Deciding Feasible-Function Existence for β-Normalized, Radius-1 LCLs on Directed Paths is in NEXPTIME

Abstract

We give a nondeterministic $2^{\mathrm{poly}(\beta)}$-time algorithm that decides, for a β-normalized, radius-1 LCL on globally oriented paths, whether there exists a feasible function. This decides both gaps $O(\log^* n)$ vs. $\Theta(n)$ (mid-consistency) and $O(1)$ vs. $\Omega(\log^* n)$ (bridging). Our proof is self-contained modulo standard pumping/replacement facts for path types; we restate the exact type bound, give a finite-checks lemma with an explicit z-bound $Z \leq \ell_{\mathrm{pump}}^2$, spell out the DP used by the verifier (with boundary handling and tight $O(k\beta^2)$ complexity), and prove soundness/completeness. We use Section 4 and Lemmas 10–15, 11, 12, as well as Theorems 8–9 from the reference.[1]

1 Model and Normalization (r=1)

We work on globally oriented paths. A β-*normalized* LCL is given by

$$\Sigma_{\mathrm{in}} = \{0,1\}, \quad |\Sigma_{\mathrm{out}}| = \beta, \quad C_{\mathrm{in-out}} \subseteq \Sigma_{\mathrm{in}} \times \Sigma_{\mathrm{out}}, \quad C_{\mathrm{out-out}} \subseteq \Sigma_{\mathrm{out}} \times \Sigma_{\mathrm{out}},$$

and a labeling is legal iff $(\mathrm{in}(v), \mathrm{out}(v)) \in C_{\mathrm{in-out}}$ for each node v and $(\mathrm{out}(u), \mathrm{out}(v)) \in C_{\mathrm{out-out}}$ for each directed edge $u \to v$. The input size is $N = \mathrm{poly}(\beta)$.

Types and pumping for $r = 1$. For a path $P = (u_1, \ldots, u_k)$ with $k \geq 4$, define $B_1 = \{u_1, u_k\}$ and $B_2 = \{u_2, u_{k-1}\}$. Two paths are *type-equivalent* if

[1] All background references (lemmas, constructions, and the two gap theorems) refer to the uploaded PDF: *The distributed complexity of locally checkable problems on paths is decidable.* See Section 4 (pp. 19–32), especially Lemmas 10–15 (types/pumping), Lemma 11 (replacement), Lemma 12 (type composition), and Theorems 8–9 (gap characterizations). :contentReference[oaicite:1]index=1

(i) inputs on $B_1 \cup B_2$ match and (ii) the same boundary-output assignments on $B_1 \cup B_2$ are extendible. Let Type(P) be the type. Counting: there are 2^4 choices for boundary inputs and a yes/no choice for each of β^4 boundary-output 4-tuples, so

$$M := |\{\text{Type}(P)\}| \;\le\; 2^4 \cdot 2^{\beta^4} \;=\; 2^{\beta^4+4} \;=\; 2^{\Theta(\beta^4)}.$$

Set $\ell_{\text{pump}} := M$. The DFA view of types and pumping/replacement lemmas are in Section 4: Lemma 12 (type composition), Lemmas 10–11 (replacement), and Lemmas 14–15 (pumping/periodicity).[2]

2 Contexts and Feasible Functions (no circularity)

Fix ℓ_{pump}. A *context* is any triple (w_1, S, w_2) with $|w_1|, |w_2| \in \{\ell_{\text{pump}}, \ell_{\text{pump}}+1\}$ and $S \in \Sigma_{\text{in}}^2$. We identify contexts by the *type of the concatenation* $w_1 S w_2$. Thus the set $\mathcal{C}$ of context types is a *subset* of the global type space, independent of f.

