclass $\mathcal{S}$ of 2^ω is a simulation class iff there is a recursive function sim from ω to ω where for all e in ω we have*

- *$\varphi_{sim(e)}$ is in $\mathcal{S}$.*
- *If $\varphi_e(0)$ is equal to 0, then $\varphi_{sim(e)}$ is a delayed simulation of φ_e.*

Proof. Let $\mathcal{S}$ be a subclass of 2^ω. Obviously, a function *sim* as in Proposition 48 witnesses that $\mathcal{S}$ is a simulation class. In order to show the reverse implication, we recall from Example 14 that the class of functions computable in polynomial time is a functional simulation class, and hence the class of sets computable in polynomial time is a simulation class. We let sim_0 be a witnessing function for the latter fact which corresponds to the type of simulation described in Example 14. Then the two conditions in Proposition 48 are satisfied with *sim* replaced by sim_0. Thus, if sim_1 witnesses that $\mathcal{S}$ is a simulation class, then $sim{:=}sim_1 \circ sim_0$ witnesses that $\mathcal{S}$ satisfies the two conditions from the proposition. More precisely, given some index e in ω, then $\varphi_{sim_1(sim_0(e))}$ is obviously in $\mathcal{S}$. Further, if function l_0 witnesses that $\varphi_{sim_0(e)}$ is a delayed simulation of φ_e, and l_1 witnesses that $\varphi_{sim_1(sim_0(e))}$ is a delayed simulation of $\varphi_{sim_0(e)}$, then $l_0 \circ l_1$ witnesses that $\varphi_{sim(e)}$ is a delayed simulation of φ_e. □

While Proposition 48 shows that simulation classes can be equivalently defined by conditions which on first sight might appear more restrictive, we now derive another equivalent definition of the concept simulation class which is seemingly weaker, and which in particular is formulated in terms of a more liberal version of delayed simulations.

Definition 49. Let A and S be subsets of ω. S is a WEAK DELAYED SIMULATION of set A iff S is many-one reducible to A via some nondecreasing and unbounded function l from ω to ω.

Delayed simulations as introduced in Definition 12 differ from weak delayed simulations precisely by the fact that for a delayed simulation the function l is required to be onto ω, while for a weak delayed simulation the range of l might have gaps.

Proposition 50. *A subclass $\mathcal{S}$ of 2^ω is a simulation class iff there is a recursive function sim from ω to ω where for all e in ω we have*

- $\varphi_{sim(e)}$ *is in $\mathcal{S}$.*
- *If φ_e is a finite or co-finite set where $\varphi_e(0)$ is equal to 0, then $\varphi_{sim(e)}$ is a weak delayed simulation of φ_e.*

Proposition 50 can be viewed as extension of an observation due to Mueller [37], for details see Remark 58. We recommend to read the proof of Proposition 50 only after the material presented in Sect. 4. The reason for this is that, firstly, we employ the recursion theorem in a way similar to proofs in Sect. 4.6 where, however, the treatment there is more detailed. Secondly, in the proof we use the concept block as introduced in Definition 61 below. In connection with the following proof, observe that we count blocks starting with block 0, and hence for example block 1 of some set is actually its second block.

Proof of Proposition 50. If some function sim witnesses that the class $\mathcal{S}$ is a simulation class, then $\mathcal{S}$ and sim obviously satisfy the two conditions from the proposition. So given $\mathcal{S}$ and sim as in the proposition, we have to show that $\mathcal{S}$ is a simulation class, that is, that there is some recursive function sim_1 where $\varphi_{sim_1(e)}$ is in $\mathcal{S}$ for all e, and is a delayed simulation of φ_e whenever φ_e is a set where $\varphi_e(0)$ is equal to 0. We proceed by specifying uniformly effective in e some partial recursive function γ. Extending the argument in Remark 84 to the recursion theorem with parameters as used in the proof of Lemma 44, we obtain that already during the specification of γ we can use an index $d(e)$ for γ where d is some recursive function, that is, we have

$$\gamma \;=\; \varphi_{r(d(e),e)} \;=\; \varphi_{d(e)} \;\;,$$

where by the smn-theorem we can choose r to be the recursive, and where the equality on the right-hand side expresses that d yields fixed points of r according to the recursion theorem with parameters. In the construction we use the notation

$$S_i := \varphi_{sim_0(i)} \qquad \text{and} \qquad W_i := \varphi_{sim(sim_0(i))}$$

where we choose the recursive function sim_0 according to Proposition 48 such that it witnesses that in some arbitrary fixed simulation class, say, the class of sets computable in polynomial time, we can find delayed simulations not only for recursive sets, but also for partial recursive functions. Observe that S_i and W_i are sets for all i in ω and that by assumption on sim, if S_i is finite or co-finite, then W_i is a weak delayed simulation of S_i.

Now, by assumption on sim, the sets W_i are in $\mathcal{S}$ for all i in ω. Further, we will argue that by our construction $W_{d(e)}$ is a delayed simulation of φ_e whenever

φ_e is a set where $\varphi_e(0)$ is equal to 0. As a consequence, the recursive function defined by

$$sim_1(e) := sim(sim_0(d(e)))$$

witnesses that $\mathcal{S}$ is a simulation class, and we are done.

We perform a construction in stages where on termination of stage s we specify the value of γ at place $s+1$. In an initialization stage, firstly, we let $\gamma(0)$ be equal to 0, and secondly, we check whether $W_{d(e)}(0)$ is equal to 0. If this is not the case, we end the construction, and we proceed to stage 0, otherwise. Stage s of the construction, $s \geq 0$, consists of two substages:

1. We try to verify that block s of $W_{d(e)}(0)$ contains at least as many elements as block s of φ_e; on success, we proceed to the second substage.
2. We let $\gamma(s+1)$ be equal to $1 - \gamma(s)$.

In the remainder of this proof, we use the notation

$$\alpha_s := \omega \oplus \emptyset \mid \{0, \ldots, s\} \;\; ,$$

that is, the course of value of the partial characteristic functions α_s alternates between 0 and 1, where the starting 0 corresponds to $\alpha_s(\lambda)$.

Claim 1. We have $\alpha_{s+1} \sqsubseteq \varphi_{d(e)}$ iff stage s of the construction terminates. If stage s of the construction does not terminate, then $\varphi_{d(e)}$ is undefined on all places $x > s$.

Proof. Claim 1 follows by an easy induction argument, because we have chosen $d(e)$ to be an index for the partial recursive function under construction.

Claim 2. The set $W_{d(e)}$ has the value 0 at place 0.

Proof. Assuming that $W_{d(e)}(0)$ differs from 0, we infer that the construction ends during the initialization stage, and thus $\varphi_{d(e)}$ is just defined at place 0. Consequently, $S_{d(e)}$ is equal to the empty set, and then so is its weak delayed simulation $W_{d(e)}$, contrary to our assumption.

Claim 3. If s is minimal such that stage s of the construction does not terminate, then $W_{d(e)}$ has exactly $s+1$ blocks.

Proof. Given s as in the claim, then $\varphi_{d(e)}$ is equal to α_s by Claim 1, that is, the partial function $\varphi_{d(e)}$ has exactly $s+1$ blocks. But then its delayed simulation $S_{d(e)}$ has exactly $s+1$ blocks, too, and in particular $S_{d(e)}$ is either finite or co-finite. Thus $W_{d(e)}$ is a weak delayed simulation of $S_{d(e)}$, and hence can have at most $s+1$ blocks, because intuitively speaking, it cannot change values more often than $S_{d(e)}$. On the other hand, $W_{d(e)}$ has at least s blocks, because either

s is 0 or stage $s-1$ terminated after checking that block $s-1$ of $W_{d(e)}$ exists and has the required size. Now, $S_{d(e)}(0)$ and $W_{d(e)}(0)$ are both equal to 0, and hence if any of these two sets has an odd number of blocks, then it is finite, and likewise, for an even number of blocks, it is co-finite. But $W_{d(e)}$ is a weak delayed simulation of $S_{d(e)}$, and hence either both are finite or both are infinite. Thus we obtain a contradiction, if we assume that $S_{d(e)}$ has $s+1$ blocks, while $W_{d(e)}$ has s blocks. The preceding discussion then leaves us with $s+1$ as the only possible number of blocks for $W_{d(e)}$.

Claim 4. If block s of φ_e exists and is finite, then stage s of the construction terminates.

Proof. By induction on s we can assume that stage s is reached during the construction. But if stage s does not terminate, then $\varphi_{d(e)}$ is equal to α_s, and hence block s of $W_{d(e)}$ is infinite by Claim 3. But then, because block s of φ_e is finite, the first substage of stage s must eventually terminate, contradicting our assumption.

Now, assume that φ_e is a set where block s of φ_e exists. In case block s is finite, then by Claim 4, stage s of the construction terminates, and by the first substage this can only happen if block s of $W_{d(e)}$ contains at least as many elements as block s of φ_e. On the other hand, if block s is infinite, then all earlier stages terminate by Claim 4, but stage s does not, because the check in the first substage fails. Then block s of $W_{d(e)}$ is infinite by Claim 3. By this cardinality considerations, we obtain that $W_{d(e)}$ is in fact a delayed simulation of φ_e, whenever $\varphi_e(0)$ is equal to 0. □

Proposition 51 comes in handy while showing that specific resource bounded reducibilities are closed under delayed patching, see Example 16 and Sect. 2.7.

Proposition 51. *Let the recursive function b from ω to ω be nondecreasing and unbounded, and let $\mathcal{F}$ be a functional simulation class. Then the class*

$$\{f \in \mathcal{F} : f(x) \leq b(x) \text{ for all } x \in \omega\} \tag{33}$$

is again a functional simulation class.

Proof. The idea of the proof is quite simple: given some recursive function h in class $\mathcal{F}$, we construct a recursive delayed simulation g of h where $g(x) \leq b(x)$ holds for all x in ω. Then, given some delayed simulation f of g where this fact is witnessed by some nondecreasing function l with range ω we have for all x in ω

$$f(x) \ = \ g(l(x)) \ \leq \ b(l(x)) \ \leq \ b(x) \ ,$$

where the relations hold, from left to right, by choice of l, because g is bounded by b, and finally, because b is nondecreasing while we have $l(x) \leq x$. Thus in order to show that the class defined in (33) is a functional simulation class, we map a recursive function not directly to a delayed simulation h in $\mathcal{F}$, but first to a recursive delayed simulation g of h which is bounded by b, and then to a delayed simulation f of g in $\mathcal{F}$.

Formally, given some function h in $\mathcal{F}$, we let $l(0)$ be equal to 0 and for x in ω we let

$$l(x+1) := \begin{cases} l(x)+1 & \text{in case } h(y) \leq b(x+1) \text{ for all } y \leq l(x)+1 \\ l(x) & \text{otherwise} \end{cases} ,$$

and further

$$g(x) \ := \ h(l(x)) \ .$$

The function g is a delayed simulation of h and is by construction bounded by b. Furthermore, an index for g can be obtained effectively from an index for h, that is, there is some recursive function r which maps an index for some function h in $\mathcal{F}$ to an index of the corresponding delayed simulation g as above. Now, given some function sim which witnesses that $\mathcal{F}$ is a functional simulation class, then by the preceding discussion, the function

$$sim' := sim \circ r \circ sim$$

witnesses that the class defined in (33) is a functional simulation class. □

We conclude this section by an easy technical remark which we will use in subsequent proofs.

Remark 52. *The definition of simulation class does not specify how the witnessing function sim should behave on arguments e where the value $\varphi_e(0)$ is equal to 1 or is undefined. However, given some witnessing function sim, there are several ways to "normalize" the function sim in order to achieve some reasonable behaviour for such indices e. For example given an index e, we can first change φ_e in order to enforce the value 0 at place 0, and then assign to e a delayed simulation of this altered partial characteristic function. More precisely, let ζ be the partial characteristic function where $dom(\zeta) = \{0\}$ and $\zeta(0) = 0$, and let g be some recursive function such that for all e in ω we have*

$$\varphi_{g(e)} := \langle \varphi_e, \zeta \rangle \ .$$

Now, if the function sim_0 witnesses that $\mathcal{S}$ is a simulation class, then so does $sim{:=}sim_0 \circ g$. In addition, for all e in ω, the set $\varphi_{sim(e)}$ is a delayed simulation of $\langle \varphi_e, \zeta \rangle$, and thus is a delayed simulation of φ_e in case $\varphi_e(0)$ is equal to 0.

3.7 Previous Work on Simulation Classes

The axiomatic approach of Schmidt [41] is based on the concept recursive gap closure which we introduce in Definition 53 in a slightly modified form. The difference is that we refer by the term block to a maximal set of consecutive numbers which all are in or all are out of some set, while Schmidt requires in addition that strings of the same length either all are in or all are out of the block.

Definition 53.
- A GAP LANGUAGE is a subset of ω which is infinite and co-infinite.
- A BLOCK of a subset A of ω is a maximal set of consecutive numbers which either all are in A or all are out of A.
- A subclass $\mathcal{C}$ of 2^ω is RECURSIVELY GAP CLOSED iff for every recursive gap language R there is some set G in $\mathcal{C}$ such that the set G, as well as its complement, contain infinitely many blocks of R.

