Lecture 17

Language classes with respect to time and space complexity

Language classes with respect to time and space complexity

- **Nondeterministic TM**

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **space complexity $S(n)$**

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **space complexity $S(n)$**
    - —**for any string of lenght n**

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **space complexity $S(n)$**
    - —for any string of lenght n
    - —any sequence of moves

Language classes with respect to time and space complexity

- Nondeterministic TM
  - space complexity $S(n)$
    - —for any string of lenght n
    - —any sequence of moves
    - —on any work tape

Language classes with respect to time and space complexity

- Nondeterministic TM
  - space complexity $S(n)$
    - —for any string of lenght n
    - —any sequence of moves
    - —on any work tape
    - —uses at most $S(n)$ cells

Language classes with respect to time and space complexity

- Nondeterministic TM
  - space complexity $S(n)$
    - for any string of lenght n
    - any sequence of moves
    - on any work tape
    - uses at most $S(n)$ cells
- Language L

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **space complexity $S(n)$**
    - —**for any string of lenght n**
    - —**any sequence of moves**
    - —**on any work tape**
    - —**uses at most $S(n)$ cells**

- **Language L**
  - **nondeterministic space complexity $S(n)$**

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **space complexity $S(n)$**
    - —for any string of lenght n
    - —any sequence of moves
    - —on any work tape
    - —uses at most $S(n)$ cells

- **Language L**
  - **nondeterministic space complexity $S(n)$**
    - —if and only if there is a nondeterministic TM M with space complexity $S(n)$ that accepts the language $L(M) = L$

Language classes with respect to time and space complexity

Language classes with respect to time and space complexity

- **Nondeterministic TM**

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **time complexity $T(n)$**

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **time complexity *T*(*n*)**
    - —**for any string of lenght *n***

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **time complexity $T(n)$**
    - —for any string of lenght n
    - —any sequence of moves

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **time complexity $T(n)$**
    - —for any string of lenght n
    - —any sequence of moves
    - —moves the head at most $T(n)$ times

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **time complexity $T(n)$**
    - —for any string of lenght n
    - —any sequence of moves
    - —moves the head at most $T(n)$ times
- **Language L**

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **time complexity $T(n)$**
    - —for any string of lenght n
    - —any sequence of moves
    - —moves the head at most $T(n)$ times

- **Language L**
  - **nondeterministic time complexity $T(n)$**

Language classes with respect to time and space complexity

- **Nondeterministic TM**
  - **time complexity $T(n)$**
    - —for any string of lenght n
    - —any sequence of moves
    - —moves the head at most $T(n)$ times

- **Language L**
  - **nondeterministic time complexity $T(n)$**
    - —if and only if there is a nondeterministic TM M with time complexity $T(n)$ that accepts the language $L(M) = L$

Language classes with respect to time and space complexity

Language classes with respect to time and space complexity

- **Four main classes of languages:**

Language classes with respect to time and space complexity

- **Four main classes of languages:**

  - **DSPACE($S(n)$)**
    - **a set of languages with deterministic space complexity $S(n)$**

Language classes with respect to time and space complexity

- **Four main classes of languages:**

  - **DSPACE($S(n)$)**
    - **a set of languages with deterministic space complexity $S(n)$**

  - **NSPACE($S(n)$)**
    - **a set of languages with nondeterministic space complexity $S(n)$**

Language classes with respect to time and space complexity

- **Four main classes of languages:**
  - **DSPACE($S(n)$)**
    — **a set of languages with deterministic space complexity $S(n)$**
  - **NSPACE($S(n)$)**
    — **a set of languages with nondeterministic space complexity S(n)**
  - **DTIME($T(n)$)**
    — **a set of languages with deterministic time complexity $T(n)$**

Language classes with respect to time and space complexity

- **Four main classes of languages:**

  - **DSPACE($S(n)$)**
    - a set of languages with deterministic space complexity $S(n)$
  - **NSPACE($S(n)$)**
    - a set of languages with nondeterministic space complexity $S(n)$
  - **DTIME($T(n)$)**
    - a set of languages with deterministic time complexity $T(n)$
  - **NTIME($T(n)$)**
    - a set of languages with nondeterministic time complexity $T(n)$

Relations between language classes

Relations between language classes

- If $L \in \mathrm{DTIME}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f(n))$

Relations between language classes

- If $L \in \mathrm{DTIME}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f(n))$
    - **Deterministic TM *M* with time complexity *f*(*n*)**

Relations between language classes

- If $L \in \mathrm{DTIME}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f(n))$

  - **Deterministic TM *M* with time complexity $f(n)$**
    - —Using $f(n)$ head moves, TM M cannot pass more than $f(n)+1$ cells

Relations between language classes

- If $L \in \mathrm{DTIME}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f(n))$

  - **Deterministic TM *M* with time complexity *f*(*n*)**
    - **—Using *f*(*n*) head moves, TM *M* cannot pass more than *f*(*n*)+1 cells**

  - **Merging two cells into one**

Relations between language classes

- If $L \in \mathrm{DTIME}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f(n))$

  - **Deterministic TM *M* with time complexity *f*(*n*)**
    - Using *f*(*n*) head moves, TM *M* cannot pass more than *f*(*n*)+1 cells

  - **Merging two cells into one**
    - we can build TM that does not use more than $\lceil (f(n)+1)/2 \rceil$ cells

Relations between language classes

- **If $L \in \mathrm{DTIME}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f(n))$**
  - **Deterministic TM *M* with time complexity $f(n)$**
    - **—Using $f(n)$ head moves, TM *M* cannot pass more than $f(n)+1$ cells**
  - **Merging two cells into one**
    - **—we can build TM that does not use more than $\lceil (f(n)+1)/2] \rceil$ cells**
  - **$(\lceil (f(n)+1)/2] \rceil < f(n)) \Rightarrow L \in \mathrm{DSPACE}(f(n))$**

Relations between language classes

Relations between language classes

- $(\text{If } L \in \text{DSPACE}(f(n)) \wedge f(n) \geq \log_2 n\ ) \Rightarrow L \in \text{DTIME}(\ c^{f(n)}\ )$

Relations between language classes

- $(\text{If } L \in \text{DSPACE}(f(n)) \wedge f(n) \geq \log_2 n\ ) \Rightarrow L \in \text{DTIME}(\ c^{f(n)}\ )$
  - **Deterministic TM M_1 with space complexity $f(n)$**

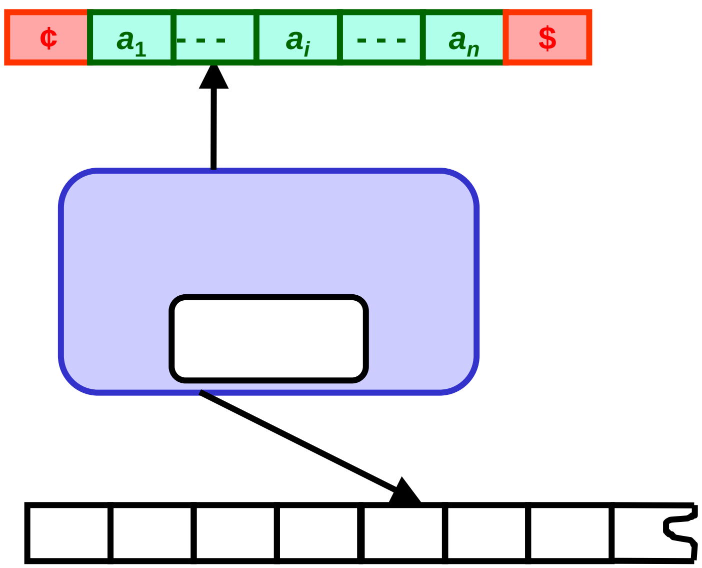
Relations between language classes

- $(\text{If } L \in \text{DSPACE}(f(n)) \wedge f(n) \geq \log_2 n\ ) \Rightarrow L \in \text{DTIME}(\ c^{f(n)}\ )$

  - **Deterministic TM M_1 with space complexity $f(n)$**
    - —**one input tape and one work tape**

Relations between language classes

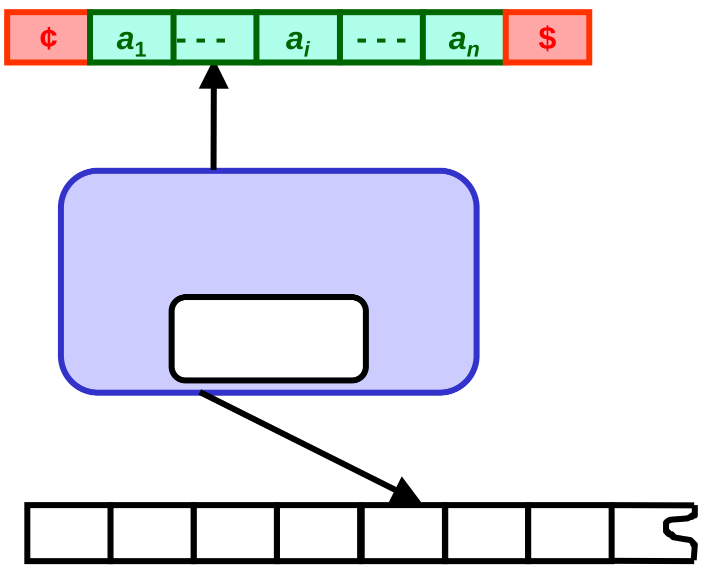
Maximal number of different configurations of TM M_1



Relations between language classes

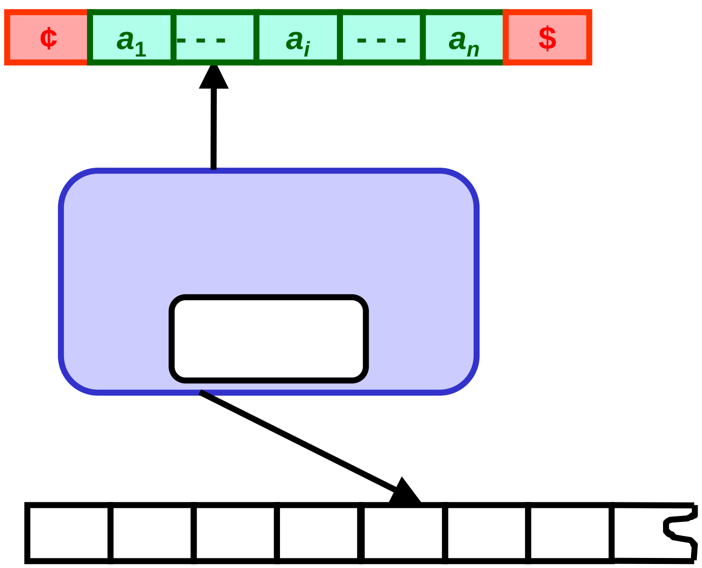
Maximal number of different configurations of TM M_1

- $s\ (n+2)\ f(n)\ t^{\ f(n)}$



Relations between language classes

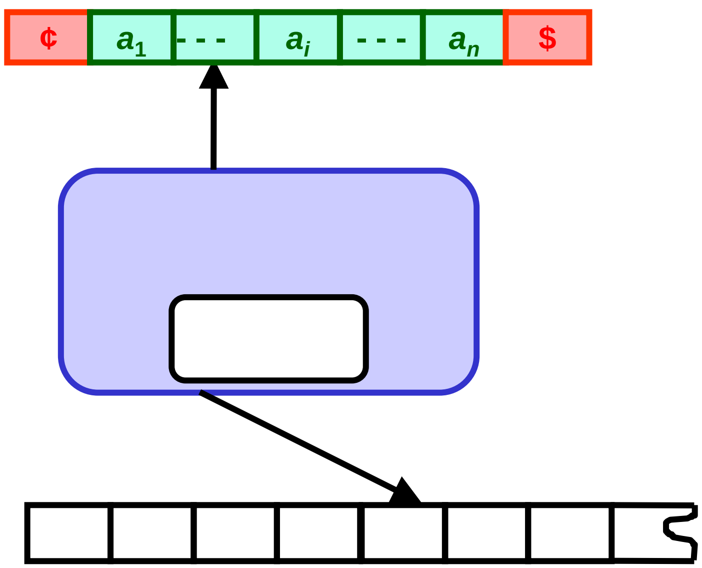
Maximal number of different configurations of TM M_1



- $s\ (n+2)\ f(n)\ t^{\ f(n)}$
  - s - number of states in Q

Relations between language classes

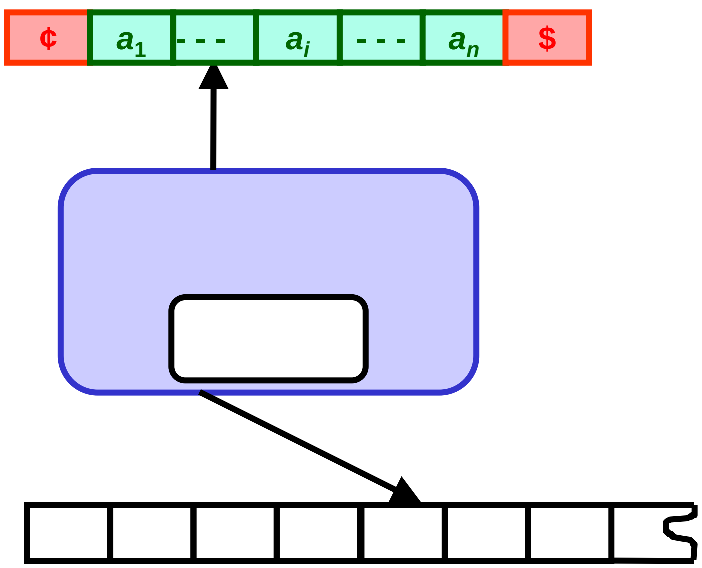
Maximal number of different configurations of TM M_1



- $s\ (n+2)\ f(n)\ t^{\ f(n)}$
  - — s - number of states in Q
  - — $n+2$ - number of different head positions on the input tape

Relations between language classes

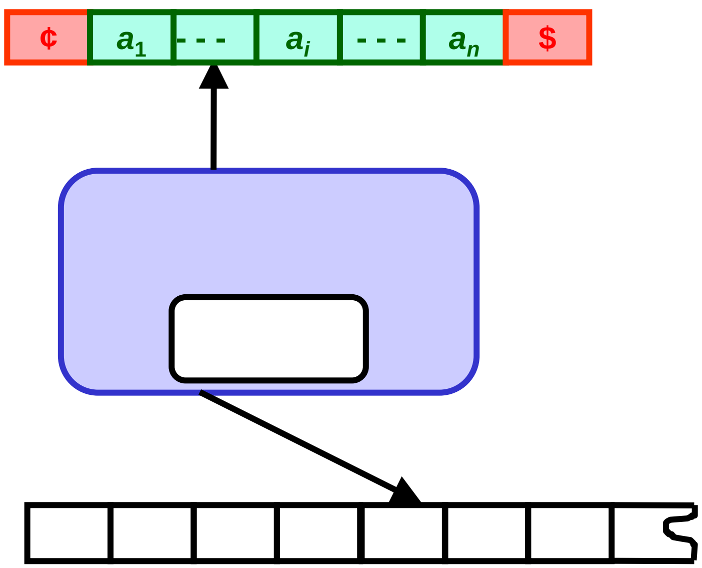
Maximal number of different configurations of TM M_1



- $s\ (n+2)\ f(n)\ t^{\,f(n)}$
  - s - number of states in Q
  - $n+2$ - number of different head positions on the input tape
    - n - length of string w

Relations between language classes

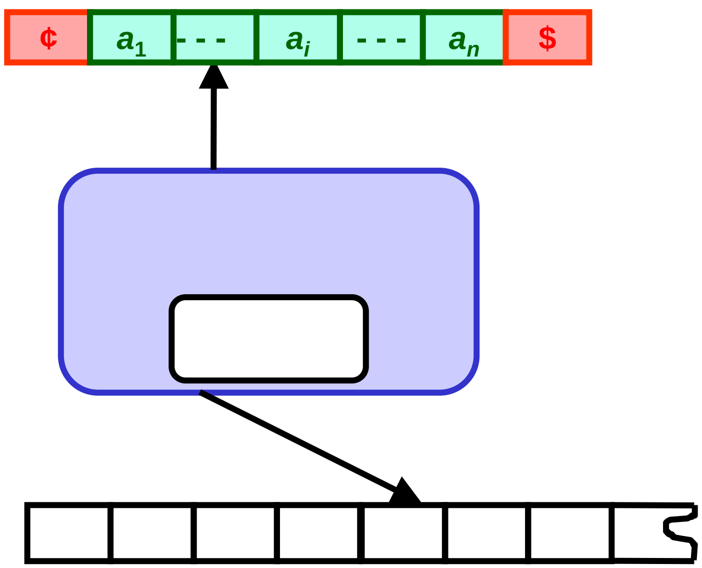
Maximal number of different configurations of TM M_1



- $s\ (n+2)\ f(n)\ t^{\ f(n)}$
  - — s - number of states in Q
  - — $n+2$ - number of different head positions on the input tape
    - - n - length of string w
  - — $f(n)$ - number of different head positions on the work tape

Relations between language classes

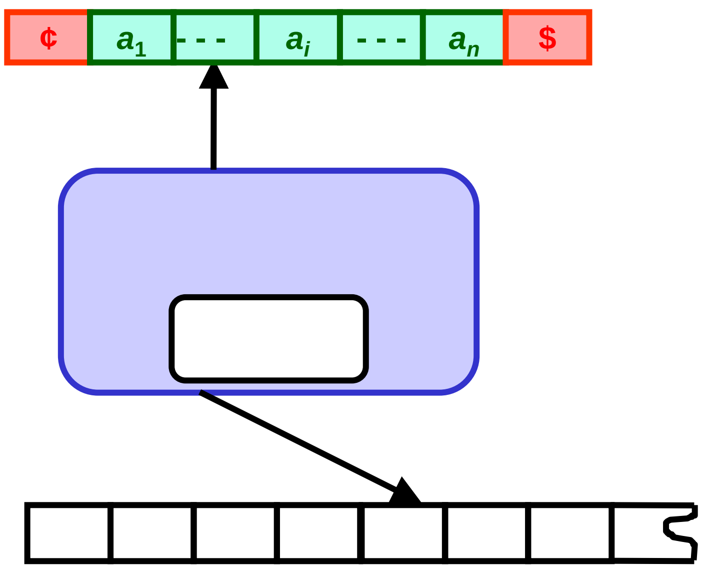
Maximal number of different configurations of TM M_1



- $s\ (n+2)\ f(n)\ t^{\ f(n)}$
  - s - number of states in Q
  - $n+2$ - number of different head positions on the input tape
    - n - length of string w
  - $f(n)$ - number of different head positions on the work tape
  - $t^{\ f(n)}$ - number of different work tape contents

Relations between language classes

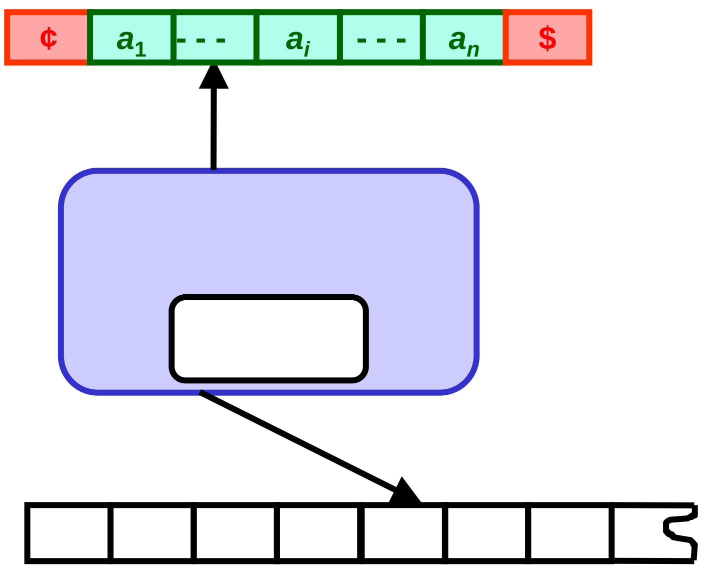
Maximal number of different configurations of TM M_1



- $s\,(n+2)\,f(n)\,t^{f(n)}$
  - s - number of states in Q
  - $n+2$ - number of different head positions on the input tape
    - n - length of string w
  - $f(n)$ - number of different head positions on the work tape
  - $t^{f(n)}$ - number of different work tape contents
    - t - cardinal number of the symbol set Γ

