A note on SLDNF-resolution

Wilfried Buchholz
Mathematisches Institut
Universität München
Theresienstr. 39
D-80333 München
email: buchholz@rz.mathematik.uni-muenchen.de

1 Introduction

In this paper, starting from Definition 8.8 in [3], we design a new (and as it seems to us rather compact and elegant) notion of SLDNF-tree together with the appropriate definition of fairness such that the following "strong completeness theorem" can be established:

Theorem
Let S be an input/output specification, P an S-correct logic program, T a fair SLDNF-tree for G w.r.t. P.

a) If G is S-correct and $\mathrm{comp}(P) \models_3 G\sigma$ then T is successfull and yields a computed answer substitution θ such that $G\sigma$ is an instance of $G\theta$.

b) If G is S-closed and $\mathrm{comp}(P) \models_3 \neg G$ then T is finitely failed.

This theorem extends Stärk's completeness result from [7] (cf. also Theorem 8.52 in [3]) where only the existence of *some* successfull (finitely failing, resp.) SLDNF-tree is proved. It also partly extends Drabent's result in [4], since it refers to the extension of SLDNF-resolution (used in [7] and [3]) where also non-ground negative literals may be selected. Actually we do not prove the Theorem as it stands but refer to the intermediate sets YES_P and NO_P introduced in [7]. These sets correspond to a sequent calculus introduced by the author in [1]. From [7] it follows that for S-correct P we have $\mathrm{comp}(P) \models_3 G \Leftrightarrow G \in \mathrm{YES}_P$, for all S-correct G, and $\mathrm{comp}(P) \models_3 \neg G \Leftrightarrow G \in \mathrm{NO}_P$, for all S-closed G.

The second purpose of this note is to demonstrate the usefulness (at least for theoretical investigations) of a certain notion of SLD(NF)-derivation implicitely introduced in ([2], pg. 495, Definition 3.18) and ([5], pp. 151ff). Instead of a derivation $G_0 \xrightarrow{\alpha_1} G_1 \cdots G_i \xrightarrow{\alpha_{i+1}} G_{i+1} \cdots$ in the usual sense we consider the corresponding sequence of resultants $\Phi_i := G_i \rightarrow G_0\alpha_1 ... \alpha_i$. The main advantage of this approach is that the relation between two adjacent members of the sequence $\Phi_0, \Phi_1, \ldots$ can be described *locally* without a condition on the "previously released variables" (or some "standardizing apart" condition): *Φ_{i+1} is a most general unrestricted resolvent of Φ_i (w.r.t. some rule K and position ν in G_i).*

2 Closure properties of the sets YES_P, NO_P

Basic Notions
We use the following syntactic variables:
$\sigma, \tau, \rho, \theta, \vartheta$ for substitutions (Subst := set of all substitutions),
A, B for *atoms*,
$L, L_1, L_2, \ldots$ for *literals*, i.e. atoms A and negated atoms $\neg A$,
G, H, Q, Π for *queries*, i.e. formulas of the shape $L_1 \wedge \ldots \wedge L_n$ (the empty query is denoted by $\Box$),
K for *rules*, i.e. formulas of the shape $Q \rightarrow A$ ($\Box \rightarrow A$ is identified with A).

For (quantifierfree) formulas F, F':
$F \geq F' \;:\Leftrightarrow\; F' \leq F \;:\Leftrightarrow\; F'$ is an instance of F, i.e. $\exists\sigma(F\sigma = F')$.
$\mathrm{var}(F)$:= set of all variables occurring in F.

A rule $\Pi \rightarrow B$ is *applicable* to an atom A iff A and B have a common instance, i.e. if $\exists\sigma, \rho(\, B\sigma = A\rho\,)$. Remark: K is applicable to A iff $\exists Q, \rho(\, K \geq Q \rightarrow A\rho\,)$.

A *logic program* is a finite set of rules.
If P is a logic program then
$P' := \{K\sigma : K \in P \;\&\; \sigma \in \mathrm{Subst}\}$,
$P(A) := \{K \in P : K$ is applicable to $A\}$.

Expressions of the shape $G \wedge * \wedge G'$ are called *q-forms*.
If $\Gamma = G \wedge * \wedge G'$ is a q-form then
$\Gamma\sigma := G\sigma \wedge * \wedge G'\sigma$,
$\Gamma[Q] := G \wedge Q \wedge G'$,
$\Gamma[\,] := \Gamma[\Box] = G \wedge G'$,
$\nu(\Gamma) := i$ where $G = L_1 \wedge \ldots \wedge L_{i-1}$.

We use $\Gamma, \Gamma', \ldots$ as syntactic variables for q-forms.

From now on let P be an arbitrary but fixed logic program.

Definition 1 (Inductive definition of the sets YES_P^n and NO_P^n)

(Y1) $\Box \in \mathrm{YES}_P^n$

(Y2) $Q \rightarrow A \in P' \;\&\; \Gamma[Q] \in \mathrm{YES}_P^n \implies \Gamma[A] \in \mathrm{YES}_P^{n+1}$

(Y3) $\Gamma[\,] \in \mathrm{YES}_P^k \;\&\; A \in \mathrm{NO}_P^m \implies \Gamma[\neg A] \in \mathrm{YES}_P^{k+m+1}$

(N1) $(\forall\rho, Q)(\, Q \rightarrow A\rho \in P' \;\Rightarrow\; \Gamma\rho[Q] \in \mathrm{NO}_P^n\,) \implies \Gamma[A] \in \mathrm{NO}_P^{n+1}$.

