Cost functions

André Nies
The University of Auckland

FRG workshop, Madison, May 2009

0.1 What cost functions are good for

- Cost functions are a great tool for analyzing certain classes of Δ^0_2 sets.
- Mostly, these classes are lowness properties such as being K-trivial, or strongly jump traceable.

Cost functions help a lot to understand the following results (I will shortly explain the notions involved).

- Each K-trivial set is Turing below a *c.e.* K-trivial set (Nies).
- Each null Σ^0_3 class of ML-random sets has a simple Turing lower bound. Moreover, this lower bound is obtained via an injury-free construction (Hirschfeldt, Miller).
- *Each* strongly jump traceable c.e. set is Turing below *each* ω-c.e. ML-random set (Greenberg, Nies).

1 Introduction to cost functions

1.1 Definition of cost functions

Definition 1. *A* cost function *is a computable function*

$$c : \mathbb{N} \times \mathbb{N} \to \{x \in \mathbb{Q}\colon\ x \geq 0\}.$$

We view $c(x, s)$ as the cost of changing $A(x)$ at stage s.

1.2 Obeying a cost function

Recall that A is Δ^0_2 *iff* $A \leq_T \emptyset'$ *iff* $A(x) = \lim_s A_s(x)$ for a computable approximation $(A_s)_{s\in\mathbb{N}}$ (Limit Lemma).

Definition 2. *The computable approximation* $(A_s)_{s\in\mathbb{N}}$ obeys *a cost function* c *if* $\infty > \sum_{x,s} c(x, s)\, [\![x < s \ \&\ x \text{ is least s.t. } A_{s-1}(x) \neq A_s(x)]\!]$. *We write* $A \models c$ *(*A *obeys* c*) if some computable approximation of* A *obeys* c*.*

Usually we use this to construct some auxiliary object of finite "weight", such as a bounded request set (aka Kraft-Chaitin set), or a Solovay test.

1.3 Basic existence theorem

For a cost function $c : \mathbb{N} \times \mathbb{N} \to \mathbb{Q}$, let $c(x) = \sup_s c(x, s)$.
We say that c has the *limit condition* if $lim_x c(x) = 0$.

Theorem 3 (Various authors)**.** *If a cost function c satisfies the limit condition, then some (promptly) simple set A obeys c.*

Proof. Let W_e be the e-th c.e. set. If W_e is infinite we want some $x \in W_e$ to enter A. We define a computable enumeration $(A_s)_{s\in\mathbb{N}}$ as follows. Let $A_0 = \emptyset$. For $s > 0$,
$A_s = A_{s-1} \cup \{x : \exists e$

$W_{e,s} \cap A_{s-1} = \emptyset$	We haven't met e-th simplicity requirement.
$x \in W_{e,s}$	We can meet it via x.
$x \geq 2e$	We make A co-infinite.
$c(x, s) \leq 2^{-e}\}$.	We ensure that A obeys c.

1.4 An infinite number of visits to K-Mart

Here is a real-life analog of the foregoing construction.

- We want to buy a shirt of each color e at K-Mart, provided that there is a sufficient number of shipments of that color from China.
- For the shirt of color e we can spend at most 2^{-e}.
- Eventually, a sufficiently cheap shirt of color e will arrive, unless that color is discontinued.
- We will be able to buy all shirts that are not discontinued.
- We will spend at most $\sum_e 2^{-e} = 2$ dollars in total.

2 Cost functions and K-trivialilty

2.1 Machines and K

- All strings are binary. A *machine* is a partial recursive function M from strings to strings.
- M is called *prefix free* if its domain is an antichain under the prefix relation of strings.
- There is a universal prefix-free machine $\mathbb{U}$.
- The prefix free version $K(y)$ of descriptive string complexity (a.k.a. Kolmogorov complexity) is the length of a shortest $\mathbb{U}$-description of y:

$$K(y) = \min\{|\sigma| : \mathbb{U}(\sigma) = y\}.$$

- Also, $K_s(y) = \min\{|\sigma| : \mathbb{U}(\sigma) = y \text{ in } s \text{ steps}\}$.

2.2 Definition of K-triviality

- For a string y, up to constants, $K(|y|) \le K(y)$, since we can compute $|y|$ from y (here we write numbers in binary).
- A set A is *K-trivial* if, for some $b \in \mathbb{N}$

$$\forall n\, K(A \upharpoonright_n) \le K(n) + b,$$

  namely, the K complexity of all initial segments is minimal.

