

String Reduction Schemas

Ari Feiglin¹ and Yoni Zohar¹

Bar-Ilan University, Ramat Gan, Israel

Abstract. In this paper we introduce the concept of *string reduction schemas* in varying levels of strength, creating a new hierarchy of classes of formal languages. We compare this hierarchy with the well-known Chomsky Hierarchy, and show various inclusions and equalities.

The main result of this paper is to show that **DCFL**, the class of deterministic context-free languages, is contained in **1-SR**. This is significant as **1-SR** schemas can be naturally produced by humans and parsers which work with them are simple to implement. Thus string reduction schema are useful in the theory of parsing. We further demonstrate this by utilizing **1-SR** schema to create parsers in OCaml as well as in T_EX, demonstrating the ease of implementation.

Keywords: String Reduction Schema · Context-Free Language · Parsing

1 Introduction

We begin our discussion by briefly giving an overview of parsing.

TODO: write introduction

2 A note on notation

While we don't use much special notation in this paper, it nevertheless is worthwhile to discuss some potentially non-standard notation we do use.

First we have some language-theoretic notational definitions:

- Given two words u, v , we denote their concatenation by juxtaposition (i.e. uv).
- Given a set of symbols Σ , we denote Σ^* the free monoid over Σ (i.e. the set of finite-length words over Σ).
- ε denotes the empty word, which is the unit in the Σ^* (i.e. $u\varepsilon = \varepsilon u = u$ for every $u \in \Sigma^*$).
- Given a set of symbols X and an integer $n \geq 0$, we denote by X^n the set of n -length words in X . For $n = 0$, $X^0 = \{\varepsilon\}$.
- Similarly, $X^{\leq n}$ denotes the set of words over X of length $\leq n$. Equivalently, $X^{\leq n} = \bigcup_{m \leq n} X^m$.
- Similarly, $X^{(0, n]} = \bigcup_{0 < m \leq n} X^m$.

We also have some set-theoretic notational definitions:

- We denote by $A \multimap B$ the type of partial functions from A to B (i.e. functions which need not produce an output for every input).
- $\mathcal{P}(A)$ is the powerset of A : the set of all subsets of A .
- $\mathcal{P}_{<\omega}(A)$ is the finitary powerset of A : the set of all finite subsets of A .
- $\mathcal{P}_{>0}(A)$ is the non-empty powerset of A : the set of all non-empty subsets of A .
- Let X be a set and x be some symbol (which in context cannot be mistaken for a set), then we write $X \cup x$ for $X \cup \{x\}$.

3 Nondeterministic string reduction schemas

We begin our introduction of string reduction schema by defining their most general variation. As discussed, string reduction schema deviate from the norm of using grammars as input and instead use string rewriting systems. This has the benefit of being straightforward to implement programmatically.

Definition 1. A (\mathbf{n}, \mathbf{m}) -NSR schema (short for (n, m) -nondeterministic string rewriting schema) is a tuple

$$B = (\Sigma, \Gamma, \square, \text{Acc}, \bar{\beta})$$

where

- (1) Σ is the alphabet of input terminals;
- (2) Γ is the alphabet of intermediary variables (or just variables). It must contain Σ (although in general we find it more useful to define an embedding of Σ into Γ);
- (3) $\square \in \Gamma$ is the string endmarker symbol;
- (4) $\text{Acc} \subseteq \Gamma$ is the set of acceptance symbols;
- (5) $\bar{\beta}$ is the initial reduction function, a function:

$$\bar{\beta}: \Gamma^{(0,n]} \longrightarrow \mathcal{P}_{>0}(\Gamma^{\leq m} \cup \perp)$$

That is, $\bar{\beta}$ accepts strings over Γ of length $\leq n$ and returns a set of strings of length $\leq m$, as well as potentially the symbol \perp (which is a symbol not in any other set of symbols we are to consider).

Of course, if we neglect \perp , then $\bar{\beta}$ can just be viewed as a string rewrite rule \rightarrow where the lhs of \rightarrow must be strings of length $\leq n$ and the rhs must be strings of length $\leq m$. The idea of this a (\mathbf{n}, \mathbf{m}) -NSR schema is to define a method to rewrite strings. This is done by iterating over an input string from left to right and rewriting any substring non-deterministically which matches an input of $\bar{\beta}$. \perp is a symbol telling us we can skip over this substring (nondeterministically).

For example, if $\Gamma = \{a, b\}$ and $\bar{\beta}(ab) = \{a, \perp\}$, $\bar{\beta}(aa) = \{b\}$, and $\bar{\beta}(ba) = \{a\}$ (assume $\bar{\beta}(w) = \{\perp\}$ for all other words w). Then we can rewrite (we underline the substrings being rewritten)

$$abaa \rightarrow \underline{aaa} \rightarrow \underline{ba} \rightarrow a$$

Notice that in the first rewrite we skip over \underline{abaa} , as we can since $\perp \in \bar{\beta}(ab)$.