Relations between language classes

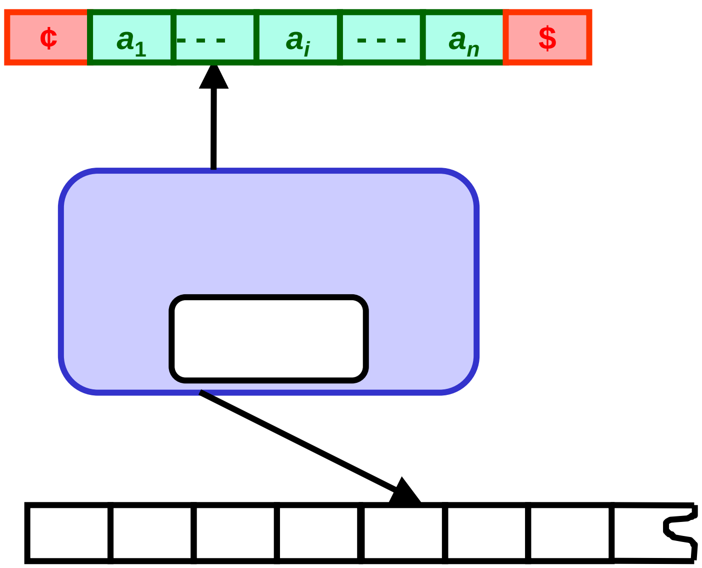
Maximal number of different configurations of TM M_1



- $s\ (n+2)\ f(n)\ t^{f(n)}$
  - s - number of states in Q
  - $n+2$ - number of different head positions on the input tape
    - n - length of string w
  - $f(n)$ - number of different head positions on the work tape
  - $t^{f(n)}$ - number of different work tape contents
    - t - cardinal number of the symbol set Γ
    - $f(n)$ – maximal number of cells used by TM M_1

Relations between language classes

Maximal number of different configurations of TM M_1



- $s\,(n+2)\,f(n)\,t^{f(n)}$
  - s - number of states in Q
  - $n+2$ - number of different head positions on the input tape
    - n - length of string w
  - $f(n)$ - number of different head positions on the work tape
  - $t^{f(n)}$ - number of different work tape contents
    - t - cardinal number of the symbol set Γ
    - $f(n)$ – maximal number of cells used by TM M_1

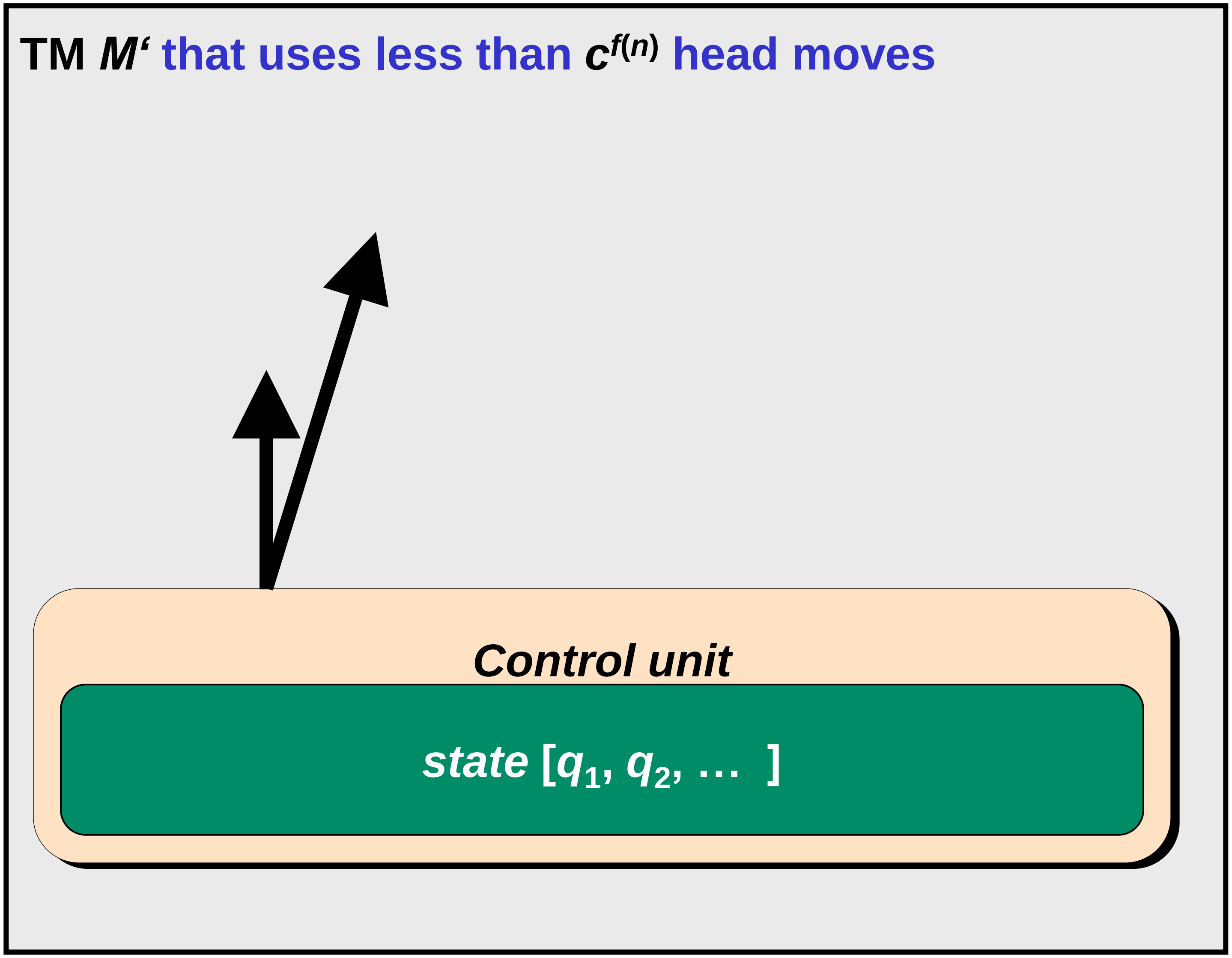
$$f(n)\geq\log_2 n \Rightarrow (c^{f(n)} \geq s(n+2)f(n)t^{f(n)}) \Rightarrow L\in \mathrm{DTIME}(c^{f(n)})$$