This is *opposite* to ML-randomness:

- Z is ML-random if $\exists d \forall n\, K(Z \upharpoonright_n) \ge n - d$ (Schnorr's Theorem). That is, all complexities $K(Z \upharpoonright_n)$ are near the upper bound $n + K(n)$;
- Z is K-trivial if each $K(Z \upharpoonright_n)$ has the minimal possible value $K(n) \le^+ 2 \log n$ (all within constants).

2.3 The cost function for K-triviality

Definition 4. *The* standard cost function $c_{\mathcal{K}}$ *is given by*

$$c_{\mathcal{K}}(x, s) = \sum_{x < w \le s} 2^{-K_s(w)}.$$

We could also use $c(x, s) = \text{Prob}[\{\sigma\colon\ \mathbb{U}_s(\sigma) \ge x\}]$, the chance that the universal machine prints a string $\ge x$ within s steps.

Lemma 5. *$c_{\mathcal{K}}$ satisfies the limit condition.*

Proof. Given $e \in \mathbb{N}$, since $\sum_w 2^{-K(w)} \le 1$, there is an x_0 such that $\sum_{w \ge x_0} 2^{-K(w)} < 2^{-e}$. Hence $c_{\mathcal{K}}(x, s) < 2^{-e}$ for all $x \ge x_0$ and all s.

2.4 Cost function characterization of the K-trivials

Theorem 6 (Nies 05)**.** *A is K-trivial* $\Leftrightarrow$
some computable approximation of A obeys $c_{\mathcal{K}}$.

'$\Leftarrow$' is not too hard.
'$\Rightarrow$' is also not too hard for c.e. sets. For Δ^0_2 sets, in contrast, '$\Rightarrow$' needs a non-uniform method known as the *golden run*.

Corollary 7. *For each K-trivial set A, there is a* c.e. *K-trivial set $D \ge_T A$.*

D is the *change set* $\{\langle x, i \rangle\colon\ A(x)$ *changes at least i times*$\}$. One verifies that D obeys $c_{\mathcal{K}}$ as well.
Actually this works for any cost function in place of $c_{\mathcal{K}}$!

2.5 The Machine Existence Theorem

We use this tool:

- A c.e. set $L \subseteq \mathbb{N} \times \{0,1\}^*$ is a *bounded request set* if

$$1 \geq \sum_{r,y} 2^{-r} [\![\langle r, y\rangle \in L]\!].$$

- *From a bounded request set L, one can (effectively) obtain a prefix free machine M such that*

$$\forall r, y[\langle r, y\rangle \in L \Leftrightarrow \exists w\, (|w| = r \,\&\, M(w) = y)].$$

2.6 Cost function criterion for K-triviality

Lemma 8. *Suppose a computable approximation $(A_s)_{s\in\mathbb{N}}$ of a set A obeys the standard cost function $c_{\mathcal{K}}(x, s) = \sum_{x<w\leq s} 2^{-K_s(w)}$.*
Then A is K-trivial.

Proof. We use the Machine Existence Theorem to implicitly build a prefix-free machine showing that A is K-trivial.
We may assume the total cost of A-changes is at most 1. We build a bounded request set. At stage s we enumerate the request

$$\langle K_s(w) + 1, A_s{\upharpoonright}_w\rangle$$

whenever $w \leq s$ and

(a) $K_s(w) < K_{s-1}(w)$, *or (b)* $K_s(w) < \infty \,\&\, A_{s-1}{\upharpoonright}_w \neq A_s{\upharpoonright}_w$.

In either case, the implicitly built prefix-free machine provides a description of $A_s{\upharpoonright}_w$ of length $K_s(w)$.
The total weight for (a) is at most $\Omega/2$. The total for (b) is at most 1/2.

