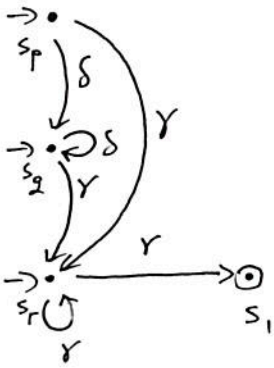
12.1 As an alternative to using the given algorithm for determining the set of configurations from which an infinite accepting run is possible, one can relatively easy (by human ingenuity) check which (q, γ) configurations satisfy (2).

Since from (p, γ) and (q, γ) we are unable to loop through p, q, r again, (p, δ) and (q, δ) cannot reach r and (r, δ) is entirely stuck, we conclude that (r, γ) is the only configuration satisfying (2) out of the ones we should consider.

Then the set of configurations from which an infinite accepting run is possible are $pre^*((r, \gamma))$ below:



Alternatively written, the wanted configurations are

$$(r, \gamma\Gamma^*),\ (q, \delta^*\gamma\gamma\Gamma^*), \text{ and } (p, \delta^*\gamma\gamma\Gamma^*).$$

12.2. (a) Let BUTA $A = (Q, \to, Q_F)$ such that $L = L(A)$ and define $G = (N, T, \to, S)$ where $N = \{N_q \mid q \in Q\}$, $T = \{a \in \Sigma \mid rk(a) = 0\}$, and

$S \to N_{q_f}$ for all $q_f \in Q_F$

$N_q \to N_{q_0} \dots N_{q_{m-1}}$ if $(q_0, \dots, q_{m-1}) \to_? q$, $m \geq 1$

$N_q \to a$ if $() \to_a q$

(b) Wlog. assume $G = (N, T, \to, S)$ is a grammar in Chomsky normal form. Define $Q = \{ q_n \mid n \in N \}$, $Q_F = \{ S \}$, $\Sigma = \{ t/0 \mid t \in T \} \cup \{ ?/2 \}$ where „?" is a new symbol. Then by setting (in BUTA A)

$$() \to_a q_n \quad \text{if} \quad n \to a \quad , \quad n \in N, a \in T$$

$$(q_{n_1}, q_{n_2}) \to_? q \quad \text{if} \quad n \to n_1 n_2 \quad , \quad n, n_1, n_2 \in N$$

we can conclude that $L(G) = \text{yield}(A)$.

12.3.(a) It was proved in class that $\forall k \in \mathbb{N}$, if $p \in R_k$ then there is an A-tree t_p for p of height $\leq k$.

This in particular implies there is a $\leq k$ A-tree for $p \in R_k \setminus R_{k-1}$ when $R_k \setminus R_{k-1} \neq \emptyset$.

Assume, by contradiction, there is a smaller height $k' < k$ of some A-tree t'_p for p. Then, by definition of $(R_i)_{i \geq 0}$ we get that $p \in R_{k'} \subseteq R_{k-1}$ ↯.

(b) <u>Claim</u>: $\exists$ A-tree for p of height $\leq k \Rightarrow p \in R_k$

<u>Proof</u>. IA. $k = 0 \Rightarrow R_0 = \emptyset \Rightarrow$ „$\exists$ A-tree for p of h. ≤ 0" = false ✓

$k = 1 \Rightarrow R_1 = \bigcup_{a \in \Sigma} \{ q \in Q \mid \to_a p \} \Rightarrow$ for any $p \in Q$, if $\exists$ A-tree for p of h. $\leq k$, the tree will be $\circ_p$ and the corresponding run will imply that $p \in R_1$.

IS. Assume $\exists$ A-tree for p of height $\leq k+1$ and the IH: $\forall p' \in Q$ $\exists$ A-tree for p' of height $\leq k \Rightarrow p' \in R_k$.

If $\exists$ A-tree for p of height $\leq k$ then $p \in R_k \subseteq R_{k+1}$ by IH, o/w if $\exists$ A-tree for p of height $= k+1$ then

$$p_0, \ldots, p_{n-1} \to_a q \quad \text{for some } a \in \Sigma$$

such that $\exists$ A-tree for p_i of height $\leq k$ so by IH $p_i \in R_k$ and $\in R_{k+1}$.

(c) $R_0 = \emptyset$, $R_1 = \{\ell_0\}$, $R_2 = \{\ell_0, \ell_1\}$, $R_3 = \{\ell_0, \ell_1, \ell_2\}$, $R_4 = R_3$.

Since $R_3 \cap Q_F \neq \emptyset$ it means ~~t~~ $L(A) \neq \emptyset$.

R.4. One can compute $(R_i)_{i \geq 0}$ as in the emptiness check. Let R be the stabilizing set of states. This set describes the set of productive states.

One can also check by looking through the tree automata rules entirely in R (i.e. $\rightarrow_{|R}$) whether there is any A-tree for ℓ containing ℓ as a „sub"-node. Equivalently, there will be a sequence of transition starting with ℓ on lhs and discovering ℓ on rhs.

A possible summary of an algorithm answering whether „L(A) is finite ~~fo~~ given BUTA A" is

```
compute R
if R ∩ Q_F = ∅ return ~~false~~ finite
else if ∃ ℓ ∈ R s.t. ℓ can be pumped
    then
        return infinite
    else
        return finite
    endif
endif.
```