(1) Set-up:

n: number of atoms

$\mathcal{C}$: a set of clauses in n variables

π: a tree-like DNF-R refutation (i.e. $R^*(id)$-refutation) of $\mathcal{C}$

k: the number of steps in π

s: size of π

c: a parameter bounding the number of conjunctions in any line in π

We allow as initial clauses also all clauses containing some $\{\ell, \neg\ell\}$.

(2) Lemma: *Assume the set-up (1). Then $\mathcal{C}$ has an R^*-refutation π^* (i.e. tree-like R) with at most $n^{O(c \log k)}$ steps.*

The lemma follows from Lemmas (5) and (6) below. Note that it is not claimed that π^* is balanced.

(3) Remark: Lemma (2) implies an analogous statement about depth $d+1$ LK refutations with k and n in the estimate replaced by $O(s)$: use limited extension for all depth $\leq d$ formulas in π to reduce to $R^*(id)$. Then substitute in π^* back formulas for the corresponding extension atoms (this changes n by adding the number of extension atoms and k when deriving a formula from the associated extension atom - in both case it is bounded above by $O(s)$).

(4) Game: Consider the Prover-Liar game where Prover asks for the truth-value of a clause D and the Liar either replies *true*, in which case D is added to his set $\mathcal{D}$ of replies, or *false*, in which case all singleton clauses $\{\neg\ell\}$, all $\ell \in D$, are added. The game stops with Prover winning the moment $\mathcal{C} \cup \mathcal{D}$ contains some clause D and at the same time also all $\{\neg\ell\}$, all $\ell \in D$.

(5) Lemma: *Assume that Prover has a winning strategy S that wins over each Liar in at most r rounds. Then $\mathcal{C}$ has an R^*-refutation with at most $(n+1)^{r+1}$ steps.*

Proof :

Think of S as of a binary tree branching according to Liar's answers. For a partial path σ in S ending in vertex v_σ denote:

S_σ: the subtree with root v_σ,

$\mathcal{D}_\sigma$: Liar's answers given on path σ,

r_σ: the height of S_σ.

Note that for the empty path Λ, $S_\Lambda = S$, $\mathcal{D}_\Lambda = \emptyset$ and $r_\Lambda = r$.

We shall prove by induction on r_σ the following

Claim: *$\mathcal{C} \cup \mathcal{D}_\sigma$ has an R^*-refutation ρ_σ with at most $(n+1)^{r_\sigma+1}$ steps.*

Assume $r_\sigma = 0$, i.e. σ is a complete path in S. By the definition of the game the set $\mathcal{D}_\sigma$ contains some clause D and also all singleton clauses $\{\neg\ell\}$, all $\ell \in D$. Define ρ_σ to be $|D| \leq n$ resolutions removing from D subsequently all literals.

Assume $r_\sigma > 0$. Let D be the clause S asks at node v_σ and denote by $S_{\sigma 1}$ the subtree corresponding to the positive answer (hence $D \in \mathcal{D}_{\sigma 1}$) and by $S_{\sigma 0}$ the negative subtree (hence $\{\neg l\} \in \mathcal{D}_{\sigma 0}$ for all $\ell \in D$). Let ρ_1 and ρ_0, resp., be the two R^*-refutations attached to the two subtrees satisfying the induction assumption, having k_0 and k_1 steps, respectively.

Change in ρ_0 all $\{\neg\ell\}$, $\ell \in D$, into $\{\neg\ell, \ell\}$ and carry the extra literals along the whole ρ_0: this yields an R^*-derivation ρ_0' of D from $\mathcal{C} \cup \mathcal{D}_{\sigma 0}$ with the same number of steps as in ρ_0.

For all $\ell \in D$ construct an R^*-derivation $\rho_{1,\ell}$ of $\{\neg\ell\}$ from $\mathcal{C} \cup \mathcal{D}_\sigma$ as follows: add to each occurrence of D as initial clause in ρ_1 literal $\neg\ell$ (hence the clause becomes an instance of free logic initial clauses - see (1)) and carry it along. Note that all $\rho_{1,\ell}$ have the same number of steps as ρ_1.

The resulting R^*-refutation ρ_σ starts as ρ_0' deriving D and the using subsequently all subproofs $\rho_{1,\ell}$ ($|D|$ of them) to cut out all literals $\ell \in D$. The number of steps in ρ_σ is bounded above by

$$|D| \cdot k_1 + k_0 \leq nk_1 + k_0 \leq n(n+1)^{\rho_\sigma} + (n+1)^{\rho_\sigma} \leq (n+1)^{\rho_\sigma + 1} .$$

This proves the claim.

The lemma follows from the claim for $\sigma := \Lambda$.

q.e.d.

(6) Lemma: *Under the set-up (1) there is a winning strategy for Prover that wins over any Liar in at most $O(c \log k)$ rounds.*

Proof :

Note that Prover can find the truth value of a DNF-clause by asking separately for the truth values of all (clauses that are negations of) conjunctions in the clause (at most c) and then for the truth value of the sub-clause consisting of the remaining literals. Use this to navigate in π in a Spira-like fashion. Hence Prover needs to ask for the values of $O(\log k)$ DNF-clauses, getting each by asking for the values of $\leq c+1$ ordinary clauses.

q.e.d.