Definition 1 (Feasible function)**.** *A map* $f : \mathcal{C} \to \Sigma_{\text{out}}^2$ *is* feasible *if:*

- ***(F1) Mid-consistency.*** *For any contexts* $w_a S_1 w_b$ *and* $w_c S_2 w_d$*, the fixed outputs* $f(\text{Type}(w_a S_1 w_b))$ *on* S_1 *and* $f(\text{Type}(w_c S_2 w_d))$ *on* S_2 *extend to a legal labeling of* $w_a S_1\, w_b w_c\, S_2 w_d$*.*
- ***(F2) Bridging.*** *For any context* (w_1, S, w_2) *and all* $z \ge 1$*, the fixed outputs* $f(\text{Type}(w_1 S w_2))$ *on* S *extend to a legal labeling of* $w_1^z S w_2^z$*.*

These are precisely the certificates that characterize the two gaps on cycles/paths: (F1) $\Leftrightarrow O(\log^* n)$ vs. $\Theta(n)$, and (F1)+(F2) $\Leftrightarrow O(1)$ vs. $\Omega(\log^* n)$.[3]

3 Finite Checks for Bridging: a clean periodic bound

We now reduce the universal quantifier in (F2) to finitely many z.

Explicit statement of Lemma 15 and why it holds. In the reference, Lemma 15 states: for each nonempty word $w \in \Sigma_{\text{in}}^+$ there exist integers

[2] See Lemmas 10–15 and Lemma 12, Section 4 (pp. 20–23). :contentReference[oaicite:2]index=2

[3] See Theorems 8–9 (pp. 19–20, 27–32) for the constructions from feasible functions. :contentReference[oaicite:3]index=3

$a, b > 0$ with $a + b \le \ell_{\text{pump}}$ such that $\text{Type}(w^{at+b})$ is independent of $t \ge 0$.[4] *Why:* consider the finite set of types $\mathcal{T}$, $|\mathcal{T}| = \ell_{\text{pump}}$, and the self-map $F_w : \mathcal{T} \to \mathcal{T}$ that composes by w. The sequence $T_i := F_w^i(T_0)$ (here T_0 is the type of the empty prefix) has a repetition $T_p = T_q$ with $1 \le p < q \le \ell_{\text{pump}}$; setting $a = q - p$, $b = p$ gives $a + b = q \le \ell_{\text{pump}}$ and T_{at+b} constant for all $t \ge 0$. This is the standard orbit argument behind Lemma 15.

Lemma 1 (Finite z for bridging)**.** *Fix a context* (w_1, S, w_2)*. There exist parameters* (a_i, b_i) *for* w_i *with* $a_i+b_i \le \ell_{\text{pump}}$ *such that* $\text{Type}(w_i^{a_i t+b_i})$ *is* t*-invariant. Hence the pair* $\big(\text{Type}(w_1^z), \text{Type}(w_2^z)\big)$ *takes at most* $a_1 a_2 \le \ell^2_{\text{pump}}$ *distinct values, and by type composition (Lemma 12)* $\text{Type}(w_1^z S w_2^z)$ *also takes at most* ℓ^2_{pump} *values. Consequently, there exists* $Z \le \ell^2_{\text{pump}}$ *with*

$$\forall z \ge 1 \ \exists z' \in \{1, \dots, Z\} : \ \text{Type}(w_1^z S w_2^z) = \text{Type}(w_1^{z'} S w_2^{z'}).$$

Thus (F2) holds iff the partial outputs on S *extend for all* $z \in \{1, \dots, Z\}$*.*

Proof. Apply Lemma 15 to each w_i (periodicity), and use Lemma 12 (type composition) to conclude that the type of $w_1^z S w_2^z$ is determined by the pair of side types, whence the bound by pigeonhole. Extendibility depends only on the resulting type and the fixed outputs on S (by the replacement lemma, Lemma 11), so a single representative per type suffice.[5] □

How we pick Z in practice. We will *use the worst-case bound* $Z = \ell^2_{\text{pump}}$; this keeps the algorithm simple and does not affect the asymptotic bound. (Optional refinement: for a fixed context, compute the cycle structure of F_{w_1} and F_{w_2} in the type DFA and set Z to the lcm of periods times the max preperiod—still $\le \ell^2_{\text{pump}}$—which can reduce constants while remaining $2^{\text{poly}(\beta)}$ overall.)

