Case Study: Deciding a Language

Kai Engelhardt

April 4, 2018

1 Introduction

Let $\mathcal{L}$ be the language generated by the grammar[1] $K \to \mathsf{a} \mid \mathsf{b}KK$.

There's hope for a counter-based solution. To find and justify that solution, we need to state more properties of $\mathcal{L}$. We do so using the following notation. We write $|w|$ for the *length* of word w. We write $\|w\|_\ell$ for the number of letters ℓ in word w. Formally, we define these two notions inductively by

$$|\epsilon| = 0 \qquad \|\epsilon\|_\ell = 0$$

$$|w\ell| = |w| + 1 \qquad \|w\ell'\|_\ell = \|w\|_\ell + \begin{cases} 1 & \text{if } \ell' = \ell \\ 0 & \text{if } \ell' \neq \ell \end{cases}$$

We say that v is a *proper prefix* of w and write $v < w$ if there exists a word $v' \neq \epsilon$ such that $w = vv'$.

As will become clear shortly, there's some benefit to be able to talk about *language concatenation* and its special case, *language exponentiation*. For two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ we define their concatenation $\mathcal{L}_1 \cdot \mathcal{L}_2 = \{\, vw \mid v \in \mathcal{L}_1 \wedge w \in \mathcal{L}_2 \,\}$. For $k \in \mathbb{N}$ we define $\mathcal{L}^k$ inductively by

$$\mathcal{L}^0 = \{\epsilon\} \qquad \mathcal{L}^{k+1} = \mathcal{L}^k \cdot \mathcal{L} \ . \tag{1}$$

We are now ready to formulate some interesting properties of our language $\mathcal{L}$.

Lemma 1 For all $w \in \{\mathsf{a}, \mathsf{b}\}^*$ and $k > 0$ we have

$$\mathsf{a}w \in \mathcal{L} \Leftrightarrow w = \epsilon \tag{2}$$

$$w \in \mathcal{L} \Rightarrow |w| \bmod 2 = 1 \tag{3}$$

$$w \in \mathcal{L} \Leftrightarrow \|w\|_\mathsf{b} + 1 = \|w\|_\mathsf{a} \wedge \forall v < w\,(\|v\|_\mathsf{a} \leq \|w\|_\mathsf{b}) \tag{4}$$

$$\mathsf{a}w \in \mathcal{L}^k \Leftrightarrow w \in \mathcal{L}^{k-1} \tag{5}$$

$$\mathsf{b}w \in \mathcal{L}^k \Leftrightarrow w \in \mathcal{L}^{k+1} \tag{6}$$

[1]If we knew more about formal languages, we'd realise that $\mathcal{L}$ is context-free but not regular, and we could employ a universal construction of a stack-based program for deciding $\mathcal{L}$. Alas, we as COMP2111 don't know these things. Take COMP3131 or COMP4141 to learn more.

Proof: For (2) we observe that $\mathsf{a} \in \mathcal{L}$ establishes the "$\Leftarrow$" direction of the claimed "$\Leftrightarrow$". The "$\Rightarrow$" direction follows from the fact that any word $w \in \mathcal{L}$ with $|w| > 1$ must have been generated by first using the production $K \to \mathsf{b}KK$, resulting in a leading b.

We prove (3) by induction on words[2] in $\mathcal{L}$. Our base case is the word a, which is indeed of odd length, generated by the production $K \to \mathsf{a}$. Our inductive hypothesis is that for some $k > 0$ we have that (3) holds for all words of length up to k. For the inductive step, consider a word $w \in \mathcal{L}$ of length $|w| = k > 1$. We already know that its generation necessarily starts with the production $K \to \mathsf{b}KK$. Each of the symbols K on the right then produces a word in $\mathcal{L}$, that is, $w = \mathsf{b}vv'$ for some $v, v' \in \mathcal{L}$. Both $|v|$ and $|v'|$ must be less than k, so our inductive hypothesis applies. We calculate $|w| \bmod 2 = |\mathsf{b}vv'| \bmod 2 = (1 + |v| + |v'|) \bmod 2 = (1 + |v| \bmod 2 + |v'| \bmod 2) \bmod 2 = (1 + 1 + 1) \bmod 2 = 1$.

We prove (4) by induction on words. The base case is $w = \epsilon$. Both sides of (4) evaluate to FALSE because $\epsilon \notin \mathcal{L}$ by (3) and $\|\epsilon\|_{\mathsf{b}} + 1 = 0 + 1 = 1 \neq 0 = \|\epsilon\|_{\mathsf{a}}$. It is easy the check that the two words of length 1, a and b similarly make (4) true: for a both sides are true—for b both are false.

For the inductive hypothesis, assume (4) holds for all words of length up to $k > 0$. Let w be a word with $|w| = k + 1$.