Let us now formalize this idea.

Definition 2. Let $B = (\Sigma, \Gamma, \square, \text{Acc}, \bar{\beta})$ be an (\mathbf{n}, \mathbf{m}) -NSR schema. We define its reduction function to be the function

$$\beta: \Gamma^* \longrightarrow \mathcal{P}_{<\omega}(\Gamma^*)$$

defined in the following way. Let S be the set of all triplets of words t_i, u_i, v_i where

- (1) $w = t_i u_i v_i$;
- (2) t_i has no subwords t' such that $\perp \notin \bar{\beta}(t')$;
- (3) $\bar{\beta}(u_i) \neq \{\perp\}$;

then we define

$$\beta(w) = \{ t_i u' v_i \mid t_i u_i v_i \in S, u' \in \bar{\beta}(u_i) - \perp \}$$

Now it is indeed simpler to view this as a string rewrite relation. So let us define \rightarrow_B to be the binary relation on Γ^* defined by $u \rightarrow_B v$ if and only if $v \in \beta(u)$. Let \rightarrow_B^* be its reflexive and transitive closure. We define the language of B (or the language accepted by B) to be

$$\mathcal{L}(B) = \{ w \in \Sigma^* \mid \square w \square \rightarrow^* \top \in \text{Acc} \}$$

The class of all languages accepted by an (\mathbf{n}, \mathbf{m}) -NSR-schema is denoted (\mathbf{n}, \mathbf{m}) -NSR.

TODO: prove (\mathbf{n}, \mathbf{m}) -NSR = $(2, 2)$ -NSR = RE and CFL $\subseteq (2, 1)$ -NSR

4 Deterministic string reduction schemas

Definition 3. A (\mathbf{n}, \mathbf{m}) -NSR schema B is deterministic if $|\bar{\beta}(w)| = 1$ for every $w \in \Gamma^{\leq n}$, and if $\bar{\beta}(w)$ is defined, $\bar{\beta}(w')$ is undefined for all subwords w' of w . We call such a schema a (\mathbf{n}, \mathbf{m}) -SR-schema.

Notice that if B is a (\mathbf{n}, \mathbf{m}) -NSR schema, then $\bar{\beta}(w)$ is either a singleton of the form $\{u\}$ or $\{\perp\}$. Thus we can consider $\bar{\beta}$ to be a partial function

$$\bar{\beta}: \Gamma^{\leq n} \rightharpoonup \Gamma^{\leq m}$$

where $\bar{\beta}(w)$ is undefined if (for the previous $\bar{\beta}$), $\bar{\beta}(w) = \{\perp\}$. Then our definition of β stays similar, except now S is the set of triplets t_i, u_i, v_i where

- (1) $w = t_i u_i v_i$;
- (2) t_i has no subwords t' such that $\bar{\beta}(t')$ is defined;
- (3) $\bar{\beta}(u_i)$ is defined.

Now, why can't a subword of a word in the domain of $\bar{\beta}$ have its image defined? Suppose $w = uw'v$ and $\bar{\beta}(w), \bar{\beta}(w')$ both be defined. Assume that $\bar{\beta}(u)$ is undefined, then $\beta(w)$ may have a size greater than 1. This is because we can reduce both w as a whole as well as w' . That is, $\bar{\beta}(w), u\bar{\beta}(w')v \in \beta(w)$.

This cannot happen in our definition of deterministic schema:

Lemma 1. *Let B be a (n, m) -SR schema. Then for every $w \in \Gamma^*$, $\beta(w)$ has at most one word.*

TODO: formally prove this

TODO: show (n, m) -SR = $(2, 2)$ -SR and describe relation between $(2, 2)$ -SR and CFL

5 1-SR schemas

Our main focus of this section is **1-SR** schemas.

The goal of this section is to show that every deterministic context-free language is **1-SR**. To do so, we first define what we mean by a deterministic context-free language.

5.1 Deterministic context-free languages

We borrow the following definition from Sipser.

Definition 4. A deterministic pushdown automaton (DPDA) is a tuple $P = (\Sigma, \Gamma, Q, q_0, F, \delta)$ where

- (1) Σ is the alphabet of terminals;
- (2) Γ is the alphabet of stack symbols;
- (3) Q is the set of states;
- (4) $q_0 \in Q$ is the initial state;
- (5) $F \subseteq Q$ is the set of accepting states;
- (6) δ is the transition function:

$$\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow Q \times \Gamma_\varepsilon$$

such that for every $q \in Q, a \in \Sigma, A \in \Gamma$, exactly one of the following is defined:

$$\delta(q, a, A), \delta(q, a, \varepsilon), \delta(q, \varepsilon, A), \delta(q, \varepsilon, \varepsilon)$$