4 Deterministic DP for extendibility (with precise complexity)

All checks reduce to deciding if a partially labeled path admits a completion. For $r = 1$:

[4] Lemma 15 (p. 23). :contentReference[oaicite:4]index=4

[5] Lemma 12 (type composition) and Lemma 11 (replacement) are in Section 4 (pp. 21–22, 20–21). :contentReference[oaicite:5]index=5

Layered automaton. Given a path $P = (v_1, \ldots, v_k)$ and a set F of forced positions with outputs $o_i \in \Sigma_{\text{out}}$, build k layers, one per position. Layer i keeps the set

$$L_i \;:=\; \{x \in \Sigma_{\text{out}} : \;(\,\text{in}(v_i), x\,) \in C_{\text{in–out}} \text{ and if } i \in F,\; x = o_i\}.$$

Insert a directed edge $x \to y$ from layer i to $i+1$ iff $(x, y) \in C_{\text{out–out}}$. There is a legal completion iff the layered graph has a path from some state in layer 1 to some state in layer k.

Running time and memory. Let $k = |P|$. Constructing layers and edges takes $O(k \cdot |C_{\text{out–out}}|)$ time, which is $O(k\beta^2)$ in the worst case, and $O(k\beta)$ memory. Forced nodes *reduce* the number of states (they do not increase complexity): the check is a standard reachability on this DAG, also in $O(k\beta^2)$ time.

5 Nondeterministic algorithm and correctness

Precomputation (deterministic, $2^{\text{poly}(\beta)}$)

1. Enumerate all types $\mathcal{T}$ by exploring the boundary-extendibility DFA (Lemma 12), set $\ell_{\text{pump}} = |\mathcal{T}| \le 2^{\Theta(\beta^4)}$.[6]
2. For each type $t \in \mathcal{T}$ and each boundary assignment on $B_1 \cup B_2$, precompute *extendible?* via DP on a fixed representative of type t (length $O(\ell_{\text{pump}})$).
3. Compute the set $\mathcal{C} \subseteq \mathcal{T}$ of *context types*: those realized by some $w_1 S w_2$ with $|w_i| \in \{\ell_{\text{pump}}, \ell_{\text{pump}}+1\}$ and $|S| = 2$. Note $|\mathcal{C}| \le \ell_{\text{pump}}$.

Guess

Nondeterministically guess $f(\tau) \in \Sigma^2_{\text{out}}$ for every $\tau \in \mathcal{C}$.

Verify (deterministic, $2^{\text{poly}(\beta)}$)

(V1) Mid-consistency. For each ordered pair of contexts $(w_a S_1 w_b,\, w_c S_2 w_d)$, run the DP on $w_a S_1\, w_b w_c\, S_2 w_d$ with the two 2-node windows S_1, S_2 forced to f; accept only if the DP succeeds. *Why no extra buffers:* by definition of contexts,

[6] Type DFA and counting bound: Lemma 13 (p. 22) specialized to r=1 gives $|\mathcal{T}| \le 2^4 \cdot 2^{\beta^4}$. :contentReference[oaicite:6]index=6

w_b, w_c each has length ℓ_{pump} or $\ell_{\text{pump}}+1$, so the underlined bridge $w_b w_c$ is already a long buffer for the replacement lemmas.

(V2) Bridging. For each context (w_1, S, w_2) and each $z = 1, \ldots, Z$ with $Z = \ell_{\text{pump}}^2$ from Lemma 1, run the DP on $w_1^z S w_2^z$ with the two nodes of S forced to f. (All seams internal to w_1^z and w_2^z are covered by this DP; no separate "seam test" is needed.)

Theorem 1 (Soundness and completeness)**.** *If the verifier accepts some guessed f, then f satisfies (F1) and (F2); conversely, if a feasible f exists, some guess passes (V1)–(V2).*

Proof sketch. (V1) is exactly (F1). For (F2), Lemma 1 shows it suffices to check $z \leq Z$; (V2) performs those checks with a DP that validates *all* local constraints, including seams created by repetition. Conversely, a feasible f passes (V1) and by Lemma 1 also passes (V2) for all $z \leq Z$, hence for all z. □

Complexity and main theorem

Precomputation explores $\mathcal{T}$ and runs DPs on $O(\ell_{\text{pump}})$-length words, all in $2^{\text{poly}(\beta)}$ time. Verification runs a polynomial number of DPs on strings of length $O(\ell_{\text{pump}}^3)$ (worst case, from $z \leq \ell_{\text{pump}}^2$ and context width $O(\ell_{\text{pump}})$), each in $O(k\beta^2)$; altogether $2^{\text{poly}(\beta)} = 2^{\text{poly}(N)}$ time.