(N2) $A \in \mathrm{YES}_P^n \implies \Gamma[\neg A] \in \mathrm{NO}_P^{n+1}$.

Corollary 2 $\mathrm{YES}_P^n \subseteq \mathrm{YES}_P^{n+1}$ and $\mathrm{NO}_P^n \subseteq \mathrm{NO}_P^{n+1}$.

Definition 3 $\mathrm{YES}_P := \bigcup_{n<\omega} \mathrm{YES}_P^n$, $\mathrm{NO}_P := \bigcup_{n<\omega} \mathrm{NO}_P^n$.

Remark 4 Our (Y3) differs a little bit from (Y3) in [7], but one easily verifies that the resulting sets YES_P, NO_P are the same as in [7].

Lemma 5 a) $G \in \mathrm{YES}^n_P$ $(\mathrm{NO}^n_P) \implies G\sigma \in \mathrm{YES}^n_P$ (NO^n_P).

b) $\Gamma[H] \in \mathrm{YES}^n_P \iff \exists k, m(n = k{+}m \;\&\; \Gamma[\,] \in \mathrm{YES}^k_P \;\&\; H \in \mathrm{YES}^m_P\,)$.

c) $\Gamma[\,] \in \mathrm{NO}^n_P \implies \Gamma[H] \in \mathrm{NO}^n_P$.

Proof by induction on n:
a) We only consider (N1). In all other cases the claim follows immediately from the IH (or is trivial). Let $G = \Gamma[A]$ and $(\forall\rho, Q)(\, Q{\to}A\rho \in P' \Rightarrow \Gamma\rho[Q] \in \mathrm{NO}^n_P\,)$. Then $(\forall\rho, Q)(\, Q{\to}A\sigma\rho \in P' \Rightarrow \Gamma\sigma\rho[Q] \in \mathrm{NO}^n_P\,)$ and thus by (N1) $\Gamma\sigma[A\sigma] \in \mathrm{NO}^{n+1}_P$.
b) straightforward.
c) Let $\Gamma \in \mathrm{NO}^{n+1}_P$ and w.l.o.g. $\Gamma = G\wedge *$.
Then $\Gamma[\,] = G$ and one of the following two cases holds.
1. $G = \Gamma'[A]$ with $(\forall\rho, Q)(\, Q{\to}A\rho \in P' \Rightarrow \Gamma'\rho[Q] \in \mathrm{NO}^n_P\,)$:
Then by IH $(\forall\rho, Q)(Q{\to}A\rho \in P' \Rightarrow \Gamma'\rho[Q]\wedge H\rho \in \mathrm{NO}^n_P)$, and thus by (N1) $\Gamma[H] = \Gamma'[A]\wedge H \in \mathrm{NO}^{n+1}_P$.
2. $G = \Gamma'[\neg A]$ with $A \in \mathrm{YES}^n_P$: Then (N2) yields $\Gamma[H] = \Gamma'[\neg A]\wedge H \in \mathrm{NO}^{n+1}_P$.

Lemma 6 (Splitting-Lemma)
$G\wedge G' \in \mathrm{NO}^n_P \;\&\; \mathrm{var}(G) \cap \mathrm{var}(G') = \emptyset \implies G \in \mathrm{NO}^n_P$ or $G' \in \mathrm{NO}^n_P$.

Proof by induction on n:
Let $G\wedge G' \in \mathrm{NO}^{n+1}_P$. Then w.l.o.g. one of the following two cases holds:
1. $G = \Gamma[A]$ and $(\forall\rho, Q)(\, Q{\to}A\rho \in P' \Rightarrow \Gamma\rho[Q] \wedge G'\rho \in \mathrm{NO}^n_P\,)$.
Assume $G' \notin \mathrm{NO}^{n+1}_P$ (otherwise we are finished).
Then also $G' \notin \mathrm{NO}^n_P$, and by Lemma 5a we get $G'' \notin \mathrm{NO}^n_P$ for each variant G'' of G'.
We now prove $\forall\rho, Q(\, Q{\to}A\rho \in P' \Rightarrow \Gamma\rho[Q] \in \mathrm{NO}^n_P\,)$, which by (N1) yields $G \in \mathrm{NO}^{n+1}_P$. Let $Q{\to}A\rho \in P'$. Since $\mathrm{var}(\Gamma[A]) \cap \mathrm{var}(G') = \emptyset$, there is a substitution ρ_0 such that $A\rho_0 = A\rho$, $\Gamma\rho_0 = \Gamma\rho$, $\mathrm{var}(\Gamma\rho[Q]) \cap \mathrm{var}(G'\rho_0) = \emptyset$ and $G'\rho_0 \geq G'$.
We now conclude as follows:
$Q{\to}A\rho_0 = Q{\to}A\rho \in P' \Rightarrow \Gamma\rho[Q]\wedge G'\rho_0 = \Gamma\rho_0[Q]\wedge G'\rho_0 \in \mathrm{NO}^n_P \overset{\mathrm{IH}}{\Rightarrow} \Gamma\rho[Q] \in \mathrm{NO}^n_P$.
2. $G = \Gamma[\neg A]$ and $A \in \mathrm{YES}^n_P$: Then $\Gamma \in \mathrm{NO}^{n+1}_P$ by (N2).