Note that every simulation class $\mathcal{S}$ is recursively gap closed: given some recursive gap language R, then by Lemma 66 below there is some gap cover G for R in $\mathcal{S}$, and here in particular G and its complement contain infinitely many blocks of R. On the other hand, there are recursively gap closed classes which are not simulation classes and which in fact do not even contain delayed simulations of all gap languages. For example consider the subclass $\mathcal{D}$ of the recursive sets where for each set in the class there are infinitely many x where x is in X, but $x-1$ and $x+1$ are not. Then for all sets in $\mathcal{D}$, the course of value contains infinitely often the "pattern" 010, and hence the class $\mathcal{D}$ for example does neither contain a delayed simulation of the empty set nor of the gap language $\{3x+1 : x \text{ in } \omega\} \cup \{3x+2 : x \text{ in } \omega\}$.

Schmidt shows recursive gap closure for several complexity classes, where from her proofs it is immediate that the classes considered are in fact simulation classes. In particular, we thus obtain that the class of sets simultaneously computable in linear time and logarithmic space is a simulation class. The content of Theorem 54 is basically the same as Schmidt's Theorem 3.1 in [41], up to minor modifications and to reformulation in our terms. Note in this connection that most of the further results in [41] are variants or applications of this theorem.

Theorem 54. *[Schmidt] Let $\mathcal{C}$ be a subclass of REC and let the class $\mathcal{M}$ be recursively gap closed where $\mathcal{C}$ is closed under definition by number-dependent cases with sets in $\mathcal{M}$, that is, for all sets M in $\mathcal{M}$ we have*

$$A, B \text{ in } \mathcal{C} \quad \text{implies} \quad \langle A, B\rangle^M \text{ in } \mathcal{C} \;.$$

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be recursively presentable classes which are c.f.v. and where $\mathcal{C}$ is neither contained in $\mathcal{C}_1$, nor in $\mathcal{C}_2$. Then $\mathcal{C}$ is not contained in the union of $\mathcal{C}_1$ and $\mathcal{C}_2$.

Proof. Let $C^1_0, C^1_1, \ldots$ and $C^2_0, C^2_1, \ldots$ be effective enumerations of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. By assumption, we choose sets D_1 and D_2 which are in $\mathcal{C}$, but not in $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. We construct in stages a recursive gap language R, where during stage s we specify which numbers are in block s of R; this then determines R by letting $R(0)$ be equal to 0. We let I_0 be the block which contains just 0. Then, given the first s blocks $I_0, \ldots, I_{s-1}$, we choose the next block I_s so large that, firstly, every set which agrees with D_1 on I_s disagrees on I_s with C^1_0 through C^1_s and, secondly, likewise every set which agrees with D_2 on I_s disagrees on I_s with C^2_0 through C^2_s. Such a set exists and can be found effectively by searching through all partial characteristic functions with domain equal to the union of $I_0, \ldots, I_{s-1}$, because $\mathcal{C}_1$ and $\mathcal{C}_2$ are c.f.v. We leave the details of the argument, which is similar to the one used in the proof of Lemma 65, to the reader. Next, we choose some set M in the recursively gap closed $\mathcal{M}$ where both, M and its complement, contain infinitely many blocks of R. Then the set

$$\langle D_1, D_2 \rangle^M$$

is in $\mathcal{C}$ by assumption on $\mathcal{C}$ and $\mathcal{M}$, but by our construction is neither contained in $\mathcal{C}_1$, nor in $\mathcal{C}_2$. □

Corollary 55. *Let $\leq_r$ be some bounded reducibility which is faithful and c.f.v. and where the class $\mathcal{M}_r$ of admissible cases is recursively gap closed. Then the structure $(REC, \leq_r)$ is dense.*

Proof. Given recursive sets A and B where $A \leq_r B$, we assume for a proof by contradiction that there are no recursive sets strictly between A and B. Then all sets in the lower cone of B are in one of the classes $\mathcal{C}_1$ and $\mathcal{C}_2$ defined by

$$\mathcal{C}_1 := \ \leq_r (A) \qquad\qquad \mathcal{C}_2 := \{X \subset \omega : X \leq_r B \text{ and } B \leq_r X \oplus A\} \ ,$$

because given some set X which is reducible to B, but is not in $\mathcal{C}_1$ or $\mathcal{C}_2$, the set $X \oplus A$ is strictly between A and B. But then, Theorem 54 yields a contradiction. First, observe that the lower cone of B is closed under definition by number-dependent cases with the sets in the recursively gap closed class $\mathcal{M}_r$, and is neither contained in $\mathcal{C}_1$ as is witnessed by B, nor in $\mathcal{C}_2$, as is witnessed by A. Furthermore, the classes $\mathcal{C}_1$ and $\mathcal{C}_2$ are both c.f.v. and recursively presentable by assumption on $\leq_r$. □

The work done by Mehlhorn in [29] and [31] has been most influential in the design of our generalized approach to resource bounded reducibilities, and in particular so by his concept delayed simulation. We conclude this section by a sketchy comparison of Mehlhorn's account to ours. For the sake of simplicity, we restrict this comparison to reducibilities on 2^ω, while similar remarks hold for the treatment of reducibilities on ω^ω. In summary, we find that Mehlhorn's account differs from ours in the following respects

- Mehlhorn's axioms are designed to be applied to reducibilities of Turing type, and in fact they are neither satisfied for reducibilities of bounded truth-table type, nor for honest reducibilities such as $\leq^{\mathcal{P}}_{h-T}$.
- We do not presuppose transitivity of the relations involved.
- By our condition on delayed patching, we are able to construct sets which have a specified greatest lower bound as is needed for lattice embeddings and for constructing exact pairs.
- In order to show the density of the recursive degrees, Mehlhorn uses the looking-back technique in a rather involved, Ladner-style argument. In the proofs of our embedding results we employ the subsequently developed gap language technique, which renders proofs more modular.

Mehlhorn [31] introduces his concept delayed simulation in order to formulate his Axiom 6, which if restricted to reducibilities on 2^ω and restated in our terms, reads as follows:

> There is some effective reduction cover $\mathcal{R}$ for $\leq_r$ and some recursive function *sim* from ω to ω such that for all e in ω we have
> - $\Phi_{sim(e)}$ is in $\mathcal{R}$.
> - In case $\Phi_e(B)$ is total for some B where $\Phi_e(B,0)$ is equal to 0, then $\Phi_{sim(e)}(B)$ is a delayed simulation of $\Phi_e(B)$.

While rephrasing the original condition we corrected a flaw, as the original axiom requires that $\Phi_{sim(e)}(B)$ is a delayed simulation of $\Phi_e(B)$ whenever $\Phi_e(B)$ is total. By an argument similar to the one given in Remark 13, we can show that this is impossible for a function *sim* where $\Phi_{sim(e)}$ is total for all e. Mehlhorn's Axiom 6 is satisfied for the reducibility $\leq^{\mathcal{P}}_T$, as well as for other bounded reducibilities of Turing or truth-table type. In order to prove this in the case of $\leq^{\mathcal{P}}_T$, we proceed as in Example 14, where we replace the Turing machine $T_{sim(e)}$ with an oracle Turing machine which tries to compute the values $\Phi_e(B,0), \Phi_e(B,1), \ldots$ by simulating the oracle Turing machine T_e on oracle B. Mehlhorn's axiom is for example not satisfied for reducibilities of bounded truth-table type, as we show in Example 56 for the case of one-question truth-table reducibility. The intuitive reason for this is that for example in the case of $\leq^{\mathcal{P}}_{1-tt}$, while the available resources are unbounded as the number inputs increase, the restriction on the oracle access is *not* relaxed, that is, even for large number arguments, not more than one place of the oracle might be used.

Example 56. *We show that Mehlhorn's Axiom 6 on delayed simulations is not satisfied for reducibilities of one-question truth-table type, that is, for binary relations on 2^ω which have a reduction cover $\mathcal{R}$ where we have for all Δ in $\mathcal{R}$ and all x in ω*

$$|u(\Delta, x)| \leq 1 \ . \tag{34}$$

We consider the recursive functional Γ defined by

$$\Gamma(B,x) := \begin{cases} 0 & \textit{in case } B \cap \{0,\ldots,x\} = \emptyset \\ 1 & \textit{otherwise} \end{cases} .$$

For a contradiction, we assume that there is some functional Δ which satisfies (34) and where $\Delta(B)$ is a delayed simulation of $\Gamma(B)$ whenever the latter set does not contain 0, or equivalently, by definition of Γ, is a delayed simulation of $\Gamma(B)$ for all B which do not contain 0. Observe that Δ maps the empty set to itself, and hence for all sets B and all x where B agrees at the single place in $u(\Delta,x)$ with the empty set, $\Delta(B,x)$ is equal to 0. Next we fix some nonempty set X which does not contain 0. Then $\Delta(X)$ is co-finite and in particular there is some z in ω where $\Delta(X,z)$ is equal to 1. We let

$$C := X \cap u(\Delta,z) \ ,$$

and hence $\Delta(C,z)$ is equal to 1. If we assume that $u(\Delta,x)$ is contained in $u(\Delta,z)$ for all $x > z$, we obtain a contradiction, because in this case Δ maps all sets which do not intersect $u(\Delta,z)$ to the empty set. On the other hand, if there are $x > z$ where $u(\Delta,x)$ is not contained in $u(\Delta,z)$, then C agrees on $u(\Delta,x)$ with the empty set, and $\Delta(C,x)$ is equal to 0. So in this case, $\Delta(C)$ is not the complement of an initial segment of the natural numbers, and is hence not a delayed simulation of $\Gamma(C)$.

Mehlhorn derives from his axioms that any countable p.o. can be embedded into any proper interval of the corresponding structure on the recursive sets. The proof of this results in particular relies on the axiom on delayed simulations stated above and on an axiom which entails that for all sets A, B, C and M we have

$$A \leq_r C,\ B \leq_r C,\ M \leq_r C \quad \text{implies} \quad \langle A,B\rangle^M \leq_r C \ , \tag{35}$$

that is, the lower cone of every set C is closed under definitions by number-dependent cases with sets in the lower cone of C. Example 57 shows in the special case of polynomial time bounded reducibilities that (35) is neither satisfied for reducibilities of many-one type, nor for reducibilities of k-tt-type where $k \geq 1$ is fixed.

Example 57. *We consider the reducibilities $\leq_m^{\mathcal{P}}$ and $\leq_{k-tt}^{\mathcal{P}}$ for some fixed $k \geq 1$, that is, the polynomial time bounded reducibilities of many-one and k-question truth table type, respectively. We show that both reducibilities do not satisfy (35). Concerning $\leq_m^{\mathcal{P}}$, recall from Example 39 that there is a recursive non-selfdual set N, that is, N is not $\leq_m^{\mathcal{P}}$ - reducible to its complement $\overline{N}$. Thus we obtain a contradiction, if we assume that the relation $\leq_m^{\mathcal{P}}$ satisfies (35), because we have for all x in ω*

$$N(x) \ = \ \begin{cases} 0 & \textit{in case } x \in \overline{N} \\ 1 & \textit{in case } x \notin \overline{N} \end{cases} \ = \ \langle \emptyset, \omega\rangle^{\overline{N}}(x) \ .$$

Concerning $\leq^{\mathcal{P}}_{k-tt}$, we let for all m and l in ω the functionals $\Gamma_{(m,l)}$ be defined by

$$\Gamma_{(m,l)}(B,x) := \begin{cases} 1 & \text{in case } B \cap \{x+m, \ldots, x+m+l-1\} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} .$$

The functionals $\Gamma_{(m,k)}$ are obviously $\leq^{\mathcal{P}}_{k-tt}$ - reductions for all m in ω, however, by the standard methods for separating bounded reducibilities due to Ladner et al. [24], for all k, k_0 and m in ω, the functional $\Gamma_{(m,k_0)}$ is not a $\leq^{\mathcal{P}}_{k-tt}$ - reduction in case $k_0 > k$. So we obtain a contradiction, if we assume that the relation $\leq^{\mathcal{P}}_{k-tt}$ satisfies (35), because we have for all sets B and all x in ω

$$\begin{aligned} \Gamma_{(m,2k)}(B,x) &= \begin{cases} 1 & \text{in case } x \in \Gamma_{(m,k)}(B) \\ \Gamma_{(m+k,k)}(B,x) & \text{otherwise} \end{cases} . \\ &= \langle \omega, \Gamma_{(m+k,k)}(B) \rangle^{\Gamma_{(m,k)}(B)}(x) \end{aligned}$$

Remark 58. *Mueller [37] introduces a variant of Mehlhorn's Axiom 6 on delayed simulations where, if restricted to the case of sets, the function sim is not required to yield delayed simulations for all recursive sets, but only for all finite and co-finite sets. Mueller then shows that based on this altered axiom we can still perform the Ladner-style construction used by Mehlhorn for showing his density result. Mueller's observation can be viewed as a special case of our Proposition 50, where we use a seemingly even weaker formulation in terms of weak delayed simulations.*

Remark 59. *While Mueller [37] obtains results about minimal pairs only in the context of time bounds, Mehlhorn claims in his Theorem 5.1 in [31], without giving a proof, that his axioms imply the existence of minimal pairs in the recursive degrees. Observe in this connection that the usual minimal pair constructions are based on oracle simulation techniques as described in Sect. 3.4 and it is at least not obvious whether the axiom system proposed by Mehlhorn entails the applicability of this techniques.*

In a context of constructive measure theory, Mehlhorn shows in [30] for certain bounded reducibilities that every recursive set is half of a minimal pair of recursive sets. Mehlhorn remarks in [31, p.164] that the proof of this result relies on the fact that while performing a reduction, we can search trough an interval the length of which is exponential in the size of the number input, and that hence the proof applies to reducibilities such as elementary recursive in, *but not to polynomial time bounded reducibilities. Using different methods from [2], it is shown in [33] that the result indeed holds for all standard reducibilities.*

3.8 Structural Properties of Faithful Reducibilities

While in the following sections we derive more involved properties of the structure (REC, $\leq_r$) from the assumption that $\leq_r$ is a standard reducibility, we state in Proposition 60 some easy structural properties which follow from weaker assumptions on the relation $\leq_r$ under consideration.