Theorem 2 (Main)**.** Feasible-Function Existence *for β-normalized, radius-1 LCLs on directed paths is in* NEXPTIME.

6 Connection to the two gaps

We use the standard implications:

- (F1) $\Rightarrow O(\log^* n)$ and failure of (F1) $\Rightarrow \Theta(n)$ (Theorem 8, with MIS-based decomposition, Lemma 16).[7]
- (F1)+(F2) $\Rightarrow O(1)$ and failure of (F2) $\Rightarrow \Omega(\log^* n)$ (partition Lemmas 20–22 and fill-in Lemmas 26–27 culminating in Theorem 9).[8]

[7] Theorem 8 and Lemma 16 (pp. 19–20, 22–24). :contentReference[oaicite:7]index=7

[8] Lemmas 20–22 and Lemmas 26–27 with Theorem 9 (pp. 27–32). :contentReference[oaicite:8]index=8

7 Positioning and bounds (summary)

Known lower bound	PSPACE-hard to distinguish $O(1)$ vs. $\Omega(n)$ on oriented paths (Sec. 3, Thr
This work (upper bound)	NEXPTIME for Feasible-Function Existence under poly(β) encoding.
Tightness	Open under this encoding (EXPTIME? PSPACE?).

See pp. 14–16 for hardness and β-normalized constant-time lower bounds.[9]

Notes

Non-circularity. Context types are defined as a subset of global types (those realized by $w_1 S w_2$ with $|w_i| \in \{\ell_{\text{pump}}, \ell_{\text{pump}}+1\}$), independently of f.

General $r > 1$ (remark). All arguments lift to any fixed radius r: the type space size becomes $2^{\text{poly}(\beta^{O(r)})}$ (cf. Lemma 13), the DP has $\beta^{O(r)}$ states per layer, and Lemma 1 still yields a finite Z from the same periodicity argument.[10]

A Optional algebraic view (boolean matrices with the seam)

Let $A \in \{0,1\}^{\beta\times\beta}$ be the adjacency of $C_{\text{out–out}}$: $A[x,y] = 1 \iff (x,y) \in C_{\text{out–out}}$. For a word w, define $C_w \in \{0,1\}^{\beta\times\beta}$ by $(C_w)_{ij} = 1$ iff there is a legal labeling of w whose first/last outputs are i/j and nodewise $(\text{in}(\cdot), \text{out}(\cdot)) \in C_{\text{in–out}}$. Then boolean multiplication gives the exact composition law

$$C_{xy} \;=\; C_x\, A\, C_y, \qquad \text{hence} \qquad C_{w^z} \;=\; C_w\, (AC_w)^{z-1}.$$

Proof: $(i,k) \in C_{xy}$ iff there exists j such that $(i,j) \in C_x$, $(j,k) \in C_y$, and the seam output pair is legal, i.e. $(j,\cdot)$ to $(\cdot,k)$ goes through an edge in A—which is exactly the middle factor. This algebraic view is not needed for the algorithm (the DP already checks seams), but it explains the role of the seam explicitly.

[9] Hardness and β-normalized lower bounds: Theorems 4–5 (pp. 14–16). :contentReference[oaicite:9]index=9

[10] Lemma 13 (p. 22) gives the general type bound. :contentReference[oaicite:10]index=10

B NEXPTIME-hardness (explicit β-normalized radius-1 model)

We prove that Feasible-Function Existence (mid-consistency (F1) only) for β-normalized, radius-1 LCLs on globally oriented paths is NEXPTIME-hard. Together with Theorem 2, this yields NEXPTIME-completeness. The reduction is from *Succinct-3SAT*: a circuit C of size s defines, for each index $j \in \{0,1\}^B$, a 3-clause clause(j) over variables $x_1, \ldots, x_{2^B}$; the question is whether the 2^B-variable CNF Φ_C is satisfiable.