Lemma 7 (Inversion-Lemma)

a) $\Gamma[A] \in \mathrm{YES}^n_P \implies n > 0 \;\&\; \exists Q(\, Q{\to}A \in P' \;\&\; \Gamma[Q] \in \mathrm{YES}^{n-1}_P\,)$.

b) $\Gamma[A] \in \mathrm{NO}^n_P \;\&\; Q{\to}A \in P' \implies \Gamma[Q] \in \mathrm{NO}^n_P$.

Proof a) By L.5b there are k, m such that $n = k{+}m$ and $\Gamma[\,] \in \mathrm{YES}^k_P$, $A \in \mathrm{YES}^m_P$. The latter yields $m > 0$ and $\exists Q(\, Q \in \mathrm{YES}^{m-1}_P \;\&\; Q{\to}A \in P'\,)$. Again by L.5b we obtain $\Gamma[Q] \in \mathrm{YES}^{k+m-1}_P = \mathrm{YES}^{n-1}_P$.
b) Induction on n: Let $\Gamma[A] \in \mathrm{NO}^{n+1}_P$. W.l.o.g. we may assume that $\Gamma = G\wedge *$.
Then one of the following cases holds:
1. $\forall Q', \rho(Q'{\to}A\rho \in P' \Rightarrow \Gamma\rho[Q'] \in \mathrm{NO}^n_P)$:
Then $\Gamma[Q] \in \mathrm{NO}^n_P \subseteq \mathrm{NO}^{n+1}_P$, since $Q{\to}A \in P'$.
2. $G = \Gamma'[B]$ and $\forall Q', \rho(Q'{\to}B\rho \in P' \Rightarrow \Gamma'\rho[Q']\wedge A\rho \in \mathrm{NO}^n_P)$:
Since $\forall\rho(Q\rho{\to}A\rho \in P')$, by IH we get $\forall Q', \rho(Q'{\to}B\rho \in P' \Rightarrow \Gamma'\rho[Q']\wedge Q\rho \in \mathrm{NO}^n_P)$.
Hence by (N1) $\Gamma[Q] = \Gamma'[B]\wedge Q \in \mathrm{NO}^{n+1}_P$.
3. $G = \Gamma'[\neg B]$ and $B \in \mathrm{YES}^n_P$: Then $\Gamma[Q] = \Gamma'[\neg B]\wedge Q \in \mathrm{NO}^{n+1}_P$ by (N2).

Note
Lemmata 5,6,7 and their proofs have been extracted from section 2.1 of [6].

3 SLDNF-resolution

Formulas of the shape $G \dashrightarrow H$ are called *r-formulas.*
Expressions of the shape $\Gamma \dashrightarrow H$, where Γ is a q-form, are called *r-forms.*
If $\Delta = \Gamma \dashrightarrow H$ is an r-form then $\Delta\sigma := \Gamma\sigma \dashrightarrow H\sigma$, $\Delta[Q] := \Gamma[Q] \dashrightarrow H$, $\nu(\Delta) := \nu(\Gamma)$.
Syntactic variables: Δ for r-forms, and Φ, Ψ for r-formulas.

Definition 8 Φ' is called a *u-resolvent of* Φ *w.r.t.* (ν, K) (in symbols $\Phi \xrightarrow[(\nu,K)]{u} \Phi'$),
if there are Δ, A, Q, ρ such that $\Phi = \Delta[A]$, $\Phi' = \Delta\rho[Q]$, $Q \rightarrow A\rho \leq K$ and $\nu(\Delta) = \nu$.

Φ' is called a *resolvent of* Φ *w.r.t.* (ν, K) (in symbols $\Phi \xrightarrow[(\nu,K)]{} \Phi'$),
if Φ' is a most general u-resolvent of Φ w.r.t. (ν, K),
i.e. $\Phi \xrightarrow[(\nu,K)]{u} \Phi'$ & $\forall\Phi''(\Phi \xrightarrow[(\nu,K)]{u} \Phi'' \Rightarrow \Phi' \geq \Phi'')$.

Φ' is called *resolvent of* Φ *w.r.t.* P (in symbols $\Phi' \xrightarrow[P]{} \Phi$),
if $\Phi \xrightarrow[(\nu,K)]{} \Phi'$ for some $K \in P$, $\nu \in \mathbb{N}$.

$(\Phi_K)_{K\in J}$ is called a *complete family of resolvents for* Φ *w.r.t.* P
(in symbols $\Phi \xrightarrow[P]{*} (\Phi_K)_{K\in J}$), if there are Δ, A such that
$\Phi = \Delta[A]$, $J = P(A)$ and $(\forall K{\in}P(A))\ \Phi \xrightarrow[(\nu,K)]{} \Phi_K$ where $\nu := \nu(\Delta)$.

Lemma 9 For $\Phi = \Delta[A]$, $\nu := \nu(\Delta)$ and K the following statements are equivalent:

(i) $K \in P(A)$.

(ii) $\Phi \xrightarrow[(\nu,K)]{u} \Phi'$ for some Φ'.