Definition 5. Let P be a DPDA, then we define a configuration of P to be an element of $Q \times \Sigma^* \times \Gamma^*$. We define the relation \vdash_P on the set of configurations like so. Given a configuration (q, aw, AW) (where $a \in \Sigma_\varepsilon, A \in \Gamma_\varepsilon$), if $\delta(q, a, A) = (p, B)$, then

$$(q, aw, AW) \vdash_P (p, w, BW)$$

Let \vdash_P^* be the transitive and reflexive closure of \vdash_P . Then the language of P (or the language recognized by P) is

$$\mathcal{L}(P) = \{ w \in \Sigma^* \mid (q_0, w, \varepsilon) \vdash_P^* (f, \varepsilon, W) \text{ for some } f \in F \text{ and } W \in \Gamma^* \}$$

The class of languages recognized by a DPDA is denoted **DCFL**.

We take P to be a DPDA, and we hold it constant for the remainder of this section. Our goal is to show that there exists a **1-SR** schema B such that $\mathcal{L}(B) = \mathcal{L}(P)$.

Notice that transitions in a DPDA are one of the following:

- (1) $\delta(q, a, A) = (p, B)$
- (2) $\delta(q, a, A) = (p, \varepsilon)$
- (3) $\delta(q, a, \varepsilon) = (p, A)$
- (4) $\delta(q, a, \varepsilon) = (p, \varepsilon)$
- (5) $\delta(q, \varepsilon, A) = (p, B)$
- (6) $\delta(q, \varepsilon, A) = (p, \varepsilon)$
- (7) $\delta(q, \varepsilon, \varepsilon) = (p, A)$
- (8) $\delta(q, \varepsilon, \varepsilon) = (p, \varepsilon)$

We can cut some of these out. That is,

- (1) $\delta(q, a, A) = (p, B)$ is unnecessary. We can add another state p_B and replace this with two transitions: $\delta(q, a, A) = (p_B, \varepsilon)$ and $\delta(p_B, \varepsilon, \varepsilon) = (p, B)$.
- (3) $\delta(q, a, \varepsilon) = (p, A)$ is unnecessary. We can add another state p_A and replace this with two transitions: $\delta(q, a, \varepsilon) = (p_A, \varepsilon)$ and $\delta(p_A, \varepsilon, \varepsilon) = (p, A)$.
- (5) $\delta(q, \varepsilon, A) = (p, B)$ is unnecessary for the same reason as (1).
- (8) $\delta(q, \varepsilon, \varepsilon) = (p, \varepsilon)$ is unnecessary, as it is the only transition between q and p , so we can simply unify q and p .

So we are left with four types of transitions:

- (2) a read & pop transition (read a character from the stream, and accordingly pop a character from the stack and change states);
- (4) a read transition (read a character from the stream, and accordingly change states);
- (6) a pop transition (pop a character from the stack without reading from the stream, and change states accordingly);
- (7) a push transition (push a character onto the stack without reading from the stream, and change states accordingly).

Notice the following interesting, yet simple, observation. Define \widehat{P} to be the same DPDA as P , except we now consider ε -moves as moves with an actual letter of the alphabet (call it, say $\widehat{\varepsilon}$). For $w \in \Sigma^*$, let us define $E(w)$ to be all the possible ways of placing $\widehat{\varepsilon}$ in w :

$$E(w) = \{ \widehat{\varepsilon}^{n_0} a_0 \widehat{\varepsilon}^{n_1} \dots \widehat{\varepsilon}^{n_k} a_k \widehat{\varepsilon}^{n_{k+1}} \mid w = a_0 \dots a_k, n_i \geq 0 \}$$

Then $w \in \mathcal{L}(P)$ if and only if there exists a $\hat{w} \in E(w)$ such that $\hat{w} \in \mathcal{L}(\hat{P})$.

Let's formalize the concept of \hat{P} . Given a DPDA $P = (\Sigma, \Gamma, Q, q_0, F, \delta)$, define $\hat{P} = (\Sigma \cup \hat{\varepsilon}, \Gamma, Q, q_0, F, \hat{\delta})$ where for $q \in Q, a \in \Sigma, A \in \Gamma, \hat{\delta}(q, a, A) = \delta(q, a, A)$. And instead of ε -transitions we have explicit $\hat{\varepsilon}$ -transitions:

$$\hat{\delta}(q, \hat{\varepsilon}, A) = \delta(q, \varepsilon, A)$$

Let us denote by $\hat{\vdash}$ the derivation relation of \hat{P} (contrasting with P 's, which we use \vdash for). We call \hat{P} the ε -free counterpart of P .