What we use from the toolbox. We rely on (i) the *types/replacement/pumping* machinery (Lemmas 10–12, 14–15 in §4 of the reference), and (ii) the *error-chain* technique to make malformed encodings harmless (as in §3 of the reference).[11]

B.1 Target: FFE(F1) for β-normalized r=1

Recall our feasible-function existence problem FFE(F1): given $\Sigma_{\text{in}} = \{0,1\}$, $|\Sigma_{\text{out}}| = \beta$, $C_{\text{in–out}} \subseteq \Sigma_{\text{in}} \times \Sigma_{\text{out}}$, $C_{\text{out–out}} \subseteq \Sigma_{\text{out}} \times \Sigma_{\text{out}}$, decide if there is a map $f : \mathcal{C} \to \Sigma_{\text{out}}^2$ (on context types) satisfying mid-consistency (F1), cf. the definition in the upper bound section.

B.2 Source and parameters

Let C be a succinct 3CNF generator of size s. Set

$$B = s^{c_0}, \qquad \beta = s^{c_1},$$

for large enough absolute constants $c_0 \ll c_1$ fixed below. We will build, in poly(s) time, a β-normalized LCL $P_C = (\Sigma_{\text{in}}, \Sigma_{\text{out}}, C_{\text{in–out}}, C_{\text{out–out}})$ such that

$$\Phi_C \text{ is satisfiable} \iff \text{FFE(F1) holds for } P_C. \tag{1}$$

Since $|\Sigma_{\text{out}}| = \beta = \text{poly}(s)$, the *type space* size $\ell_{\text{pump}} := |\{\text{Type}(\cdot)\}|$ satisfies $\ell_{\text{pump}} \leq 2^{\Theta(\beta^4)} = 2^{\text{poly}(s)}$, and hence the contexts in $\mathcal{C}$ (those realized by $w_1 S w_2$ with $|w_i| \in \{\ell_{\text{pump}}, \ell_{\text{pump}}+1\}$) cover exponentially many patterns, which we will exploit.[12]

[11] Types, replacement, pumping: §4, Lemmas 10–15; error-chains and bounded-tape execution checking: §3. :contentReference[oaicite:0]index=0

[12] Type bound and composition/replacement: §4, Lemma 13 (bound), Lemma 12 (composition), Lemma 11 (replacement), and Lemmas 14–15 (pumping/periodicity). :contentReference[oaicite:1]index=1

B.3 The block language and the verifiers

We describe Σ_{out} and the two verifiers in the β-normalized form. The construction follows the standard "copy input and check a tableau by nearest-neighbor rules" recipe, together with the *error-chain* gadget that guarantees malformed inputs never constrain f.[13]

Alphabet Σ_{out}. Each output symbol encodes a constant number of *tracks*:

(T1) a bit in which must copy the node's input bit (enforced by $C_{\text{in-out}}$);

(T2) a *role* from a constant set $\{\mathsf{HEAD}, \mathsf{RID}, \mathsf{GID}, \mathsf{RUN}, \mathsf{PAD}\}$ and a constant-size *phase* (synchronization);

(T3) three *colors* $\{\text{RED}, \text{GRN}, \text{BLU}\}$ (used only at the 2-node window S);

(T4) a special *plumbing* symbol $\perp$;

(T5) a bounded collection of *error* symbols (the error-chain alphabet) as in §3;

(T6) (for RUN) the working alphabet for a fixed universal TM U running in $B^{O(1)}$ time.

Thus $|\Sigma_{\text{out}}| = \beta = \text{poly}(s)$.