(iii) $\Phi \xrightarrow[(\nu,K)]{} \Psi$ for some Ψ.

Proof (i)$\Rightarrow$(iii): Let $\Pi \rightarrow B$ a variant of K such that $\mathrm{var}(\Phi) \cap \mathrm{var}(\Pi \rightarrow B) = \emptyset$. Since $K \in P(A)$ and $\mathrm{var}(A) \cap \mathrm{var}(B) = \emptyset$, the atoms A, B are unifiable. Let θ be a most general unifier of A, B, and let $\Psi := \Delta\theta[\Pi\theta]$. We prove $\Phi \xrightarrow[(\nu,K)]{} \Psi$: Since $\Pi\theta \rightarrow A\theta \leq K$, Ψ is a u-resolvent of Φ w.r.t. (ν, K). Now let Φ' be an arbitrary u-resolvent of Φ w.r.t. (ν, K). Then $\Phi' = \Delta\rho[Q]$ with $Q \rightarrow A\rho \leq K \leq \Pi \rightarrow B$. Since $\mathrm{var}(\Delta[A]) \cap \mathrm{var}(\Pi \rightarrow B) = \emptyset$, we may assume that $Q = \Pi\rho$ and $A\rho = B\rho$. Hence $\rho = \theta\sigma$ for some σ and thus $\Phi' = \Delta\theta\sigma[\Pi\theta\sigma] \leq \Psi$.
(iii)$\Rightarrow$(ii) and (ii)$\Rightarrow$(i) are trivial.

Lemma 10 $G \dashrightarrow H \xrightarrow[P]{*} (G_K \dashrightarrow H_K)_{K\in J}$ & $\forall K{\in}J(G_K \in \mathrm{NO}_P) \implies G \in \mathrm{NO}_P$.

Proof By assumption there are Γ, A such that $G = \Gamma[A]$, $J = P(A)$ and
$G \dashrightarrow H \xrightarrow[(\nu,K)]{} G_K \dashrightarrow H_K$ for all $K \in J$ (with $\nu := \nu(\Gamma)$). By the second assumption (and since J is finite) there exists an n such that $G_K \in \mathrm{NO}_P^n$ for all $K \in J$. Now let $Q \rightarrow A\rho \in P'$. We have to prove: $\Gamma\rho[Q] \in \mathrm{NO}_P^n$. (Then the claim follows by (N1).) For some $K \in J$ we have $Q \rightarrow A\rho \leq K$. Then $G \dashrightarrow H \xrightarrow[(\nu,K)]{u} \Gamma\rho[Q] \dashrightarrow H\rho$ and $G \dashrightarrow H \xrightarrow[(\nu,K)]{}$ $G_K \dashrightarrow H_K$ which yields $G_K \geq \Gamma\rho[Q]$. Together with $G_K \in \mathrm{NO}_P^n$ by Lemma 5a we now obtain $\Gamma\rho[Q] \in \mathrm{NO}_P^n$.

Definition 11 An *SLDNF-tree* for P is a function T such that

I. $\emptyset \neq \mathrm{dom}(T) \subseteq \{\langle \iota_0, ..., \iota_{n-1}\rangle : n \in \mathbb{N} \ \&\ \iota_0, ..., \iota_{n-1} \in P \cup \{0,1\}\}$ and
$\forall \langle \iota_0, ..., \iota_n\rangle \in \mathrm{dom}(T)(\, \langle \iota_0, ..., \iota_{n-1}\rangle \in \mathrm{dom}(T)\,)$.

II. For each $\xi \in \mathrm{dom}(T)$ $\ T(\xi)$ is a r-formula and one of the following cases holds:

(0) $T(\xi) = \Box \!\rightarrow\! H$ and ξ is a leaf of T,

(1) There are Δ, A such that $T(\xi) = \Delta[A]$, $\{\iota : \xi{*}\langle\iota\rangle \in \mathrm{dom}(T)\} = P(A)$ and, $T(\xi) \xrightarrow[(\nu,K)]{} T(\xi{*}\langle K\rangle)$ for all $K \in P(A)$ (where $\nu := \nu(\Delta)$)
(We say that the literal A is *selected at* ξ.)

(2) There are Δ, A such that $T(\xi) = \Delta[\neg A]$, $\{\iota : \xi{*}\langle\iota\rangle \in \mathrm{dom}(T)\} = \{0,1\}$, $T(\xi{*}\langle 1\rangle) = A\!\rightarrow\! A$, and

$$T(\xi{*}\langle 0\rangle) = \begin{cases} \Delta[\,] & \text{if } \mathrm{var}(A) = \emptyset \\ \Delta[\neg A] & \text{otherwise} \end{cases}, \qquad (\neg A \text{ is } \textit{selected at } \xi).$$

In other words: An SLDNF-tree for P is a finitely branching downward growing (possibly infinite) tree of r-formulas which is correct with respect to the following rules:

(0) $\dfrac{\Box\!\rightarrow\! H}{}$

(1) $\dfrac{\Phi}{....\Phi_K....(K\in J)}$ $\qquad$ if $\Phi \xrightarrow[P]{*} (\Phi_K)_{K\in J}$

(2) $\dfrac{\Delta[\neg A]}{\Delta[\,] \qquad A\!\rightarrow\! A}$ $\qquad$ if $\mathrm{var}(A) = \emptyset$

(2') $\dfrac{\Delta[\neg A]}{\Delta[\neg A] \qquad A\!\rightarrow\! A}$ $\qquad$ if $\mathrm{var}(A) \neq \emptyset$.