Proposition 60. *Let $\leq_r$ be a faithful bounded reducibility which is c.f.v. and let A be some recursive set.*

- *The lower $\leq_r$ - cone of A is recursively presentable, and in particular, contains only recursive sets.*
- *There is a recursive set B which is strictly above A w.r.t. the relation $\leq_r$, that is, we have $A \leq_r B$ and $B \not\leq_r A$.*
- *If A is not in the class of least sets $\mathcal{L}_r$, then there is some recursive set B which is incomparable to A w.r.t. the relation $\leq_r$. Consequently, $(REC, \leq_r)$ is not a linear preordering.*
- *Let $(K, \leq)$ be a countable partial ordering such that every member of K has at most finitely many predecessors. Then $(K, \leq)$ can be embedded as a p.o. into $(REC, \leq_r)$.*

Proof. Let $\{\Delta_0, \Delta_1, \ldots\}$ be an effective reduction cover for $\leq_r$. The first assertion is immediate, because the lower cone of the recursive set A can be written as $\{\Delta_i(A) : i \in \omega\}$ where the sets $\Delta_i(A)$ are uniformly recursive in i.

The second assertion follows from the first assertion: there is some recursive set D which is not $\leq_r$ - reducible to A, because if REC were contained in the recursively presentable lower cone of A, one could effectively diagonalize against all recursive set, which is a plain contradiction. Now, the join operator provides locally transitive l.u.b.'s w.r.t. the faithful relation $\leq_r$, and thus A is reducible to $B{:=}A \oplus D$, while the set $A \oplus D$ cannot be reducible to A, because otherwise, D were reducible to A, thus contradicting our choice of D.

We obtain a recursive set B which is incomparable with A by a standard injury-free finite extension construction, that is, we give an effective procedure which successively computes finite partial characteristic functions $\beta_0 \sqsubseteq \beta_1 \sqsubseteq \ldots$ where $dom(\beta_i)$ converges to ω, and then we let $B{:=}\bigsqcup_{i\in\omega} \beta_i$. In the construction, we assure $B \not\leq_r A$ by letting for all i in ω

$$\beta_{2i}(y_i) := 1 - \Delta_i(A, y_i)$$

where y_i is chosen to be the least natural number not contained in the domain of β_j for $j < 2i$. We assure $A \not\leq_r B$ by defining β_{2i+1} for all i in ω to be an extension of β_{2i} where there is some natural number z_i such that, firstly, $u(\Delta_i, z_i)$ is contained in $dom(\beta_{2i+1})$ and, secondly, we have

$$\Delta_i(\langle \emptyset, \beta_{2i+1}\rangle, z_i) \; = \; 1 - A(z_i) \;\; ,$$

where we include the empty set as a dummy constant in order to obtain some valid set argument for the functional Δ_i. Such an extension always exists, because by assumption on A there is some set X to which A is not reducible, and hence by $\leq_r$ being c.f.v., A is not reducible to $\langle X, \beta_{2i} \rangle$. Furthermore, as the Δ_i are uniformly recursive, we can obtain such an extension uniformly effective in β_{2i} and i by a parallelized search trough all extensions of β_{2i}.

We show the last assertion in Proposition 60 by constructing a sequence $A_0, A_1, \ldots$ of recursive sets where for all i in ω and all finite sets J we have

$$i \notin J \text{ implies } A_i \not\leq_r \bigoplus_{j \in J} A_j \quad .$$

Then given a partial ordering $\leq_K$ on some countable domain $K = \{k_0, k_1, \ldots\}$, we map k_i to the join of the finitely many sets $\{A_j : k_j \leq_K k_i\}$. Here we exploit the fact that for a faithful relation the join of finitely many sets is a l.u.b. for the sets joined. The proof of this fact is an easy induction argument and can be found in [33] and [32]. The sets $A_0, A_1, \ldots$ are defined simultaneously by a finite extension construction similar to the construction of the set B in the proof of the third statement. We leave the details of the proof to the interested reader, because in the case of a standard reducibility $\leq_r$, the fourth statement in Proposition 60 will be a corollary to the more general results on partial order and lattice embeddings in Sect. 4.

4 Lattice Embeddings for Bounded Reducibilities

4.1 Gap Languages

In the following, we consider embeddings of countable partial orderings and of countable distributive lattices into proper intervals of the structure $(\mathrm{REC}, \leq_r)$ where $\leq_r$ is some bounded reducibility. Such embeddings have been constructed before for several specific bounded reducibilities. In corresponding proofs, the use of gap languages has become a standard technique: see the section on uniform diagonalization in [6], as well as Schöning [42, 43] and further corresponding references cited in Sect. 1.2. Recall from Definition 53 that a *gap language* is a subset of the natural numbers which is infinite and co-infinite. Recall further that a gap language A can be conceived as a partition of ω into infinitely many finite *blocks*, where each block corresponds to a maximal set of consecutive natural numbers which either all are in A or all are in the complement of A. Now, given a gap language G, we can number the blocks of G in the natural way, and we can assign to each x in ω the number of its block w.r.t. G. Definition 61 provides a formal account of this correspondence between gap languages and partitions of ω. For further use, we introduce these concepts for arbitrary partial characteristic functions, not just for gap languages.

Definition 61. Let $\alpha : \omega \to \{0, 1\}$ be a partial characteristic function. For all x in ω we let the BLOCK NUMBER $\mathbf{bn}(\boldsymbol{\alpha}, \boldsymbol{x})$ of x w.r.t. α be equal to

$$\mathrm{bn}(\alpha, x) \;:=\; |\{y \in \omega : y < x \;\&\; \alpha(y) \neq \alpha(y+1)\}| \;\;,$$

in case $\alpha(y)$ is defined for all $y \leq x$, and we let $\mathrm{bn}(\alpha, x)$ be undefined, otherwise. The set $\{x \in \omega : \mathrm{bn}(\alpha, x) = j\}$ is denoted as BLOCK $\boldsymbol{j}$ OF $\boldsymbol{\alpha}$ for all j in ω. We say, block j of α EXISTS iff block j of α is nonempty.

Observe that the blocks of a partial characteristic function α are "numbered" by the function $\mathrm{bn}(\alpha, .)$, starting with block number zero. Obviously, the function $\mathrm{bn}(\alpha, .)$ is total iff α is total, and α has infinitely many blocks iff α is a gap language.

Definition 62. Let A and B be gap languages. Then B is a GAP COVER for A iff every block of B contains some block of A.

It is immediate from Definition 62 that the gap cover relation is transitive, that is, if B is a gap cover for A and so is C for B, then also C is gap cover for A.

4.2 Density of $(\mathrm{REC}, \leq_r)$

Given an effective reduction cover $\Delta_0, \Delta_1, \ldots$ for some bounded reducibility $\leq_r$, then by definition of the notion reduction cover, a set E is $\leq_r$ - reducible to a set

F iff there is *some* reduction Δ_j where E is equal to $\Delta_j(F)$. As a consequence, we can construct sets E and F where E is not reducible to F by *diagonalizing* against all reductions Δ_j, that is, it is sufficient to ensure that for all j in ω there is some x_j in ω where we have

$$E(x_j) \neq \Delta_j(F, x_j) \quad . \tag{36}$$

Now, the functionals Δ_j are continuous, and thus for given j we can enforce (36) by specifying $E(x_j)$ and a finite part of F, that is, we are led to a finite extension construction. We employ such a construction in the proof of Lemma 65. Before, we introduce some notation.

Definition 63. Let G be a gap language and let E, E', F, and F' be sets.

1. The sets E and E' are G-similar, written $E \simeq^G E'$, iff E and E' agree on infinitely many blocks of G.
2. The pairs (E,F), and (E',F') are G-similar, written $(E,F) \simeq^G (E',F')$, iff we have for infinitely many blocks I of G

$$E \mid I \;=\; E' \mid I \qquad \text{and} \qquad F \mid I \;=\; F' \mid I \;\; .$$

For further use, we state the following easy fact which is immediate from the definitions of the concepts involved.

Proposition 64. *Let the gap language G be a gap cover for H. Then for all sets E, E', F, and F'*

$$\begin{array}{rcl} E \simeq^G E' & \textit{implies} & E \simeq^H E' \;\; , \\ (E,F) \simeq^G (E',F') & \textit{implies} & (E,F) \simeq^H (E',F') \;\; . \end{array}$$

Lemma 65 – Diagonalization lemma. *Let $\leq_r$ be a bounded reducibility such that $\leq_r$ is c.f.v. and let E and F be recursive sets where $E \not\leq_r F$. Then there is a recursive gap language G such that we have for all sets E' and F'*

$$(E,F) \simeq^G (E',F') \quad \textit{implies} \quad E' \not\leq_r F' \;\; .$$

We will use the diagonalization lemma in connection with results about partial order embeddings in order to ensure that the constructed embeddings preserve non-order. More precisely, by the diagonalization lemma, given sets E and F where E is not reducible to F, in order to construct a set E' which is not reducible to F', it is sufficient to ensure that E' and F' agree with E and F, respectively, on some set which contains infinitely many blocks of the gap language G we obtain from the diagonalization lemma.

Proof of lemma 65. Let $\{\Delta_0, \Delta_1, \ldots\}$ be some effective reduction cover for $\leq_r$, and assume $E \not\leq_r F$. We construct in stages a gap language G as required in the lemma. During stage s, we specify which numbers are in the block s of G. This then determines G by letting $G(0)$ be equal to 0.

We denote block s of G by I_s, and at stage 0, we let $I_0 = \{0\}$. At stage $s > 0$, for all $j \leq s$ and for all pairs α and β of finite partial characteristic functions with domain equal to the union of the blocks $I_0, \ldots, I_{s-1}$ we let $z_{\alpha,\beta,j}$ be the least number which satisfies

$$\langle E, \alpha \rangle (z_{\alpha,\beta,j}) \neq \Delta_j(\langle F, \beta \rangle, z_{\alpha,\beta,j}) \ . \tag{37}$$

Observe that there is always some number $z_{\alpha,\beta,j}$ which satisfies (37), because due to $\leq_r$ being c.f.v., $\langle E, \alpha \rangle = \Delta_j(\langle F, \beta \rangle)$ would imply $E \leq_r F$. Furthermore, the least such number can be found effectively in α, β, and j, because the Δ_j are uniformly recursive. Then, we choose I_s so large that for all such j, α, and β the set $\{z_{\alpha,\beta,j}\} \cup u(\Delta_j, z_{\alpha,\beta,j})$ is contained in the union of $I_0, \ldots, I_s$.

In order to show that the gap language G has the required properties, assume for a proof by contradiction that there are sets E' and F' where firstly, (E', F') and (E, F) are G-similar, and secondly, the set E' is $\leq_r$ - reducible to F', say via reduction Δ_k. Choose some $s \geq k$ where E' and F' agree with E and F, respectively, on block s of G. Let α and β be the restrictions of E' and F', respectively, to the union of the blocks $I_0, \ldots, I_{s-1}$. Now, the witness $z_{\alpha,\beta,k}$ for $\langle E, \alpha \rangle \neq \Delta_k(\langle B, \beta \rangle)$ we found during stage s of the construction of G witnesses $E' \neq \Delta_k(F')$, because $\langle E, \alpha \rangle$ and $\langle F, \beta \rangle$ agree with E' and F', respectively, on the relevant blocks $I_0, \ldots, I_s$ of G. □

We apply the diagonalization lemma in the proof of Theorem 68 in order to derive the density of the structure $(\mathrm{REC}, \leq_r)$ for all faithful bounded reducibilities which are c.f.v. and where $\mathcal{M}_r$ is a simulation class. We then show in subsequent sections that these assumptions in fact imply the embeddability of all countable partial orderings into any proper interval of $(\mathrm{REC}, \leq_r)$, and further that for standard reducibilities this result can be extended to embeddings of arbitrary countable distributive lattices. However, in the proof of this extension we use a more involved argument, and we prefer to show first separately the special case of density in order to demonstrate the methods used. In the proof of the density result, we use Lemma 66, the cover lemma, which is in fact a special case of the coding lemma stated as Lemma 79 below. For the sake of continuity of the current section, we postpone the proofs of the cover and coding lemma to Sect. 4.6.

Lemma 66 – Cover Lemma. *Let $\mathcal{S}$ be a simulation class and let G be a recursive gap language. Then $\mathcal{S}$ contains a gap cover of G.*

Further, we use the easy Lemma 67, which we state separately for further use.

Lemma 67. *Let the binary relation $\leq_r$ on 2^ω be faithful.*

1. *If B is not $\leq_r$ - reducible to A, then $A \oplus B$ is not $\leq_r$ - reducible to $A \oplus \emptyset$.*
2. *If A is $\leq_r$ - reducible to B, then for every set M in $\mathcal{M}_r$ we have*

$$A \ \leq_r \ \langle A \oplus \emptyset, A \oplus B \rangle^M \ \leq_r \ B \ .$$

Proof of Lemma 67. We show the first assertion by contraposition. If $A \oplus B$ is reducible to $A \oplus \emptyset$, we obtain by definition of faithfulness that $A \oplus B$ is reducible to A. But $A \oplus B$ is a locally transitive upper bound for set B, and hence also B is reducible to A.