It would be nice to have a function mapping $w \in \mathcal{L}(P)$ to $\varepsilon(w) \in \mathcal{L}(\hat{P})$. We can construct such a function inductively. Since P is deterministic, for every configuration c_0 , there is at most one configuration c_1 such that $c_0 \vdash c_1$. Thus for a given configuration c_0 we can construct a unique path $c_0 \vdash c_1 \vdash \dots \vdash c_n \not\vdash$. We define $\varepsilon(c_0)$ by induction on n .

- (1) For $n = 0$, meaning $(p, w, W) \not\vdash$, we define $\varepsilon(p, w, W) = w$.
- (2) We split into two cases for $n > 0$:
 - (a) if $(p, w, W) \vdash (q, w, W')$, we define $\varepsilon(p, w, W) = \hat{\varepsilon}\varepsilon(q, w, W')$.
 - (b) if $(p, aw, W) \vdash (q, w, W')$, we define $\varepsilon(p, aw, W) = a\varepsilon(q, w, W')$.

And we let $\varepsilon(w) = \varepsilon(q_0, w, \varepsilon)$.

We prove the correctness of this definition in the following lemmas:

Lemma 2. $(p, w, W) \vdash (q, w', W')$ iff $(p, \varepsilon(p, w, W), W) \hat{\vdash} (q, \varepsilon(q, w', W'), W')$

Proof. If $(p, w, W) \vdash (q, w', W')$, we split into cases:

- (1) If $w' = w$, then $\varepsilon(p, w, W) = \hat{\varepsilon}\varepsilon(q, w, W')$ and the transition is an ε -transition. So

$$(p, \varepsilon(p, w, W), W) = (p, \hat{\varepsilon}\varepsilon(q, w, W'), W) \hat{\vdash} (q, \varepsilon(q, w, W'), W')$$

- (2) If $w = aw'$ then $\varepsilon(p, w, W) = a\varepsilon(q, w, W')$ and

$$(p, \varepsilon(p, w, W), W) = (p, a\varepsilon(q, w, W'), W') \hat{\vdash} (q, \varepsilon(q, w, W'), W')$$

And conversely, suppose $(p, w, W) \vdash (p', w'', W'')$. Then we just showed that $(p, \varepsilon(p, w, W), W) \hat{\vdash} (p', \varepsilon(p', w'', W''), W'')$. Thus $p' = q$, $W'' = W'$. Then we just split into cases regarding if $w = w''$, or $w = aw''$. ■

Immediately from this lemma we get by induction on $n \geq 0$ that

$$(p, w, W) \vdash^n (q, w', W') \iff (p, \varepsilon(p, w, W), W) \hat{\vdash}^n (q, \varepsilon(q, w', W'), W')$$

Now, $(p, w, W) \not\vdash$ if and only if $\varepsilon(p, w, W) = w$ by definition. So we get that

$$(p, w, W) \vdash^n (q, w', W') \not\vdash \iff (p, \varepsilon(p, w, W), W) \hat{\vdash}^n (q, w', W') \hat{\not\vdash}$$

In particular, for $f \in F$ (since we are assuming there are no ε -transitions from final states):

$$(q_0, w, \varepsilon) \vdash^* (f, \varepsilon, W) \iff (q_0, \varepsilon(w), \varepsilon) \hat{\vdash}^* (f, \varepsilon, W)$$

Thus we get the desired result:

Theorem 1. $w \in \mathcal{L}(P)$ if and only if $\varepsilon(w) \in \mathcal{L}(\hat{P})$.

We now prove a related result.

Lemma 3. For $\hat{w} \in E(w)$, if $(p, \hat{w}, W) \hat{\vdash} (q, \hat{u}, U)$ then $(p, w, W) \vdash (q, u, U)$. By induction this means that $(p, \hat{w}, W) \hat{\vdash}^* (q, \hat{u}, U)$ implies $(p, \hat{w}, W) \hat{\vdash} (q, u, U)$.

Proof. Notice that substrings of strings from $E(w)$ are strings in $E(u)$ for some substring u of w , so this statement is well-formed. Now, if $\hat{w} = \hat{\varepsilon}\hat{u}$ then we have $(p, \hat{\varepsilon}\hat{u}, W) \hat{\vdash} (q, \hat{u}, U)$, so by definition this means that (since $\hat{u} \in E(w)$), $(p, u, W) \vdash (q, u, U)$. And if $\hat{w} = a\hat{u}$ then we have the desired result as well, since the transitions of \hat{P} and P on elements of Σ are identical. ■

Theorem 2. $w \in \mathcal{L}(P)$ if and only if there exists a $\hat{w} \in E(w)$ such that $\hat{w} \in \mathcal{L}(\hat{P})$.