Remark 12 1. If (ξ, Φ) is a leaf of an SLDNF-tree then either $\Phi = \Box\!\rightarrow\! H$ or $\Phi \xrightarrow[P]{*} (\Phi_K)_{K\in J}$ with $J = \emptyset$.
2. Each subtree of an SLDNF-tree is itself an SLDNF-tree.

Convention: In the following T always denotes an SLDNF-tree for P.

Abbreviations:
$\varepsilon := \langle\,\rangle$ (the empty sequence, i.e. the root)
$T = (\Phi; (T_\iota)_{\iota\in I})$ $\ :\Leftrightarrow\ T(\varepsilon) = \Phi$ and $(T_\iota)_{\iota\in I}$ is the family of immediate subtrees of T.
In case that $I = \{0, ..., n{-}1\}$ we also write $(\Phi; T_0, ..., T_{n-1})$ instead of $(\Phi; (T_\iota)_{\iota\in I})$.

For $\xi \in \mathrm{dom}(T)$ let $T|_\xi$ be the subtree determined by ξ,
i.e., $\mathrm{dom}(T|_\xi) := \{\eta : \xi * \eta \in \mathrm{dom}(T)\}$ and $T|_\xi(\eta) := T(\xi * \eta)$.

Let no be a new symbol. A *generalized answer* is either a query or the symbol no. We now define a relation $T \Vdash X$ between SLDNF-trees T and generalized answers X. As one easily verifies this relation $\Vdash$ has the property that if $T(\varepsilon) = G\!\rightarrow\! H \ \&\ T \Vdash H'$ then H' is an instance of H. Further if $T(\varepsilon) = G\!\rightarrow\! G$ then "$T \Vdash G\theta$" corresponds to "T is successful and yields $\theta|G$ as a computed answer substitution", and "$T \Vdash$ no" corresponds to "T is finitely failed" in [3].

Definition 13 (Inductive definition of $T \Vdash X$)

0. $T = (\Box \!\to\! H;\ ) \;\Rightarrow\; T \Vdash H$

1. $T = (\Delta[A]; (T_K)_{K \in P(A)})$:

1.1. $(\exists K \in J)\, T_K \Vdash H \;\Rightarrow\; T \Vdash H$.

1.2. $(\forall K \in J)\, T_K \Vdash \mathsf{no} \;\Rightarrow\; T \Vdash \mathsf{no}$.

2. $T = (\Delta[\neg A]; T_0, T_1)$ and $T_0(\varepsilon) \in \{\Delta[\,], \Delta[\neg A]\}$, $T_1(\varepsilon) = A \!\to\! A$:

2.1. $T_0 \Vdash H$ & $T_1 \Vdash \mathsf{no} \;\Rightarrow\; T \Vdash H$.

2.2. $T_1 \Vdash A' \geq A \;\Rightarrow\; T \Vdash \mathsf{no}$.

2.3. $T_0 \Vdash \mathsf{no} \;\Rightarrow\; T \Vdash \mathsf{no}$.

2.4. $T_0 \Vdash H$ & $T_0(\varepsilon) = T(\varepsilon) \;\Rightarrow\; T \Vdash H$.

The following figures are intended to make the above definition more transparent.

0. $\Box \!\to\! H \colon H$, 1.1. $\dfrac{\Phi \colon H}{\Phi_K \colon H}$, 1.2. $\dfrac{\Phi \colon \mathsf{no}}{\ldots \Phi_K \colon \mathsf{no} \ldots {\scriptstyle (K \in J)}}$

2.1. $\dfrac{\Delta[\neg A] \colon H}{\Delta[\,] \colon H \qquad A \!\to\! A \colon \mathsf{no}}$, 2.2. $\dfrac{\Delta[\neg A] \colon \mathsf{no}}{\Delta[({\scriptstyle \neg A})] \qquad A \!\to\! A \colon A'}$ with $A' \geq A$

2.3. $\dfrac{\Delta[\neg A] \colon \mathsf{no}}{\Delta[({\scriptstyle \neg A})] \colon \mathsf{no} \qquad A \!\to\! A}$, 2.4. $\dfrac{\Delta[\neg A] \colon H}{\Delta[\neg A] \colon H \qquad A \!\to\! A}$

Remark 14 1. In clause 2.2 of the above definition actually A' is a variant of A, since $T_1(\varepsilon) = A \!\to\! A$ & $T_1 \Vdash A'$ implies $A \geq A'$.

2. If $T \Vdash X$ holds then this fact is already established by a finite initial segment of T. In other words: Assume that for finite initial segments $\tilde{T}$ of SLDNF-trees we have defined the relation $\tilde{T} \Vdash X$ by the same clauses as for SLDNF-trees. Then for each SLDNF-tree T we have $T \Vdash X$ if and only if $\tilde{T} \Vdash X$ for some finite initial segment $\tilde{T}$ of T.