Relations between language classes

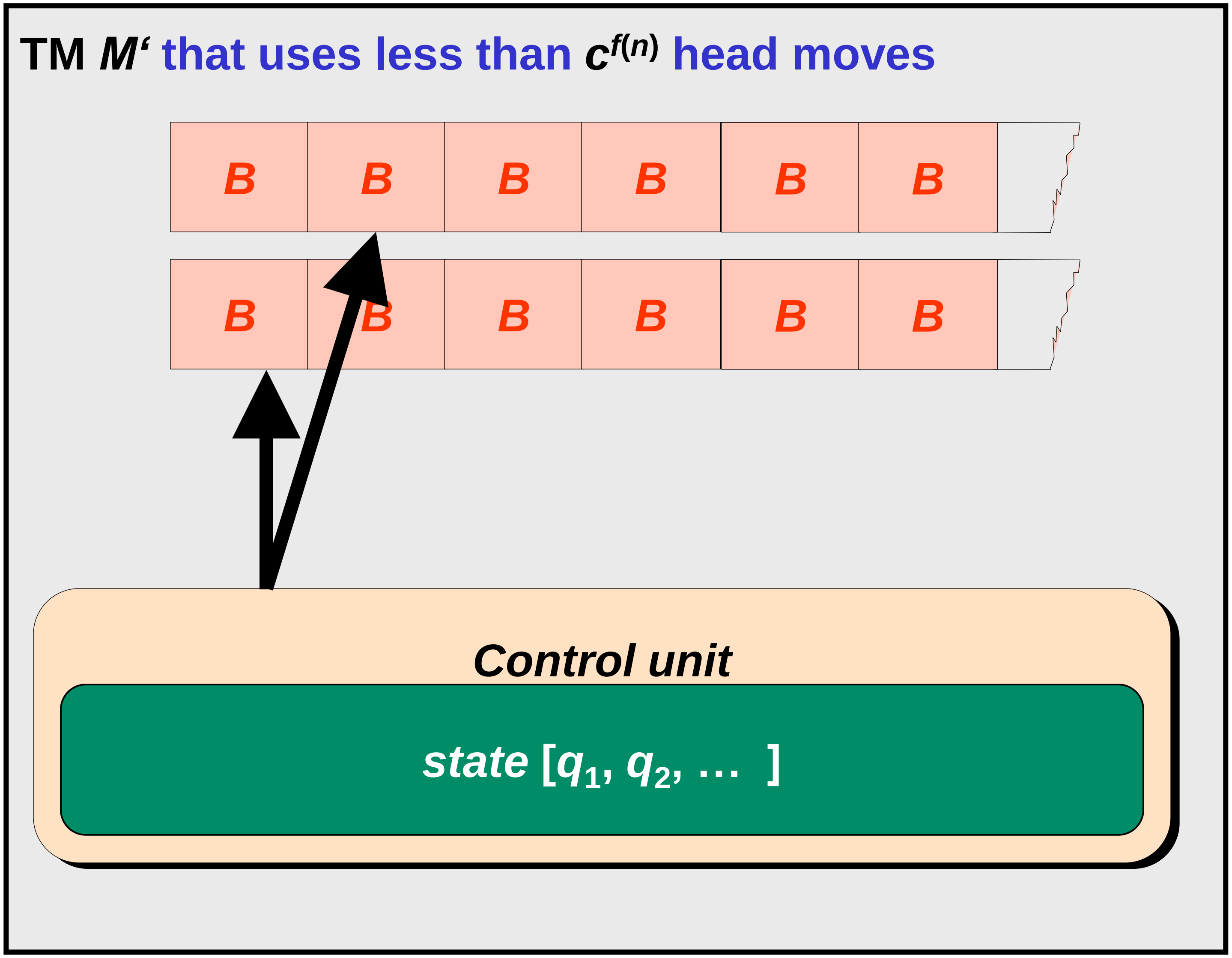
Relations between language classes

TM M' that uses less than $c^{f(n)}$ head moves

Relations between language classes

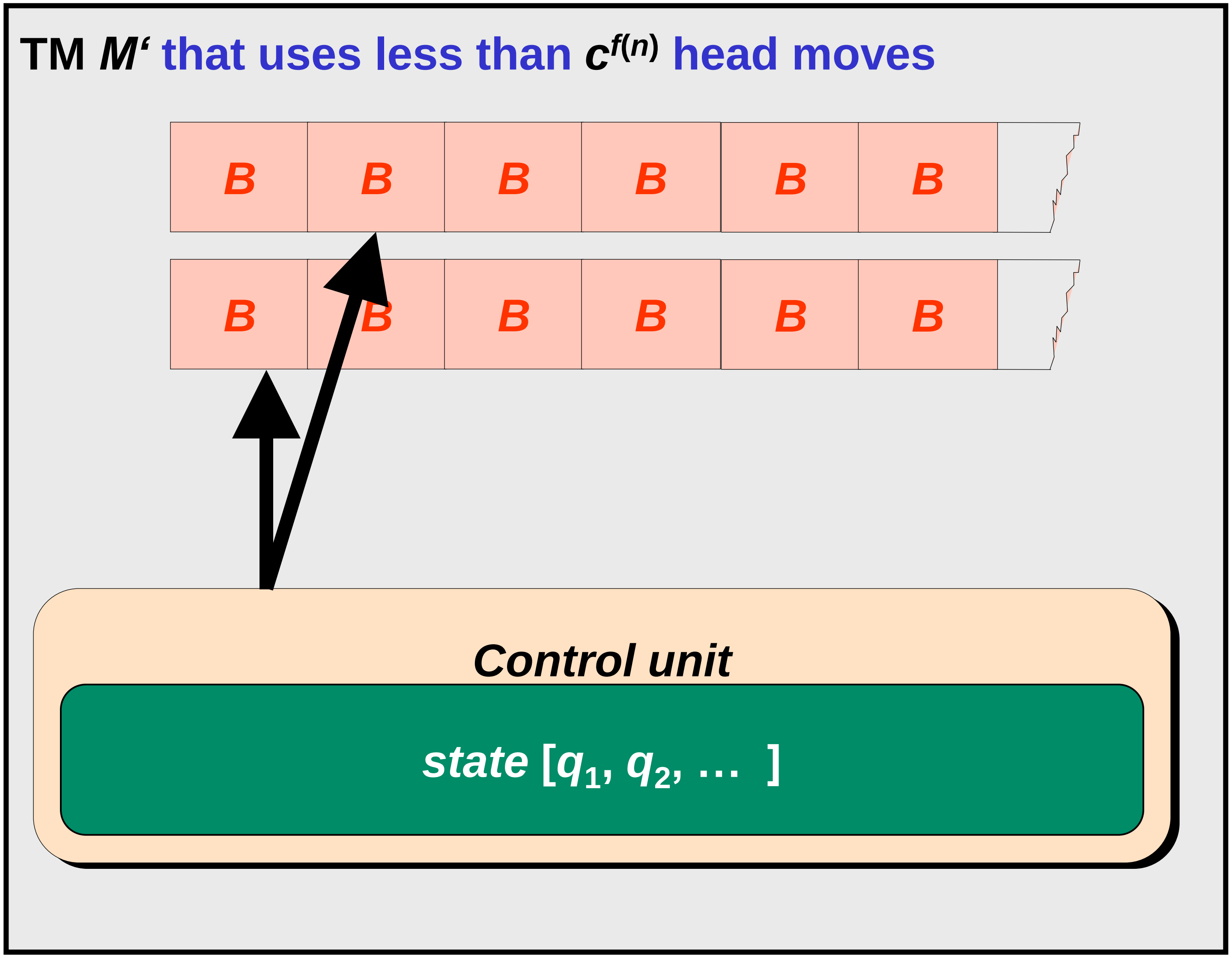


Relations between language classes



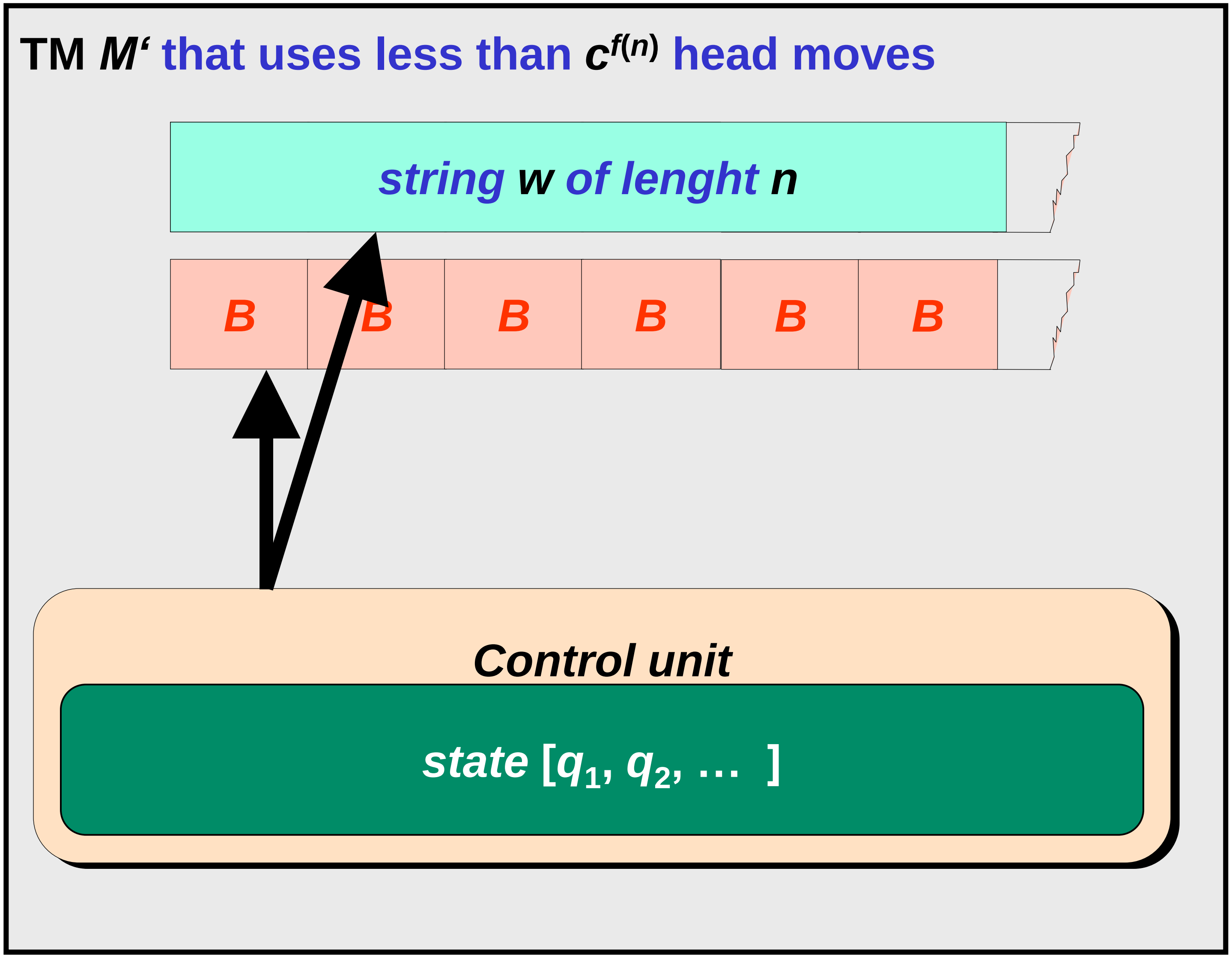
Relations between language classes

$c^{f(n)}$



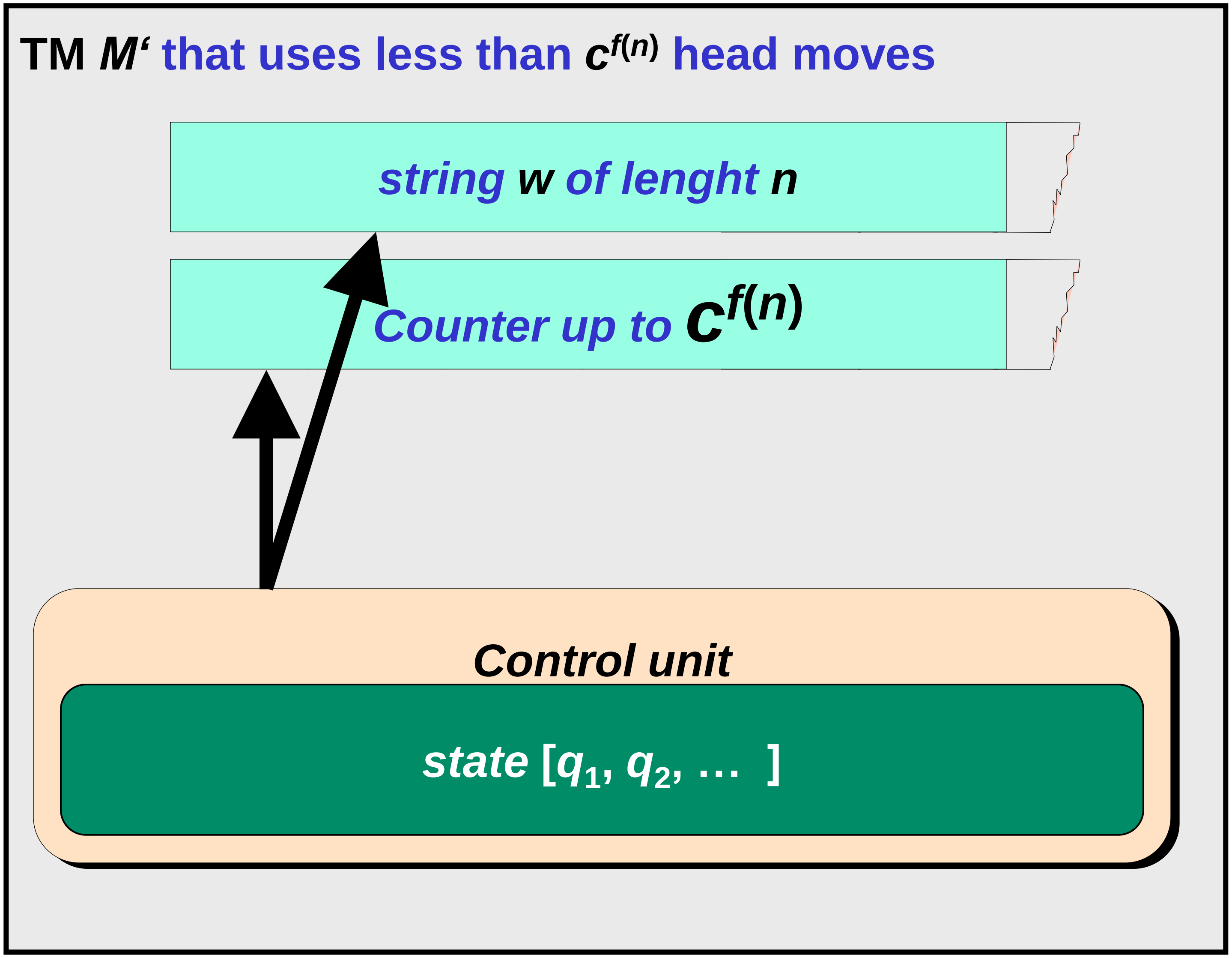
Relations between language classes

$c^{f(n)}$



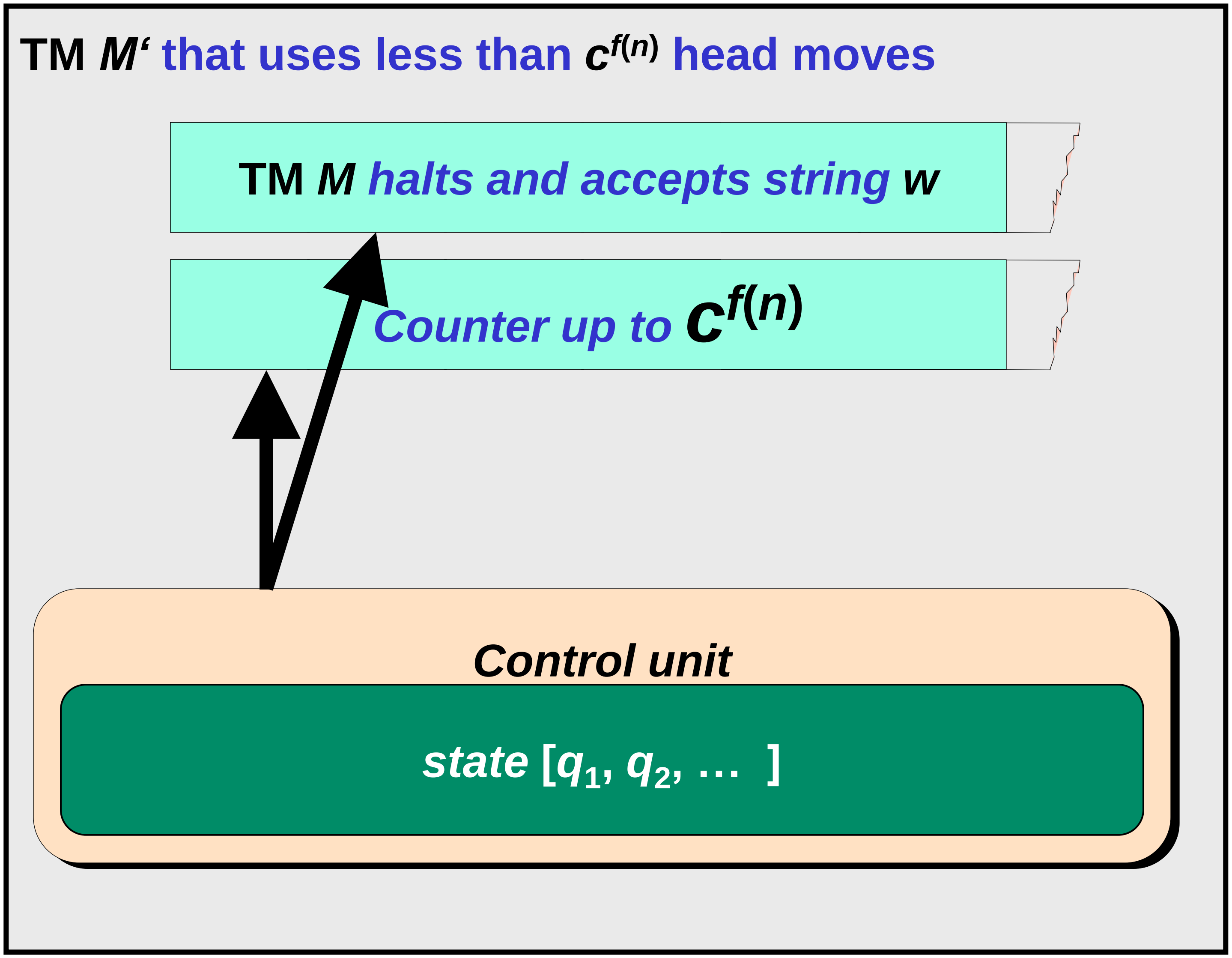
Relations between language classes

$$c^{f(n)}$$



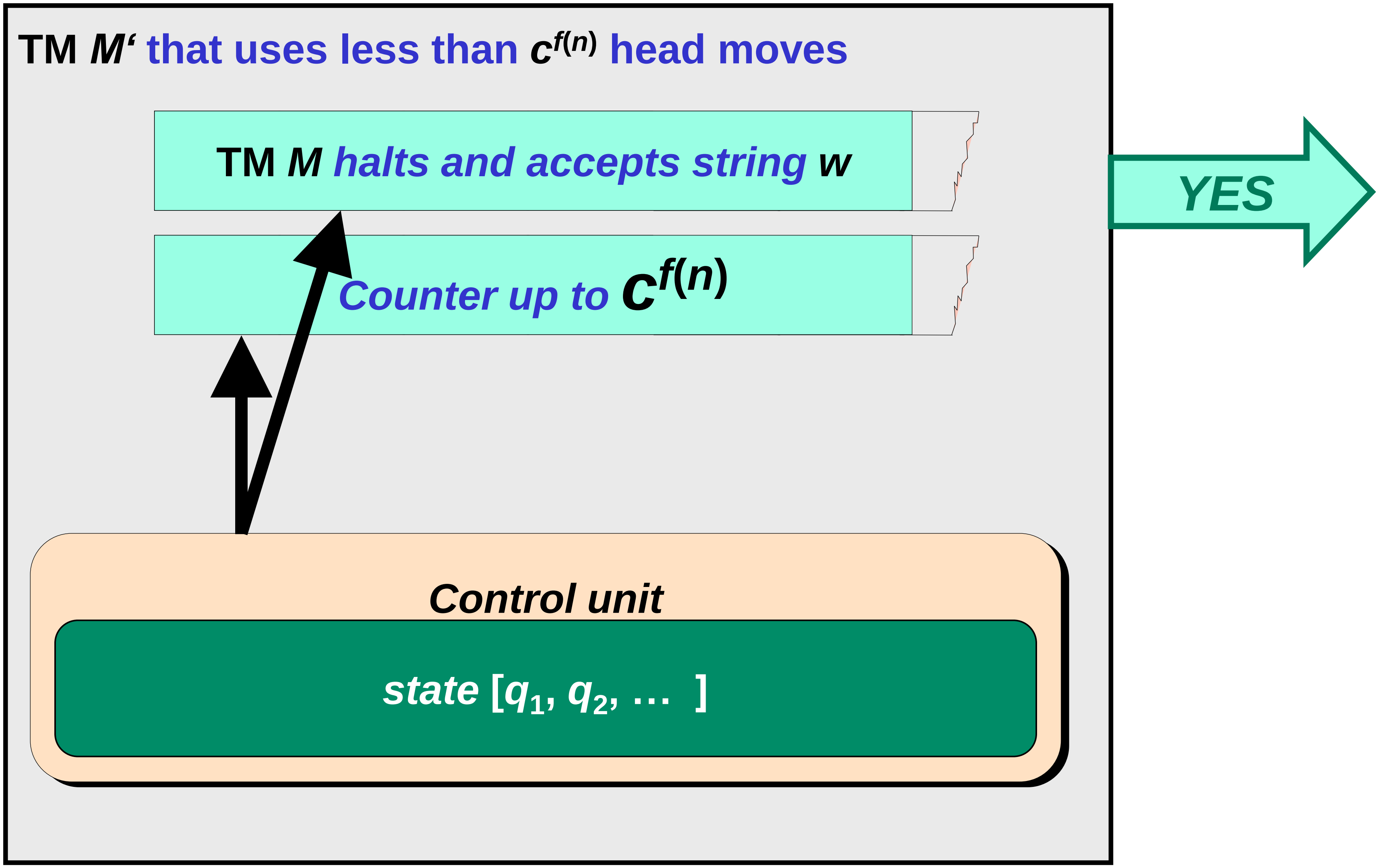
Relations between language classes

$$c^{f(n)}$$



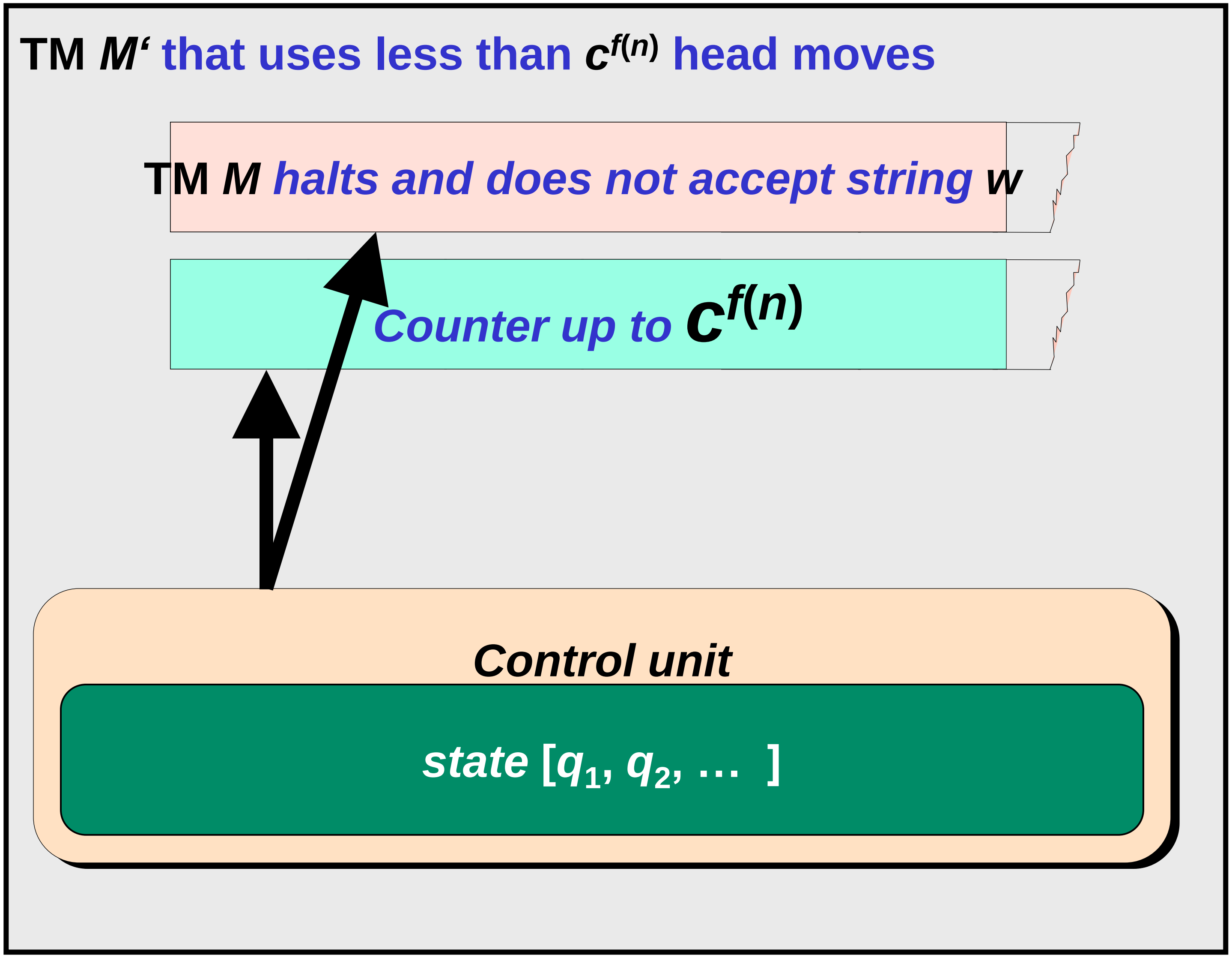
Relations between language classes

$c^{f(n)}$



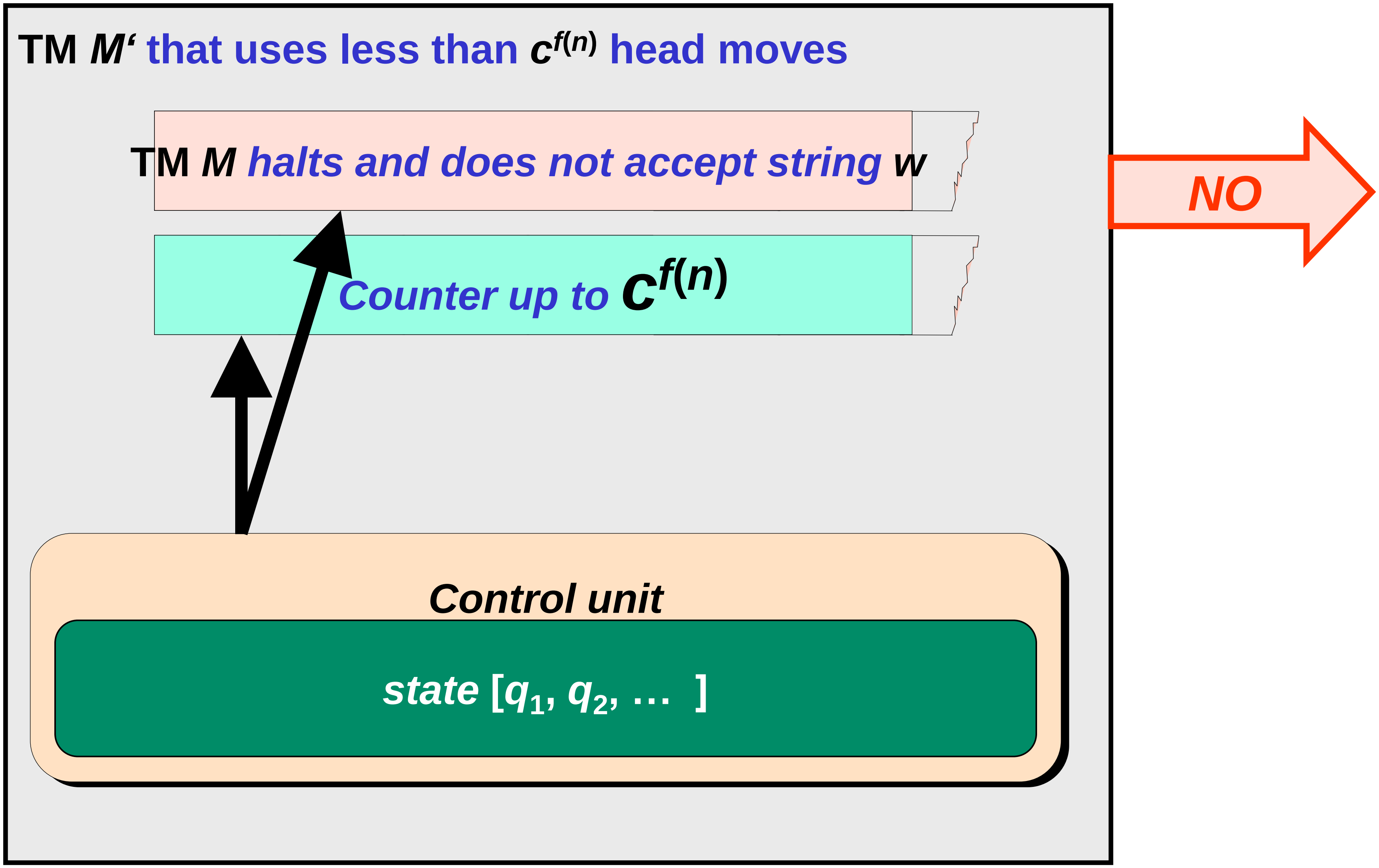
Relations between language classes

$c^{f(n)}$



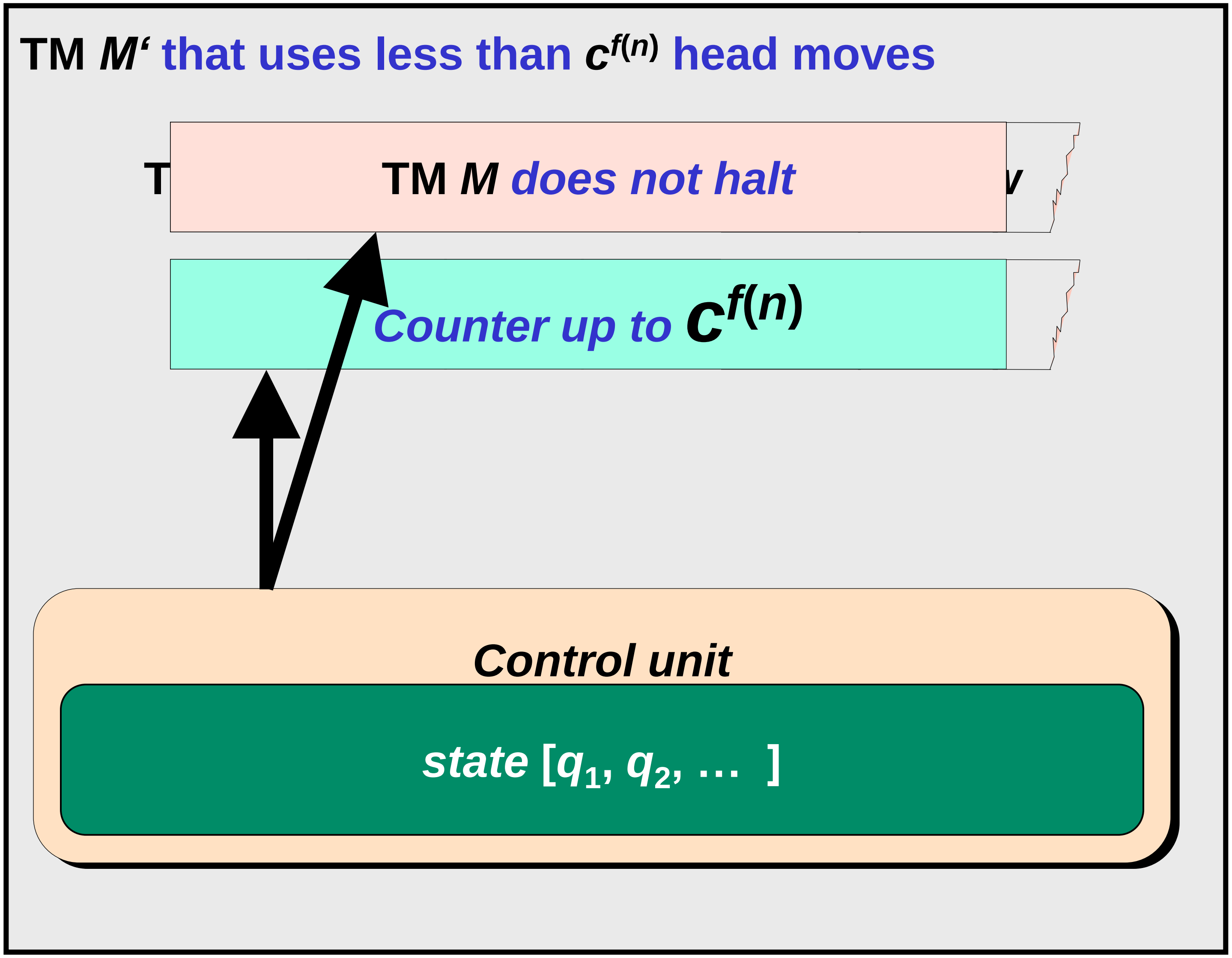
Relations between language classes

$$c^{f(n)}$$



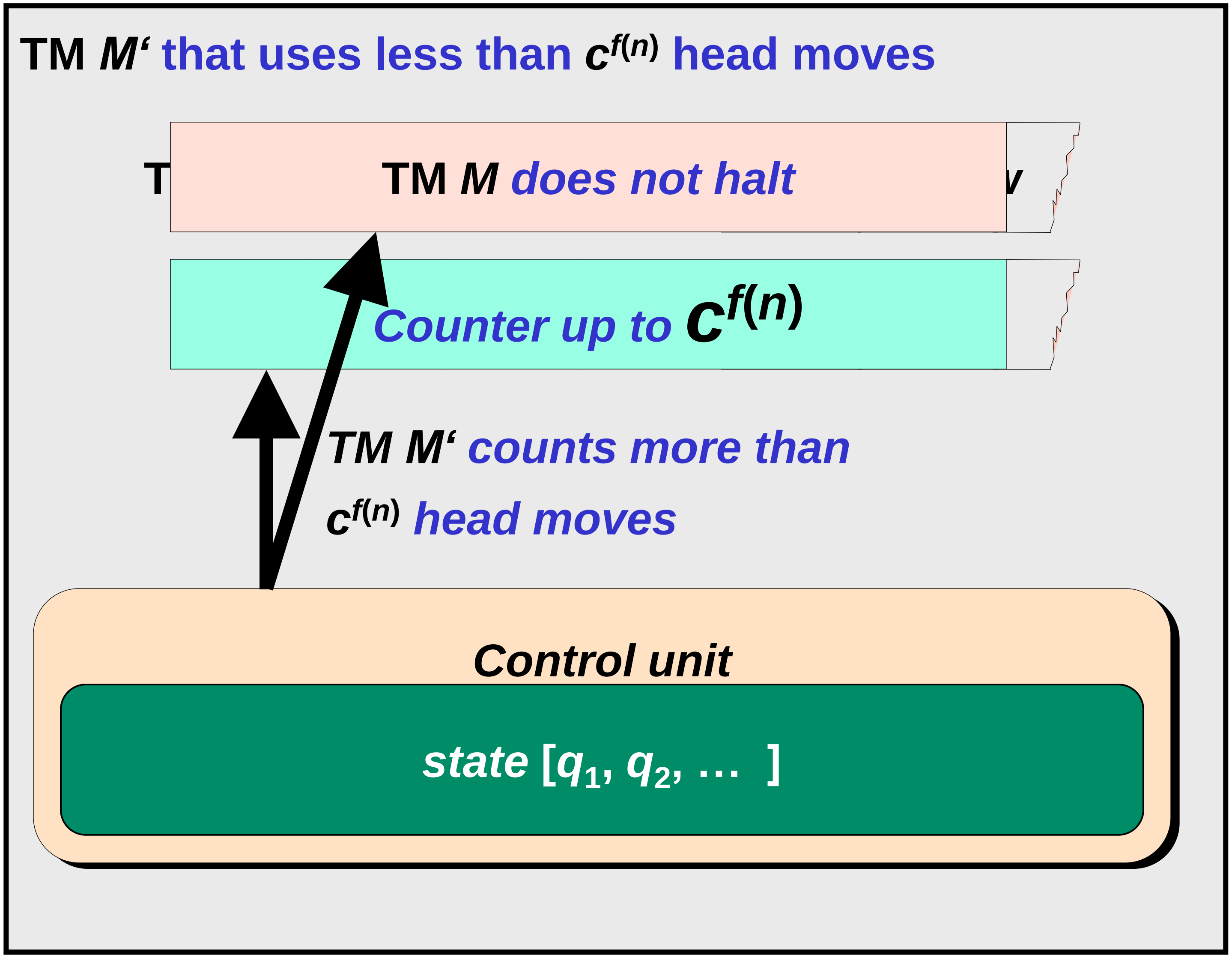
Relations between language classes

$c^{f(n)}$



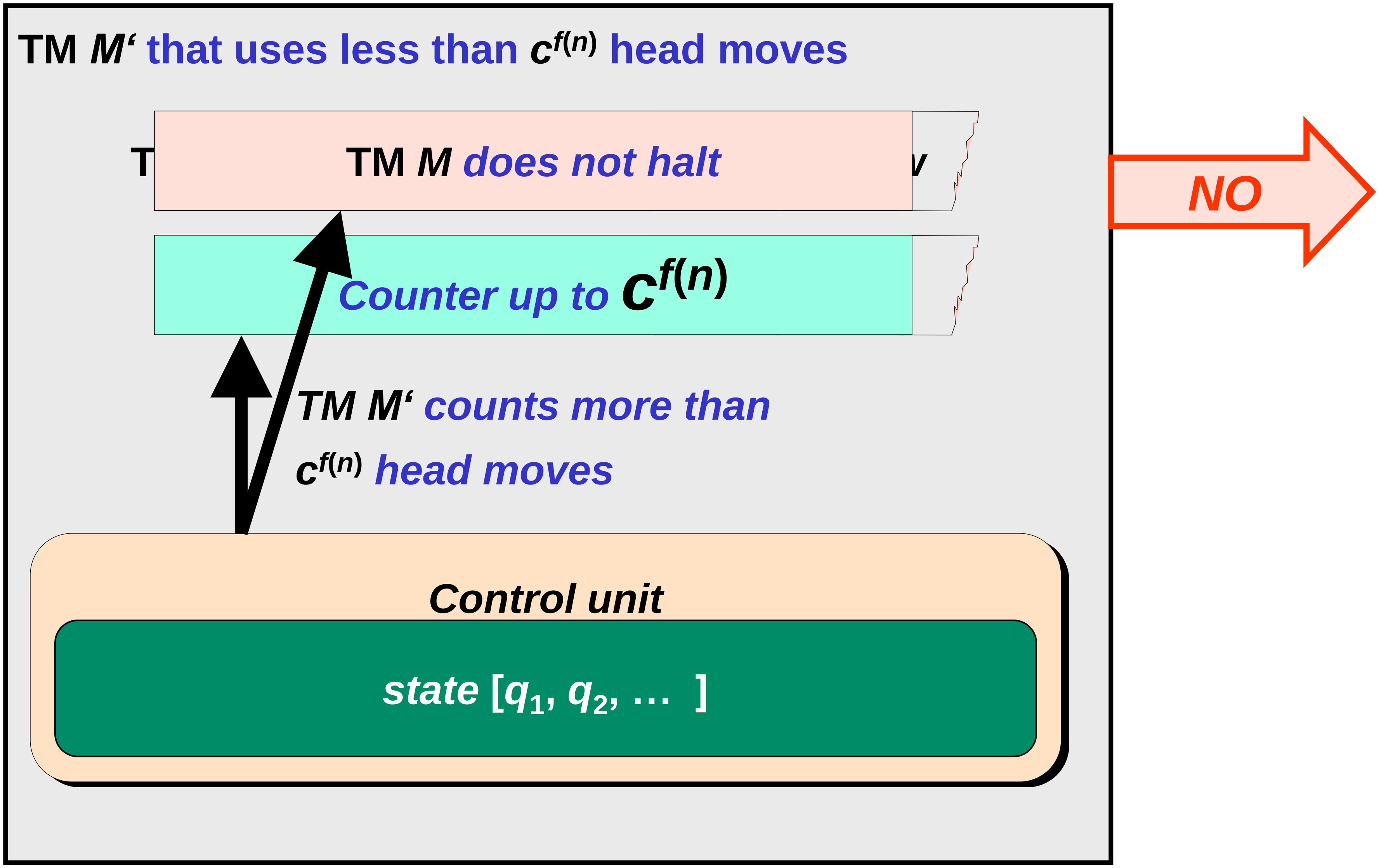
Relations between language classes

$$c^{f(n)}$$



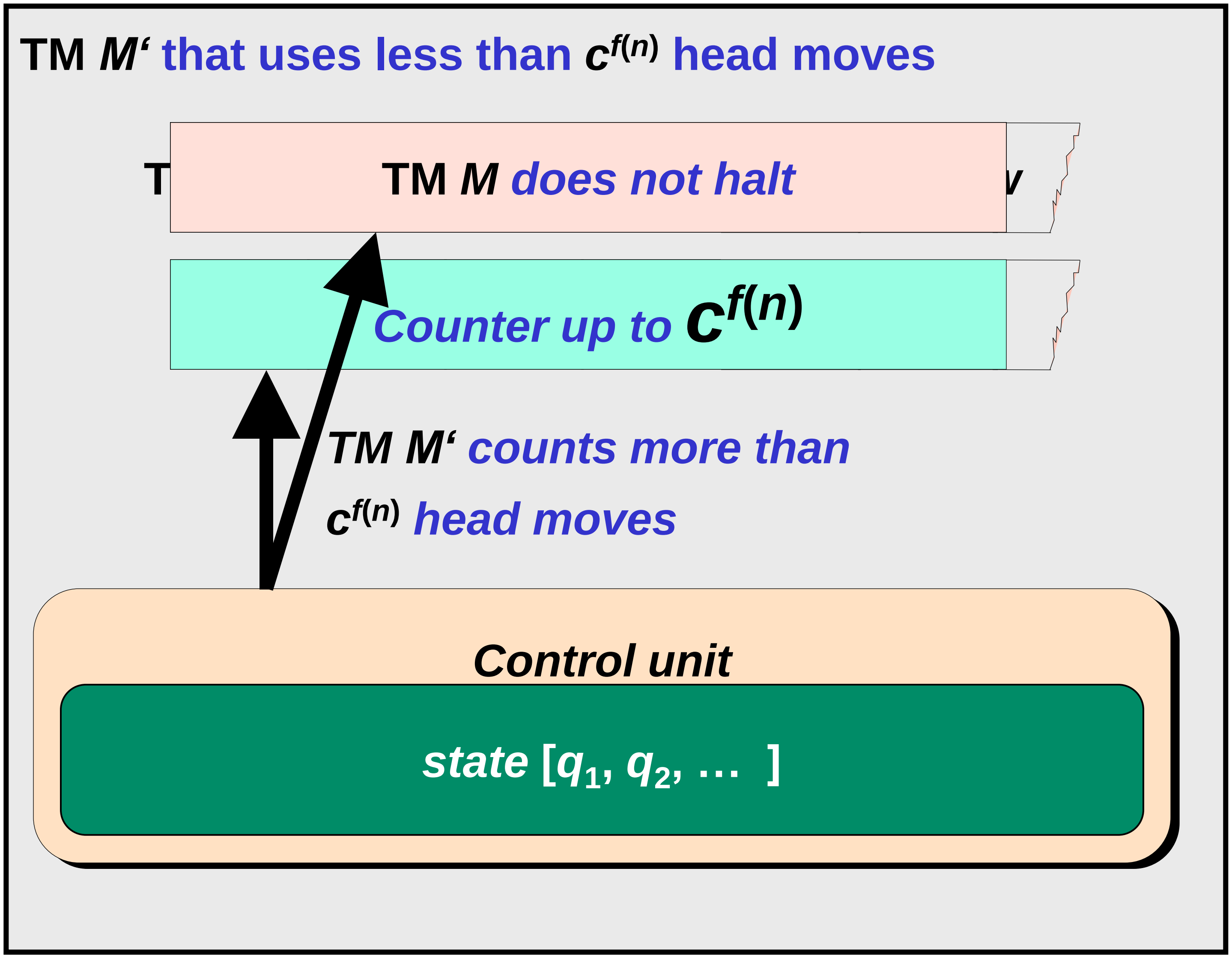
Relations between language classes

$c^{f(n)}$



Relations between language classes

$$c^{f(n)}$$



Relations between language classes

Relations between language classes

- $(\text{If } L \in \text{NTIME}(f(n)) \wedge f(n) \geq \log_2 n) \Rightarrow L \in \text{DTIME}(c^{f(n)})$