The intended block. A *well-formed block* (a regular language over Σ_{out}) has the form

$$\mathsf{HEAD}_C \ \ \mathsf{RID} \ \ \mathsf{GID} \ \ \mathsf{RUN} \ \ \mathsf{PAD},$$

where:

- HEAD_C is a self-delimiting binary header that literally *stores* C on the in track (copied to output by (T1)) and marks the start of a block.
- RID is a B-bit *variable index* region; GID is a B-bit *clause index* region.
- RUN is a row-by-row encoding of a *tableau* of the fixed machine U which, given $(C, \mathsf{RID}, \mathsf{GID})$, runs for $B^{O(1)}$ steps and *computes the three literals*

[13] The radius-1 checking of a 1D flattening of a time poly(B) computation and the error-chain idea are lifted verbatim from §3 (LBA encoding + locally checkable refutations). :contentReference[oaicite:2]index=2

of clause(GID) *and their signs*, and *writes* three length-B *one-hot vectors* $V^{(+)}$, $V^{(-)}$ and $V^{(\perp)}$ on dedicated subtracks:

$$V^{(+)}[i] = 1 \iff x_i \text{ occurs positively in clause}(\mathsf{GID}), \quad V^{(-)}[i] = 1 \iff \neg x_i \text{ occurs},$$

and $V^{(\perp)}$ is always all zero (reserved for BLU).

- In *variable* blocks we also run U on (C, RID) to materialize a *variable one-hot* vector $\mathsf{Hot}_{\mathsf{RID}} \in \{0,1\}^B$ with $\mathsf{Hot}_{\mathsf{RID}}[i] = 1$ iff $i = \mathsf{RID}$.

The adjacency table $C_{\text{out–out}}$ permits exactly nearest-neighbor pairs consistent with (i) the grammar above (placement of regions, row separators, phases), and (ii) single-step correctness of U's transition across each row seam; malformed pieces are *locally refutable* by emitting an *error-chain* (as in §3), which we also allow in $C_{\text{out–out}}$ and which always leads to a successful completion.[14]

The window S and activation. The two nodes of the window S (in contexts $w_1 S w_2$) may be labeled at f's discretion by one of the *colors* $\{\text{RED}, \text{GRN}, \text{BLU}\}$ *or* by $\perp$. No other output is allowed at S. The out–out grammar is arranged so that:

> *The choice $\perp$ at S is always extendible. A non-$\perp$ color at S is extendible* only *if somewhere inside w_1 (resp. w_2) there is a well-formed block whose HEAD_C matches the fixed header for C, and whose RID (resp. GID) and tableau are valid.*

We say such contexts are *active*; all other contexts are *inactive*. Inactive contexts are thus tautologically satisfied by setting $f(\cdot) = \perp$.

Bridge semantics (the "clause satisfied by variable" test). For an *active* ordered pair (variable on the left, clause on the right), the bridge (F1) instance $w_a S_1 w_b w_c S_2 w_d$ is allowed to complete *iff*

$$\exists i \in [B] \text{ with } \mathsf{Hot}_{\mathsf{RID}}[i] = 1 \text{ and } \begin{cases} V^{(+)}[i] = 1 & \text{if } f(\text{Type(left)}) = \text{RED}, \\ V^{(-)}[i] = 1 & \text{if } f(\text{Type(left)}) = \text{GRN}, \\ V^{(\perp)}[i] = 1 & \text{if } f(\text{Type(left)}) = \text{BLU}. \end{cases}$$

[14] Exactly as in §3: a fixed finite set of error types and local rules allow every malformed tableau to be disproved by a short, locally checkable chain; on well-formed encodings such chains cannot be produced. :contentReference[oaicite:3]index=3

This ”$\exists i$” is checked by a standard radius-1 *witness chain*: upon a color choice at S_1 and S_2, the out–out grammar enables a path that (i) guesses an index i, (ii) walks to bit i inside the left block and requires $\mathsf{Hot}_{\mathsf{RID}}[i] = 1$, and (iii) walks to bit i inside the right block and requires the corresponding $V^{(\cdot)}[i] = 1$. If any subcheck fails, the chain gets stuck (no completion). This is exactly the same local ”existential witness” mechanism as in the error-chain gadget, so it is implementable with radius 1 and a constant alphabet overhead.[15]

All other ordered pairs of context types are trivialized: either at least one side is inactive (then $\perp$ suffices), or the role pairing is not (variable, clause), in which case the grammar *ignores* the colors and accepts with $\perp$ buffers.