Theorem 15 (Correctness)

a) $T(\varepsilon) = G \!\to\! H$ & $T \Vdash H' \implies (\exists \vartheta)\, H\vartheta \geq H'$ & $G\vartheta \in \mathrm{YES}_P$.

b) $T(\varepsilon) = G \!\to\! H$ & $T \Vdash \mathsf{no} \implies G \in \mathrm{NO}_P$.

Proof by induction on the definition of $\Vdash$:
a) Let $T = (G \!\to\! H; (T_\iota)_{\iota \in J})$.
0. $G = \Box$ and $H' = H$: $\vartheta := \mathrm{id}$.
1.1. $G = \Gamma[A]$, $Q \!\to\! A\rho \leq K \in P$ and $T_K(\varepsilon) = \Gamma\rho[Q] \!\to\! H\rho$ & $T_K \Vdash H'$:
By IH there is a ϑ such that $H\rho\vartheta \geq H'$ & $\Gamma\rho[Q]\vartheta \in \mathrm{YES}_P$.
Since $Q\vartheta \!\to\! A\rho\vartheta \in P'$, it follows that $G\rho\vartheta = \Gamma\rho\vartheta[A\rho\vartheta] \in \mathrm{YES}_P$.
2.1. $G = \Gamma[\neg A]$, $T_0(\varepsilon) = \Gamma[\,] \!\to\! H$ & $T_0 \Vdash H'$, $T_1(\varepsilon) = A \!\to\! A$ & $T_1 \Vdash \mathsf{no}$:
By IHa there is a ϑ such that $H\vartheta \geq H'$ & $\Gamma[\,]\vartheta \in \mathrm{YES}_P$.
By IHb we get $A \in \mathrm{NO}_P$ and thus (by L.5a) also $A\vartheta \in \mathrm{NO}_P$. Hence $\Gamma[\neg A]\vartheta \in \mathrm{YES}_P$.

2.4. $G = \Gamma[\neg A]$, $T_0(\varepsilon) = T(\varepsilon)$ & $T_0 \Vdash H'$:
In this case the claim follows immediately from IH.

b) 1.2. $T_K(\varepsilon) = G_K \dashv H_K$ & $T_K \Vdash \mathsf{no}$ for all $K \in J = P(A)$:
By IHb $G_K \in \mathrm{NO}_P$ for all $K \in J$. By Lemma 10 this implies $G \in \mathrm{NO}_P$.
2.2. $G = \Gamma[\neg A]$ and $T_1(\varepsilon) = A \dashv A$ & $T_1 \Vdash A' \geq A$:
By IHa there is a ϑ such that $\mathrm{YES}_P \ni A\vartheta \geq A' \geq A$.
By Lemma 1a from this we get $A \in \mathrm{YES}_P$ and thus $G \in \mathrm{NO}_P$.
2.3. $G = \Gamma[\neg A]$, $T_0(\varepsilon) = \Gamma[(\neg A)] \dashv H$ & $T_0 \Vdash \mathsf{no}$:
By IHb $\Gamma[(\neg A)] \in \mathrm{NO}_P$. This implies $G \in \mathrm{NO}_P$ (cf. L.5c).

Corollary 16 $T(\varepsilon) = G \dashv G$ & $T \Vdash G\theta \implies G\theta \in \mathrm{YES}_P$.

Definition 17 A *main branch* in T is a sequence $(\xi_j)_{j<N}$ (with $0 < N \leq \omega$) of nodes in T, such that for all $j < N$:

(i) $j + 1 < N \Rightarrow \xi_{j+1} = \xi_j * \langle\iota\rangle$ for some $\iota \in P \cup \{0\}$,

(ii) $j + 1 = N \Rightarrow \xi_j$ is a leaf node in T.

A main branch is called *fair* if either it terminates with a leaf $(\xi, \Gamma[A])$ such that $P(A) = \emptyset$ or for each literal L in it after finitely many steps a descendent of L is selected. T is called *fair*, if all its main branches are fair.

Obviously for each program P and goal G a fair SLDNF-tree can be defined.

Definition 18
Let $\mathcal{C}^+, \mathcal{C}^-$ be sets of queries.
P is called a $(\mathcal{C}^+, \mathcal{C}^-)^*$*-program*, if the following conditions are satisfied:

(S1) $G \in \mathcal{C}^+ \ (\mathcal{C}^-) \Rightarrow G\sigma \in \mathcal{C}^+ \ (\mathcal{C}^-)$,

(S2) $Q \rightarrow A \in P'$ & $\Gamma[A] \in \mathcal{C}^+ \ (\mathcal{C}^-) \Rightarrow \Gamma[Q] \in \mathcal{C}^+ \ (\mathcal{C}^-)$,

(S3) $\Gamma[\neg A] \in \mathcal{C}^+$ & $\mathrm{var}(A) \neq \emptyset \Rightarrow \Gamma$ contains at least one positive literal,

(S4) $\Gamma[\neg A] \in \mathcal{C}^+$ & $\mathrm{var}(A) = \emptyset \Rightarrow \Gamma[\,] \in \mathcal{C}^+$ & $A \in \mathcal{C}^-$,

(S5) $\Gamma[\neg A] \in \mathcal{C}^- \Rightarrow \Gamma[\,] \in \mathcal{C}^-$ & $A \in \mathcal{C}^+$.