3 Basic properties of cost functions

We introduce monotonicity and give some examples.
We obtain some simple closure properties for the class of sets obeying a cost function

Definition 9. *A cost function $c(x, s)$ is called* monotonic *if it is nonincreasing in x and nondecreasing in s. That is, $c(x + 1, s) \leq c(x, s) \leq c(x, s + 1)$ for all x, s.*

Exercise 10. *Show that $c_{\mathcal{K}}$ is monotonic.*

Exercise 11. *There is a computable enumeration $(A_s)_{s\in\mathbb{N}}$ of $\mathbb{N}$ in the order $0, 1, 2, \ldots$ (i.e., each A_s is an initial segment of $\mathbb{N}$) such that $(A_s)_{s\in\mathbb{N}}$ does not obey $c_{\mathcal{K}}$.*

Exercise 12. *Prove the converse of the Existence Theorem 3 for a monotonic cost function c: if an incomputable Δ^0_2 set A obeys c, then c satisfies the limit condition.*

3.1 The class $\{A\colon\ A \models c\}$

We say that Y is ω-*c.e.* if $Y(x) = \lim_s Y_s(x)$ with a computably bounded number of changes. Equivalently, $Y \leq_{wtt} \emptyset'$. Let V_e be the e-th ω-c.e. set (given by an index of a wtt reduction to $\emptyset'$).
In the following let c *be a monotonic cost function.*

Exercise 13. *(i) The index set* $\{e\colon\ V_e \models c\}$ *is* Σ^0_3.
(ii) If $\forall x\, \exists s\, [c(x,s) > 0]$*, then* $A \models c$ *implies that* A *is* ω*-c.e.*

For $X \subseteq \mathbb{N}$ let $2X$ denote $\{2x\colon\ x \in X\}$. Recall that $A \oplus B = 2A \cup (2B+1)$.

Exercise 14. $A \models c\ \&\ B \models c$ *implies* $A \oplus B \models c$.

3.2 Changing early is good

Proposition 15 (Nies)**.** *Let* c *be a monotonic cost function. Suppose* $A \leq_{ibT} B$ *and* $B \models c$*. Then* $A \models c$.

The argument is fairly typical. We change $A(x)$ as early as possible because earlier changes are cheaper.
For a computable approximation (E_s), let $\mathsf{TC}((E_s), c)$ denote the total cost of changes.
Let $A = \Gamma^B$ where Γ is a Turing reduction with use bounded by the identity. We define a computable increasing sequence of stage $(s(i))_{i \in \mathbb{N}}$ by $s(0) = 0$ and

$$s(i+1) = \mu s > s(i)\, [\Gamma^B \restriction_{s(i)} [s] \downarrow].$$

We define $A_{s(k)}(x)$ for each $k \in \mathbb{N}$. Then we let $A_s(x) = A_{s(k)}(x)$ where k is maximal such that $s(k) \leq s$.
Suppose $s(i) \leq x < s(i+1)$.

- Let $A_{s(k)}(x) = v$ for $k < i$ where $v = \Gamma^B(x)[s(i+2)]$.
- For $k \geq i$, let $A_{s(k)}(x) = \Gamma^B(x)[s(k+2)]$.

(Note that these values are defined.)
Clearly $\lim_s A_s(x) = A(x)$. We show

$$\mathsf{TC}((A_s), c) \leq \mathsf{TC}((B_t), c).$$

Suppose that $A_{s(k)}(x) \neq A_{s(k)-1}(x)$. Since the reduction is ibT, there is $y \leq x$ such that $B_t(y) \neq B_{t-1}(y)$ for some t, $s(k+1) < t \leq s(k+2)$. Then $c(x, s(k)) \leq c(y,t)$.

4 Cost functions, Kučera's Theorem, Diamond Classes

- We consider pairs of sets A, Y such that A is c.e., Y is ML-random, and $A \leq_T Y$.
- If $Y \not\geq_T \emptyset'$, then it is hard for A to get anything out of Y: the set A must be K-trivial (Hirschfeldt, Nies, Stephan 2007).

4.1 Kučera's Theorem

Theorem 16 (Kučera1986)**.** *Let Y be Δ^0_2 and* ML-random. *Then there is a (promptly) simple set $A \leq_T Y$. Moreover, the use is bounded by the identity.*

Kučera actually proved this for any Δ^0_2 set computing a d.n.c. function. The recursion theorem is needed in the more general case.

To prove the theorem, we need a test concept that is equivalent to ML-tests.

- A *Solovay test* $\mathcal{G}$ is given by an effective enumeration of strings $\sigma_0, \sigma_1, \ldots$, such that $\sum_i 2^{-|\sigma_i|} < \infty$.
- Y *passes* $\mathcal{G}$ if $\sigma_i \not\preceq Y$ for almost all i.