Relations between language classes

- $(\text{If } L \in \text{NTIME}(f(n)) \wedge f(n) \geq \log_2 n\ ) \Rightarrow L \in \text{DTIME}(\ c^{f(n)}\ )$
  - **Nondeterministic TM M_1 with time complexity $f(n)$**

Relations between language classes

- $(\text{If } L \in \text{NTIME}(f(n)) \wedge f(n) \geq \log_2 n) \Rightarrow L \in \text{DTIME}(c^{f(n)})$

  - **Nondeterministic TM M_1 with time complexity $f(n)$**
    —***k* work tapes**

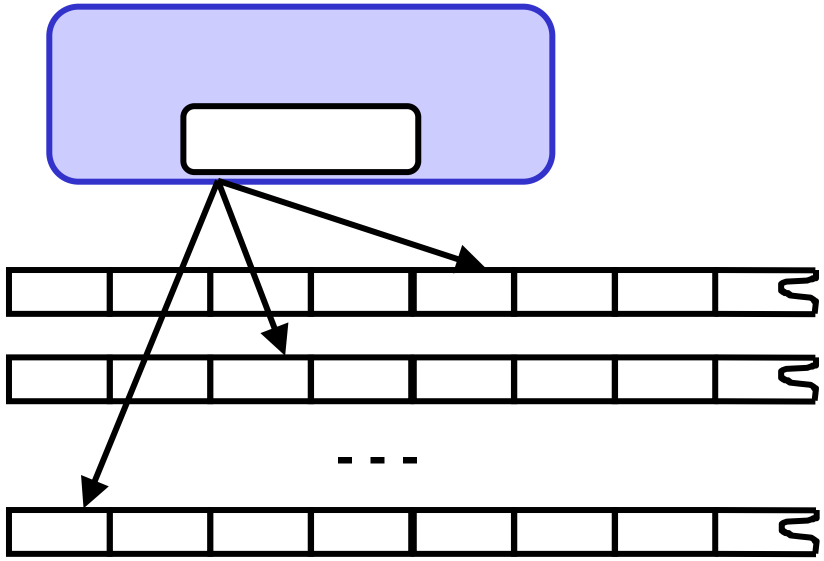
Relations between language classes

- (If $L \in \mathrm{NTIME}(f(n)) \wedge f(n) \geq \log_2 n$) $\Rightarrow L \in \mathrm{DTIME}(c^{f(n)})$

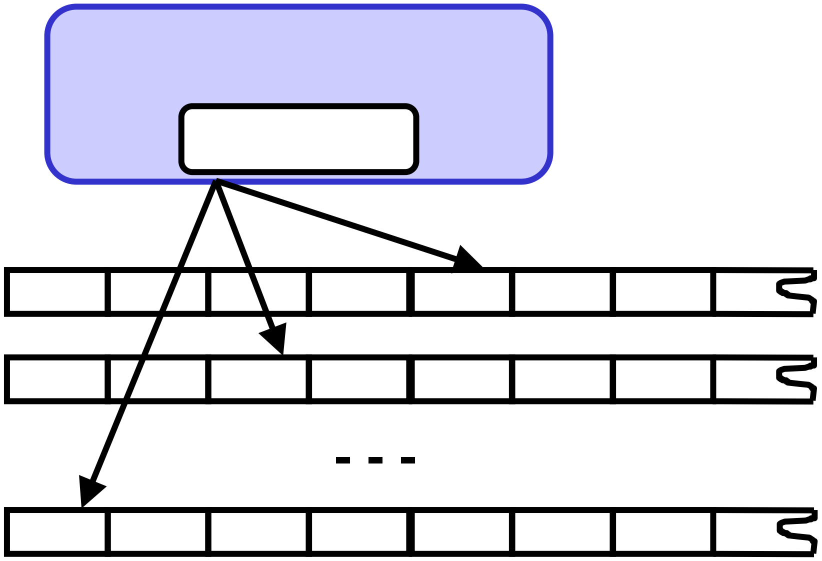
  - **Nondeterministic TM M_1 with time complexity $f(n)$**
    - —***k*** **work tapes**
    - —**by applying $f(n)$ moves, the head cannot pass more than $f(n)$+1 cells**

Relations between language classes

Maximal number of different configurations of TM M_1



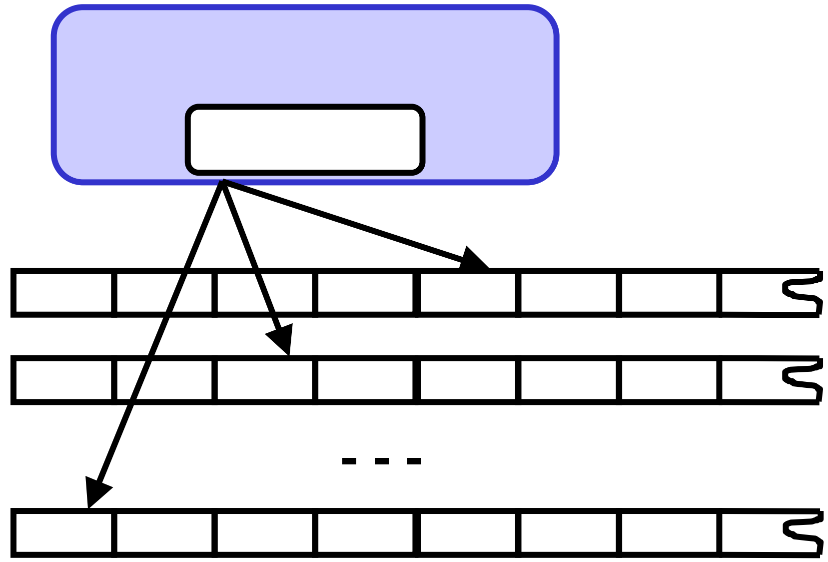
Relations between language classes



Maximal number of different configurations of TM M_1

- $s\ (f(n)+1)^k\ \ t^{\,k\,f(n)}$

Relations between language classes

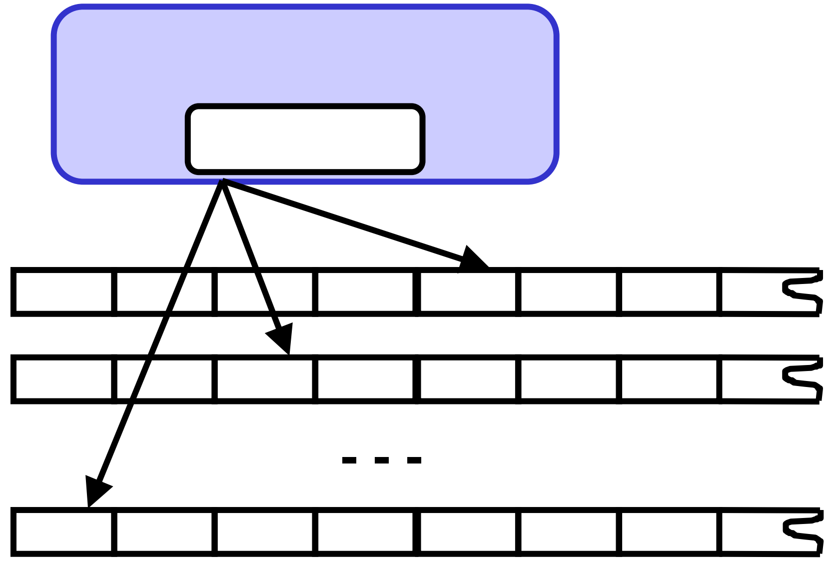


Maximal number of different configurations of TM M_1

- $s\,(f(n)+1)^k \;\; t^{\,k\,f(n)}$
  - s - number of states in Q

Relations between language classes

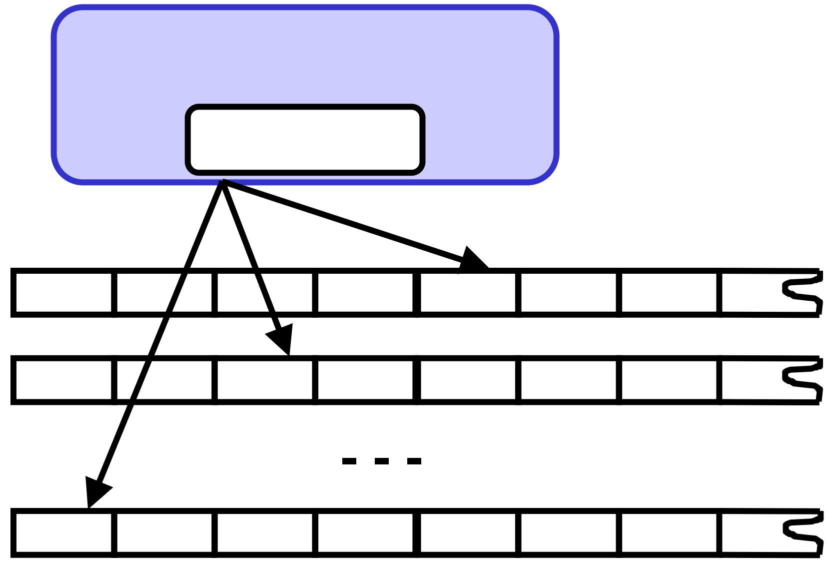
Maximal number of different configurations of TM M_1