For "$\Rightarrow$" suppose $w \in \mathcal{L}$. Then, as we argued before, $w = \mathsf{b}vv'$ for some $v, v' \in \mathcal{L}$ each of length less than k. So the inductive hypothesis applies to both, v and v'. We calculate $\|w\|_{\mathsf{b}} + 1 = \|\mathsf{b}vv'\|_{\mathsf{b}} = 1 + \|v\|_{\mathsf{b}} + \|v'\|_{\mathsf{b}} + 1 = 1 + \|v\|_{\mathsf{a}} - 1 + \|v'\|_{\mathsf{a}} - 1 + 1 = \|vv'\|_{\mathsf{a}} = \|\mathsf{b}vv'\|_{\mathsf{a}} = \|w\|_{\mathsf{a}}$. Next let $w' < w$.

Case $w' = \epsilon$**:** we clearly have $\|w'\|_{\mathsf{a}} = 0 \leq 0 = \|w'\|_{\mathsf{b}}$.

Case $w' = \mathsf{b}w'' < v$**:** the inductive hypothesis for v yields $\|w'\|_{\mathsf{a}} = \|\mathsf{b}w''\|_{\mathsf{a}} = \|w''\|_{\mathsf{a}} \leq \|w''\|_{\mathsf{b}} < 1 + \|w''\|_{\mathsf{b}} = \|\mathsf{b}w''\|_{\mathsf{b}} = \|w'\|_{\mathsf{b}}$.

Case $w' = \mathsf{b}v$**:** the inductive hypothesis for v yields $\|w'\|_{\mathsf{a}} = \|v\|_{\mathsf{a}} \leq \|v\|_{\mathsf{b}} + 1 = \|w'\|_{\mathsf{b}}$.

Case $\mathsf{b}v < w' = \mathsf{b}vw''$**:** by the inductive hypothesis we have that $\|w'\|_{\mathsf{a}} = \|\mathsf{b}vw''\|_{\mathsf{a}} = 1 + \|v\|_{\mathsf{b}} + \|w''\|_{\mathsf{a}} \leq 1 + \|v\|_{\mathsf{b}} + \|w''\|_{\mathsf{b}} = \|\mathsf{b}vw''\|_{\mathsf{b}} = \|w'\|_{\mathsf{b}}$.

For "$\Leftarrow$" suppose $\|w\|_{\mathsf{b}} + 1 = \|w\|_{\mathsf{a}} \wedge \forall v < w\,(\|v\|_{\mathsf{a}} \leq \|w\|_{\mathsf{b}})$. The instance $|v| = 1$ yields that w's first letter is a b. Let v and v' such that $w = \mathsf{b}vv'$ and $\mathsf{b}v$ is the shortest prefix of w satisfying $\|v\|_{\mathsf{a}} = \|v\|_{\mathsf{b}} + 1$. (That split exists because setting $v' = \mathsf{a}$ gives one such split that fails at most the minimality condition for v's length.) The inductive hypothesis yields that $v \in \mathcal{L}$. Next observe that $\|v'\|_{\mathsf{b}} + 1 = \|w\|_{\mathsf{b}} - \|\mathsf{b}v\|_{\mathsf{b}} + 1 = (\|w\|_{\mathsf{b}} + 1) - (\|v\|_{\mathsf{b}} + 1) = \|w\|_{\mathsf{a}} + \|v\|_{\mathsf{a}} = \|v'\|_{\mathsf{a}}$. Hence the inductive hypothesis also yields $v' \in \mathcal{L}$. Thus $w = \mathsf{b}vv' \in \mathcal{L}$.

(5) is a consequence of (2).

(6) follows from the definition of the grammar: there's exactly one production to generate a b and that is $K \to \mathsf{b}KK$. ■

[2] A mathematically more precise way to express this inductive argument is by talking about induction on the length of the sequence of productions to generate a word from K.

2 Specification

$$b : [w \in \{\mathsf{a}, \mathsf{b}\}^*, b \Leftrightarrow w \in \mathcal{L}] \tag{7}$$

3 Program

We suggest the following program. We use k as loop counter, indicating which letter of the word $w = w[0]..w[|w| - 1]$ is checked next. Inspired by (5) and (6) the counter c holds how many words in $\mathcal{L}$ we still have to see, whence it is initialised to 1. (This is simpler/more elegant than initialising to 0 and counting the difference between as and bs seen so far in $w[0]..w[k - 1]$ — as I did in class.)

```
**var** $k, c \cdot$
    $k := 0;$
    $c := 1;$
    **while** $k < |w| \wedge c > 0$ **do**
        **if** $w[k] = \mathsf{a}$ **then** $c := c - 1$ **else** $c := c + 1$ **fi**
        $k := k + 1$
    **od**
    $b := k = |w| \wedge c = 0$
```