Concerning the second assertion, assume $A \leq_r B$. Now, by faithfulness of $\leq_r$, $A \oplus \emptyset$ and $A \oplus B$ are both reducible to B, and then so is $X := \langle A \oplus \emptyset, A \oplus B \rangle^M$ for every set M in $\mathcal{M}_r$. Further, by definition of the join operator, for all sets M the set X thus defined can be written in the form $A \oplus H$ for some set H, and hence A is reducible to X.

Theorem 68. *Let $\leq_r$ be a faithful bounded reducibility which is c.f.v. and where $\mathcal{M}_r$ is a simulation class. Then the structure $(REC, \leq_r)$ is dense, that is, for all recursive sets A and B where $A <_r B$, there is some recursive set H such that we have*

$$A <_r H <_r B \ .$$

Note again that we show Theorem 68 just in order to demonstrate the methods used. In particular, Corollary 55 shows that the density of the structure $(\text{REC}, \leq_r)$ follows from weaker assumptions than are used here.

Proof of Theorem 68. Given sets A and B where $A <_r B$, we let the functional Π be defined by

$$\Pi(M) := \langle A \oplus B, A \oplus \emptyset \rangle^M \ . \tag{38}$$

Then by the second assertion in Lemma 67, Π maps every set M in $\mathcal{M}_r$ to some set in the closed interval between A and B. So we are done, if we can show that there is some M in $\mathcal{M}_r$ where we have $\Pi(M) \not\leq_r A$ and $B \not\leq_r \Pi(M)$. By the first assertion in Lemma 67, $A \oplus B$ is not reducible to $A \oplus \emptyset$. Thus, by applying the diagonalization lemma to these sets, we obtain a gap language G such that for all sets X and Y we have

$$(X, Y) \ \simeq^G \ (A \oplus B, A \oplus \emptyset) \quad \text{implies} \quad X \not\leq Y \ . \tag{39}$$

By letting Y be equal to $A \oplus \emptyset$ we obtain from (39)

$$X \ \simeq^G \ A \oplus B \text{ implies } X \not\leq A \oplus \emptyset, \text{ and hence implies } X \not\leq A \ .$$

Likewise, by letting X be equal to $A \oplus B$, we obtain

$$Y \ \simeq^G \ A \oplus \emptyset \text{ implies } A \oplus B \not\leq Y, \text{ and hence implies } B \not\leq Y \ .$$

Now, we are done, because by the cover lemma we can choose some gap cover M of G in $\mathcal{M}_r$, where by definition of Π the set $\Pi(M)$ is M-similar to $A \oplus \emptyset$ and to $A \oplus B$, and hence by Proposition 64, $\Pi(M)$ is also G-similar to these sets. □

4.3 Embeddings of countable partial orderings

We show next that the assumptions of Theorem 68 in fact imply that every countable p.o. can be embedded into every proper interval of $(\mathrm{REC}, \leq_r)$. In Sect. 4.4, we then show that for standard reducibilities all countable distributive lattices can be so embedded. In the proofs of these embedding results we exploit properties of the countable atomless Boolean algebra stated in Fact 70. The corresponding techniques were used before by Ambos-Spies [1] in connection with lattice embeddings for polynomial time bounded reducibilities; see there for references to results about Boolean algebras.

Definition 69. An element a of some Boolean algebra is an atom iff there is exactly one element (that is, the least element 0) strictly below a. A Boolean algebra is atomless iff it does not contain atoms.

Fact 70.
- *The theory of the atomless Boolean algebra is ω-categorical, that is, all countable atomless Boolean algebras are isomorphic.*
- *Every countable p.o. can be embedded (as a p.o.) into the countable atomless Boolean algebra. Every countable distributive lattice can be embedded (as a lattice) into the countable atomless Boolean algebra with least and greatest element preserved.*

Next, we exhibit a representation of the countable atomless Boolean algebra which is in some sense effective.

Definition 71.
- For a set A, let $[A]$ be the equivalence class of A w.r.t. to the equivalence relation $=^*$ on 2^ω, that is, we have $[A] = [B]$ for two sets A and B iff A is a finite variation of B.
- For a subclass $\mathcal{M}$ of 2^ω we let

$$\mathcal{M}^* := \{[A] : A \in \mathcal{M}\} .$$

- We let $\leq^*$ denote the p.o. induced on $(2^\omega)^*$ by the relation $\subseteq^*$, that is,

$$[A] \leq^* [B] \quad \text{iff} \quad A \subseteq^* B .$$

We will show in Proposition 74 that for a standard reducibility $\leq_r$ the "degree structure" induced on $\mathcal{M}_r$ by the relation $\subseteq^*$ is the atomless Boolean algebra.

Fact 72. *Let $(\mathcal{M}, \subseteq)$ be a subalgebra of $(2^\omega, \subseteq)$ which contains all finite sets. Then $(\mathcal{M}^*, \leq^*)$ is a Boolean algebra.*

Concerning Fact 72, observe that if the structure $(\mathcal{M}, \subseteq)$ is an upper semi-lattice which contains all finite sets, then $(\mathcal{M}^*, \leq^*)$ is in fact the quotient upper semi-lattice of $(\mathcal{M}, \subseteq)$ over the ideal of finite sets. Now, it is known from lattice theory that a quotient upper semi-lattice inherits the property of being a Boolean algebra from the lattice we start with. Using Fact 72, we derive Lemma 73, by which then in turns we infer that in fact for every standard reducibility $\leq_r$ the structure $({\mathcal{M}_r}^*, \leq^*)$ is the countable atomless Boolean algebra. The latter result extends a corresponding result in the setting of polynomial time shown in [1] and like there, the idea used in the proof of Lemma 73 for showing atomlessness is basically the same as for Breidbart's splitting theorem [14].

Lemma 73. *Let $\mathcal{M}$ be a simulation class which contains all finite and only recursive sets, and which is a subalgebra of $(2^\omega, \subseteq)$. Then $(\mathcal{M}^*, \leq^*)$ is the countable atomless Boolean algebra.*

Proof. By Fact 72, it remains to show that $(\mathcal{M}, \subseteq^*)$ is atomless. So choose some M in $\mathcal{M}$ where $\emptyset \subset^* M$. Then M is infinite and is by assumption recursive, and consequently there is some recursive gap language G such that each block of G contains at least one element of M. By Lemma 66 there is some gap cover C of G in $\mathcal{M}$. This finishes our proof, because the set $M \cap C$ is obviously in $\mathcal{M}$, and by choice of C we have $\emptyset \subset^* M \cap C \subset^* M$. □

Proposition 74. *For every standard reducibility $\leq_r$, the structure $({\mathcal{M}_r}^*, \leq^*)$ is the countable atomless Boolean algebra.*

Proof. The proposition follows by Lemma 73, and because due to the results of Sect. 3.3 for a standard reducibility $\leq_r$ the class $\mathcal{M}_r$ of admissible cases is a subalgebra of $(2^\omega, \subseteq)$, is a simulation class, and contains only recursive sets due to being contained in the class $\mathcal{L}_r$ of least sets.

By applying Propositions 38 and 74 to the reducibility $\leq_T^{\mathcal{P}}$, for which $\mathcal{M}_r$ coincides with the lower cone of the empty set and is hence recursively presentable, we obtain Fact 75. Observe that Fact 75 is essentially a reformulation of a result in [1], where it was shown that the structure $(\mathcal{P}^*, \leq^*)$ is the countable atomless Boolean algebra.

Fact 75. *There is a recursively presentable subclass $\mathcal{D}$ of 2^ω where $(\mathcal{D}^*, \leq^*)$ is the countable atomless Boolean algebra and such that $(\mathcal{D}, \subseteq)$ is a subalgebra of $(2^\omega, \subseteq)$, that is, $\mathcal{D}$ contains the empty set and is closed under union, intersection, and complement.*

Using the above mentioned facts about Boolean algebras, we show next that the assumptions of Theorem 68 imply not just the density of the the recursive sets, but in fact are sufficient to derive the embeddability of every countable p.o.

into every proper interval of the recursive sets. Observe in connection with Theorem 76 that there are reducibilities which satisfy the assumptions of the theorem, but which probably fail to be standard reducibilities. For example consider the variant of Turing reducibility which is restricted simultaneously to linear time and logarithmic space. The corresponding class of admissible cases contains all sets which are simultaneously computable in linear time and logarithmic space, and this latter class is in fact a simulation class as is implicitly shown by Schmidt [41], see the discussion in Sect. 3.7. Using this result, we obtain easily that the reducibility satisfies the assumptions of Theorem 76. However, it is not obvious to us whether the reducibility is in fact a standard reducibility, because it might lack an effective reduction cover which is closed under delayed patching. Observe that while the imposed resource bounds are sufficient to compute delayed simulations of arbitrary recursive functions, it is not clear whether this delayed simulations might be used to patch the oracle, because, intuitively speaking, looking up the computed values might require too much resources.

Theorem 76. *Let $\leq_r$ be a faithful bounded reducibility which is c.f.v. and where $\mathcal{M}_r$ is a simulation class. Let A and B be recursive sets where $A <_r B$. Then any countable p.o. can be embedded (as a p.o.) into the interval of $(REC, \leq_r)$ between A and B.*

In the proof of Theorem 76 we use the rather technical Lemma 77, which, for further use, we state separately and in a stronger form than required here.

Lemma 77. *Let $\leq_r$ be a binary relation on 2^ω and let A and B be recursive sets where*

- *the relation $\leq_r$ is faithful and c.f.v.,*
- *$\mathcal{M}_r$ is a simulation class,*
- *A is $\leq_r$ - reducible to B,*
- *there is some recursive gap language G where we have for all sets E and F*

$$(A \oplus B, A \oplus \emptyset) \simeq^G (E, F) \quad \textit{implies} \quad E \not\leq_r F \ .$$

Let the subclass $\mathcal{D}$ of 2^ω be recursively presentable and such that $(\mathcal{D}, \subseteq)$ is a subalgebra of $(2^\omega, \subseteq)$. Then the structure $(\mathcal{D}^*, \leq^*)$ can be embedded as an u.s.l. into the interval of $(REC, \leq_r)$ between A and B.

Proof of Theorem 76. Using Lemmas 65 and 67, we infer that the assumptions of Theorem 76 imply the assumptions of Lemma 77 below. Thus by choosing $\mathcal{D}$ according to Fact 75, we can embed the countable atomless Boolean algebra as a p.o. into the interval of (REC, $\leq_r$) between A and B. Now, we are done, because firstly, by Fact 70 every countable p.o. can be embedded into the countable atomless Boolean algebra, and secondly, the composition of two p.o. embeddings is again a p.o. embedding.

In the proof of Lemma 77 we will use, firstly, an extension of the usual concept of degree to not necessarily transitive reducibilities which is discussed in Remark 78, and, secondly, Lemma 79, the coding lemma.

Remark 78. *Recall from the introduction that two sets A and B are $\leq_r$ - EQUIVALENT, written $A \equiv_r B$, iff their respective upper and lower $\leq_r$ - cones are identical, and that the relation $\equiv_r$ is by definition an equivalence relation on 2^ω, that is, it is reflexive, transitive, and symmetric. Let $\leq_r$ be a binary relation on 2^ω and let A be some set. We denote the equivalence class*

$$deg_r(A) := \{B \subseteq \omega : A \equiv_r B\}$$

of A w.r.t. the equivalence relation $\equiv_r$ as $\leq_r$ - degree of A.

In connection with reducibilities which are reflexive and transitive, degrees are usually defined in terms of interreducibility, that is, sets A and B are in the same degree iff A is reducible to B and vice versa. This is justified, because for a p.p.o. $\leq_r$ the corresponding notion of interreducibility is an equivalence relation. In fact, as we have mentioned in the introduction, equivalence and interreducibility coincide for a p.p.o., and consequently our definition of degrees in terms of equivalence yields just the usual concept in case $\leq_r$ is a p.p.o.

Observe that in connection with specific resource bounded reducibilities which are reflexive and transitive, results about the structure induced on the recursive sets are usually formulated in terms of the corresponding degree structures, that is, for example results about embeddings are usually formulated in terms of embeddings into the degree structure. In the sequel, however, we refrain from formulating results in terms of degrees, as we feel that such a formulation does not really simplify the picture in the case of a not necessarily transitive relation $\leq_r$.

Lemma 79, the coding lemma, is a generalization of Lemma 66, the cover lemma; as for the latter, the proof of Lemma 79 is postponed to Sect. 4.6.

Lemma 79 – Coding lemma. *Let $A_0, A_1, \ldots$ be a sequence of uniformly recursive sets and let G be a gap language. Let $\mathcal{M}$ be a simulation class which contains all finite sets and where the structure $(\mathcal{M}, \subseteq)$ is a subalgebra of $(2^\omega, \subseteq)$. Then there are sets $R_0, R_1, \ldots$ in $\mathcal{M}$ and a gap language M in $\mathcal{M}$ such that*

- *The set M is a gap cover for G.*
- *For all i and s in ω, and for all x in block s of M we have $R_i(x) = A_i(s)$.*

The point of the coding lemma is that it yields delayed simulations R_i of the sets A_i which are "synchronized" via the gap language M, that is, for all i and s the set R_i is constant on the s-th block of M and has the value $A_i(s)$ there. Observe that the sets R_i in the coding lemma are uniformly recursive, because the sets A_i are, and due to the second condition in the conclusion of the coding lemma.