We want to meet the requirements

$$S_e: \ |W_e| = \infty \Rightarrow A \cap W_e \neq \emptyset.$$

Construction. *At stage s, if S_e is not satisfied yet, see if there is an x, $2e \leq x < s$, such that*

$$x \in W_{e,s} - W_{e,s-1} \ \& \ \forall t_{x<t<s} \, Y_t \restriction_e = Y_s \restriction_e .$$

If so, put x into A. Put the string $\sigma = Y_s \restriction_e$ into $\mathcal{G}$. Declare S_e satisfied.

- Clearly A is (promptly) simple.
- To see that $A \leq_T Y$, choose s_0 such that $\sigma \not\preceq Y$ for any σ enumerated into $\mathcal{G}$ after stage s_0. Given an input $x \geq s_0$, using Y as an oracle, compute $t > x$ such that $Y_t \restriction_x = Y \restriction_x$. Then $x \in A \ \leftrightarrow \ x \in A_t$. For if we put x into A at a stage $s > t$ for the sake of S_e then $x > e$, so we list σ in $\mathcal{G}$ where $\sigma = Y_s \restriction_e = Y \restriction_e$; this contradicts the fact that $\sigma \not\preceq Y$.

4.2 A proof of Kučera's Theorem using a cost function

Let $c_Y(x,s) = 2^{-x}$ for each $x \geq s$. If $x < s$, and $e < x$ is least such that $Y_{s-1}(e) \neq Y_s(e)$, let

$$c_Y(x,s) = \max(c_Y(x,s-1), 2^{-e}).$$

Since Y is Δ^0_2, the cost function c_Y satisfies the limit condition.

Fact 17 (Greenberg and Nies)**.** *If the Δ^0_2 set A obeys c_Y, then $A \leq_T Y$ with use function bounded by the identity.*

We build the Solovay test as follows. When $A_{s-1}(x) \neq A_s(x)$ and $c_Y(x,s) = 2^{-e}$, we list the string $Y_s \restriction_e$ in $\mathcal{G}$. Since A obeys c_Y, $\mathcal{G}$ is indeed a Solovay test. Now as before one shows $A \leq_T Y$ with use bounded by the identity.

Some promptly simple A obeys c_Y. So $A \leq_T Y$.

4.3 The arithmetical hierarchy for classes

- A Π^0_1 class is of the form $\{X\colon\ \forall y\, T(X\upharpoonright_y)\}$
- A Σ^0_2 class is of the form $\{X\colon\ \exists y_1 \forall y_2\, V(y_1, X\upharpoonright_{y_2})\}$
- A Π^0_2 class is of the form $\{X\colon\ \forall y_1 \exists y_2\, S(y_1, X\upharpoonright_{y_2})\}$,
- a Σ^0_3 class is of the form $\{X : \exists y_1 \forall y_2 \exists y_3\, R(y_1, y_2, X\upharpoonright_{y_3})\}$, where T, V, S and R are computable relations.

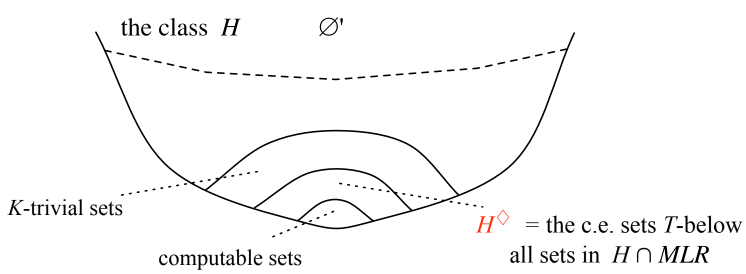
Examples:

- Π^0_1 means the complement of a c.e. open class.
- The class MLR is Σ^0_2.
- The class of c.e. sets is Σ^0_3.
- The class of computable sets is Σ^0_3.
- The class of cofinite sets is Σ^0_2 and not Π^0_2.