- $s\,(f(n)+1)^k\ \ t^{\,k\,f(n)}$
  - s - number of states in Q
  - $(f(n)+1)^k$ - number of head positions on k work tapes

Relations between language classes

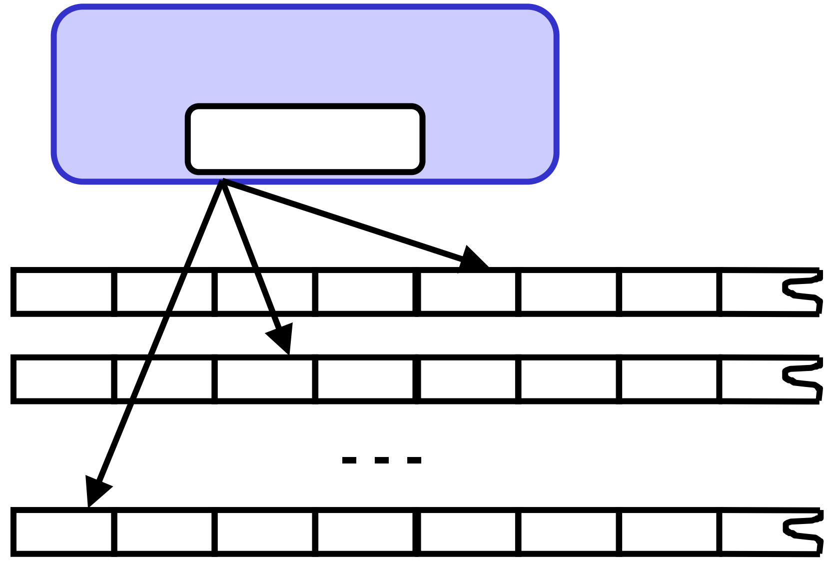
Maximal number of different configurations of TM M_1



- $s\,(f(n)+1)^k\ \ t^{\,k\,f(n)}$
  - s - number of states in Q
  - $(f(n)+1)^k$ - number of head positions on k work tapes
    - $f(n)+1$ - number of head positions on one work tape

Relations between language classes

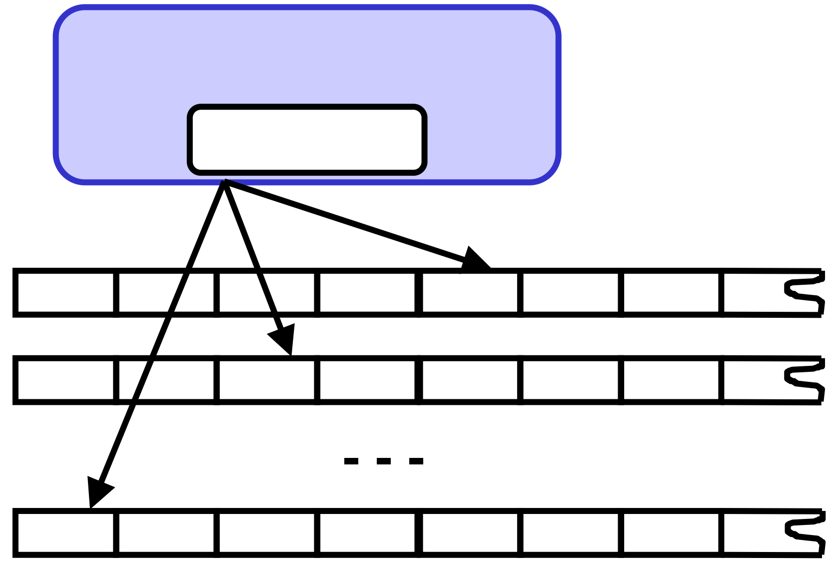
Maximal number of different configurations of TM M_1



- $s\,(f(n)+1)^k\ \ t^{\,k\,f(n)}$
  - — s - number of states in Q
  - — $(f(n)+1)^k$ - number of head positions on k work tapes
    - - $f(n)+1$ - number of head positions on one work tape
    - - k – number of work tapes

Relations between language classes

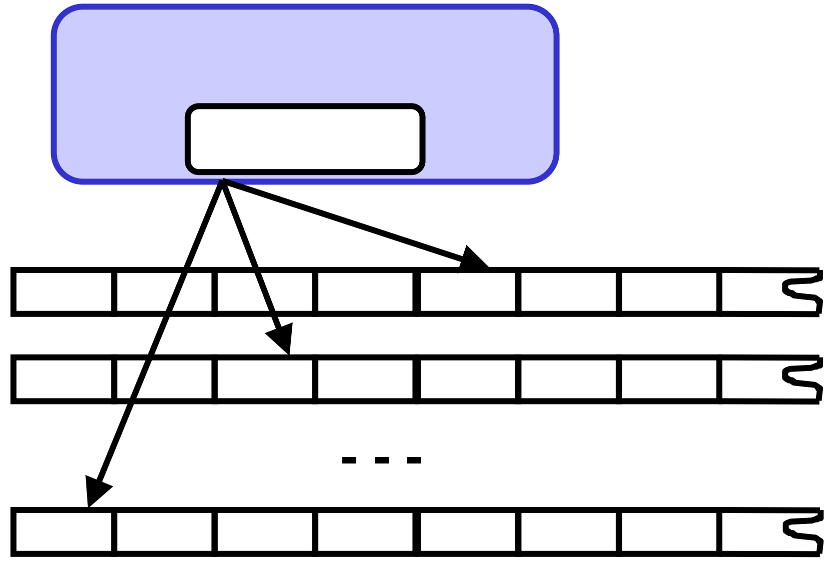
Maximal number of different configurations of TM M_1



- $s\,(f(n)+1)^k\ \ t^{k\,f(n)}$
  - s - number of states in Q
  - $(f(n)+1)^k$ - number of head positions on k work tapes
    - $f(n)+1$ - number of head positions on one work tape
    - k – number of work tapes
    - n - length of string w

Relations between language classes

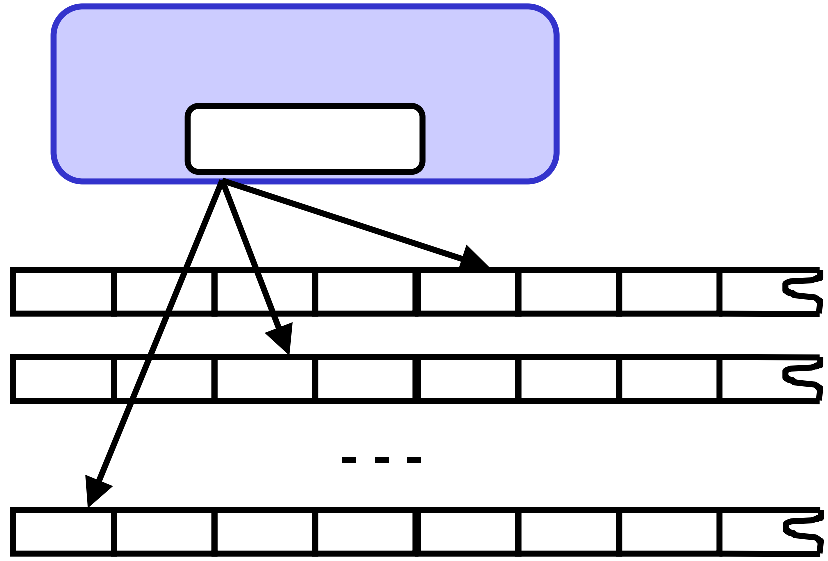
Maximal number of different configurations of TM M_1



- $s\,(f(n)+1)^k\ \ t^{\,k\,f(n)}$
  - s - number of states in Q
  - $(f(n)+1)^k$ - number of head positions on k work tapes
    - $f(n)+1$ - number of head positions on one work tape
    - k – number of work tapes
    - n - length of string w
  - $t^{\,k\,f(n)}$ - number of different contents of k work tapes

Relations between language classes

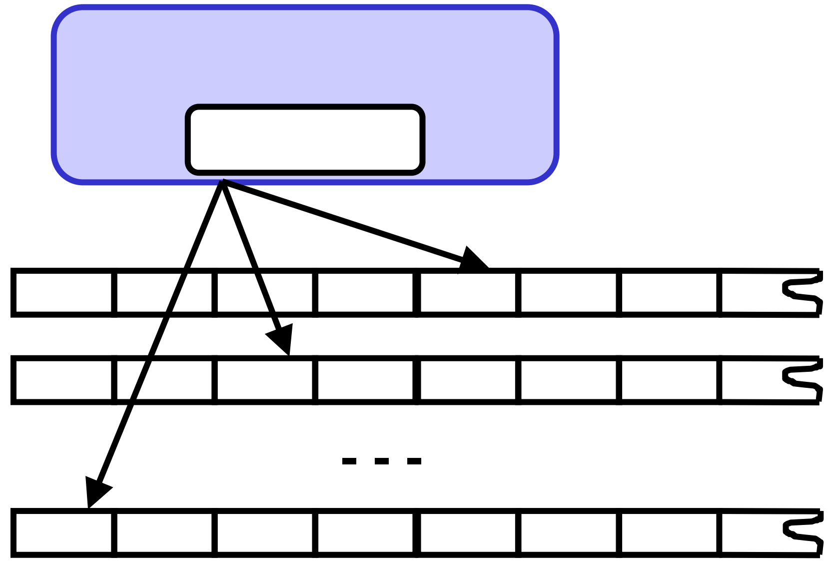
Maximal number of different configurations of TM M_1



- $s\,(f(n)+1)^k\ \ t^{\,k\,f(n)}$
  - s - number of states in Q
  - $(f(n)+1)^k$ - number of head positions on k work tapes
    - $f(n)+1$ - number of head positions on one work tape
    - k – number of work tapes
    - n - length of string w
  - $t^{\,k\,f(n)}$ - number of different contents of k work tapes
    - $t^{\,f(n)}$ - number of different contents of one work tape

Relations between language classes

Maximal number of different configurations of TM M_1



- $s\,(f(n)+1)^k\;\; t^{\,k\,f(n)}$
  - s - number of states in Q
  - $(f(n)+1)^k$ - number of head positions on k work tapes
    - $f(n)+1$ - number of head positions on one work tape
    - k – number of work tapes
    - n - length of string w
  - $t^{\,k\,f(n)}$ - number of different contents of k work tapes
    - $t^{\,f(n)}$ - number of different contents of one work tape
    - t – number of tape symbols in Γ