Remark 19 The above definition is a modification of Stärks definition of $(\mathcal{C}^+, \mathcal{C}^-)$-programs in [7]. As one easily verifies, the following holds: if S is an input/output specification, P is S-correct, and $\mathcal{C}^+$ ($\mathcal{C}^-$) is the set of S-correct (S-closed) queries in the sense of [7], then P is a $(\mathcal{C}^+, \mathcal{C}^-)^*$-program. But, as one of the referees noticed, not every $(\mathcal{C}^+, \mathcal{C}^-)$-program is a $(\mathcal{C}^+, \mathcal{C}^-)^*$-program.

Theorem 20 (Strong Completeness)
Assume that T is a fair SLDNF-tree with $T(\varepsilon) = G \dashv H$, and P is a $(\mathcal{C}^+, \mathcal{C}^-)^*$-program.

a) $G\sigma \in \mathrm{YES}_P^n$ & $G \in \mathcal{C}^+ \implies (\exists H')\, T \Vdash H' \geq H\sigma$ (in particular if G=H then $(\exists\theta)\, T \Vdash G\theta \geq G\sigma$).

b) $G \in \mathrm{NO}_P^n$ & $G \in \mathcal{C}^- \implies T \Vdash \mathsf{no}$.

Proof by induction on n: Let $T = (G \rightarrow H; (T_\iota)_{\iota \in J})$.
a)
Proposition 1:
There exists a node $\xi = \langle 0, ..., 0\rangle \in \mathrm{dom}(T)$ (ξ may be empty) such that $T(\varepsilon) = T(\langle 0\rangle) = T(\langle 0, 0\rangle) = \ldots = T(\xi)$ and $(\xi * \langle 0\rangle \in \mathrm{dom}(T) \Rightarrow T(\xi * \langle 0\rangle) \neq T(\xi))$.

Proof: Assume the contrary. Then for all $k \in \mathbb{N}$ we have $\xi_k := \overbrace{\langle 0, ..., 0\rangle}^{k} \in \mathrm{dom}(T)$ and $T(\xi_k) = T(\varepsilon)$ which means that at every node ξ_k a nonclosed negative literal is selected and in particular G contains a nonclosed negative literal. Now, since $G \in \mathcal{C}^+$, by (S3) it follows that G contains at least one positive literal B a descendent of which will be selected at some ξ_k (due to the fairness of T). Contradiction.

Now let ξ be as in Proposition 1. Obviously we have $(T|_\xi \Vdash X \Rightarrow T \Vdash X)$. Therefore w.l.o.g. we may assume that $\xi = \varepsilon$ and thus one of the following cases holds.
0. $G = \Box$ and $J = \emptyset$: $H' := H$.
1. $G = \Gamma[A]$, $\nu = \nu(\Gamma)$, and $G \rightarrow H \underset{(\nu,K)}{\longrightarrow} T_K(\varepsilon)$ for all $K \in J = P(A)$:
From $G\sigma \in \mathrm{YES}^n_P$ it follows by L.7a that there exists a query Q and a rule $K \in P(A)$ such that $\Gamma\sigma[Q] \in \mathrm{YES}^{n-1}_P$ and $K \geq Q \rightarrow A\sigma$. Then $G \rightarrow H \overset{u}{\underset{(\nu,K)}{\longrightarrow}} \Gamma\sigma[Q] \rightarrow H\sigma$ and $G \rightarrow H \underset{(\nu,K)}{\longrightarrow} T_K(\varepsilon) =: G_K \rightarrow H_K$. Hence $G_K\sigma' = \Gamma\sigma[Q] \in \mathrm{YES}^{n-1}_P$ and $H_K\sigma' = H\sigma$ for some σ'. From $G \in \mathcal{C}^+$ we get $G_K \in \mathcal{C}^+$ by (S1),(S2). Now by IH there is an H' such that $T_K \Vdash H' \geq H\sigma$. And $T_K \Vdash H'$ implies $T \Vdash H'$.
2. $G = \Gamma[\neg A]$, $T_0(\varepsilon) = \Gamma[\,] \rightarrow H$, $T_1(\varepsilon) = A \rightarrow A$ and $\mathrm{var}(A) = \emptyset$:
From $G \in \mathcal{C}^+$ we get $\Gamma[\,] \in \mathcal{C}^+$ and $A \in \mathcal{C}^-$ by (S4). From $G\sigma \in \mathrm{YES}^n_P$ by L.5b we get $n > 0$, $\Gamma[\,]\sigma \in \mathrm{YES}^{n-1}_P$, $\neg A = \neg A\sigma \in \mathrm{YES}^n_P$. The latter implies $A \in \mathrm{NO}^{n-1}_P$. Hence by IH $T_1 \Vdash \mathsf{no}$ and $T_0 \Vdash H' \geq H\sigma$ for some H'. From $T_0 \Vdash H'$ and $T_1 \Vdash \mathsf{no}$ we get $T \Vdash H'$.
b)
Proposition 2:
Every main branch of T starting with ε contains a node ξ such that $T|_\xi \Vdash \mathsf{no}$.

From this we get $T \Vdash \mathsf{no}$, since otherwise there would be a main branch $(\xi_j)_{j<N}$ starting with ε such that $T|_{\xi_j} \nVdash \mathsf{no}$ for all $j < N$.