Proof of Lemma 77. Assume $\mathcal{D} = \{D_0, D_1, \ldots\}$ and for a set D in $\mathcal{D}$ let $[D]$ be the equivalence class of D under the finite variation relation. We apply the coding lemma to the gap language G and the sets in $\mathcal{D}$ in order to obtain a gap language M in $\mathcal{M}_r$ and sets R_i in $\mathcal{M}_r$. We define a function from $\mathcal{D}$ to 2^ω by

$$\begin{array}{rl} \Pi_0 : \ \mathcal{D} & \to 2^\omega \\ D_i & \mapsto \langle A \oplus B, A \oplus \emptyset \rangle^{R_i} \end{array} .$$

We let Π_1 be a function which maps each equivalence class $[D_i]$ to one of its elements, that is, we have

$$\Pi_1([D_i]) \text{ is in } [D_i] \qquad \text{for all } D_i \text{ in } \mathcal{D} \ .$$

Finally, we let

$$\Pi \ := \ \Pi_0 \circ \Pi_1 \ ,$$

that is, intuitively speaking, Π maps an equivalence classes first to some of its elements, and then applies the function Π_0. We will show that Π is the embedding of $(\mathcal{D}^*, \leq^*)$ we are looking for.

Claim 1. The function Π is well-defined.

Proof. Claim 1 is immediate by definition of Π.

Claim 2. For all i and j we have

$$D_i \subseteq^* D_j \quad \text{iff} \quad R_i \subseteq^* R_j \ .$$

Proof. Claim 2 is immediate by choice of the sets R_i and R_j and by the second condition in the conclusion of Lemma 79.

Claim 3. For all i in ω, the set $\Pi_0(D_i)$ is a finite variation of $\Pi([D_i])$, and in particular the two sets are $\leq_r$ - equivalent.

Proof. Assuming that $\Pi_1([D_i])$ is equal to D_j for some j, we obtain by definition of Π_1 that D_i and D_j are finite variations of each other, and then so are R_i and R_j. Claim 3 then follows by the definitions of Π and Π_0.

Claim 4. For every set H in the range of Π we have $A \leq_r H \leq_r B$.

Proof. Immediate from Lemma 67 and the definition of Π, because the sets R_i are all in $\mathcal{M}_r$.

In the following claims, sets in the image of Π figure in such a way that the truth values of the assertions under considerations are not changed by replacing these sets with $\leq_r$ - equivalent sets. As a consequence, by Claim 3, in the corresponding proofs we can replace the images under Π with the corresponding images under Π_0.

Claim 5. The function Π respects ordering.

Proof. Assume $[D_i] \leq^* [D_j]$, that is, $D_i \subseteq^* D_j$. By Claim 2, we obtain $R_i \subseteq^* R_j$. By definition of Π_0, we have that the set $\Pi_0(D_j)$ is equal to $A \oplus B$ for all places x in R_j, and hence for almost all places x in R_i. Thus we obtain

$$\Pi_0(D_i) := \langle A \oplus B, A \oplus \emptyset \rangle^{R_i} =^* \langle \Pi_0(D_j), A \oplus \emptyset \rangle^{R_i} \leq_r \Pi_0(D_j) \quad , \tag{40}$$

from which $\Pi([D_i]) \leq_r \Pi([D_j])$ follows by Claim 3 and because $\leq_r$ is c.f.v. The relations in (40) hold, from left to right, by definition of Π_0, by the discussion preceding (40), and finally because R_i is in $\mathcal{M}_r$ and because by Claim 4 the set A and hence by faithfulness also $A \oplus \emptyset$ are $\leq_r$ - reducible to $\Pi_0(D_j)$.

Claim 6. The function Π respects non-ordering.

Proof. Assuming $[D_i] \not\leq^* [D_j]$, that is, $D_i \not\subseteq^* D_j$, we obtain $R_i \not\subseteq^* R_j$ by Claim 2. Thus there are infinitely many numbers, and hence, because the sets R_i and R_j are constant on the blocks of M, there are in fact infinitely many blocks of M where R_i is equal to 1, while R_j is equal to 0. Then, by definition of Π_0, on each such block of M, $\Pi_0(D_i)$ is equal to $A \oplus B$, and $\Pi_0(D_j)$ is equal to $A \oplus \emptyset$, and we obtain

$$(A \oplus B, A \oplus \emptyset) \ \simeq^M \ (\Pi_0(D_i), \Pi_0(D_j)) \quad . \tag{41}$$

Now we have chosen M as a gap cover for the gap language G from the assumption of Lemma 77, that is, each block of M contains some block of G. As a consequence, (41) remains valid with M replaced by G. Thus, by assumption on G, the set $\Pi_0(D_i)$ is not reducible to $\Pi_0(D_j)$, and the same holds for the corresponding images under Π according to Claim 3.

So we have shown that Π embeds $(\mathcal{D}^*, \leq^*)$ as a p.o. into the given interval. In order to obtain the full statement of Lemma 77, it remains to show that Π also respects least upper bounds.

Claim 7. The function Π respects least upper bounds.

Proof. The l.u.b. of $[D_i]$ and $[D_j]$ in $(\mathcal{D}^*, \leq^*)$ is $[D_i \cup D_j]$. By Claim 5, the set $\Pi_0(D_i \cup D_j)$ is an upper bound for $\Pi_0(D_i)$ and $\Pi_0(D_j)$, and so it remains to show that if the two latter sets are both reducible to some set Y, then so is $\Pi_0(D_i \cup D_j)$. But this follows from

$$\begin{aligned} \Pi_0(D_i \cup D_j) &= \langle A \oplus B, A \oplus \emptyset \rangle^{R_i \cup R_j} = \langle A \oplus B, \langle A \oplus B, A \oplus \emptyset \rangle^{R_j} \rangle^{R_i} \\ &= \langle A \oplus B, \Pi_0(D_j) \rangle^{R_i} \quad = \langle \Pi_0(D_i), \Pi_0(D_j) \rangle^{R_i} \leq_r Y \end{aligned} \quad ,$$

where the relations hold by definition of Π_0 and the choice of the sets R_k, by the properties of definition by number-dependent cases, by definition of $\Pi_0(D_j)$, because $\Pi_0(D_i)$ agrees with $A \oplus B$ on all numbers in R_i, and finally by assumption on Y and because R_i is in $\mathcal{M}_r$. □

4.4 Embeddings of countable distributive lattices

We show next that for standard reducibilities the embedding results of Sect. 4.3 can be extended to lattice embeddings.

Theorem 80. *Let $\leq_r$ be a standard reducibility and let A and B be recursive sets where $A <_r B$. Then any countable distributive lattice can be embedded (as a lattice) into the interval between A and B of $(REC, \leq_r)$ with least or greatest element preserved.*

In addition, given recursively presentable subclasses $\mathcal{E}_0$ and $\mathcal{E}_1$ of 2^ω which are c.f.v. and where $\mathcal{E}_0$ does not contain $A \oplus \emptyset$ and $\mathcal{E}_1$ does not contain $A \oplus B$, then the range of the embedding can be chosen to be disjoint from the union of $\mathcal{E}_0$ and $\mathcal{E}_1$, except that in case we want to preserve the minimum, the minimal element might be mapped to an element of $\mathcal{E}_1$, and likewise for the maximum and $\mathcal{E}_0$.

The proof of Theorem 80 follows the lines of the proof of the corresponding results for polynomial time bounded reducibilities in [1]. The proof amounts to construct an embedding of the countable atomless Boolean algebra into the given interval, which is sufficient because by Fact 70 every countable distributive lattice can be embedded in turn into the countable atomless Boolean algebra.

Proof. We first construct an embedding of the countable atomless Boolean algebra which preserves the least element, and afterwards we indicate the minor changes necessary in the symmetric case where we want to preserve the greatest element. The embedding we construct is essentially the same as in the proof of Lemma 77, however, we have to show that we can choose a gap language G and some class $\mathcal{D}$ where $(\mathcal{D}^*, \leq^*)$ is the countable atomless Boolean algebra such that the resulting embedding of $(\mathcal{D}^*, \leq^*)$ is not just an u.s.l. embedding, but is in fact a lattice embedding, that is, preserves also greatest lower bounds.

- We choose the subclass $\mathcal{D}_0$ of 2^ω according to Fact 75, and we let

  $$\mathcal{D} := \{\{3\,x : x \in D\} : D \in \mathcal{D}_0\} \ ,$$

  that is, the sets in $\mathcal{D}$ contain only multiples of three. We leave the routine task to the reader, to show that $\mathcal{D}$ inherits the following properties of $\mathcal{D}_0$:
  - $\mathcal{D}$ is recursively presentable,
  - $(\mathcal{D}^*, \leq^*)$ is the countable atomless Boolean algebra,
  - $(\mathcal{D}, \subseteq)$ is a sublattice of $(2^\omega, \subseteq)$.

  Observe in connection with the last item that $(\mathcal{D}, \subseteq)$ is not a subalgebra of $(2^\omega, \subseteq)$, because complementation in the former structure differs from complementation in the latter. Observe further that in the proof of Lemma 77 we use only the properties of $\mathcal{D}$ listed above.

- The recursive gap language G is a gap cover for three recursive gap languages G_0, G_1, and G_2 where
  - we define the gap language G_0 below,
  - we choose the gap language G_1 according to the diagonalization lemma such that for all sets E and F

$$(A \oplus B, A \oplus \emptyset) \simeq^{G_1} (E, F) \quad \text{implies} \quad E \not\leq_r F \ ,$$

  - we choose the recursive gap language G_2 such that, firstly, every set which is G_2-similar to $A \oplus \emptyset$ is not contained in $\mathcal{E}_0$ and, secondly, every set which is G_2-similar to $A \oplus B$ is not contained in $\mathcal{E}_1$. In order to do so, we choose block k of G_2 so large that both of $A \oplus \emptyset$ and $A \oplus B$ disagree on this block with the first k sets in $\mathcal{E}_0$ and $\mathcal{E}_1$, respectively.

Using G and $\mathcal{D}$, we define the embedding Π as in the proof of Lemma 77, and like there we infer that Π is an an upper semi-lattice embedding of $(\mathcal{D}^*, \leq^*)$ into the interval between A and B. Observe that Π preserves the least element, because it maps the least equivalence class in $\mathcal{D}^*$, which contains all finite sets, to a set which is $\leq_r$ - equivalent to $A \oplus \emptyset$. Further, all equivalence classes in $\mathcal{D}^*$ which are strictly above the least class contain only sets which are infinite and co-infinite, and thus the assertion about avoiding $\mathcal{E}_0$ and $\mathcal{E}_1$ is immediate from the definition of Π and by choice of G as a gap cover of G_2. Now, $(\mathcal{D}^*, \leq^*)$ is the countable atomless Boolean algebra, into which, according to Fact 70, every countable distributive lattice can be embedded with least and greatest element preserved. So we are done, if we can show that we can choose some recursive gap language G_0 such that the embedding Π in fact respects meets.

In the remainder of this proof, we use the expression "every reduction to a set in $\mathcal{C}$ is witnessed by some $\leq_r$ - reduction where ... " to express that for all sets C in $\mathcal{C}$ and all sets E which are reducible to C, there is an $\leq_r$ - reduction Δ where E is equal to $\Delta(C)$ and such that Δ has the properties in question. We fix some effective reduction cover $\mathcal{R}$ for $\leq_r$ which is closed under delayed patching.

Claim 1. There is a recursive function h such that for each Δ in $\mathcal{R}$ and for almost all places x we have

$$\max\ u(\Delta, x) \ \leq\ h(x) \ .$$

Proof. Let $\mathcal{R}$ be equal to $\{\Delta_0, \Delta_1, \ldots\}$ where the reductions Δ_k are uniformly recursive. Then $u(\Delta_k, x)$ is finite for all k and x and can be computed effectively from k and x, and thus it is sufficient to let

$$h(x) \ = \ \max_{i \leq x} u(\Delta_i, x) \ .$$

Claim 2. Let the class $\mathcal{C}$ be recursively presentable. Then there is an recursive function b which is nondecreasing and unbounded and such that every reduction to some set C in $\mathcal{C}$ is witnessed by some Δ in $\mathcal{R}$ where the generalized use of Δ is almost everywhere bounded from below by b, that is, we have for almost x

$$b(x) \ \leq \ \min u(\Delta, x) \ .$$

Proof. By Proposition 43, the reduction cover $\mathcal{R}$ is closed under oracle simulation, so we fix some modulus of oracle simulation m which witnesses this fact. We let r be some recursive function where $\mathcal{C}$ is equal to $\{\varphi_{r(i)} : i \in \omega\}$. For all i and x in ω, we let

$$v(i,x) := \max_{y \leq x} \ \left[\{0,\ldots,y\} \subseteq dom(\sigma_{m(r(i),x)})\right] \ .$$

The function v is recursive, because m and r are. Further, for each fixed i in ω, the function $v(i,.)$ is nondecreasing and unbounded, because $\varphi_{r(i)}$ is total and hence the domains of the partial characteristic functions $\sigma_{m(r(i),x)}$ converge nondecreasingly to ω as x goes to infinity. Next, we let

$$b(x+1) := \begin{cases} b(x)+1 & \text{in case } b(x)+1 \leq v(i,x) \text{ for all } i \leq b(x) \\ b(x) & \text{otherwise} \end{cases} \ .$$

The function b is by definition nondecreasing, and due to the functions $v(i,.)$ being unbounded, we obtain by an easy induction argument that b is unbounded, too. Then, given a set $C = \varphi_{r(i)}$ in $\mathcal{C}$ and some set which is reducible to C, we fix some witnessing reduction Δ in $\mathcal{R}$. The functional

$$\Delta' := \Delta \otimes m(r(i),.) \ ,$$

the $m(r(i),.)$-patch of Δ, is in $\mathcal{R}$ by choice of the modulus m. The functionals Δ and Δ' agree on the set C, because the patches $\sigma_{m(r(i),x)}$ agree with C on their entire domain, and hence we have for all x

$$\Delta'(C,x) \ = \ [\Delta \otimes m(r(i),.)](C,x) \ = \ \Delta(\langle C, \sigma_{m(r(i),x)}\rangle, x) \ = \ \Delta(C,x) \ .$$

Furthermore, by definition of b, for almost all x the numbers less or equal to $b(x)$ are all contained in the domain of $\sigma_{m(r(i),x)}$, and hence are not in the generalized use $u(\Delta', x)$.