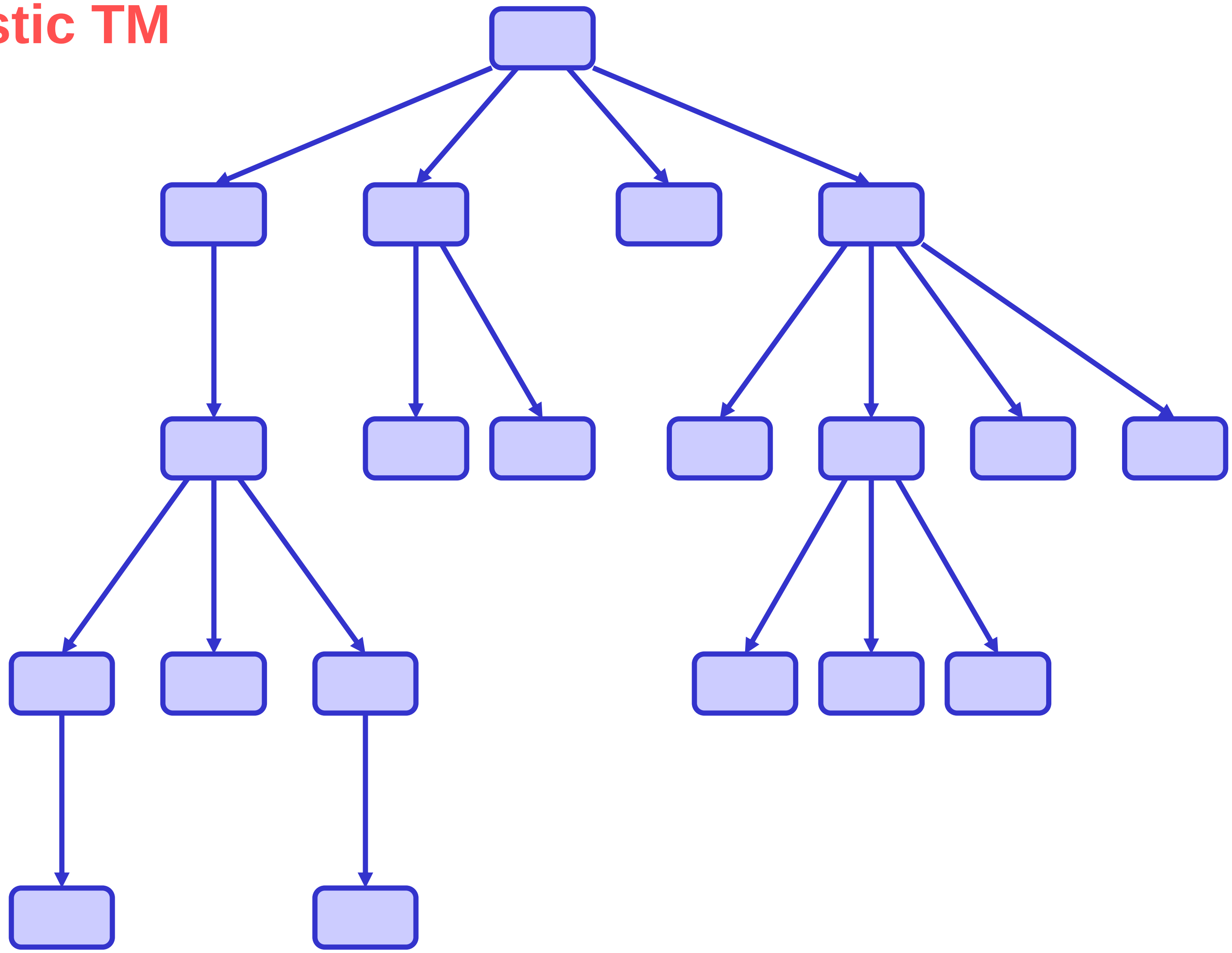
Relations between language classes

Relations between language classes

Nondeterministic TM

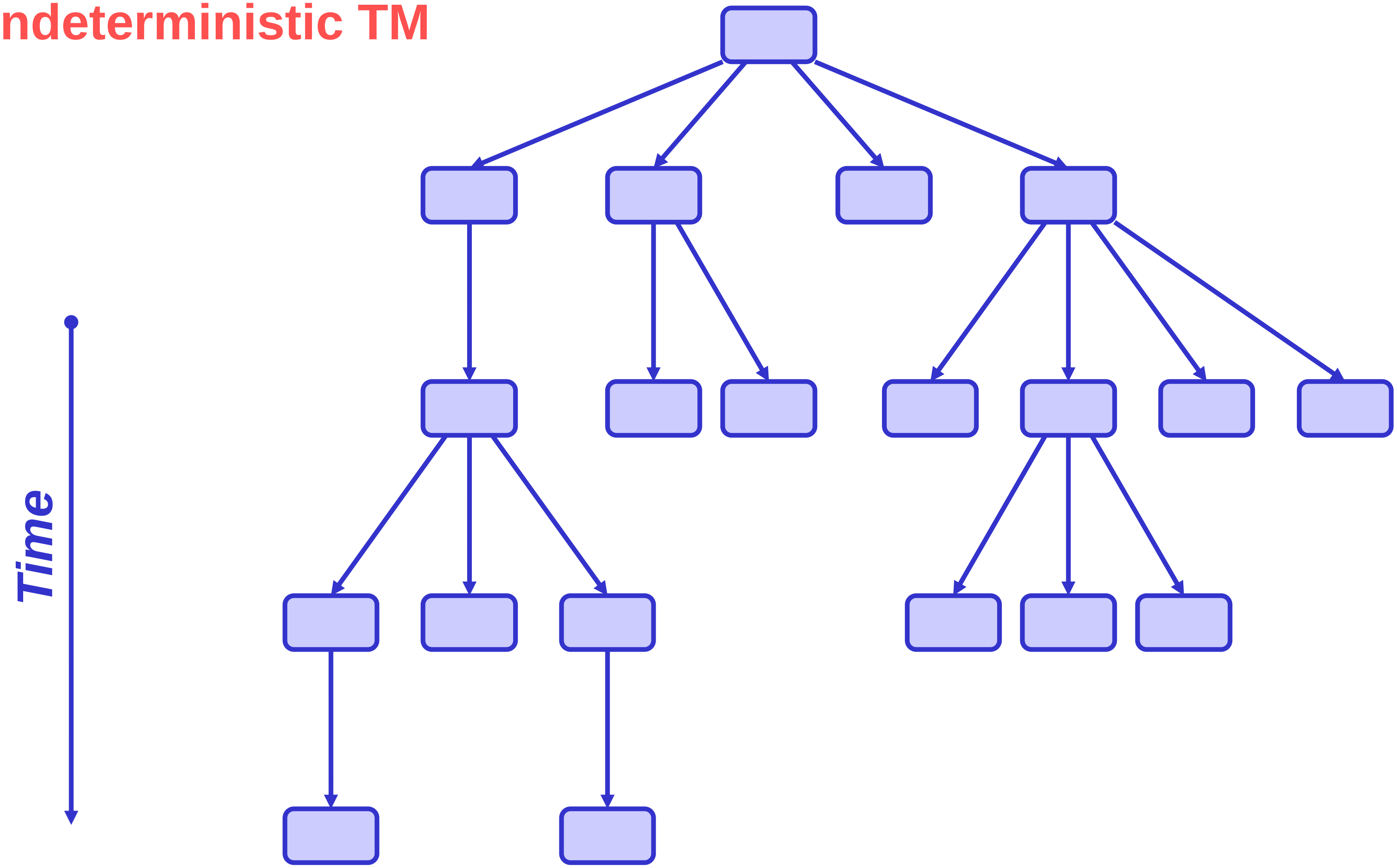
Relations between language classes

Nondeterministic TM



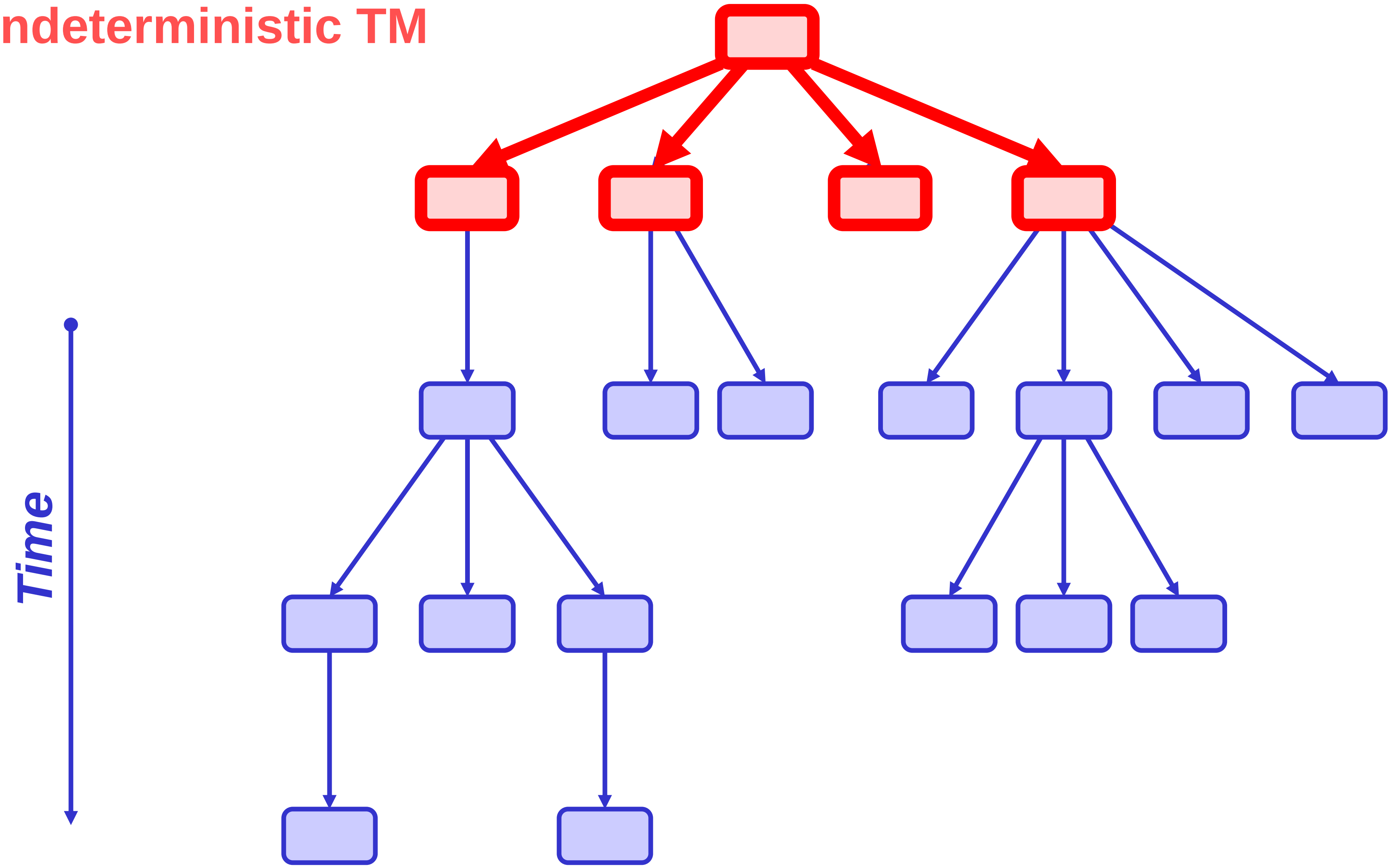
Relations between language classes

Nondeterministic TM



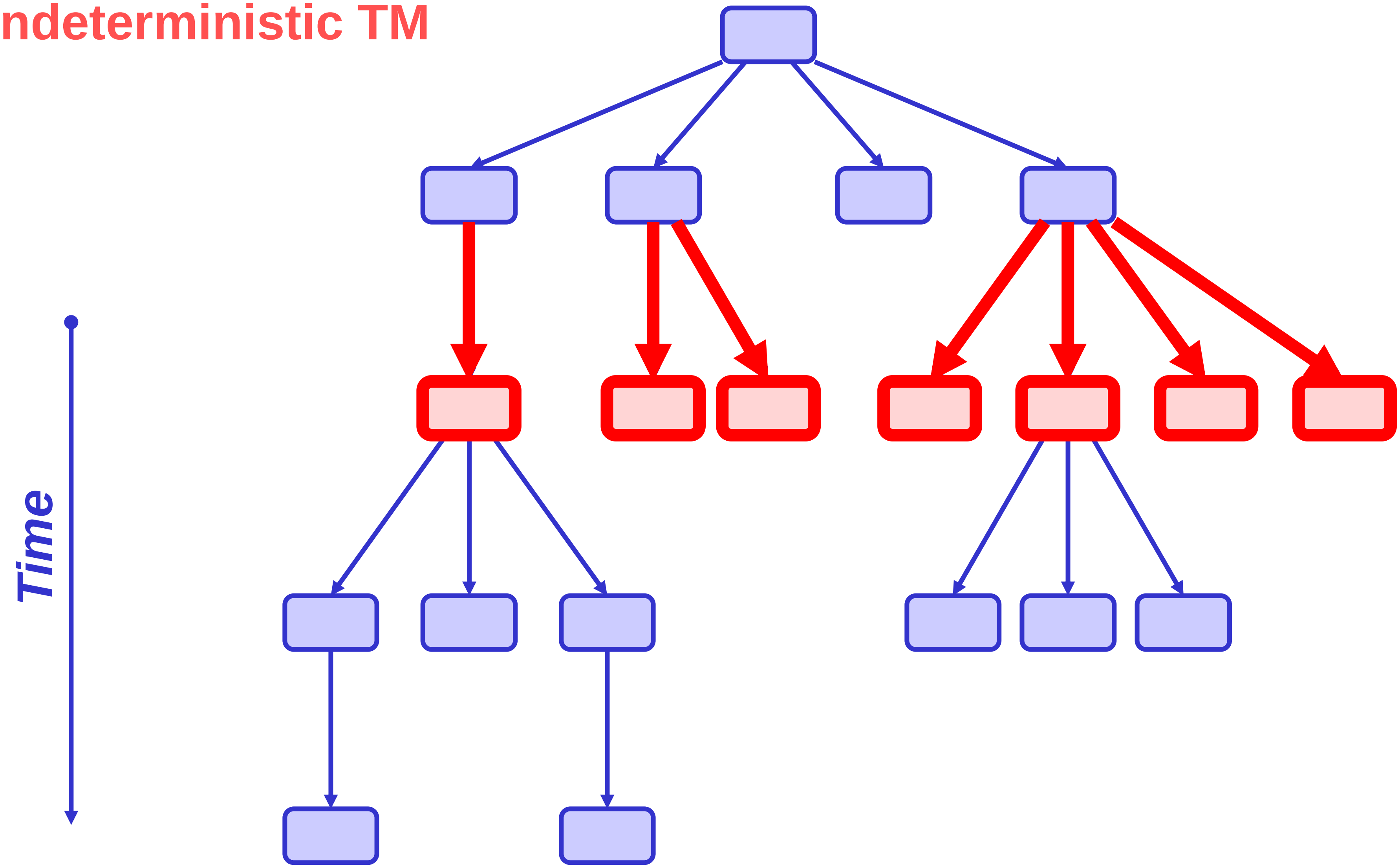
Relations between language classes

Nondeterministic TM



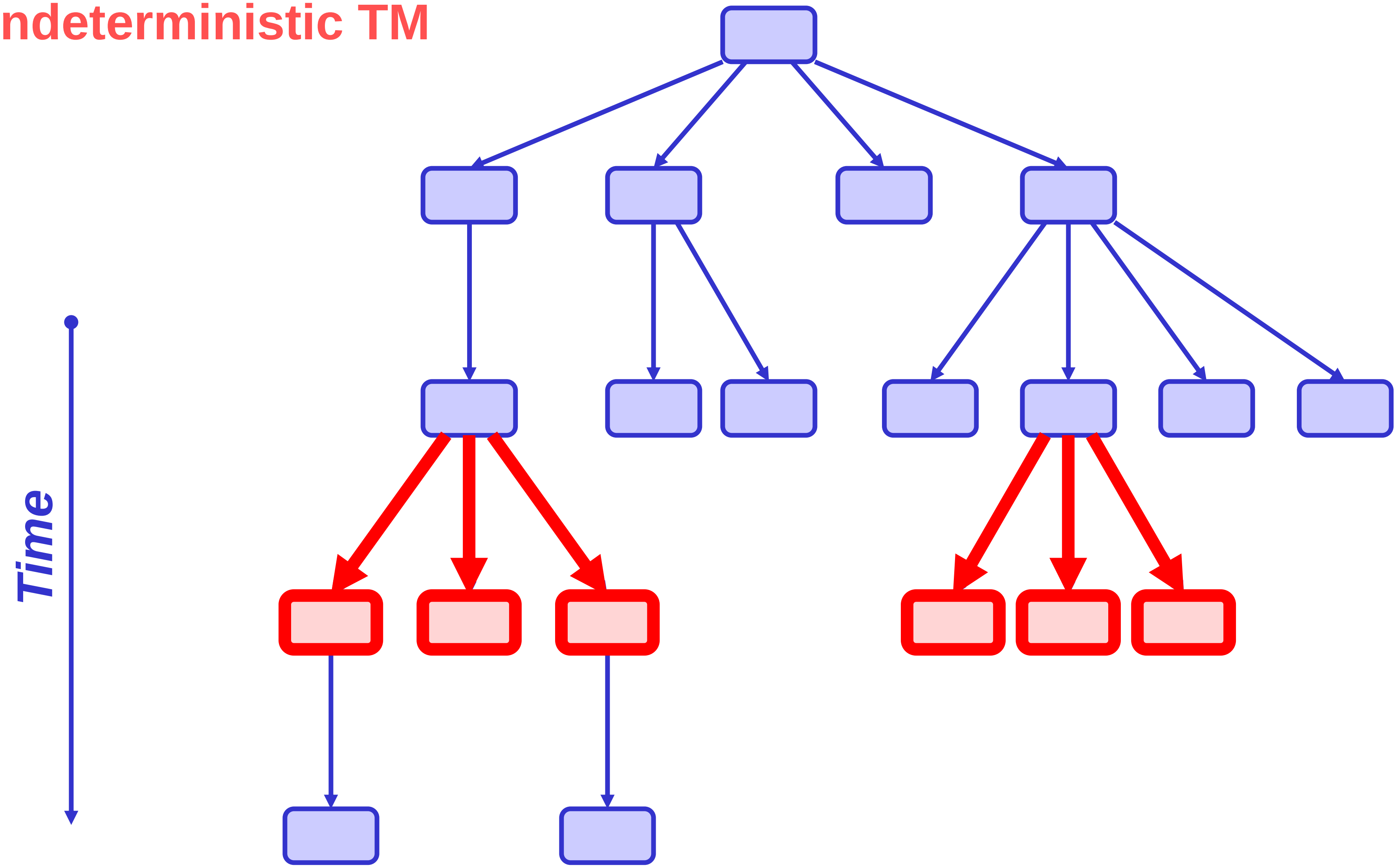
Relations between language classes

Nondeterministic TM



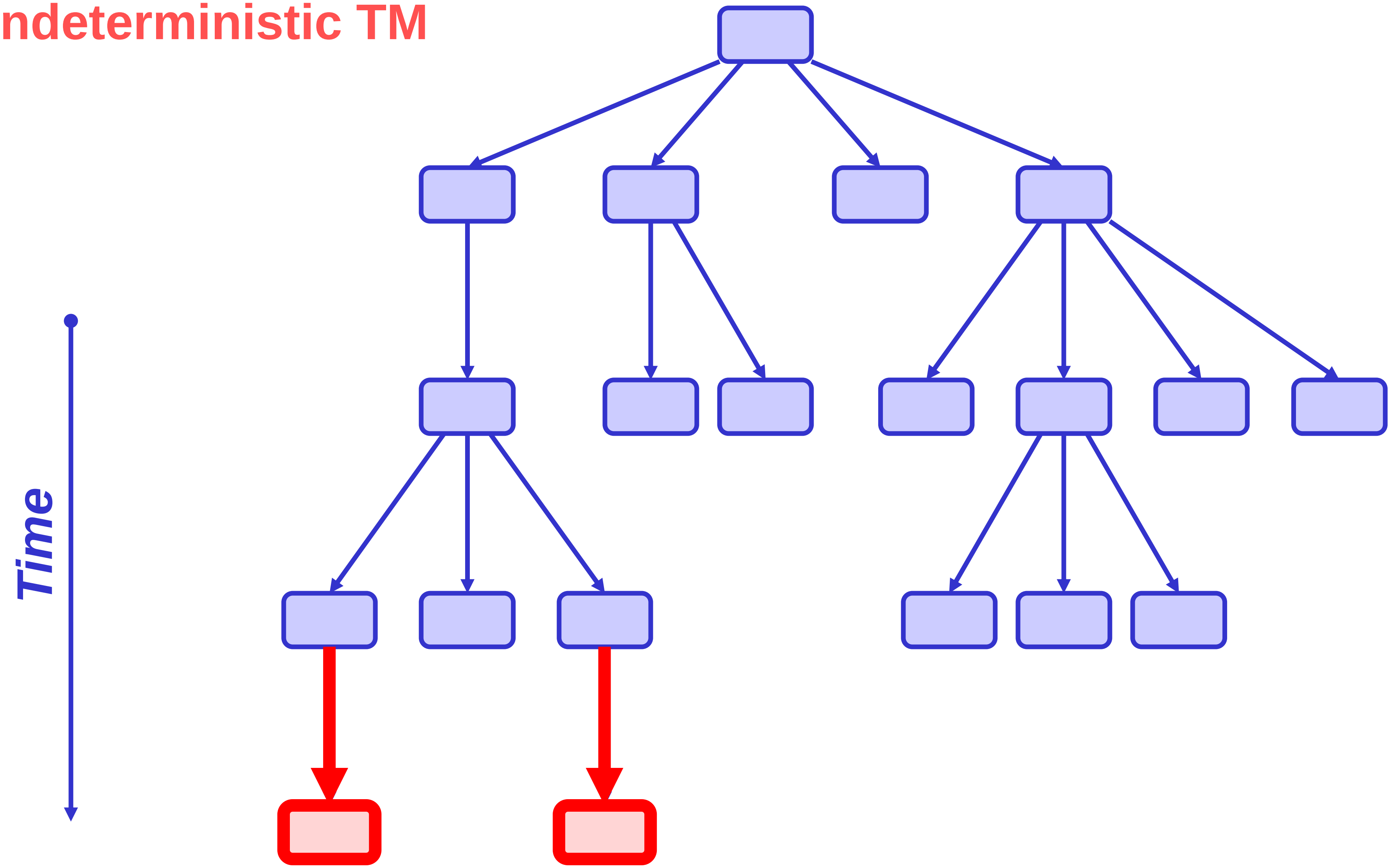
Relations between language classes

Nondeterministic TM



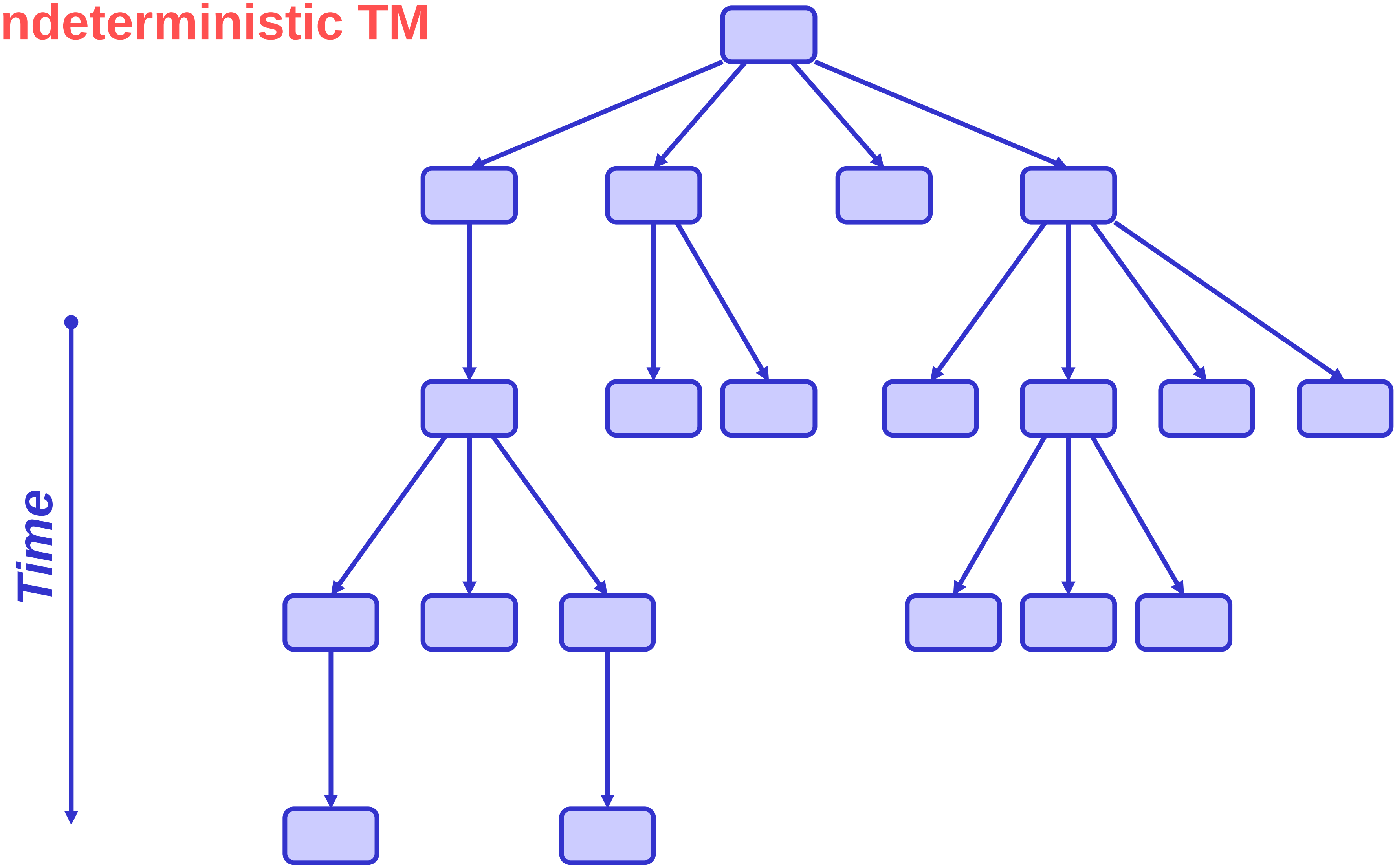
Relations between language classes

Nondeterministic TM



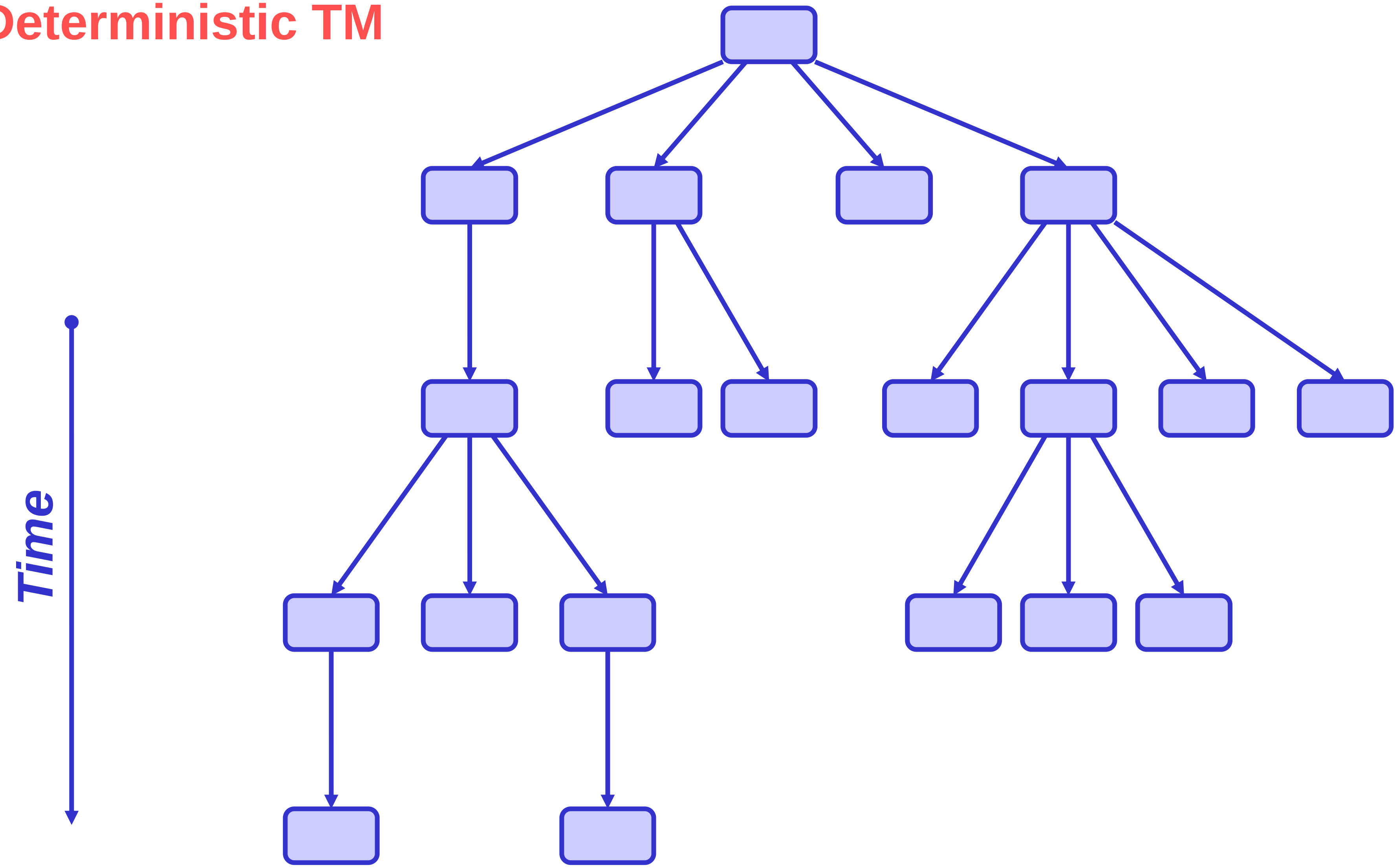
Relations between language classes

Nondeterministic TM



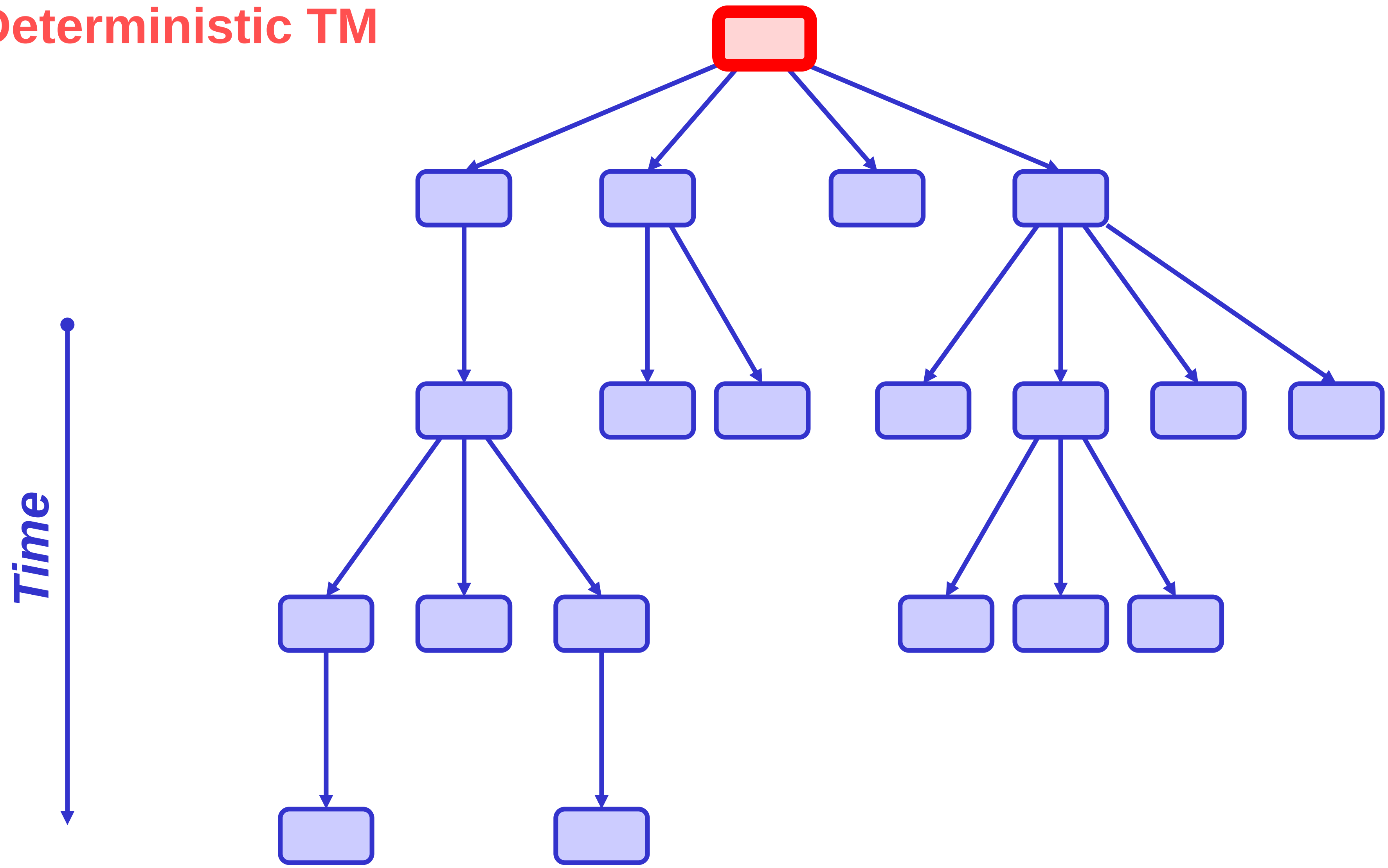
Relations between language classes

Deterministic TM



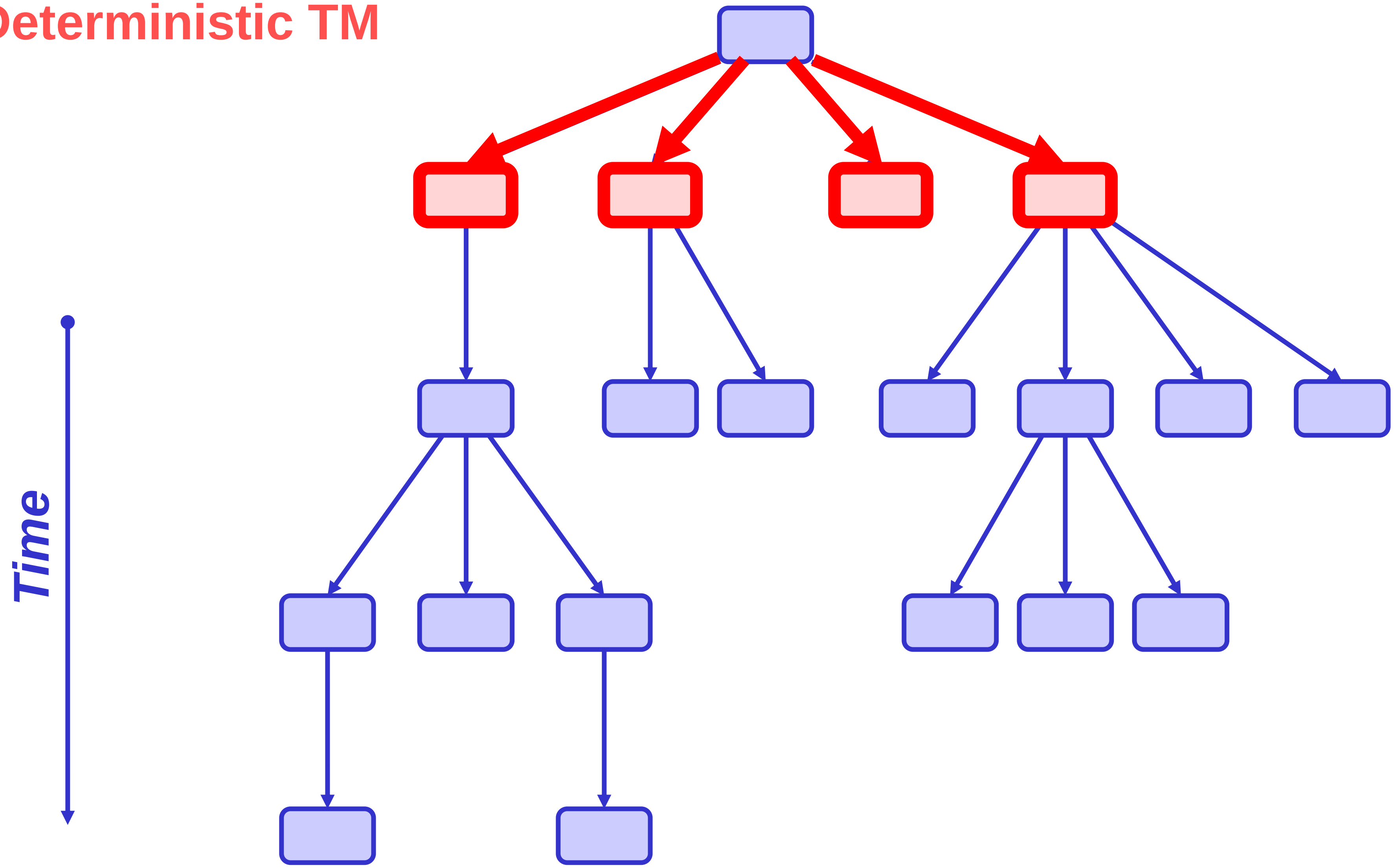
Relations between language classes

Deterministic TM



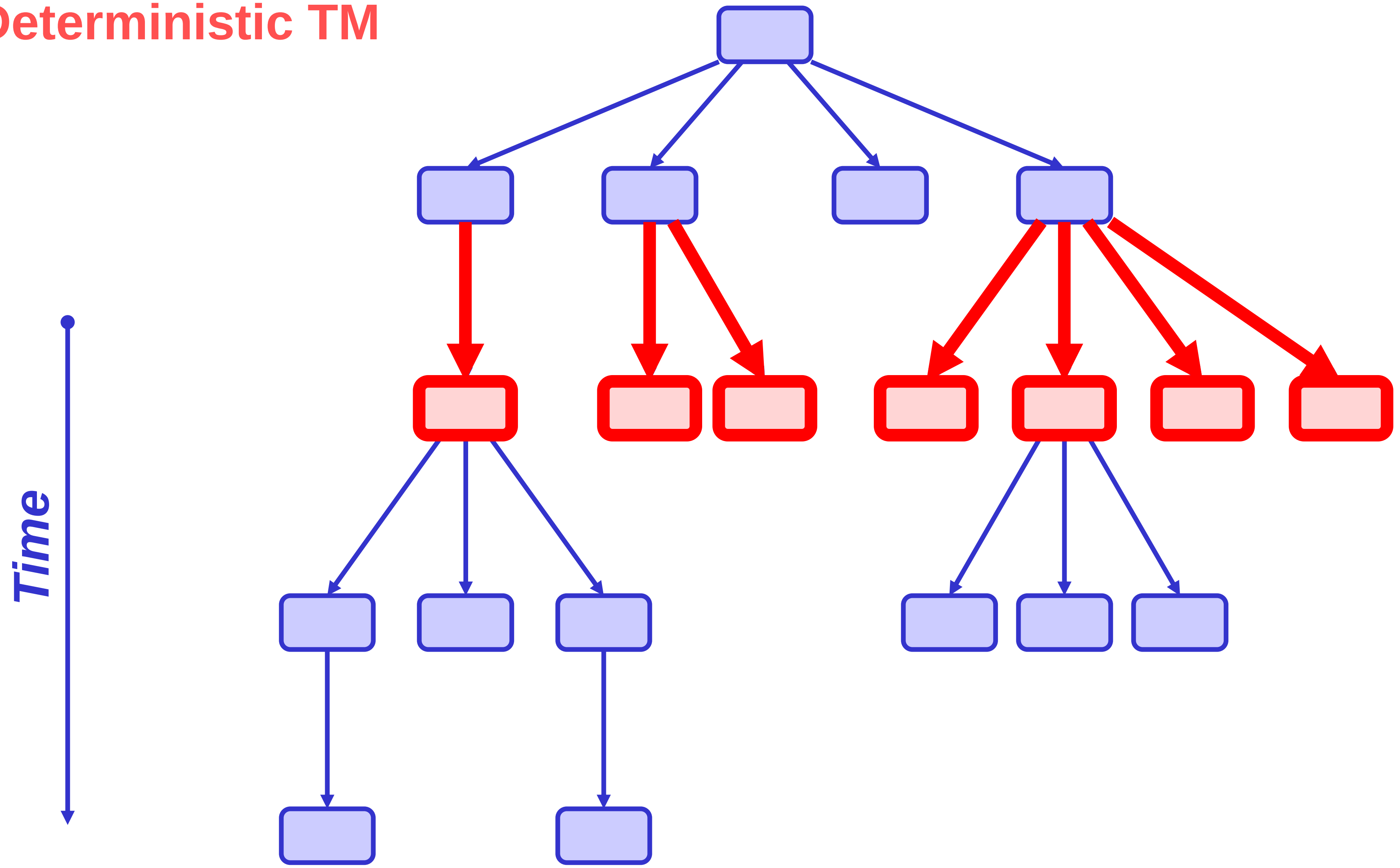
Relations between language classes

Deterministic TM



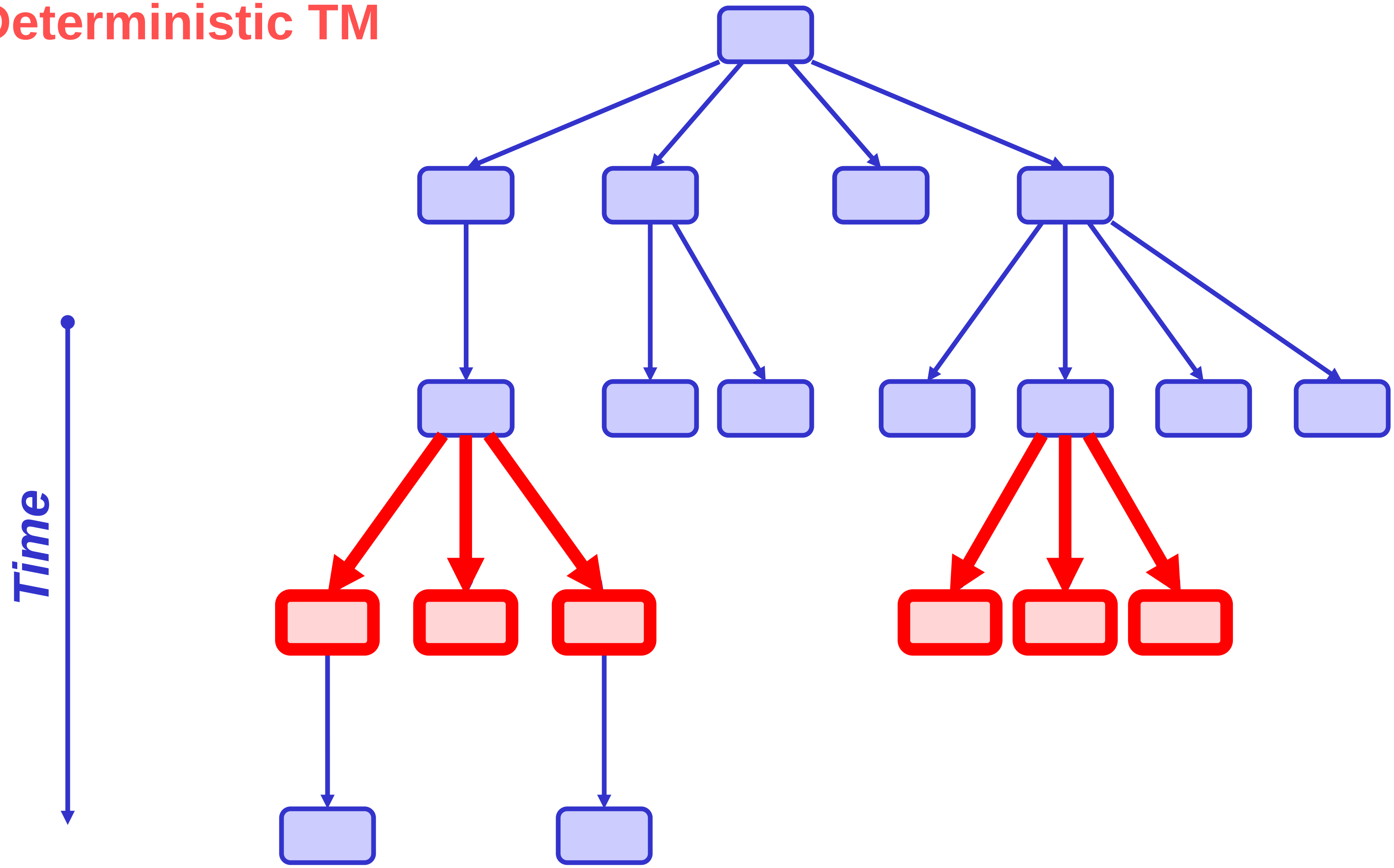
Relations between language classes

Deterministic TM



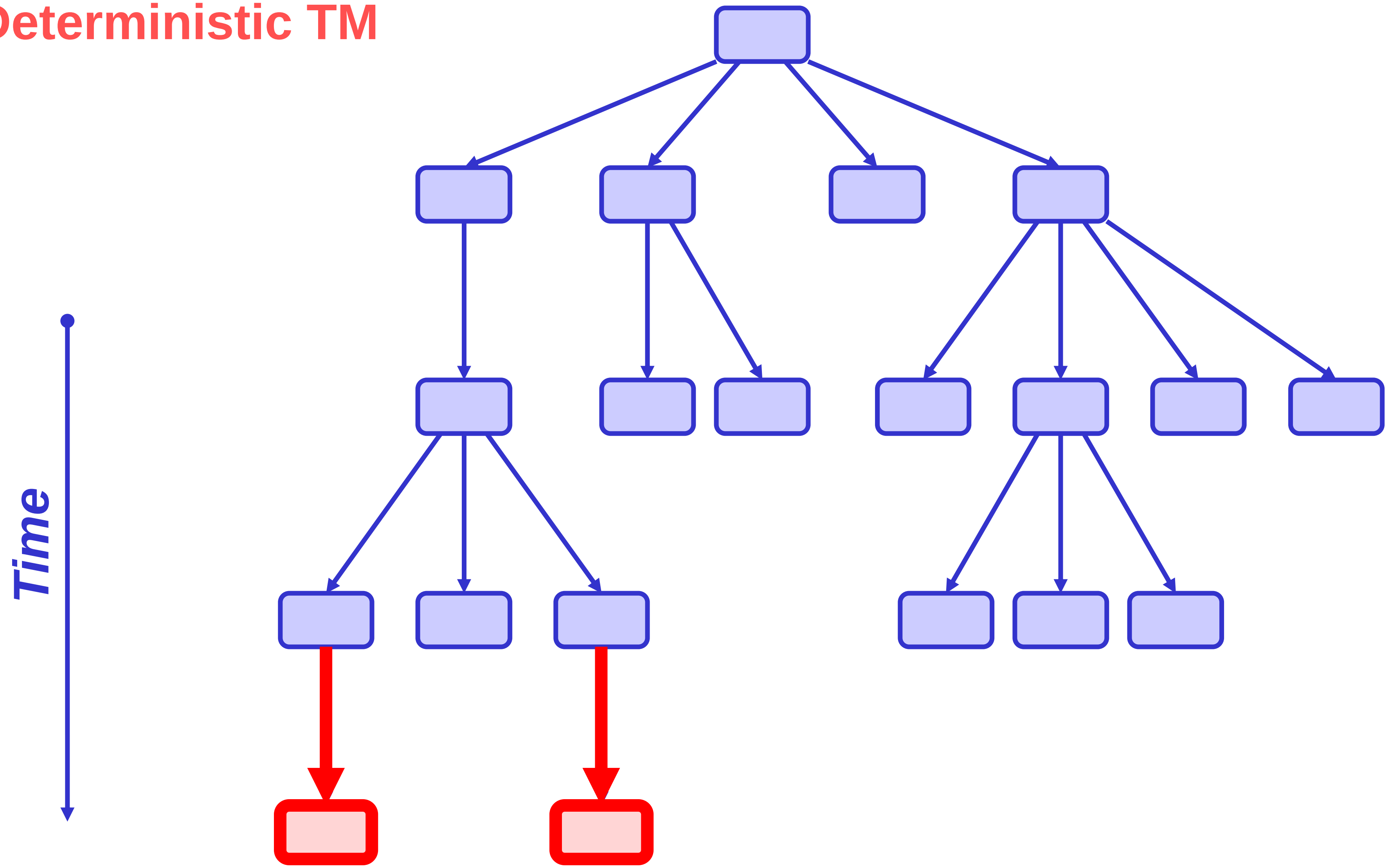
Relations between language classes

Deterministic TM



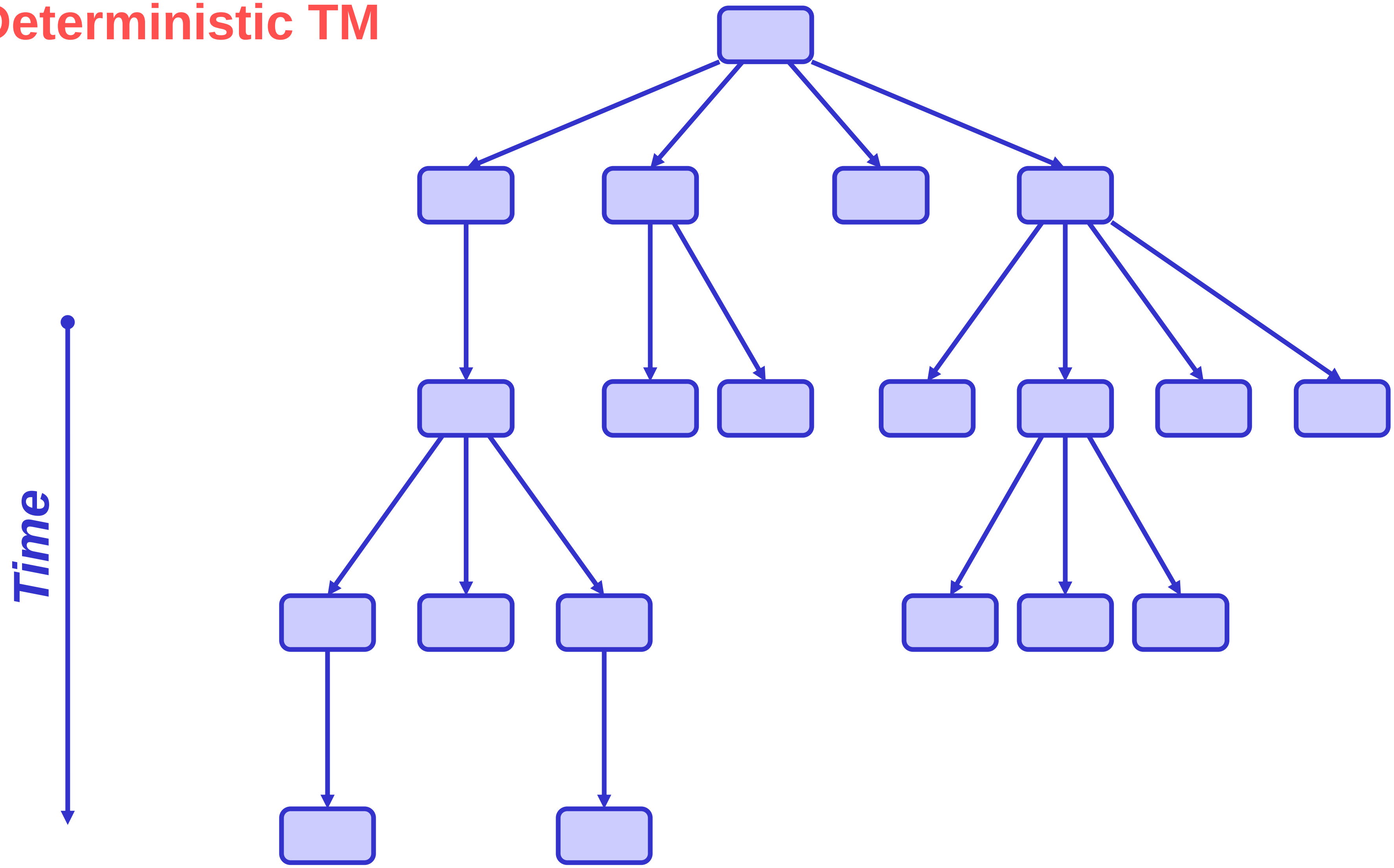
Relations between language classes

Deterministic TM



Relations between language classes

Deterministic TM



Relations between language classes

Relations between language classes

- A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration

Relations between language classes

- A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration
- Maximal number of cells needed to store a list of configurations

Relations between language classes

- A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration
- Maximal number of cells needed to store a list of configurations
  - Maximal number of configurations

Relations between language classes

- A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration
- Maximal number of cells needed to store a list of configurations
  - Maximal number of configurations
    - — $V = \mathrm{s}(f(n)+1)^k\, t^{k\ f(n)}$

Relations between language classes

- A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration
- Maximal number of cells needed to store a list of configurations
  - Maximal number of configurations
    - $V = s(f(n)+1)^k\, t^{k\ f(n)}$
  - number of cells needed to store one configuration

Relations between language classes

- **A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration**
- **Maximal number of cells needed to store a list of configurations**
  - **Maximal number of configurations**
    - — $V = s(f(n)+1)^k \, t^{k f(n)}$
  - **number of cells needed to store one configuration**
    - — $W = k(f(n)+1)+1$

Relations between language classes

- A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration
- Maximal number of cells needed to store a list of configurations
  - Maximal number of configurations
    - $V = s(f(n)+1)^k\, t^{k\ f(n)}$
  - number of cells needed to store one configuration
    - $W = k(f(n)+1)+1$
  - $B = V\,W$

Relations between language classes

- A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration
- Maximal number of cells needed to store a list of configurations
  - Maximal number of configurations
    - $V = s(f(n)+1)^k\, t^{k\ f(n)}$
  - number of cells needed to store one configuration
    - $W = k(f(n)+1)+1$
  - $B = V\, W$
- Head moves between list beginning and list end

Relations between language classes

- A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration
- Maximal number of cells needed to store a list of configurations
  - Maximal number of configurations
    - — $V = s(f(n)+1)^k\ t^{k\ f(n)}$
  - number of cells needed to store one configuration
    - — $W = k(f(n)+1)+1$
  - $B = V\ W$
- Head moves between list beginning and list end
  - number of head moves = B^2

Relations between language classes

- A deterministic TM simulates M_1 by constructing a list of all configurations reachable from the initial configuration
- Maximal number of cells needed to store a list of configurations
  - Maximal number of configurations
    - $V = s(f(n)+1)^k\, t^{k\ f(n)}$
  - number of cells needed to store one configuration
    - $W = k(f(n)+1)+1$
  - $B = V\,W$
- Head moves between list beginning and list end
  - number of head moves = B^2
  - $f(n) \geq \log_2 n \ \Rightarrow (c^{f(n)} \geq B^2) \Rightarrow L \in \mathrm{DTIME}(c^{f(n)})$

Relations between language classes

Relations between language classes

- **Space-constructible functions**

Relations between language classes

- **Space-constructible functions**
  - **Function $S(n)$ is *space-constructible***
    - —**for each $n > n_0$**
    - —**there is a TM M that computes $S(n)$ using exactly $S(n)$ cells**

Relations between language classes

- **Space-constructible functions**
  - **Function $S(n)$ is *space-constructible***
    - —for each $n > n_0$
    - —there is a TM M that computes $S(n)$ using exactly $S(n)$ cells
  - **Function $S(n)$ is *fully space-constructible***
    - —the above definition holds for each n
- **Time-constructible functions**

Relations between language classes

- **Space-constructible functions**
  - **Function $S(n)$ is *space-constructible***
    - —for each $n > n_0$
    - —there is a TM M that computes $S(n)$ using exactly $S(n)$ cells
  - **Function $S(n)$ is *fully space-constructible***
    - —the above definition holds for each n
- **Time-constructible functions**
  - **Function $T(n)$ is *time-constructible***
    - —for each $n > n_0$
    - —there is a TM M that computes$T(n)$ in exactly $T(n)$ steps (halts after $T(n)$ steps)

Relations between language classes

- **Space-constructible functions**
  - **Function $S(n)$ is *space-constructible***
    - —for each $n > n_0$
    - —there is a TM M that computes $S(n)$ using exactly $S(n)$ cells
  - **Function $S(n)$ is *fully space-constructible***
    - —the above definition holds for each n
- **Time-constructible functions**
  - **Function $T(n)$ is *time-constructible***
    - —for each $n > n_0$
    - —there is a TM M that computes$T(n)$ in exactly $T(n)$ steps (halts after $T(n)$ steps)