4.4 Diamond Classes

For a null class $\mathcal{H} \subseteq 2^{\mathbb{N}}$, we define

$$\mathcal{H}^{\Diamond} = \textit{the c.e. sets } A \textit{ Turing below each ML-random set in } \mathcal{H}.$$



- The larger $\mathcal{H}$ is, the smaller is $\mathcal{H}^{\Diamond}$.
- $\mathcal{H}^{\Diamond}$ induces an ideal in the c.e. Turing degrees.
- If some ML-random set $Y \not\geq_T \emptyset'$ is in $\mathcal{H}$, then $\mathcal{H}^{\Diamond} \subseteq K$-trivial.

4.5 An existence Theorem

Theorem 18 (Hirschfeldt/Miller)**.** *For each null Σ^0_3 class $\mathcal{H}$, there is a promptly simple set in $\mathcal{H}^\Diamond$.*

For instance, there is a promptly simple set in (ω-c.e.)$^\Diamond$.

- The theorem is proved by defining an appropriate cost function $c_\mathcal{H}$ with the limit condition.
- Whenever a c.e. set A obeys $c_\mathcal{H}$, then A is in $\mathcal{H}^\Diamond$.
- Now recall that some promptly set A obeys $c_\mathcal{H}$.

This implies that a ML-random set Y that is not weakly 2-random bounds an incomputable c.e. set: for $\mathcal{H}$ choose a null Π^0_2 class containing Y.
Kučera's Theorem is the special case where $\mathcal{H} = \{Y\}$ for ML-random Δ^0_2 set Y. Note that this $\mathcal{H}$ is Π^0_2.

4.6 The cost function $c_\mathcal{H}$

We may at first assume that $\mathcal{H}$ is a Π^0_2 class. That is, $\mathcal{H} = \bigcap_x V_x$ where V_x is c.e. open uniformly in x, and $V_x \supseteq V_{x+1}$. Let

$$c_\mathcal{H}(x, s) = \lambda V_{x,s}.$$

We want to show that $A \models c_\mathcal{H} \Rightarrow A \in \mathcal{H}^\Diamond$.

- Let $Y \in \mathcal{H} \cap \mathsf{MLR}$. Intuitively, we enumerate a Turing functional Γ such that $A = \Gamma^Y$. At stage t we define $\Gamma^Y(x) = A_t(x)$ for all Y in $V_{x,t}$. When $A_s(x) \neq A_{s-1}(x)$ for $s > t$, we have to remove all those oracles by declaring them non-random.
- Thus, we enumerate the Solovay test $\mathcal{G}$ as follows: when $A_s(x) \neq A_{s-1}(x)$, we enumerate $V_{x,s}$ into $\mathcal{G}$ (more precisely, we enumerate all strings σ of length s such that $[\sigma] \subseteq V_{x,s}$).

Extending this to a Σ^0_3 class $\mathcal{H}$ is left as an exercise.

4.7 Adaptive cost functions

A cost function construction can only be regarded as *injury-free* if the underlying cost function is *non-adaptive*, that is, the cost at a stage s does not depend on A_{s-1}. The usual construction of a low simple set has the lowness requirements

$$L_e\colon\ \exists^\infty s\, J^A(e)[s-1]\!\downarrow\, \Rightarrow\ J^A(e)\!\downarrow.$$

The following adaptive cost function encodes the restraint imposed by L_e: if $J^A(e)$ newly converges at stage $s-1$, then define

$$c(x, s) = \max\{c(x, s-1), 2^{-e}\}$$

for each $x < \mathsf{use}\ J^A(e)[s-1]$. If A is enumerated in such a way that the total cost of changes is finite, then L_e is injured only finitely often, so that A is low.
In contrast, a cost function c given in advance cannot be used to simulate restraints.

4.8 Some cost functions

Cost function	Definition	Purpose	Ref.
$c_{\mathcal{K}}(x,s)$	$\sum_{x<w\leq s} 2^{-K_s(w)}$	characterize the K-trivials	Def. 4
$c_Y(x,s)$	$= \max(c_Y(x,s-1), 2^{-e})$ where $Y_{s-1}(e) \neq Y_s(e)$	build a set below a Δ^0_2 set $Y \in \mathsf{MLR}$	Fact 17
$c_{\mathcal{H}}(x,s)$	$\lambda V_{x,s}$, where V_x is uniformly Σ^0_1 and $\mathcal{H} = \bigcap_x V_x$ is null	build a lower bound for $\mathcal{H} \cap \mathsf{MLR}$	page 8
$c_{\mathbb{U},A}(x,s)$	$\sum_\sigma 2^{-\lvert\sigma\rvert} [\![\mathbb{U}^A(\sigma)[s-1]\downarrow \ \& \ x < \mathsf{use}\ \mathbb{U}^A(\sigma)[s-1]]\!]$	build a set that is low for K	book

Recall that $c(x) = \sup_s c(x,s)$.