B.4 Existence of exponentially many active context types

Because $\ell_{\text{pump}} = 2^{\text{poly}(s)}$ and our header+index+run occupy $O(s) + O(B) + B^{O(1)} = \text{poly}(s)$ symbols, by the pumping lemmas there are *context types* that (i) have a valid block with any desired $\mathsf{RID} \in \{0,1\}^B$ on the left and (ii) context types that have a valid block with any desired GID on the right. Moreover, types that differ in RID (or in GID) are *distinct* because, by design, extendibility from S under a color differs for some bridge depending on the bit pattern (the witness chain succeeds or not). Hence there are $2^B = 2^{\text{poly}(s)}$ distinct *variable* types $\{\text{Var}_i\}$ and 2^B distinct *clause* types $\{\text{Cl}_j\}$ present in $\mathcal{C}$.[16]

B.5 Correctness of the reduction

We prove (1).

Lemma 2 (Completeness)**.** *If Φ_C is satisfiable, then there exists $f : \mathcal{C} \to \Sigma_{\text{out}}^2$ satisfying (F1) for P_C.*

Proof. Fix a satisfying assignment $\alpha : \{x_1, \ldots, x_{2^B}\} \to \{0,1\}$. Define f as follows.

- For each *variable* type Var_i, set $f(\text{Var}_i) = \text{RED}$ if $\alpha(x_i) = 1$ and $f(\text{Var}_i) = \text{GRN}$ if $\alpha(x_i) = 0$ (either orientation of S may be chosen; the grammar handles both).

[15]Witness chains (existential pointers to positions with local checking) are the same combinatorics as the §3 error pointers; we reuse their radius-1 implementation. :contentReference[oaicite:4]index=4

[16]Realization and stability of these patterns inside w_b or w_c follow from type composition and replacement; lengths are padded using pumping (§4, Lemmas 11–12, 14–15). :contentReference[oaicite:5]index=5

- For each *clause* type Cl_j, pick any *true* literal of clause(j) under α and set $f(\mathrm{Cl}_j)$ to its color: RED for a positive literal, GRN for a negative literal (use BLU only if desired as a dummy literal, with $V^{(\perp)} \equiv 0$ never satisfying witnesses).

- For all *inactive* types, set $f(\tau) = \perp$.

Consider any ordered pair of contexts in the (F1) quantification. If at least one side is inactive or the roles are not (variable, clause), the bridge is trivial by construction. If the pair is $(\mathrm{Var}_i, \mathrm{Cl}_j)$, then by the definition of the RUN subtracks we have $\mathsf{Hot}_{\mathsf{RID}}[i] = 1$ inside Var_i, and $V^{(+)}[i] = 1$ (resp. $V^{(-)}[i] = 1$) if we chose RED (resp. GRN) at Cl_j and the picked literal is x_i (resp. $\neg x_i$). Since $f(\mathrm{Cl}_j)$ is a *true* literal of clause(j), such a choice exists, and the bridge DP succeeds via a witness chain at index i. Hence every (F1) check passes. □

Lemma 3 (Soundness)**.** *If there exists f satisfying (F1) for P_C, then Φ_C is satisfiable.*

Proof. From f, read an assignment α by

$$\alpha(x_i) = \begin{cases} 1, & f(\mathrm{Var}_i) = \mathrm{RED}, \\ 0, & f(\mathrm{Var}_i) = \mathrm{GRN}, \\ \text{arbitrary}, & f(\mathrm{Var}_i) = \mathrm{BLU} \text{ (we may fix it to 0)}, \end{cases}$$