Proof. The first direction is true due to the above theorem, taking $\hat{w} = \varepsilon(w)$. For the converse, suppose $\hat{w} \in \mathcal{L}(\hat{P})$, meaning $(q_0, \hat{w}, \varepsilon) \hat{\vdash}^* (f, \varepsilon, W)$. By the above lemma, this means $(q_0, w, \varepsilon) \vdash^* (f, \varepsilon, W)$ and so $w \in \mathcal{L}(P)$. ■

Summarizing,

Corollary 1. The following are equivalent:

- (1) $w \in \mathcal{L}(P)$
- (2) $\varepsilon(w) \in \mathcal{L}(\hat{P})$
- (3) for some $\hat{w} \in E(w)$, $\hat{w} \in \mathcal{L}(\hat{P})$.

Let P be a DPDA. Let us define the *set of transition symbols* to be

$$\text{TrS} = \{ (p \vdash q), (p \vdash q + A), (p \vdash q - A) \mid p, q \in Q, A \in \Gamma \}$$

Furthermore, we define the *projections* $\pi_b, \pi_e: \text{TrS} \rightarrow Q$ where $\pi_b(p \vdash q \circ X) = p$ and $\pi_e(p \vdash q \circ X) = q$. We extend these to $\text{TrS}^+ \rightarrow Q$ where

$$\pi_b(t_1 \cdots t_n) = \pi_b(t_1), \quad \pi_e(t_1 \cdots t_n) = \pi_e(t_n)$$

We also define the *stack function* $\text{Stk}: \text{TrS}^* \times \Gamma^* \rightarrow \Gamma^*$ recursively as follows:

- (1) $\text{Stk}(\varepsilon; W) = W$
- (2) if \bar{t} ends with $\pi_e(\bar{t}) = q$, then
 - (a) $\text{Stk}(\bar{t}(q \vdash p); W) = \text{Stk}(\bar{t}; W)$
 - (b) $\text{Stk}(\bar{t}(q \vdash p + A); W) = A\text{Stk}(\bar{t}; W)$
 - (c) if $\text{Stk}(\bar{t}; W) = AW'$ then $\text{Stk}(\bar{t}(q \vdash p - A); W) = W'$

Otherwise Stk is left undefined.

We extend Stk to $\text{Stk}: \mathcal{P}(\text{TrS})^* \times \Gamma^* \rightarrow \mathcal{P}(Q \times \Gamma^* \times Q)$ by

$$\text{Stk}(T_1 \cdots T_n; W) = \{ (\pi_b(\bar{t}), \text{Stk}(\bar{t}; W), \pi_e(\bar{t})) \mid \bar{t} = t_1 \cdots t_n, t_i \in T_i \}$$

For ε , we let

$$\text{Stk}(\varepsilon; W) = \{ (q, W, q) \mid q \in Q \}$$

this agrees (pretty much) with our definition.

An immediate consequence of our definitions is

Lemma 4. *If $\bar{S}, \bar{T} \in \mathcal{P}(\text{TrS})^*$*

$$(p_1, W_1, p_2) \in \text{Stk}(\bar{S}; W_0) \text{ and } (p_2, W_2, p_3) \in \text{Stk}(\bar{T}; W_1)$$

then $(p_1, W_2, p_3) \in \text{Stk}(\bar{S}\bar{T}; W_0)$.

Given a letter $a \in \Sigma$, we define $\underline{a} \in \mathcal{P}(\text{TrS})$ by

$$\begin{aligned} \underline{a} = & \{ p \vdash q \mid \delta(p, a, \varepsilon) = (q, \varepsilon) \} \cup \\ & \{ p \vdash q - A \mid \delta(p, a, A) = (q, \varepsilon) \} \end{aligned}$$

Similarly we define

$$\begin{aligned} \widehat{\varepsilon} = & \{ p \vdash q + A \mid \delta(p, \varepsilon, \varepsilon) = (q, A) \} \cup \\ & \{ p \vdash q - A \mid \delta(p, \varepsilon, A) = (q, \varepsilon) \} \end{aligned}$$

Given a word $w = a_1 \cdots a_n \in (\Sigma \cup \widehat{\varepsilon})^*$, we define $\underline{w} = \underline{a}_1 \cdots \underline{a}_n \in \mathcal{P}(\text{TrS})^*$.

These definitions come together in the following lemma:

Lemma 5. *For $w \in (\Sigma \cup \widehat{\varepsilon})^+$,*

$$(p, W', q) \in \text{Stk}(\underline{w}; W) \iff (p, w, W) \hat{\vdash}^* (q, \varepsilon, W').$$

Proof. First suppose that $(p, W', q) \in \text{Stk}(\underline{w}; W)$. For the base case of w having a length of one, we split into two cases:

- (1) if $w = a \in \Sigma$, then $(p, W', q) \in \text{Stk}(\underline{a}; W)$ if and only if $p \vdash q \in \underline{a}$ and $W' = W$, or if $p \vdash q - A \in \underline{a}$ and $W = AW'$. In the former, this means that $\delta(p, a, \varepsilon) = (q, \varepsilon)$ and so $(p, a, W) \hat{\vdash} (q, \varepsilon, W = W')$ as required. In the latter case, this means that $\delta(p, a, A) = (q, \varepsilon)$ and so $(p, a, W = AW') \hat{\vdash} (q, \varepsilon, W')$ as required.
- (2) if $w = \widehat{\varepsilon}$, then $(p, W', q) \in \text{Stk}(\widehat{\varepsilon}; W)$ if and only if $p \vdash q + A \in \widehat{\varepsilon}$ and $W' = AW$, or if $p \vdash q - A \in \widehat{\varepsilon}$ and $W = AW'$. Then we do the same as before and chase definitions.