Proof of Proposition 2:
Let $(\xi_j)_{j<N}$ be a main branch of T starting with $\xi_0 = \varepsilon$. Let $T(\xi_j) = G_j \rightarrow H_j$. Then by (S1),(S2),(S5) $G_j \in \mathcal{C}^-$ for all $j < N$.
1. Assume that the branch terminates with $T(\xi_l) = \Delta[A]$ such that $J = P(A) = \emptyset$. Then $T|_{\xi_l} \Vdash \mathsf{no}$.
2. Assume that there is an l s.t. $G_l = \Gamma'[\neg A']$, $T(\xi_l * \langle 1\rangle) = A' \rightarrow A'$ and $A' \in \mathrm{YES}^{n-1}_P$. From $G_l \in \mathcal{C}^-$ we get $A' \in \mathcal{C}^+$ by (S5). By IHa there is an A'' s.t. $T|_{\xi_l * \langle 1\rangle} \Vdash A'' \geq A'$. From this we get $T|_{\xi_l} \Vdash \mathsf{no}$.
3. Otherwise: Then, since $G \in \mathrm{NO}^n_P$, $n > 0$ and one of the following two cases holds:
3.1. $G = \Gamma[\neg A]$ with $A \in \mathrm{YES}^{n-1}_P$:
Since T is fair, there exists an l such that $G_l = \Gamma'[\neg A']$, $T(\xi_l * \langle 1\rangle) = A' \rightarrow A'$ and $A \geq A'$. From $A \in \mathrm{YES}^{n-1}_P$ we get $A' \in \mathrm{YES}^{n-1}_P$ and thus case 2. holds.
3.2. $G = \Gamma[A]$ with $\forall Q, \rho(Q \rightarrow A\rho \in P' \Rightarrow \Gamma\rho[Q] \in \mathrm{NO}^{n-1}_P)$:
By side induction on l we prove:
(+) $G_l = \Gamma'[A']$ & A' descendent of A & $Q \rightarrow A'\rho \in P' \implies \Gamma'\rho[Q] \in \mathrm{NO}^{n-1}_P$.

If $l = 0$, then $\Gamma' = \Gamma$ and $A' = A$, from which together with $Q \rightarrow A'\rho \in P'$ the claim follows.

Now let $l > 0$. Then one of the following three cases holds.
(i) $G_{l-1} = \Gamma''[B]$ (where B not a descendent of A) and $G_l = \Gamma''\tau[\Pi]$ with $\Pi \rightarrow B\tau \in P'$:
W.l.o.g. $G_{l-1} = B \wedge \Gamma_0[A_0]$, $\Gamma' = \Pi \wedge \Gamma_0\tau$ und $A' = A_0\tau$, where A_0 is a descendent of A. Further we have $Q \rightarrow A_0\tau\rho \in P'$, from which by side-IH we get $B\tau\rho \wedge \Gamma_0\tau\rho[Q] \in \mathrm{NO}_P^{n-1}$. Since $\Pi\rho \rightarrow B\tau\rho \in P'$, L.7b now yields $\Gamma'\rho[Q] = \Pi\rho \wedge \Gamma_0\tau\rho[Q] \in \mathrm{NO}_P^{n-1}$.
(ii) $G_{l-1} = G_l$: Then the claim follows immediately from the side-IH.
(iii) $G_{l-1} = \neg B \wedge \Gamma'[A']$ & $T(\xi_{l-1} * \langle 1 \rangle) = B \dashv B$, where B is closed:
By side-IH we have $\neg B \wedge \Gamma'\rho[Q] \in \mathrm{NO}_P^{n-1}$. By L.6 it follows that $\Gamma'\rho[Q] \in \mathrm{NO}_P^{n-1}$ or $\neg B \in \mathrm{NO}_P^{n-1}$. The latter would imply $B \in \mathrm{YES}_P^{n-1}$ and we were in case 2. So $\Gamma'\rho[Q] \in \mathrm{NO}_P^{n-1}$ holds.

Since T is fair, there exists an l, such that $G_l = \Gamma'[A']$, A' is descendent of A, and $G_{l+1} = \Gamma'\rho[Q]$ with $Q \rightarrow A'\rho \in P'$. By (+) from this we get $G_{l+1} \in \mathrm{NO}_P^{<n}$ and thus $T|_{\xi_{l+1}} \Vdash \mathsf{no}$ by IHb.

Acknowledgement
The author is indepted to Kees Doets and an anonymous referee for their valuable comments.

References

[1] W. Buchholz. *Ein Negation-as-Failure-Kalkül.* Typescript, University of Munich, 1992.

[2] K. Doets. *Levationis laus.* J. of Logic and Computation 3, pp. 487-516 (1993)

[3] K. Doets. *From Logic to Logic Programming.* MIT Press, 1994.

[4] W. Drabent. *Completeness of SLDNF-resolution for non-floundering queries.* J. of Logic Programming 27(2), pp. 89-106 (1996)

[5] J.C. Shepherdson. *The role of standardising apart in logic programming.* Theoretical Computer Science 129, pp. 143-166 (1994)

[6] R.F. Stärk. *The Proof Theory of Logic Programs with Negation.* PhD Thesis, University of Bern, 1992.

[7] R.F. Stärk. *Input/output dependencies of normal logic programs.* J. of Logic and Computation 4, pp. 249–262 (1994)