Exercise 19. *For the cost functions $c = c_{\mathcal{K}}$ and $c = c_Y$ ($Y \in \Delta^0_2$), describe $c(x)$ by giving a simple expression.*

5 Calculus of cost functions

5.1 Analogy with model theory

- A cost function c describes a class of Δ^0_2 sets: those sets with an approximation obeying the cost function.
- For instance, the standard cost function $c_{\mathcal{K}}$ describes the K-trivial sets.
- This is somewhat similar to a sentence in some formal language describing a class of structures.
- "*A obeys c*" means that A is a *model* of c.
- The *limit condition* behaves like *consistency*.
  Here we need to disregard the computable sets.
- If a cost function c has a model, it satisfies the limit condition. This is soundness.
- If c satisfies the limit condition, it has a model. This is like the completeness theorem.

5.2 The lower semilattice of cost functions

We introduce some relations and operations on *monotonic* cost functions. This corresponds to the Lindenbaum algebra on sentences.
For a cost function $c(x,s)$, recall that

$$c(x) = sup_s c(x,s).$$

We may assume $c(x)$ is finite for each x (otherwise only computable sets obey c).

5.3 Implication of cost functions

For cost functions c, d we write $c \longrightarrow d$ if $A \models c$ implies $A \models d$ for each Δ^0_2 set A. This is equivalent to $d(x) = O(c(x))$:

Theorem 20 (Nies 2009)**.** *Let c, d be cost functions. Suppose c satisfies the limit condition. Then*

$$c \longrightarrow d \ \Leftrightarrow\ \exists N \, \forall x \, [Nc(x) > d(x)].$$

In particular, whether $A \models c$ only depends on the function $c(x)$.

"$\Leftarrow$" needs a "changing early" construction similar to Prop.15. For "$\Rightarrow$" we assume that the right hand side fails. We build a counterexample: a Δ^0_2 set A such that $A \models c$ but $A \not\models d$.
Not sure whether A can be made c.e.

5.4 Relating c_Y and $c_{\mathcal{K}}$

Let the Δ^0_2 set Y be ML-random. Recall that c_Y is the cost function for being $\leq_T Y$. (Note that c_Y actually depends on a computable approximation of Y.)

Corollary 21. *Let $Y <_T \emptyset'$ be ML-random. Then $c_Y \longrightarrow c_{\mathcal{K}}$, and therefore $c_{\mathcal{K}}(x) = O(c_Y(x))$.*

Proof. Suppose $A \models c_Y$.
Let $D \geq_T A$ be the change set of the given approximation of A as in Cor. 7. Then $D \models c_Y$ and therefore $D \leq_T Y$.
Since D is c.e. and $Y <_T \emptyset'$, D is a base for ML-randomness by a result of Hirschfeldt, Nies, and Stephan. Therefore D, and hence A, is K-trivial. Thus $A \models c_{\mathcal{K}}$.

5.5 Conjunction of cost functions

The conjunction is simply the sum.

Theorem 22 (Nies 2009)**.** *Let c, d be cost functions. Then*

$$A \models c \,\&\, A \models d \ \Leftrightarrow\ A \models c + d.$$

"$\Leftarrow$" is trivial.
"$\Rightarrow$" needs some work because we have to find a computable approximation of A that obeys both c and d.

6 Benign cost functions and strong jump traceability

6.1 Strongly jump traceable sets

- An *order function* is a function $h : \mathbb{N} \to \mathbb{N}$ that is computable, nondecreasing, and unbounded.
- A *c.e. trace with bound* h is a uniformly c.e. sequence $(T_x)_{x\in\mathbb{N}}$ such that $|T_x| \leq h(x)$ for each x.
- Let $J^A(e)$ be the value of the A-jump at e, namely, $J^A(e) \simeq \Phi_e^A(e)$.
- The set A is called *strongly jump traceable* if for *each* order function h, there is a c.e. trace $(T_x)_{x\in\mathbb{N}}$ with bound h such that, whenever $J^A(x)$ it is defined, we have

$$J^A(x) \in T_x$$

  (Figueira, Nies, Stephan, 2004).
- *SJT* will denote the class of *c.e.* strongly jump traceable sets.