4 Verification

We pepper the program with assertions, leaving the crucial invariant I undefined for now. All assertions are chosen to fit the Hoare logic rules for the construct these assertion enclose. The only gaps that need any extra justification are pairs of adjacent assertions where the postcondition of the previous group does not match the precondition of the next. We hope to find some inspiration for the shape of the invariant when we inspect the gaps later.

```
$\{w \in \{\mathsf{a},\mathsf{b}\}^*\}$                                   (8)
$\{I[^1/_c][^0/_k]\}$                                                      (9)
$k := 0;$
$\{I[^1/_c]\}$                                                             (10)
$c := 1;$
$\{I\}$                                                                    (11)
while $k < |w| \wedge c > 0$ do
    $\{I \wedge k < |w| \wedge c > 0\}$                                    (12)
    $\{J\}$                                                                (13)
    if $w[k] = \mathsf{a}$ then
        $\{J \wedge w[k] = \mathsf{a}\}$                                   (14)
        $\left\{I[^{k+1}/_k][^{c-1}/_c]\right\}$                           (15)
        $c := c - 1$
        $\left\{I[^{k+1}/_k]\right\}$                                      (16)
    else
        $\{J \wedge w[k] = \mathsf{b}\}$                                   (17)
        $\left\{I[^{k+1}/_k][^{c+1}/_c]\right\}$                           (18)
        $c := c + 1$
        $\left\{I[^{k+1}/_k]\right\}$                                      (19)
    fi
    $\left\{I[^{k+1}/_k]\right\}$                                          (20)
    $k := k + 1$
    $\{I\}$                                                                (21)
od
$\{I \wedge (k = |w| \vee c \leq 0)\}$                                     (22)
$\{k = |w| \wedge c \leq 0 \Leftrightarrow w \in \mathcal{L}\}$            (23)
$b := k = |w| \wedge c \leq 0$
$\{b \Leftrightarrow w \in \mathcal{L}\}$                                  (24)
```

where

$$J = \begin{pmatrix} (w[k] = \mathsf{a} \Rightarrow (I[^{k+1}/_k])[^{c-1}/_c]) \wedge \\ (w[k] = \mathsf{b} \Rightarrow (I[^{k+1}/_k])[^{c+1}/_c]) \end{pmatrix}$$

is chosen with the help of the **if** rule of Hoare logic to ensure the implications (14) $\Rightarrow$ (15) and (17) $\Rightarrow$ (18) are trivially valid.

To find the invariant I, let us list the constraints on it.

$$\begin{array}{rc}
(8) \Rightarrow (9): & w \in \{\mathsf{a},\mathsf{b}\}^* \Rightarrow I[^1/_c][^0/_k] \\
(12) \Rightarrow (13): & I \wedge k < |w| \wedge c > 0 \Rightarrow \begin{pmatrix} (w[k] = \mathsf{a} \Rightarrow (I[^{k+1}/_k])[^{c-1}/_c]) \wedge \\ (w[k] = \mathsf{b} \Rightarrow (I[^{k+1}/_k])[^{c+1}/_c]) \end{pmatrix} \\
(22) \Rightarrow (23): & I \wedge (k = |w| \vee c \leq 0) \Rightarrow (k = |w| \wedge c \leq 0 \Leftrightarrow w \in \mathcal{L})
\end{array}$$

The first can be accommodated simply by having $w \in \{\mathsf{a},\mathsf{b}\}^*$ as a conjunct in I. It is often a good idea to add to I some constraints on the program variables that state limits on their value ranges. So we add conjuncts $0 \leq k \leq |w|$ and $0 \leq c$.

What's missing is a link between the program variables k and c on the one hand and the inherent state of knowledge after k iterations of the loop. What does it mean that c has a certain value? The partial answer is that we have seen $c - 1$ more bs than as in $w[0..k-1]$. In other words, we need to find c more as than bs in $w[k..]$ if $w \in \mathcal{L}$. More precisely, we have that $w \in \mathcal{L}$ iff $w[k..]$ is the concatenation of c words in $\mathcal{L}$, that is, $w[k..] \in \mathcal{L}^c$.

$$I = w \in \{\mathsf{a},\mathsf{b}\}^* \wedge 0 \leq k \leq |w| \wedge 0 \leq c \wedge (w \in \mathcal{L} \Leftrightarrow w[k..] \in \mathcal{L}^c)$$

We shall find out whether this I works by checking the three constraints.