Given a gap language G and some x in ω we let

$$Nb(x,G) := \{y \in \omega : \mathrm{bn}(G,x) - 1 \leq \mathrm{bn}(G,y) \leq \mathrm{bn}(G,x)+1\} \ ,$$

that is, $Nb(x,G)$ consists of the block of x w.r.t. G, and of the adjacent block to the left and to the right, respectively.

Claim 3. Let the class $\mathcal{C}$ be recursively presentable. Then there is a recursive gap language G such that every $\leq_r$ - reduction to some set in $\mathcal{C}$ is witnessed by a reduction Δ in $\mathcal{R}$ where we have for all x in ω

$$u(\Delta, x) \subseteq Nb(x, G) .$$

The point of Claim 3 is that every reduction to a set in $\mathcal{C}$ is witnessed by a reduction where for almost all places x the generalized use of Δ is bounded by the "window" $Nb(x, G)$.

Proof of Claim 3. By Claims 1 and 2, we obtain recursive functions h and b, where b is nondecreasing and unbounded and such that every reduction to some set in $\mathcal{C}$ is witnessed by some functional Δ in $\mathcal{R}$ where for almost places x the generalized use $u(\Delta, x)$ is a subset of the possibly empty interval between $b(x)$ and $h(x)$. Thus it is sufficient to construct a gap language G such that for all x the interval between $b(x)$ and $h(x)$ is contained in $Nb(x, G)$. In order to do so, we define the blocks of G inductively. We let block 0 of G be equal to $\{0\}$, and assuming that the blocks 0 to s of G are already defined where their union is U_s, we let m_s be the least number not contained in U_s where we have

$$max\ U_s < b(m_s) \quad \text{and} \quad max_{y \in U_s} h(y) \leq m_s .$$

Then, we let block $s+1$ of G be equal to $\{0, \ldots, m_s\} - U_s$.

The next claim shows that in the conclusion of Claim 3 we can replace G by any gap cover of G.

Claim 4. Let M be a gap cover for the gap language G. Then holds for all x

$$Nb(x, G) \subseteq Nb(x, M) .$$

Proof. We fix some x and assume that x is in block i of G, and in block j of M. Then block $j-1$ of M cannot contain block i of G, because the latter contains x which is in block j of M. Thus, by M being a gap cover for G, block $j-1$ of M must contain some block k of G where $k \leq i-1$. But then, obviously the union of the consecutive blocks $j-1$ and j of M contains block $i-1$ of G. By symmetry, the blocks j and $j+1$ of M contain block $i+1$ of G, from which Claim 4 then is immediate.

We define a recursively presentable class $\mathcal{C}$ to which subsequently we will apply Claim 3.

$$\mathcal{C} := \{\langle A \oplus B, A \oplus \emptyset \rangle^L : L \in \mathcal{L}_r\} .$$

The point of this definition is that the sets $\Pi_0(D_i)$ are all in $\mathcal{C}$, because the sets R_i we use in the definition of Π_0 are all in $\mathcal{M}_r$, and hence, by Proposition 38, are in $\mathcal{L}_r$. We apply Claim 3 to the class $\mathcal{C}$ and obtain a recursive gap language G_0

which we use during the definition of the embedding Π in the way described at the beginning of this proof. We now argue that Π in fact is a lattice embedding of the structure $(\mathcal{D}^*, \leq^*)$. According to the discussion at the beginning of this proof Π is an upper semi-lattice embedding, and so it remains to show that Π respects greatest lower bounds.

Claim 5. The function Π respects greatest lower bounds.

Proof. We fix sets D_i and D_j in $\mathcal{D}$. The g.l.b. of $[D_i]$ and $[D_j]$ in $(\mathcal{D}^*, \leq^*)$ is $[D_i \cap D_j]$, and so we have to show that $\Pi([D_i \cap D_j])$ is the g.l.b. of $\Pi([D_i])$ and $\Pi([D_j])$. We already know that $\Pi([D_i \cap D_j])$ is a lower bound for the two latter sets, because Π respects ordering. So it remains to show that if a set X is reducible to both of $\Pi_0(D_i)$ and $\Pi_0(D_j)$, then it is also reducible to $\Pi_0(D_i \cap D_j)$. Here Π_0 is the function from the proof of Lemma 77 where for every set in $\mathcal{D}$ the sets $\Pi_0(D)$ and $\Pi([D])$ are $\leq_r$ - equivalent. Recall that we have chosen the class $\mathcal{C}$ such that the sets $\Pi_0(D_i)$ and $\Pi_0(D_j)$ are in $\mathcal{C}$. Thus by choice of the gap language G as a gap cover for the gap language G_0 obtained by applying Claim 3 to $\mathcal{C}$ we obtain reductions Δ_0 and Δ_1 in $\mathcal{R}$ where we have

$$X \; = \; \Delta_0(\Pi_0(D_i)) \; = \; \Delta_1(\Pi_0(D_j))$$

and where for almost all places x the generalized use of Δ_0 and of Δ_1 are both contained in $Nb(x, G)$. By Claim 4, the latter holds also for the gap cover M of G which we have obtained by applying the coding lemma. For every m in ω, we let $N(m)$ be equal to the set $\{m-1, m, m+1\}$ and further

$$\begin{array}{lllll} I & := & I_{(i,j)} & := & \{m \in \omega : D_i \text{ and } D_i \cap D_j \text{ agree on } N(m)\} \\ L & := & L_{(i,j)} & := & \{x \in \omega : x \text{ is in block } k \text{ of } M \text{ where } k \text{ is in } I_{(i,j)}\} \end{array} .$$

Then given some x in L, we infer from the definition of Π_0 that $\Pi_0(D_i)$ and $\Pi_0(D_i \cap D_j)$ agree on $Nb(x, M)$, and thus by assumption on Δ_0 we obtain that

$$x \in L \quad \text{implies} \quad \Delta_0(\Pi_0(D_i \cap D_j), x) \; = \; \Delta_0(\Pi_0(D_i), x) \; = \; X(x) \; . \tag{42}$$

On the other hand, given some x not in L, where we assume that x is in block k of M, then D_i and $D_i \cap D_j$ differ on $N(k)$, and hence there is some number in $N(k)$ which is in D_i, but is not in D_j. Now, $N(k)$ contains exactly one multiple of three, and thus, as all numbers in D_i and D_j are multiples of three, the intersection of $N(k)$ with D_j is empty. As a consequence, $\Pi_0(D_j)$ and $\Pi_0(D_i \cap D_j)$ have an empty intersection with $Nb(x, M)$, and as in the case of (42), we infer

$$x \notin L \quad \text{implies} \quad \Delta_1(\Pi_0(D_i \cap D_j), x) \; = \; \Delta_1(\Pi_0(D_j), x) \; = \; X(x) \; . \tag{43}$$

From (42) and (43) we then obtain

$$X \; = \; \langle \Delta_0(\Pi_0(D_i \cap D_j)), \Delta_1(\Pi_0(D_i \cap D_j)) \rangle^L \; .$$

This finishes the proof of Claim 5, because we can assume that L is in $\mathcal{M}_r$. More precisely, if we apply the coding lemma not simply to the class $\mathcal{D}$, but to

$$\mathcal{D} \cup \{I_{(i,j)} : i, j \in \omega\} ,$$

where however for defining the function Π_0 we only use the sets R_k which correspond to elements in $\mathcal{D}$, then the sets R_k will comprise all sets $L_{(i,j)}$.

Observe that Π_0 maps the empty set to $A \oplus \emptyset$ and hence the embedding Π preserves the least element. If we want to construct an embedding which preserves the greatest element, in the construction we choose $\mathcal{D}_0$ as before, but replace $\mathcal{D}$ with

$$\mathcal{D}' := \{\{3\,x : x \in D\} \cup \{3\,x+1 : x \in D\} \cup \{3\,x+2 : x \in D\} : D \in \mathcal{D}_0\} ,$$

that is, intuitively speaking, for the sets in the image of the constructed embedding the gaps are filled with the set $A \oplus B$ instead of the set $A \oplus \emptyset$. □

We conclude this section by stating some easy consequences of Theorem 80, which correspond to corollaries obtained in [1] for the polynomial time case .

Corollary 81. *Let $\leq_r$ be a standard reducibility. Let A and B be recursive where B is not in $\mathcal{L}_r$.*

- *The set A is meet-reducible, that is, A is the greatest lower bound of some pair of recursive sets.*
- *The set B is join-reducible, that is, B is the least upper bound of some pair of recursive sets.*
- *If $\leq_r$ is transitive, then B bounds some minimal pair, that is, there are two sets which are both reducible to B such that the intersection of their lower cones is exactly the class of least sets.*
- *Assuming $A <_r B$, every recursively presentable anti-chain where all elements are strictly contained in the interval between A and B is not maximal with this property.*

Proof. The first assertions is immediate by embedding the diamond, that is, the four-element Boolean algebra, above A with least element preserved. Likewise, the second and third assertion follow by embedding the diamond into the interval between $\emptyset$ and B with greatest and least element preserved, respectively. In connection with the third assertion observe that by definition the lower cone of a greatest lower bound of two sets is equal to the intersection of the lower cones of these two sets, and that for a transitive relation the lower cone of a least set is just the class of least sets. Concerning the last assertion, given some recursively representable anti-chain $\{C_0, C_1, \ldots\}$ where we have $A <_r C_i <_r B$ for all i in ω, in order to extend the anti-chain by some set strictly between A and B, we embed the three-element chain between A and B while avoiding, according to Theorem 80, the classes

$$\mathcal{E}_0 := \bigcup_{i\in\omega}\{X \subseteq \omega : C_i \leq_r X \leq_r B\} \text{ and } \mathcal{E}_1 := \bigcup_{i\in\omega}\{X \subseteq \omega : X \leq_r C_i\} \ .$$

4.5 Lattice Embeddings for Bounded Reducibilities on ω^ω

The result about lattice embeddings for standard reducibilities on 2^ω extends to standard reducibilities on ω^ω as introduced in Sect. 2.6.

Theorem 82. *Let $\leq_r$ be a standard reducibility on ω^ω and let f and g be recursive functions where $f <_r g$. Then any countable distributive lattice can be embedded (as a lattice) into the interval between f and g of $(REC, \leq_r)$ with least or greatest element preserved.*

In addition, given a recursively presentable subclasses $\mathcal{E}_0$ and $\mathcal{E}_1$ which are c.f.v. and where $\mathcal{E}_0$ does not contain $f \oplus \emptyset$ and $\mathcal{E}_1$ does not contain $f \oplus g$, then the embedding can be chosen disjoint from the union of $\mathcal{E}_0$ and $\mathcal{E}_1$, except that in case we want to preserve the minimum, the minimal element might be mapped to an element of $\mathcal{E}_1$, and likewise for the maximum and $\mathcal{E}_0$.

We omit the proof of Theorem 82, which is is basically identical with the proof of Theorem 80, except that we have to take into account that ω^ω is not compact. More precisely, we consider variants of the Diagonalization Lemma and of Claim 3 in the proof of Theorem 80 which are relativized to some effectively compact subclass of ω^ω.

Lemma 83 – Diagonalization lemma for reducibilities on ω^ω . *Let $\leq_r$ be a bounded reducibility on ω^ω which is c.f.v., let e and f be recursive functions where $e \not\leq_r f$, and let $\mathcal{C}$ be some effectively compact subclass of ω^ω. Then there is a recursive gap language G such that we have for all functions e' and f' in $\mathcal{C}$*

$$(e, f) \simeq^G (e', f') \ \textit{implies} \ e' \not\leq_r f' \ .$$

Claim 3. Let the class $\mathcal{C}_0$ be recursively presentable such that $\mathcal{C}_0$ is contained in some effectively compact class $\mathcal{C}$. Then there is a recursive gap language G such that every $\leq_r$ - reduction to a function in $\mathcal{C}_0$ can be replaced by an $\leq_r$ - reduction Δ where $u^{\mathcal{C}}(\Delta, x)$ is contained in $Nb(x, G)$.

Lemma 83 and the adjusted version of Claim 3 can be shown in exactly the same way as the original claims. The remainder of the proof then goes through by applying these adaptations to the effectively compact class

$$\mathcal{C} := \bigotimes_{i\in\omega} \{[f \oplus \emptyset](i), [f \oplus g](i)\} \ ,$$

which in particular contains all functions obtained from $f \oplus \emptyset$ and $f \oplus g$ via definition by cases with sets in $\mathcal{M}_r$, and thus contains all functions in the range of the embedding Π_0.

4.6 The Proofs of the Cover and the Coding Lemma

In this section we give the still missing proofs of the cover lemma and the coding lemma, which we have stated as Lemmas 66 and 79. In both proofs we use the recursion theorem in a form described in Remark 84.