(To define *jump traceability*, one merely requires that the tracing works for *some* bound h.)

6.2 SJT is a proper subclass of the c.e. K-trivial sets

Theorem 23 (Cholak, Downey, Greenberg 2006)**.** *The c.e. strongly jump traceable sets form a* proper *subideal of the K-trivial sets.*

It is currently unknown what happens within the Δ_2^0 sets.

6.3 Comparing K-trivial and *SJT*

Within the c.e. sets:
- Both classes are closed downward under $\leq_T$.
- Both classes are closed under $\oplus$.
- The c.e. K-trivials have a Σ_3^0 index set;
  the (c.e.) SJTs have a Π_4^0-complete index set (Selwyn Ng).

Outside the c.e. sets:
- Each K-trivial is Turing-below a c.e. K-trivial.
- Currently we merely know that each strongly jump traceable set is *low* (Downey and Greenberg).

6.4 Benign cost functions

Let $c(x, s)$ be a monotonic cost function, that is, nonincreasing in x, and nondecreasing in s. For $\delta \in \mathbb{Q}^+$, a *δ-collection* is a set of pairwise disjoint intervals $[x, s)$ such that $c(x, s) \geq \delta$.

The limit condition is equivalent to $\forall\delta\,\exists x\,\forall s\,[c(x,s) < \delta]$. This is equivalent to: *each δ-collection is finite.*

Definition 24. *We say that the monotonic cost function c is* benign *if the cardinality of any δ-collection is bounded computably in δ.*

The standard cost function $c_{\mathcal{K}}$ is benign via the bound $\delta \to 1/\delta$.

6.5 Characterizing SJT via cost functions

- Cholak, Downey and Greenberg showed that SJT strictly implies K-trivial for c.e. sets.
- Greenberg and Nies reproved and extended this, using the language of cost functions.

Theorem 25 (Greenberg and Nies, to appear)**.** *Let A be c.e. Then*
A is strongly jump traceable $\Leftrightarrow$
A obeys each benign cost function.

- In particular, A is K-trivial.
- A single benign cost function doesn't do it, because *SJT* has Π^0_4 complete index set by a result of Selwyn Ng, while obeying a single cost function is Σ^0_3.
- We also prove directly that each benign cost function is obeyed by some c.e. set that is not strongly jump traceable.
- This gives a further proof that *SJT* is a proper subclass of the K-trivials.

For "$\Leftarrow$" we have to define the right benign cost function to ensure tracing of J^A at order h.
The harder direction is "$\Rightarrow$". It uses the "box promotion method" of Cholak, Downey and Greenberg.

6.6 A lowness property and its dual highness property

- Recall that Z is *low* if $Z' \leq_T \emptyset'$, and Z is *high* if $\emptyset'' \leq_T Z'$.
- These classes are "too big": we have

  $$\textit{(low)}^{\Diamond} = \textit{(high)}^{\Diamond} = \textit{computable}.$$

  (For instance, (high)$^{\Diamond}$= computable because there is a minimal pair of high ML-random sets.)
- So we will try somewhat smaller classes, replacing $\leq_T$ by the stronger truth-table reducibility $\leq_{tt}$.

Definition 26 (Mohrherr 1986)**.** *A set Z is* superlow *if $Z' \leq_{tt} \emptyset'$. Z is* superhigh *if $\emptyset'' \leq_{tt} Z'$.*

A random set can be superlow (low basis theorem). It can also be superhigh but Turing incomplete (Kučera coding).

6.7 SJT is contained in the two diamond classes

- *Superlow* is a countable Σ^0_3 class. *Superhigh* is contained in a null Σ^0_3 class (Simpson).
- So via the Hirschfeldt/Miller cost function $c_{\mathcal{H}}$ introduced to prove Theorem 18 we already know there is a promptly set in each of the corresponding diamond classes.
- Now we make such a cost function benign.