4.0.1 Validity of (8)$\Rightarrow$(9)

$$\begin{aligned} w \in \{\mathsf{a},\mathsf{b}\}^* &\Rightarrow w \in \{\mathsf{a},\mathsf{b}\}^* \wedge 0 \leq 0 \leq |w| \wedge 0 \leq 1 \wedge (w \in \mathcal{L} \Leftrightarrow w[0..] \in \mathcal{L}^1) \\ &\Leftrightarrow I[^1/_c][^0/_k] \end{aligned}$$

4.0.2 Validity of (12)$\Rightarrow$(13)

$$\begin{aligned}
& I \wedge k < |w| \wedge c > 0 \\
& \Leftrightarrow w \in \{\mathsf{a},\mathsf{b}\}^* \wedge 0 \leq k \leq |w| \wedge 0 \leq c \wedge (w \in \mathcal{L} \Leftrightarrow w[k..] \in \mathcal{L}^c) \wedge k < |w| \wedge c > 0 \\
& \Rightarrow \begin{pmatrix} (w[k] = \mathsf{a} \Rightarrow w \in \{\mathsf{a},\mathsf{b}\}^* \wedge 0 \leq k+1 \leq |w| \wedge 0 \leq c-1 \wedge (w \in \mathcal{L} \Leftrightarrow w[k+1..] \in \mathcal{L}^{c-1})) \wedge \\ (w[k] = \mathsf{b} \Rightarrow w \in \{\mathsf{a},\mathsf{b}\}^* \wedge 0 \leq k+1 \leq |w| \wedge 0 \leq c+1 \wedge (w \in \mathcal{L} \Leftrightarrow w[k+1..] \in \mathcal{L}^{c+1})) \end{pmatrix} \\
& \Rightarrow \begin{pmatrix} (w[k] = \mathsf{a} \Rightarrow ((w \in \{\mathsf{a},\mathsf{b}\}^* \wedge 0 \leq k \leq |w| \wedge 0 \leq c \wedge (w \in \mathcal{L} \Leftrightarrow w[k..] \in \mathcal{L}^c))[^{k+1}/_k])[^{c-1}/_c]) \wedge \\ (w[k] = \mathsf{b} \Rightarrow ((w \in \{\mathsf{a},\mathsf{b}\}^* \wedge 0 \leq k \leq |w| \wedge 0 \leq c \wedge (w \in \mathcal{L} \Leftrightarrow w[k..] \in \mathcal{L}^c))[^{k+1}/_k])[^{c+1}/_c]) \end{pmatrix} \\
& \Leftrightarrow \begin{pmatrix} (w[k] = \mathsf{a} \Rightarrow (I[^{k+1}/_k])[^{c-1}/_c]) \wedge \\ (w[k] = \mathsf{b} \Rightarrow (I[^{k+1}/_k])[^{c+1}/_c]) \end{pmatrix}
\end{aligned}$$

4.0.3 Validity of (22)⇒(23)

$$
\begin{aligned}
&I \wedge (k = |w| \vee c \leq 0) \\
&\Leftrightarrow w \in \{\mathsf{a},\mathsf{b}\}^* \wedge 0 \leq k \leq |w| \wedge 0 \leq c \wedge (w \in \mathcal{L} \Leftrightarrow w[k..] \in \mathcal{L}^c) \wedge (k = |w| \vee c \leq 0)
\end{aligned}
$$

We can now distinguish three cases of how the last conjunct $(k = |w| \vee c \leq 0)$ holds and what that means for the second to last conjunct $(w \in \mathcal{L} \Leftrightarrow w[k..] \in \mathcal{L}^c)$:

1. Both disjuncts hold so we have $w \in \mathcal{L} \Leftrightarrow \epsilon \in \mathcal{L}^0$. By (1) we have the LHS and thus also the RHS, $w \in \mathcal{L}$.
2. Only the first conjunct holds so we have $c > 0$ and $w \in \mathcal{L} \Leftrightarrow \epsilon \in \mathcal{L}^c$. But $\epsilon \notin \mathcal{L}$ implies that $\epsilon \notin \mathcal{L}^c$ for $c > 0$. Thus $w \notin \mathcal{L}$.
3. Only the second conjunct holds so we have $k < |w|$ and $w \in \mathcal{L} \Leftrightarrow w[k..] \in \mathcal{L}^0$. But $w[k..] \neq \epsilon$ and only $\epsilon \in \mathcal{L}^0$. Thus $w \notin \mathcal{L}$.

By case analysis we have thus established

$$(22) \Rightarrow k = |w| \wedge c \leq 0 \Leftrightarrow w \in \mathcal{L} \ .$$

We conclude that our I indeed works as a loop invariant. Our Hoare-logic proof together with the observation that the program only ever changes the variables k, c, and b is then sufficient to establish that our program (without the local variable introduction) refines $b, k, c : [w \in \{\mathsf{a},\mathsf{b}\}^*, b \Leftrightarrow w \in \mathcal{L}]$. An application of the local variable rule found on page 5 of slides 6 completes the proof.