We now induct. If $(p, W', q) \in \text{Stk}(\underline{wa}; W)$, suppose $t_i \in \underline{w}_i$ and $s \in \underline{a}$ such that $W' = \text{Stk}(\bar{t}s; W)$ and $\pi_b(\bar{t}) = p$ and $\pi_e(s) = q$. Now, let $W'' = \text{Stk}(\bar{t}; W)$ and $p' = \pi_e(\bar{t})$. Then $(p, W'', p') \in \text{Stk}(\underline{w}; W)$, so by induction $(p, w, W) \hat{\vdash}^* (p', \varepsilon, W'')$ and so $(p, wa, W) \hat{\vdash}^* (p', a, W'')$. Furthermore, we know that $(p', W', q) \in \text{Stk}(\underline{a}; W'')$

and so by the base case $(p', a, W'') \hat{\vdash} (q, \varepsilon, W')$. Together this tells us that $(p, wa, W) \hat{\vdash}^* (q, \varepsilon, W')$ as required.

Now we assume the converse: that $(p, w, W) \hat{\vdash}^* (q, \varepsilon, W')$. Again we induct on the length of w (which is the length of the derivation). For the base case we have that $(p, a, W) \hat{\vdash} (q, \varepsilon, W')$. Then $\delta(p, a, -) = (q, -)$, and splitting the options here into cases on $-$, we see that indeed $(p, W', q) \in \text{Stk}(\underline{a}; W)$.

Now suppose that

$$(p, wa, W) \hat{\vdash} (q', a, W'') \hat{\vdash} (q, \varepsilon, W')$$

Then we have that $(p, w, W) \hat{\vdash}^* (q', a, W'')$ so by induction $(p, W'', q') \in \text{Stk}(\underline{w}; W)$. And by the base case we have that $(q', W', q) \in \text{Stk}(\underline{a}; W'')$, so then $(p, W', q) \in \text{Stk}(\underline{wa}; W)$ as required. \blacksquare

In particular we have that for $f \in F$ and $\hat{w} \in E(w)$ (strictly speaking, we need to quantify by saying “there exists a $\hat{w} \in E(w)$ ”):

$$(q_0, W, f) \in \text{Stk}(\hat{w}; \varepsilon) \iff (q_0, \hat{w}, \varepsilon) \hat{\vdash}^* (f, \varepsilon, W)$$

and thus

$$w \in \mathcal{L}(P) \iff \hat{w} \in \mathcal{L}(\hat{P}) \iff (q_0, W, f) \in \text{Stk}(\hat{w}; \varepsilon)$$

5.2 The main theorem

We define the *transition reduction operator* $\text{tr}: \text{TrS} \times \text{TrS} \rightarrow \text{TrS}$ as follows:

$$\begin{aligned} (p \vdash q + A), (p \vdash r - A) &\mapsto (p \vdash r) \\ (p \vdash q + A), (q \vdash r) &\mapsto (p \vdash r + A) \\ (p \vdash q), (q \vdash r + A) &\mapsto (p \vdash r + A) \\ (p \vdash q), (q \vdash r) &\mapsto (p \vdash r) \end{aligned}$$

This extends to $\text{tr}: \mathcal{P}(\text{TrS}) \times \mathcal{P}(\text{TrS}) \rightarrow \mathcal{P}(\text{TrS})$ by $\text{tr}(S_1, S_2) = \{ \text{tr}(s_1, s_2) \mid s_i \in S_i \}$.

The following lemma is immediate:

Lemma 6. *If $\text{tr}(t_n, t_{n+1})$ is defined, then for every $W \in \Gamma^*$,*

$$\text{Stk}(t_1 \cdots t_{n-1} \text{tr}(t_n, t_{n+1}); W) = \text{Stk}(t_1 \cdots t_n t_{n+1}; W)$$

and the lhs is defined iff the rhs is.

This implies

Lemma 7. *If $\text{tr}(t_i, t_{i+1})$ is defined for any i , then for every $W \in \Gamma^*$,*

$$\text{Stk}(t_1 \cdots \text{tr}(t_i, t_{i+1}) \cdots t_n; W) = \text{Stk}(t_1 \cdots t_i t_{i+1} \cdots t_n; W)$$

An immediate lemma is the following:

Lemma 8. $\pi_X(t, s) = \pi_X(\text{tr}(t, s))$ for $X \in \{b, e\}$. Thus if $\bar{t} \in T^+$ and $\bar{s} \in T^+$ is the result of applying tr to \bar{t} at an arbitrary position, then $\pi_X(\bar{t}) = \pi_X(\bar{s})$ for $X \in \{b, e\}$.