Remark 84. *If we specify a procedure which computes uniformly effectively in some given index e a partial function γ_e, then, by the smn-theorem, there is some recursive function g where $\varphi_{g(e)}$ is equal to γ_e for all e in ω. According to the recursion theorem, there is some index e_0 denoted as fixed point of g, where we have*

$$\varphi_{e_0} \;=\; \varphi_{g(e_0)} = \gamma_{e_0} \;.$$

Observe that such an index e_0, in general, will not be a fixed point of g in the usual sense $g(e_0) = e_0$. Now, if we can ensure certain properties of the constructed partial function γ_e under the assumption that the given index e is an index for γ_e, we succeed in constructing a partial function with these properties, because for the fixed point e_0 our assumption will be true.

In recursion theory, the following, more convenient form of this technique is widely used: instead of viewing the index e as an argument, we assume that already during the specification of some partial recursive function γ, an index for γ is available, that is, there is some e where $\gamma = \varphi_e$ holds, and we can use e in the definition of γ. After introducing this form of the technique in the section on the recursion theorem in [47], Soare points out that in the definition of γ, we must not rely on e being an index for the function under construction. The point of his remark is that, while we are only interested in the cases where the number e used in our construction is indeed an index for the function under construction, in order to render the construction valid, we have to ensure that the constructed partial function γ is in fact partial recursive in e.

Subsequently, we will apply this technique by giving some effective procedure which enumerates the graph of some partial function γ, while using in the construction an index for γ. In order to address the problem mentioned in the last paragraph, we apply the following convention:

- *In case the partial function γ is eventually defined during the construction at some place x, then $\gamma(x)$ is equal to the value which is assigned first; otherwise, $\gamma(x)$ is undefined.*

It should be clear that by this convention there is always some recursive function g such that for all e, $\varphi_{g(e)}$ is equal to the function γ which we specify under the possibly wrong assumption that e is an index for the function under construction. For example we can choose a function g which assigns to e, intuitively speaking, a Turing machine $T_{g(e)}$ which computes $\gamma(x)$ by simulating the construction of γ on the given index e until x enters the domain of γ. Then, if for some specific *index e, the construction of γ "gets stuck", this simply means that the domain of $\gamma = \varphi_{g(e)}$ is finite and contains only the places which have been assigned values so far.*

In accordance with the fact that the density result stated in Theorem 68 is a special case of the result about p.o. embeddings stated in Theorem 76, the cover lemma shown next is a special case of the coding lemma. As for the embedding results, we show the less involved cover lemma separately in order to demonstrate the techniques used. In particular, the notation and the subroutines used in the proof of the cover lemma partially are tailored to be reused in the more involved proof of the coding lemma.

Lemma 66 - Cover Lemma. *Let $\mathcal{S}$ be a simulation class and let G be a recursive gap language. Then $\mathcal{S}$ contains a gap cover of G.*

Proof. Figure 1 shows an effective procedure which enumerates the graph of some partial characteristic function γ. By Remark 84, we can assume that an index e for the function under construction is available already during the construction, that is, we have $\gamma = \varphi_e$ and the specification of γ might depend on e. Now, $\mathcal{S}$ is a simulation class, and consequently, by Proposition 48, we can choose some recursive function sim where $\varphi_{sim(e)}$ is a set in $\mathcal{S}$ for all e in ω, and is in addition a delayed simulation of φ_e whenever $\varphi_e(0) = 0$ holds. Therefore, by letting $\gamma(0) = 0$, during the construction of γ we do not only have available some index e for γ, but also the delayed simulation $M{:=}\varphi_{sim(e)}$ of γ in $\mathcal{S}$. In the verification of the construction we then show that the set M in fact is a gap cover for G.

We first give an outline of the construction and sketch the ideas on which its verification is based. The course-of-values of γ is rather simple: if γ is defined at all at some place s, then it is equal to $\omega \oplus \emptyset$, that is, $\gamma(s)$ is 0 in case s is even, and $\gamma(s)$ is 1, otherwise. The actual load of the construction is to decide successively for $s = 1, 2, \ldots$, whether $\gamma(s)$ is to be defined or not. Thus, during the construction γ can always be written as

$$\gamma{:=}(\omega \oplus \emptyset) \mid \{0, \ldots, s\} \ , \tag{44}$$

where $s{:=}\max dom(\gamma)$ is in block s of γ (recall that we start counting blocks with number zero). In the situation of (44), informally we denote block s of M as the open block of the construction. Now, by choice of sim and by definition of M, the set $M = \varphi_{sim(e)}$ is a delayed simulation of $\gamma = \varphi_e$, and consequently at any point of the construction we have the following properties of the open block

- The open block exists, because if γ has at least s blocks, then so does its delayed simulation M.

```
The Construction

Initialization

        γ(0):=0

Stage s of the construction

        cover_some_block_of_G;
        close_the_open_block;


The Subroutines

  close_the_open_block
        m:= max dom(γ)
        γ(m + 1):=1 − γ(m)

  cover_some_block_of_G
        z:= min{i ∈ ω : bn(M, i) = max dom(γ)}
        y:=z
        while {z, . . . , y} does not contain some block of G
              if M(y) = M(y + 1) then y:=y + 1
```

Fig. 1. The construction of the set M from the cover lemma.

- The open block is finite iff φ_e is defined at place $s+1$.

The construction is based on the following idea: in the situation of (44) we refrain from defining γ at place $s+1$ unless we can verify that the open block contains some block of G. As a consequence the open block indeed contains some block of G: in case γ indeed remained undefined at place $s+1$, then the open block would be infinite, but would not contain a block of the gap language G, which is a plain contradiction. So, intuitively speaking, if we add block $s+1$ of γ only after verifying that block s of M is large enough, then block s of M indeed is large enough. Using the ideas sketched in the preceding paragraphs, we now give a formal proof of the cover lemma by showing a series of claims.

Claim 1. On entering stage s of the construction we have $\gamma{:=}(\omega \oplus \emptyset) \mid \{0, \ldots, s\}$.

Proof of Claim 1. Recall that the course of values of $\omega \oplus \emptyset$ is $01010\ldots$. Claim 1 now follows by an easy induction argument from the construction and the definition of the procedure *close_the_open_block*.

Claim 2. If the procedure *cover_some_block_of_G* is called during stage s of the construction, then the procedure terminates and block s of M contains some block of G.

Proof of Claim 2. Let the procedure be called during some stage s of the construction. By Claim 1, on entering the procedure we have $s = \max dom(\gamma)$ and γ has exactly s blocks. Thus its delayed simulation M has at least s blocks, and on start of the procedure, z and y are set equal to the minimal element of block s of M, and the while loop is entered. During the iterations of the while loop, the set $\{z, \ldots, y\}$ is always contained in block s of M, and consequently, by the condition in the head of the while loop, if the while loop is eventually left, then block s of M must contain some block of G. Now, assume for a contradiction that the while loop is never left, and that consequently γ will remain undefined at place $s+1$ for the rest of the construction. Then the delayed simulation M of γ has exactly s blocks, and $M(y) = M(y+1)$ holds for all places y which are greater than the minimal element z in block s of M. So y goes to infinity, and the condition in the head of the while loop eventually becomes false, because all blocks of the gap language G are finite. So the while loop is eventually left, contrary to our assumption.

Claim 3. M is a gap cover for G.

Proof of Claim 3. By inspection of the construction and from Claim 2 we infer that the construction passes through all stages $s = 0, 1, \ldots$. Thus Claim 1 shows that the delayed simulation M of γ has infinitely many blocks and Claim 3 implies that each block of M contains some block of G, that is, M is a gap cover for G. □

Now, using similar methods as in the proof of the Cover Lemma, we show the Coding Lemma 79.

Lemma 79 - Coding lemma. *Let $A_0, A_1, \ldots$ be a sequence of uniformly recursive sets and let G be a gap language. Let $\mathcal{M}$ be a simulation class which contains all finite sets and where the structure $(\mathcal{M}, \subseteq)$ is a subalgebra of $(2^\omega, \subseteq)$. Then there are sets $R_0, R_1, \ldots$ in $\mathcal{M}$ and a gap language M in $\mathcal{M}$ such that*

- *The set M is a gap cover for G.*
- *For all i and k in ω, and for all x in block s of M holds $R_i(x) = A_i(s)$.*

Recall that the sets R_i are uniformly recursive, because the sets A_i are and due to the second condition in the conclusion of the coding lemma. Recall further that the point of the coding lemma is that it yields delayed simulations R_i of the sets A_i which are "synchronized" via the gap language M, that is, for all i and s the set R_i is constant on the block s of M and has the value $A_i(s)$ there. The proof of the coding lemma relies on the same idea as the proof of Lemma 66, but is combinatorially more involved.

Proof. For a given gap language M, we denote blocks number $0, 2, 4, \ldots$ as even blocks of M and likewise the remaining blocks of M are denoted as odd. It is sufficient to show that given $\mathcal{M}$, G, and $A_0, A_1, \ldots$ as in the premise of the coding lemma, there is a gap language M in $\mathcal{M}$ and sets $S_1, S_2, \ldots$ in $\mathcal{M}$ where firstly, M is a gap cover for G, and secondly

(i) for all x in an odd block s of M and for all i in ω we have $S_{2i+1}(x) = A_i(s)$,
(ii) for all x in an even block s of M and for all i in ω we have $S_{2i+2}(x) = A_i(s)$,

that is, for every i, the set S_{2i+2} is constant on each even block of M and attains on these blocks the values $A_i(0), A_i(2), A_i(4), \ldots$, respectively, and a similar remark holds for the sets S_{2i+1} and the odd blocks of M. Given sets M and $S_1, S_2, \ldots$ as above, we let

$$R_i := (S_{2i+1} \cap M) \cup (S_{2i+2} \cap \overline{M}) \ ,$$

where due to $\mathcal{M}$ being closed under complement, we assume $M(0) = 0$, that is, M is the union of its odd blocks. By assumption on the sets S_k, the sets R_i satisfy the second condition required in the coding lemma, and further the sets R_i are in $\mathcal{M}$, because M and the set S_k are, and because $\mathcal{M}$ is a subalgebra of $(2^\omega, \subseteq)$. Note that it is actually sufficient to show that there are sets S_k which satisfy condition (i) and (ii) for almost all x, because by assumption the subalgebra $\mathcal{M}$ contains all finite sets and is hence c.f.v.

We denote by γ^j row j of a partial characteristic function γ, that is, γ^j denotes the partial characteristic function which maps x to $\gamma(\langle x, j \rangle)$. We let r be a recursive function such that for all e the number $r(e, j)$ is an index for row j of φ_e, and we let $\mathcal{M}$ be a simulation class via a function sim such that $\varphi_{sim(e)}$ is a delayed simulation of φ_e whenever $\varphi_e(0)$ is equal to 0.

Similar to the proof of the cover lemma, we give an effective enumeration of the graph of a partial characteristic function γ where by Remark 84, in the specification of γ we use an index e where $\gamma = \varphi_e$. In the specification of this enumeration, we will refer by γ to the finite partial characteristic function the graph of which has already been enumerated, and likewise we will refer by γ^j to row j of this intermediate partial characteristic function. For the index e given to the construction, we let

$$S_k \ := \ \varphi_{sim(r(e,k))} \qquad \text{and} \qquad M \ := \ S_0 \ = \ \varphi_{sim(r(e,0))} \ ,$$

```
The Construction

⋆ Initially, the partial functions γ^j all have empty domain

  Stage s of the Construction

      initialize_row(s)
      for all k in {j ∈ ω : 1 ≤ j ≤ s and j + s is odd}
          if A_[k div 2](s − 1) ≠ A_[k div 2](s + 1) then
              compare(0,k)
              close_block(k)
              compare(k,0)
      cover_some_block_of_G
      close_block(0)
```

Fig. 2. The construction of the sets M and S_k.

that is, for all $k \geq 0$ the set S_k, which we informally denote as row k, is a delayed simulation of row k of φ_e. Note that thus for any given index e the meaning of M and S_k is fixed ahead of the construction. In the verification of the construction we then show, under the assumption that the index e used in the construction is indeed an index for the function γ constructed, that the sets M and $S_1, S_2, \ldots$ have the required properties.

The construction is shown in Fig. 2. Before we formally verify the construction, we give an informal description of stage s of the construction. Like in the verification of the cover lemma, at any stage of the construction, we denote block $\max dom(\gamma^k)$ as the open block of S_k. We assume that s is even, while the considerations for the symmetric case where s is odd are basically the same. During stage s, first row s is initialized, and then we consider all rows $k \leq s$ where k is odd. We have to ensure that for all such k the sets S_k do not have a block-change in an odd block of M. Further, if we let i be equal to k div 2, then in case $A_i(s-1)$ differs from $A_i(s+1)$, the set S_k must have exactly one block-change within block s of M, while in case $A_i(s-1)$ is equal to $A_i(s+1)$, the set S_k must not have a block-change within block s of M. In the latter case, we leave γ^k untouched, and otherwise we enforce the necessary block change by first forcing the open block of S_k to extend beyond the minimum of the open block of M, then closing the open block of S_k, and then extending the open block of M beyond the minimum of the (new) open block of S_k. This extensions

```
Some subroutines

⋆ The following two routines are the same as in Fig. 1
⋆ with γ replaced by γ^j and γ^0, respectively
close_the_open_block(j)
        m:= max dom(γ^j)
        γ^j(m + 1):=1 − γ^j(m)

cover_some_block_of_G
        z:= min{i ∈ ω : bn(M, i) = max dom(γ^0)}
        y:=z
        while {z, . . . , y} does not contain some block of G
                if M(y) = M(y + 1) then y:=y + 1

initialize_row(k)
        γ^k(0):=0
        if A_[k div 2](s) = 1 then γ^k(1):=1
        compare(k,0)

compare(k,l)
        y:= min{i ∈ ω : bn(S_l, i) = max dom(γ^l)}
        z:= min{i ∈ ω : bn(S_k, i) = max dom(γ^k)}
        while y ≰ z
                if S_l(y) = S_l(y + 1) then y:=y + 1
```

Fig. 3. Some subroutines used in the construction of the sets M and S_i.

are handled by the procedure *compare*, where by an invocation *compare(k,l)* we force the open block of S_l to contain an element which is greater or equal to the least element in the open block of S_k. Finally, we ensure that the open block of M contains some block of G and close the open block of M.