  - **Function $T(n)$ is *fully time-constructible***
    - —the above definition holds for each n

Relations between language classes

Relations between language classes

- **Function $f(n)$**

Relations between language classes

- **Function $f(n)$**
  - **$f(n) \geq \log_2 n$, fully space-constructible**

Relations between language classes

- **Function $f(n)$**
  - **$f(n) \geq \log_2 n$, fully space-constructible**
- **If $L \in \mathrm{NSPACE}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f^2(n))$**

Relations between language classes

- **Function $f(n)$**
  - **$f(n) \geq \log_2 n$, fully space-constructible**
- **If $L \in \mathrm{NSPACE}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f^2(n))$**
  - **Off-line nondeterministic TM M_1 with space complexity $f(n)$**

Relations between language classes

- **Function $f(n)$**
  - **$f(n) \geq \log_2 n$, fully space-constructible**
- **If $L \in \mathrm{NSPACE}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f^2(n))$**
  - **Off-line nondeterministic TM M_1 with space complexity $f(n)$**
    - **—one input tape and one work tape**

Relations between language classes

- **Function $f(n)$**
  - **$f(n) \geq \log_2 n$, fully space-constructible**
- **If $L \in \mathrm{NSPACE}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f^2(n))$**
  - **Off-line nondeterministic TM M_1 with space complexity $f(n)$**
    - **—one input tape and one work tape**
  - **Maximal number of different configurations of TM M_1**

Relations between language classes

- **Function $f(n)$**
  - **$f(n) \geq \log_2 n$, fully space-constructible**
- **If $L \in \mathrm{NSPACE}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f^2(n))$**
  - **Off-line nondeterministic TM M_1 with space complexity $f(n)$**
    - **one input tape and one work tape**
  - **Maximal number of different configurations of TM M_1**
  - **$s\ (n+2)\ f(n)\ t^{f(n)} \leq c^{f(n)}$**

Relations between language classes

- Function $f(n)$
  - $f(n) \geq \log_2 n$, fully space-constructible
- If $L \in \mathrm{NSPACE}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f^2(n))$
  - Off-line nondeterministic TM M_1 with space complexity $f(n)$
    - one input tape and one work tape
  - Maximal number of different configurations of TM M_1
  - $s\,(n+2)\,f(n)\,t^{f(n)} \leq c^{f(n)}$
    - s - number of states in Q

Relations between language classes

- Function $f(n)$
  - $f(n) \geq \log_2 n$, fully space-constructible
- If $L \in \text{NSPACE}(f(n)) \Rightarrow L \in \text{DSPACE}(f^{2}(n))$
  - Off-line nondeterministic TM M_1 with space complexity $f(n)$
    - —one input tape and one work tape
  - Maximal number of different configurations of TM M_1
  - $s\,(n+2)\,f(n)\,t^{f(n)} \leq c^{f(n)}$
    - —s - number of states in Q
    - —$(n+2)$ - number of different head positions on the input tape

Relations between language classes

- Function $f(n)$
  - $f(n) \geq \log_2 n$, fully space-constructible
- If $L \in \mathrm{NSPACE}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f^2(n))$
  - Off-line nondeterministic TM M_1 with space complexity $f(n)$
    - one input tape and one work tape
  - Maximal number of different configurations of TM M_1
  - $s\,(n+2)\,f(n)\,t^{f(n)} \leq c^{f(n)}$
    - s - number of states in Q
    - $(n+2)$ - number of different head positions on the input tape
      - n - length of string w

Relations between language classes

- **Function $f(n)$**
  - **$f(n) \geq \log_2 n$, fully space-constructible**
- **If $L \in \text{NSPACE}(f(n)) \Rightarrow L \in \text{DSPACE}(f^2(n))$**
  - **Off-line nondeterministic TM M_1 with space complexity $f(n)$**
    - **one input tape and one work tape**
  - **Maximal number of different configurations of TM M_1**
  - **$s\,(n+2)\,f(n)\,t^{f(n)} \leq c^{f(n)}$**
    - **s - number of states in Q**
    - **$(n+2)$ - number of different head positions on the input tape**
      - n - length of string w
    - **$f(n)$ - number of different head positions on the work tape**

Relations between language classes

- Function $f(n)$
  - $f(n) \geq \log_2 n$, fully space-constructible
- If $L \in \mathrm{NSPACE}(f(n)) \Rightarrow L \in \mathrm{DSPACE}(f^2(n))$
  - Off-line nondeterministic TM M_1 with space complexity $f(n)$
    - one input tape and one work tape
  - Maximal number of different configurations of TM M_1
  - $s\,(n+2)\,f(n)\,t^{f(n)} \leq c^{f(n)}$
    - s - number of states in Q
    - $(n+2)$ - number of different head positions on the input tape
      - n - length of string w
    - $f(n)$ - number of different head positions on the work tape
    - $t^{f(n)}$ - number of different work tape contents

Relations between language classes

- Function $f(n)$
  - $f(n) \geq \log_2 n$, fully space-constructible
- If $L \in \text{NSPACE}(f(n)) \Rightarrow L \in \text{DSPACE}(f^2(n))$
  - Off-line nondeterministic TM M_1 with space complexity $f(n)$
    - one input tape and one work tape
  - Maximal number of different configurations of TM M_1
  - $s\,(n+2)\,f(n)\,t^{f(n)} \leq c^{f(n)}$
    - s - number of states in Q
    - $(n+2)$ - number of different head positions on the input tape
      - n - length of string w
    - $f(n)$ - number of different head positions on the work tape
    - $t^{f(n)}$ - number of different work tape contents
      - t – number of tape symbols in Γ

Relations between language classes

- Function $f(n)$
  - $f(n) \geq \log_2 n$, fully space-constructible
- If $L \in \text{NSPACE}(f(n)) \Rightarrow L \in \text{DSPACE}(f^2(n))$
  - Off-line nondeterministic TM M_1 with space complexity $f(n)$
    - one input tape and one work tape
  - Maximal number of different configurations of TM M_1
  - $s\,(n+2)\,f(n)\,t^{f(n)} \leq c^{f(n)}$
    - s - number of states in Q
    - $(n+2)$ - number of different head positions on the input tape
      - n - length of string w
    - $f(n)$ - number of different head positions on the work tape
    - $t^{f(n)}$ - number of different work tape contents
      - t – number of tape symbols in Γ
    - Content of the input tape does not change and does not affect the total number of configurations

Relations between language classes

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

- TM M_1

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

- TM M_1