We define Stk_ε similar to Stk , except we allow ε -transitions. Specifically, $\text{Stk}_\varepsilon: Q \times \text{TrS}^* \times \Gamma^* \rightarrow \mathcal{P}_{<\omega}(\Gamma^* \times Q)$ where first we define

$$\text{Stk}_\varepsilon(p_0, \varepsilon; W) = \{ (W', q) \mid (p_0, \varepsilon, W) \vdash^* (q, \varepsilon, W') \}$$

and we recurse:

- (1) for $(p \vdash q)$, if $(W', p) \in \text{Stk}_\varepsilon(p_0, \bar{t}; W)$ and $(q, \varepsilon, W') \vdash^* (s, \varepsilon, W'')$ then $(W'', s) \in \text{Stk}_\varepsilon(p_0, \bar{t}(p \vdash q); W)$.
- (2) for $(p \vdash q + A)$, if $(W', p) \in \text{Stk}_\varepsilon(p_0, \bar{t}; W)$ and $(q, \varepsilon, AW') \vdash^* (s, \varepsilon, W'')$, then $(W'', s) \in \text{Stk}_\varepsilon(p_0, \bar{t}(p \vdash q + A); W)$.
- (3) for $(p \vdash q - A)$, if $(AW', p) \in \text{Stk}_\varepsilon(p_0, \bar{t}; W)$ and $(q, \varepsilon, W') \vdash^* (s, \varepsilon, W'')$, then $(W'', s) \in \text{Stk}_\varepsilon(p_0, \bar{t}(p \vdash q + A); W)$.

We extend Stk_ε to $\mathcal{P}_{<\omega}(\text{TrS})^* \times \Gamma^* \rightarrow \mathcal{P}_{<\omega}(Q \times \Gamma^* \times Q)$ where

$$\text{Stk}_\varepsilon(\bar{S}; W) = \{ (p_0, W', p_e) \mid p_0 \in Q, (W', p_e) \in \text{Stk}_\varepsilon(p_0, \bar{S}; W) \}$$

Furthermore, for $\bar{S} = S_1 \cdots S_k \in \mathcal{P}(\text{TrS})^*$, let us define

$$\underline{E}(\bar{S}) = \{ \hat{\varepsilon}^{n_0} S_1 \hat{\varepsilon}^{n_1} \cdots S_k \hat{\varepsilon}^{n_k} \mid n_i \geq 0 \}$$

so that $\underline{E}(\underline{w}) = \underline{E}(w)$.

An immediate result by definition is given two strings of the same length \bar{S} and \bar{T} , if we define their union $\bar{S} \cup \bar{T}$ pointwise (meaning $(\bar{S} \cup \bar{T})_i = S_i \cup T_i$), then we have

$$\text{Stk}_\varepsilon(\bar{S} \cup \bar{T}; W) = \text{Stk}_\varepsilon(\bar{S}; W) \cup \text{Stk}_\varepsilon(\bar{T}; W)$$

Lemma 9.

$$\text{Stk}_\varepsilon(\bar{S}; W) = \bigcup_{\hat{S} \in \underline{E}(\bar{S})} \text{Stk}(\hat{S}; W)$$

in particular

$$\text{Stk}_\varepsilon(\underline{w}; \varepsilon) = \bigcup_{\hat{w} \in \underline{E}(w)} \text{Stk}(\hat{w}; \varepsilon)$$

Proof. This is done via induction on \bar{S} . For $\bar{S} = \varepsilon$, if $(p_0, W', p_e) \in \text{Stk}_\varepsilon(\varepsilon; W)$ then $(p_0, \varepsilon, W) \vdash^n (p_e, \varepsilon, W')$ then $(p_0, W', p_e) \in \text{Stk}(\hat{\varepsilon}^n; W)$. And so on, the idea being that when $(q, \varepsilon, W') \vdash^n (s, \varepsilon, W'')$, we replace this with $\hat{\varepsilon}^n$. ■

Lemma 10. For $w \in \Sigma^*$,

$$(p, W', q) \in \text{Stk}_\varepsilon(\underline{w}; W) \iff (p, w, W) \vdash^* (q, \varepsilon, W')$$

Proof. This is immediate due to the previous lemma and lemma 5.

We define the relation \vdash on Q , where $p \vdash q$ means that $(p, \varepsilon, \varepsilon) \vdash (q, \varepsilon, W)$. And \vdash^* is its reflexive and transitive closure.