Theorem 27 (Greenberg,Nies/ Nies)**.** *Let $\mathcal{H}$ be either superlowness or superhighness. Then $\mathsf{SJT} \subseteq \mathcal{H}^\diamond$.*

For the proofs they build appropriate benign cost functions.
The superlow case we have done already: Each superlow set is ω-c.e., and if Y is ω-c.e. then c_Y is clearly benign.
For superhigh, Nies builds a c.f. c_Γ for each tt-reduction Γ. It deals with the ML-random sets Y such that $\emptyset'' = \Gamma(Y')$.

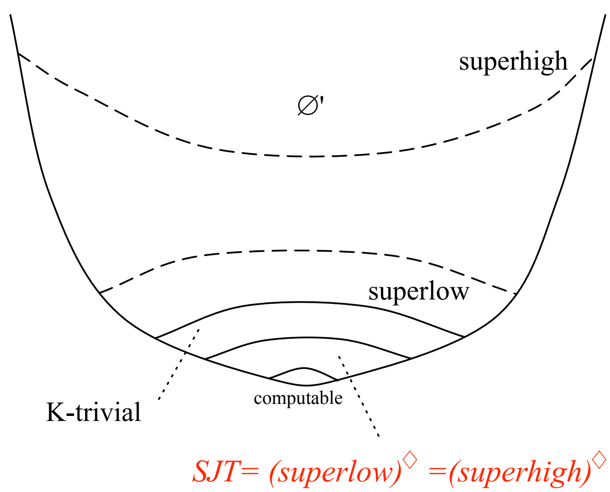
6.8 Conversely, these diamond classes are contained in SJT

The converse inclusion holds as well. Putting all together, we have

Theorem 28 (Greenberg, Hirschfeldt, Nies)**.** $\mathsf{SJT} = (\omega\text{−c.e.})^\diamond$ =superlow $^\diamond$= superhigh $^\diamond$.

To prove the remaining inclusions we use a "golden run" construction with infinitely many levels.

6.9 Diagram: *SJT* means computed by many oracles



6.10 Corollaries to the characterizations of SJT

Often new characterizations give new views of the class. We obtain:

- a new proof of the Cholak, Downey and Greenberg result that *SJT* induces an ideal in the c.e. Turing degrees (because every diamond class does that).
- a cost function construction (hence, injury-free) of a promptly simple set in *SJT* via the Hirschfeldt/MIller cost function $c_{\mathcal{H}}$ where $\mathcal{H} = \omega$-c.e., say.
  (Recall that if A obeys $c_{\mathcal{H}}$, then $A \in \mathcal{H}^{\Diamond} \subseteq$ *SJT*.)

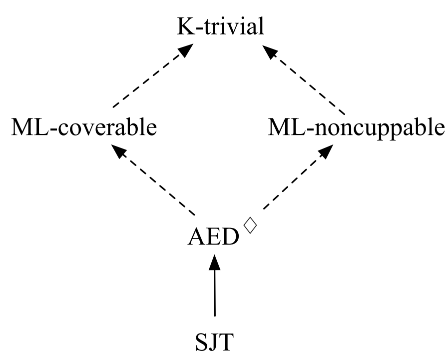
6.11 Open questions on classes between *SJT* and K-trivial

- No natural classes are currently known to lie properly between *SJT* and K-trivial
- A good candidate is *(AED)*$^{\Diamond}$. Here AED is the class of almost everywhere dominating sets Z of Dobrinen and Simpson: for almost all sets X, each function $f \leq_T X$ is dominated by a function $g \leq_T Z$. For the highness properties, there are proper implications

$$\text{Turing-complete} \Rightarrow \text{AED} \Rightarrow \text{superhigh.}$$

- For the corresponding diamond classes, Greenberg and Nies proved that *SJT* is properly contained in (AED)$^{\Diamond}$.
- However, (AED)$^{\Diamond}$ may coincide with K-trivial.
- This would imply that the classes *ML-coverable* and *ML-noncuppable* also coincide with K-trivial.

6.12 Classes of c.e. sets between *SJT* and K-trivial



(The dashed arrows may be coincidences.)

- A is ML-coverable if $A \leq_T Y$ for some ML-random $Y \not\geq_T \emptyset'$.
- A is ML-noncuppable if
  $\emptyset' \leq_T A \oplus Y$ for ML-random Y implies $\emptyset' \leq_T Y$.