for all variable types Var_i (if $f(\mathrm{Var}_i) = \perp$, the type is inactive, impossible by construction of Var_i). Fix any clause index j. If $f(\mathrm{Cl}_j) = \perp$ then Cl_j is inactive, which contradicts the existence of the header+RUN in Cl_j; hence $f(\mathrm{Cl}_j) \in \{\mathrm{RED}, \mathrm{GRN}, \mathrm{BLU}\}$. Consider the (F1) bridge for the ordered pair $(\mathrm{Var}_i, \mathrm{Cl}_j)$ where i ranges over all variables that *appear* in clause(j). The out–out grammar allows completion *only* if there exists an index i with $\mathsf{Hot}_{\mathsf{RID}}[i] = 1$ in Var_i and the corresponding $V^{(+)}[i] = 1$ (for RED) or $V^{(-)}[i] = 1$ (for GRN) in Cl_j. Because (F1) must hold for *every* such ordered pair, the choice $f(\mathrm{Cl}_j)$ must be compatible with at least one Var_i that actually occurs in clause(j); therefore the picked color is a *true* literal under α. Since j was arbitrary, every clause has a true literal: α satisfies Φ_C. □

Lemma 4 (Inactive contexts never constrain f)**.** *If a context lacks a well-formed header+indices+valid RUN (on either side), then* any *output at S except $\perp$ makes the (F1) bridge DP fail, while $\perp$ always extends. Thus such contexts are irrelevant and do not affect the existence of f.*

Proof. Both statements follow from the grammar: non-$\perp$ at S *activates* witness chains that must eventually check bits inside a block; in the absence of a valid block the chain gets stuck and no completion exists. Conversely, when S is $\perp$, the out–out table contains a self-looping $\perp$ buffer language that trivially completes the bridge (the same plumbing as in the upper-bound DP). The error-chain rules guarantee that malformed fragments admit locally checkable refutations and hence do not impose constraints.[17] $\square$

Theorem 3 (FFE(F1) is NEXPTIME-complete)**.** *For β-normalized, radius-1 LCLs on globally oriented paths (explicit model),* Feasible-Function Existence *is* NEXPTIME*-complete.*

Proof. Membership is Theorem 2. For hardness, the reduction $C \mapsto P_C$ runs in poly(s) time and produces $|\Sigma_{\text{out}}| = \beta = \text{poly}(s)$ with adjacency tables of size $\Theta(\beta^2)$. By Lemmas 2–4, (1) holds. Hence FFE(F1) is NEXPTIME-hard. $\square$

B.6 From (F1) to the full feasible-function (F1)+(F2)

To pass from mid-consistency to the *full* feasible-function existence (bridging included), add a neutral symbol $\perp$ (already present) and allow, in $C_{\text{out–out}}$, arbitrary $\perp$-labelings on repetitions w_1^z and w_2^z. Then (F2) is vacuous: for every context (w_1, S, w_2) and every $z \geq 1$, the partial labeling that fixes S to $f(\text{Type}(w_1 S w_2))$ extends by filling w_i^z with $\perp$. Hence:

Corollary 1 (Classification hardness)**.** *Deciding the distributed class ($O(1)$ vs. $O(\log^* n)$ vs. $\Theta(n)$) via feasible functions is* NEXPTIME*-hard in the explicit β-normalized, radius-1 model.*

B.7 Size and radius bookkeeping

All rules are radius 1. The only growth parameter is $\beta = \text{poly}(s)$, which controls the number of output symbols needed for (i) the fixed role/phase tracks, (ii) the three color tokens $\{\text{RED}, \text{GRN}, \text{BLU}\}$ and $\perp$, (iii) the finite error alphabet, and (iv) the universal machine U work alphabet. The header+indices+RUN occupy $O(s) + O(B) + B^{O(1)} = \text{poly}(s)$ positions,

[17] See §3 for the error-chain "disprover" gadget accepted by the verifier, ensuring malformed encodings always have completions, and §4 for the "plumbing" view of the DP. :contentReference[oaicite:6]index=6

whereas the pump length satisfies $\ell_{\text{pump}} = 2^{\Theta(\beta^4)} = 2^{\text{poly}(s)}$, which guarantees that active contexts containing such blocks exist and are stable under replacement/pumping.[18]

[18] All quantitative bounds (type count, replacement, pumping) follow from §4; the error alphabet and local checking of bounded-time executions are from §3. :contentReference[oaicite:7]index=7