Claim 1. Every call to *close_the_open_block* and to *cover_some_block_of_G* during the construction results in a terminating computation.

Proof of Claim 1. The assertion is immediate in the case of the former procedure, while in the case of *cover_some_block_of_G* the argument is basically the same as for the corresponding claim in the proof of Theorem 66.

In the remainder of the proof, we will use the following notation: a set I extends beyond the minimum of a set J iff we have $\min J \leq x$ for some x in I.

Claim 2. Assume that on a call *compare(k,l)* during the construction we have

$$m_k := \max dom(\gamma^k) \quad \text{and} \quad m_l := \max dom(\gamma^l). \ .$$

Then this call results in a terminating computation and block m_l of S_l extends beyond the minimum of block m_k of S_k.

Proof of Claim 2. Similar to the case of the procedure *cover_some_block_of_G* we obtain that on entering *compare*, y and z are indeed set to the minimal elements in the open blocks of row k and l, respectively. Obviously, if the while loop is eventually left, then block m_l of S_l extends beyond the minimum of block m_k of S_k. Now, assuming that the while loop is never left, we infer that γ and a fortiori row l of γ will never be altered afterwards, and consequently, the open block of row l is infinite and y goes to infinity, that is, the while loop is eventually left, contrary to our assumption.

Claim 3. The construction passes through infinitely many stages.

Proof of Claim 3. Immediate from the construction and by Claims 1 and 2.

Claim 4. The set M is a gap cover for G.

Proof of Claim 4. As for Claim 3 in the proof of Theorem 66 where γ is replaced with γ^0.

We say that the n-th block-change of some set C occurs within some set I iff the set C has at least $n+1$ blocks and the maximum of block n of C and the minimum of block $n+1$ are both in I.

Claim 5. Let C and D be sets, and let I be some block of D. If block n of C extends beyond the minimum of I and I extends beyond the minimum of block $n+1$ of C, then the n-th block-change of C occurs within I.

Proof of Claim 5. The claim is immediate by the definition of the concepts involved.

Claim 6. For all k, s, n in ω where $0 < k \leq s$ the number $n+1$ enters the domain of γ^k during stage s iff the n-th block-change of S_k occurs in block s of M.

Proof of Claim 6. We show first that if $n+1$ enters the domain of γ^k during stage $s \geq k$, then the n-th block-change of S_k occurs in block s of M. In case $n+1$ enters the domain of γ^k during stage s, this is due to a call *close_the_open_block(k)* in the body of the *if* statement. So this call is preceded by a call *compare(0,k)*, and is followed by a call *compare(k,0)* where on both calls $\max dom(\gamma^0)$ is equal to s, and $\max dom(\gamma^k)$ is equal to n and $n+1$ on the first and on the second call, respectively. Then, if we let I be equal to block s of M, we obtain by Claim 4 that block n of S_k extends beyond the minimum of I and that in turns I extends beyond the minimum of block $n+1$ of S_k. Thus, we infer by Claim 5 that the n-th block-change of S_k occurs in I. Conversely, if $n+1$ does not enter the domain of γ^k then S_k does not have an n-th block-change, and if $n+1$ enters the domain of γ^k during some stage s_0 different from s, then by the above argument, the n-th block-change of S_k occurs in block s_0 of M and not in block s.

Claim 7. For all k in ω, there occurs at most one block-change of S_k in each block of M.

Proof of Claim 7. Immediate from Claim 5 and the construction, because at each stage at most one element enters the domain of any γ^k.

Claim 8. For all k, s, i in ω where $k \leq s$ and i is equal to k div 2, the set S_k has a block-change in block s of M iff $k+s$ is odd and $A_i(s-1)$ differs from $A_i(s+1)$.

Proof of Claim 8. It is obvious from the construction that some element enters the domain of γ^k during stage $s \geq k$ iff $k+s$ is odd and $A_i(s-1)$ differs from $A_i(s+1)$. So the assertion follows by Claim 6.

Claim 9. For each i in ω

- for almost all x in an odd block s of M we have $S_{2i+1}(x) = A_i(s)$,
- for almost all x in an even block s of M we have $S_{2i+2}(x) = A_i(s)$.

Proof of Claim 9. We show the assertion for the odd block and leave the similar proof for the even blocks to the reader. Let $k = 2i+1$ for some $i \geq 0$ and consider stage k of the construction. During the procedure *initialize_row* row k of γ is changed such that $\gamma^k(\max dom(\gamma^k))$ is equal to $A_i(s)$, and by the call of the procedure *compare* and Claim 2 follows that block s of M extends beyond the minimum of the open block of S_k. Now, the assertion follows by an easy induction argument, because for each odd block $s = 2j+1 > k$ we have by Claim 8 that either $A_i(s-1)$ is equal to $A_i(s+1)$ and there occurs no block-change of S_k within the blocks $s-1$, s, and $s+1$ of M, or $A_i(s-1)$ differs from $A_i(s+1)$ and there occurs exactly one block-change of S_k within block s of M, but none within the blocks $s-1$ and $s+1$. □

5 Acknowledgements

We like to thank Klaus Ambos-Spies for helpful discussion.

References

1. Klaus Ambos-Spies. Sublattices of the polynomial time degrees. *Information and Control*, 65(1):63–84, 1985.
2. Klaus Ambos-Spies. Minimal pairs for polynomial time reducibilities. In Egon Börger, editor, *Computation Theory and Logic*, Lecture Notes in Computer Science 270. Springer-Verlag, 1987.
3. Klaus Ambos-Spies. Honest polynomial time reducibilities and the $P = ?NP$ problem. *Journal of Computer and System Sciences*, 39(3):250–281, 1989.
4. Klaus Ambos-Spies, Peter A. Fejer, Steffen Lempp, and Manuel Lerman. Decidability of the two-quantifier theory of the recursively enumerable weak truth-table degrees and other distributive upper semi-lattices. *Journal of Symbolic Logic*, 61(3):880–905, 1996.
5. José Luis Balcázar and Josep Díaz. A note on a theorem by Ladner. *Information Processing Letters*, 15:84–86, 1982.
6. José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity*, volume I. Springer-Verlag, Berlin Heidelberg New York, 1988.
7. José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity*, volume II. Springer-Verlag, Berlin Heidelberg New York, 1990.
8. Sanat K. Basu. On the structure of subrecursive degrees. *Journal of Computer and System Sciences*, 4:452–464, 1970.
9. Manuel Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the Association for Computing Machinery*, 14(2):322–336, 1967.
10. Ronald Book, Ding-Zhu Du, and David Russo. On polynomial and generalized complexity cores. In Juris Hartmanis, editor, *Proceedings of the 3rd Annual Conference on Structure in Complexity Theory*, pages 236–250, Washington, D.C., 1988. IEEE Computer Science, Computer Science Press of the IEEE.
11. Ronald V. Book, Jack H. Lutz, and Klaus W. Wagner. An observation on probability versus randomness with applications to complexity classes. *Mathematical Systems Theory*, 27(3):201–209, 1994.
12. Ronald V. Book, Heribert Vollmer, and Klaus W. Wagner. On type-2 probabilistic quantifiers. Technical Report 123, Institut für Informatik, Julius-Maximilians-Universität, Würzburg, Germany, 1995. Submitted for further publication.
13. Ronald V. Book, Heribert Vollmer, and Klaus W. Wagner. On type-2 probabilistic quantifiers. In *International Conference on Automata, Languages and Programming 96*, 1996.
14. Seth Breidbart. On splitting recursive sets. *Journal of Computer and System Sciences*, 17:56–64, 1978.
15. Paul Chew and Michael Machtey. A note on structure and looking back applied to the relative complexity of computable functions. *Journal of Computer and System Sciences*, 22:53–59, 1981.

16. Peter Clote. Computation models and function algebras. To appear in E. Griffor, editor, Handbook of Recursion Theory.
17. Steven Homer. Minimal degrees for polynomial time reducibilities. In *Proceedings of the 25th Annual IEEE Symposium on Foundations of Computer Science*, 1984.
18. Steven Homer. Minimal polynomial degrees of nonrecursive sets. In Heinz-Dieter Ebbinghaus, Gert H. Müller, and Gerald E. Sacks, editors, *Recursion Theory Week 84*, Lecture Notes in Computer Science 1141, pages 193–202. Springer-Verlag, Berlin Heidelberg New York, 1984.
19. Steven Homer. Minimal degrees for polynomial time reducibilities. *Journal of the Association for Computing Machinery*, 34(2):480–491, 1987.
20. John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, Massachusetts, 1979.
21. Bruce M. Kapron and Stephen A. Cook. A new characterization of Mehlhorn's polynomial time functionals. In *Proceedings of the 32th Annual IEEE Symposium on Foundations of Computer Science*, 1991.
22. Bruce M. Kapron and Stephen A. Cook. A new characterization of type-2 feasibility. *SIAM Journal on Computing*, 25(1):117–132, 1996.
23. Richard E. Ladner. On the structure of polynomial time reducibility. *Journal of the Association for Computing Machinery*, 22(1):155–171, 1975.
24. Richard E. Ladner, Nancy Ann Lynch, and Alan L. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1:103–123, 1975.
25. Larry H. Landweber, Richard J. Lipton, and Ed L. Robertson. On the structure of sets in NP and other complexity classes. *Theoretical Computer Science*, 15:181–200, 1981.
26. Nancy Ann Lynch, Albert R. Meyer, and Michael J. Fischer. Relativization of the theory of computational complexity. *Transactions of the American Mathematical Society*, 220:243–287, 1976.
27. Michael Machtey. The honest subrecursive classes are a lattice. *Information and Computation*, 24:247–263, 1974.
28. Michael Machtey. Minimal pairs of polynomial degrees with subexponential complexity. *Theoretical Computer Science*, 2:73–76, 1976.
29. Kurt Mehlhorn. *Polynomial and abstract subrecursive classes*. Ph.d. dissertation, Cornell University, Ithaca, NY, 1974.
30. Kurt Mehlhorn. The "almost all" theory of subrecursive degrees is decidable. In *International Conference on Automata, Languages and Programming 74*, pages 317–325, 1974.
31. Kurt Mehlhorn. Polynomial and abstract subrecursive classes. *Journal of Computer and System Sciences*, 12:147–178, 1976.
32. Wolfgang Merkle. *A Generalized Account of Resource Bounded Reducibilities*. Doctoral dissertation, Universität Heidelberg, Germany, 1996.
33. Wolfgang Merkle. Exact pairs for abstract bounded reducibilities. In Marc Bezeem and Dirk van Dalen, editors, *Computer Science Logic 96*, Lecture Notes in Computer Science 1258, pages 349–360. Springer-Verlag, 1997.
34. Wolfgang Merkle. Structural properties of bounded relations with an application to NP optimization problems. Submitted for publication, 1997.

35. Wolfgang Merkle and Yongge Wang. Random separations and "Almost" classes for generalized reducibilities. In *Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science 969. Springer-Verlag, Berlin Heidelberg New York, 1995.
36. Jeanleah Mohrherr. Density of a final segment of the truth-table-degrees. *Pacific Journal of Mathematics*, 115(2):409–419, 1984.
37. William Mueller. *Abstract degree structures.* Ph.d. dissertation, Mount Holyoke College, South Hadley, MA, 1991.
38. Piergiorgio Odifreddi. *Classical Recursion Theory.* North-Holland, Amsterdam, 1989.
39. Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.
40. Kenneth W. Regan and James S. Royer. On closure properties of bounded two-sided error complexity classes. *Mathematical Systems Theory*, 28(3):229–243, 1995.
41. Diana Schmidt. The recursion-theoretic structure of complexity classes. *Theoretical Computer Science*, 38:143–156, 1985.
42. Uwe Schöning. A uniform approach to obtain diagonal sets in complexity classes. *Theoretical Computer Science*, 18:95–103, 1982.
43. Uwe Schöning. Minimal pairs for P. *Theoretical Computer Science*, 31:41–48, 1984.
44. María J. Serna. *The parallel approximability of P-complete problems.* Tesis doctoral, Universitat Politècnica de Catalunya, Barcelona, Spain, 1990.
45. Richard A. Shore and Theodore A. Slaman. The p-t-degrees of the recursive sets: lattice embeddings, extensions of embeddings and the two quantifier theory. *Theoretical Computer Science*, 97:263–284, 1992.
46. Craig Smoryński. *Logical Number Theory*, volume I. Springer-Verlag, Berlin Heidelberg New York, 1991.
47. Robert I. Soare. *Recursively Enumerable Sets and Degrees.* Springer-Verlag, Berlin Heidelberg New York, 1987.
48. Heribert Vollmer. The gap-language-technique revisited. In Egon Börger, Hans Kleine-Büning, and Michael M. Richter, editors, *Computer Science Logic 90*, Lecture Notes in Computer Science. Springer-Verlag, 1991.
49. Christopher B. Wilson. Relativized circuit complexity. *Journal of Computer and System Sciences*, 31:169–181, 1985.
50. Christopher B. Wilson. Relativized NC. *Mathematical Systems Theory*, 20:13–29, 1987.