We now define our **1-SR** schema, $B = (\Sigma, A, \square, \text{Acc}, \bar{\beta})$. We begin by defining

$$A = \mathcal{P}(\text{TrS}) \cup \mathcal{P}(Q\top)$$

Now we define $\square = \{ f\top \mid f \in F \} \in \mathcal{P}(Q\top)$, and we define $\bar{\beta}$ as follows:

- (1) for $S \in \mathcal{P}(\text{TrS})$, $\bar{\beta}(S, \varepsilon) = S \cup \text{tr}(S, \hat{\varepsilon}) \cup \text{tr}(\hat{\varepsilon}, S)$ if this is not equal to S ;
- (2) for $S_1, S_2 \in \mathcal{P}(\text{TrS})$, $\bar{\beta}(S_1, S_2) = \text{tr}(S_1, S_2)$ if non-empty;
- (3) for $S_1 \in \mathcal{P}(\text{TrS})$, $S_2 \in \mathcal{P}(Q\top)$,

$$\bar{\beta}(S_1, S_2) = \{ p\top \mid p \vdash q \text{ or } p \vdash q + A \in S_1 \text{ and } q \vdash^* s \text{ such that } s\top \in S_2 \}$$

And we define $S \in \text{Acc}$ for $S \in \mathcal{P}(Q\top)$ if and only if $q_0 \vdash^* p$ for some $p\top \in S$.

Lemma 11. For $\bar{S} \in \mathcal{P}(\text{TrS})^*$, $\text{Stk}_\varepsilon(\bar{S}) = \text{Stk}_\varepsilon(\beta\bar{S})$.

Proof. We must split into two cases: the first when only a single symbol of \bar{S} is reduced, and the second when two symbols are reduced.

In the first case, suppose $\bar{S} = S_1 \cdots S_i \cdots S_n$, and $\beta\bar{S} = S_1 \cdots \bar{\beta}S_i \cdots S_n$. Now we know that $\bar{\beta}(S_i) = S_i \cup \text{tr}(S_i, \hat{\varepsilon}) \cup \text{tr}(\hat{\varepsilon}, S_i)$, thus

$$\text{Stk}_\varepsilon(\beta\bar{S}) = \text{Stk}_\varepsilon(\bar{S}) \cup \text{Stk}_\varepsilon(S_1 \cdots \text{tr}(S_i, \hat{\varepsilon}) \cdots S_n) \cup \text{Stk}_\varepsilon(S_1 \cdots \text{tr}(\hat{\varepsilon}, S_i) \cdots S_n)$$

We see then that $\text{Stk}_\varepsilon(\bar{S}) \subseteq \text{Stk}_\varepsilon(\beta\bar{S})$. Conversely, let us look at $E_1 = \underline{E}(S_1 \cdots \text{tr}(S_i, \hat{\varepsilon}) \cdots S_n)$. Since Stk is tr-invariant, taking Stk of E_1 is equal to Stk over $\underline{E}(S_1 \cdots S_i \hat{\varepsilon} \cdots S_n) \subseteq E(\bar{S})$. Thus we see that $\text{Stk}_\varepsilon(S_1 \cdots \text{tr}(S_i, \hat{\varepsilon}) \cdots S_n) \subseteq \text{Stk}_\varepsilon(\bar{S})$, and similar for $\text{tr}(\hat{\varepsilon}, S_i)$, as required.

For the second case, this is just due to Stk being tr-invariant. ■

Lemma 12. For $\bar{S} \in \mathcal{P}(\text{TrS})^*$, $\bar{S}\square \rightarrow^* \top \in \text{Acc}$ if and only if $(q_0, W, f) \in \text{Stk}_\varepsilon(\bar{S}; \varepsilon)$.

Proof. Let $\bar{T} = \hat{\beta}^*(\bar{S})$, then we know that $\bar{S}\square \rightarrow^* \bar{T}\square$ and $\text{Stk}_\varepsilon(\bar{S}) = \text{Stk}_\varepsilon(\bar{T})$.

Now, we know that $\bar{T}\square \rightarrow^* \top \in \text{Acc}$ if and only if $p_i \vdash q_i + A_i \in T_i$ such that

- (1) $A_i \in \Gamma_\varepsilon$
- (2) $q_i \vdash^* p_{i+1}$
- (3) $q_0 \vdash^* p_1$ and $q_n \vdash^* f \in F$

And it is not hard to see that this is equivalent. ■

In particular we have that $w \in \mathcal{L}(B)$ if and only if $(q_0, W, f) \in \text{Stk}_\varepsilon(\underline{w}; \varepsilon)$ if and only if $(q_0, w, \varepsilon) \vdash^* (f, \varepsilon, W')$ if and only if $w \in \mathcal{L}(P)$ as required.

Theorem 3. B and P are equivalent.

Theorem 4. $\text{DCFL} \subseteq \text{1-SR}$