  - from configuration I_1

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

- TM M_1
  - from configuration I_1
  - moves into configuration I_2

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

- TM M_1
  - from configuration I_1
  - moves into configuration I_2
  - using at most 2^i moves

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

Test(l_1, l_2, i)

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

Test(l_1, l_2, i)
{

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

Test(l_1, l_2, i)

{

<u>if</u> $(i == 0)$ && $((l_1 == l_2) \,||\, (l_1 \succ l_2)))$

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

```
Test(l1, l2, i)
{
        if (i == 0) && ((l1 = = l2) || (l1 ≻ l2)))
                return TRUE;
```

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

Test(l_1, l_2, i)
{

if $(i == 0)$ && $((l_1 == l_2) \,||\, (l_1 \succ l_2)))$
return ***TRUE;***

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

Test(l_1, l_2, i***)***

{

<u>if</u> (i **== 0) && ((**l_1 **= =** l_2**) || (**$l_1 \succ l_2$**)))**

<u>return</u> ***TRUE;***

<u>if</u> ($i \geq 1$**)**

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

Test(l_1, l_2, i)
{

<u>if</u> $(i == 0)$ **&&** $((l_1 = = l_2) \,||\, (l_1 \succ l_2)))$
<u>return</u> ***TRUE;***

<u>if</u> $(i \geq 1)$

<u>for each</u> **(*configuration*** l' ***of lenght*** $< f(n)$**)**

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

Test(I_1, I_2, i)
{

<u>if</u> (i == 0) && ((I_1 = = I_2) || ($I_1 \succ I_2$)))
<u>return</u> *TRUE;*

<u>if</u> ($i \geq 1$)

<u>for each</u> (*configuration* I' *of lenght* < $f(n)$)

<u>if</u> ((*Test*(I_1, I', i–1) = = *TRUE*) &&

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

Test(I_1, I_2, i)
{
if (i == 0) && ((I_1 = = I_2) || ($I_1 \succ I_2$)))
return *TRUE;*

if ($i \geq 1$)

for each (*configuration* I' *of lenght* < $f(n)$)

if ((*Test*(I_1, I', i–1) = = *TRUE*) &&
(*Test*(I', I_2, i–1) = = *TRUE*))

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

Test(I_1, I_2, i)

{

if $(i == 0)$ && $((I_1 == I_2) \,||\, (I_1 \succ I_2)))$

return ***TRUE;***

if $(i \geq 1)$

for each (*configuration* I' *of lenght* $< f(n)$)

if $((Test(I_1, I', i-1) == TRUE)$ && $(Test(I', I_2, i-1) == TRUE))$

return ***TRUE;***

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

Test(I_1, I_2, i)

{

if $(i == 0)$ && $((I_1 == I_2) \mid\mid (I_1 \succ I_2))$

return ***TRUE;***

if $(i \geq 1)$

for each (*configuration* I' *of lenght* $< f(n)$)

if $((Test(I_1, I', i-1) == TRUE)$ &&

$(Test(I', I_2, i-1) == TRUE))$

return ***TRUE;***

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

```
Test(I1, I2, i)
{
        if (i == 0) && ((I1 = = I2) || (I1 ≻ I2)))
                        return TRUE;

        if (i ≥ 1)

                for each (configuration I' of lenght < f(n))

                        if      ((Test(I1, I', i–1) = = TRUE) &&
                                (Test(I', I2, i–1) = = TRUE))
                                        return TRUE;

        return FALSE;
```

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

***Test*(I_1, I_2, i)**

```
Test(I1, I2, i)
{
        if (i == 0) && ((I1 = = I2) || (I1 ≻ I2)))
                        return TRUE;

        if (i ≥ 1)

                for each (configuration I' of lenght < f(n))

                        if      ((Test(I1, I', i–1) = = TRUE) &&
                                 (Test(I', I2, i–1) = = TRUE))
                                        return TRUE;

        return FALSE;

}
```

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

- TM M_2 simulates a nondeterministic TM M_1

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

- TM M_2 simulates a nondeterministic TM M_1
  - Testing a string w of lenght n

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

- TM M_2 simulates a nondeterministic TM M_1
  - Testing a string w of lenght n
  - TM M_2 calls $Test(l_1,l_2,i)$

Relations between language classes

$$l_1 \overset{i}{\succ} l_2$$

- TM M_2 simulates a nondeterministic TM M_1
  - Testing a string w of lenght n
  - TM M_2 calls $Test(l_1,l_2,i)$

    1. Based on the lenght n it calculates $f(n)$

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

- TM M_2 simulates a nondeterministic TM M_1
  - Testing a string w of lenght n
  - TM M_2 calls $Test(I_1, I_2, i)$

1. Based on the lenght n it calculates $f(n)$
2. Configurations I_1 is the initial configuration

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

- TM M_2 simulates a nondeterministic TM M_1
  - Testing a string w of lenght n
  - TM M_2 calls $Test(I_1, I_2, i)$

1. Based on the lenght n it calculates $f(n)$
2. Configurations I_1 is the initial configuration
3. Configurations I_2 is one of the configurations of TM M_1 which accept the string

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

- TM M_2 simulates a nondeterministic TM M_1
  - Testing a string w of lenght n
  - TM M_2 calls $Test(I_1, I_2, i)$

1. Based on the lenght n it calculates $f(n)$
2. Configurations I_1 is the initial configuration
3. Configurations I_2 is one of the configurations of TM M_1 which accept the string
4. Maximal number of configurations from initial configuration to the configuration that accepts the string $\leq c^{f(n)} \Rightarrow$

Relations between language classes

$$I_1 \overset{i}{\succ} I_2$$

- TM M_2 simulates a nondeterministic TM M_1
  - Testing a string w of lenght n
  - TM M_2 calls $Test(I_1, I_2, i)$

1. Based on the lenght n it calculates $f(n)$

2. Configurations I_1 is the initial configuration

3. Configurations I_2 is one of the configurations of TM M_1 which accept the string

4. Maximal number of configurations from initial configuration to the configuration that accepts the string $\leq c^{f(n)} \Rightarrow$

$$i = log_2(c^{f(n)}) = f(n)\, log_2 c$$

Relations between language classes

Relations between language classes

LIFO stack

Relations between language classes

Number of cells

LIFO stack

Relations between language classes

Beginning of work

Number of cells

LIFO stack

Relations between language classes

Beginning of work

Input tape contents

Number of cells

LIFO stack

Relations between language classes

Beginning of work

Input tape contents

Number of cells

Recursive algorithm call

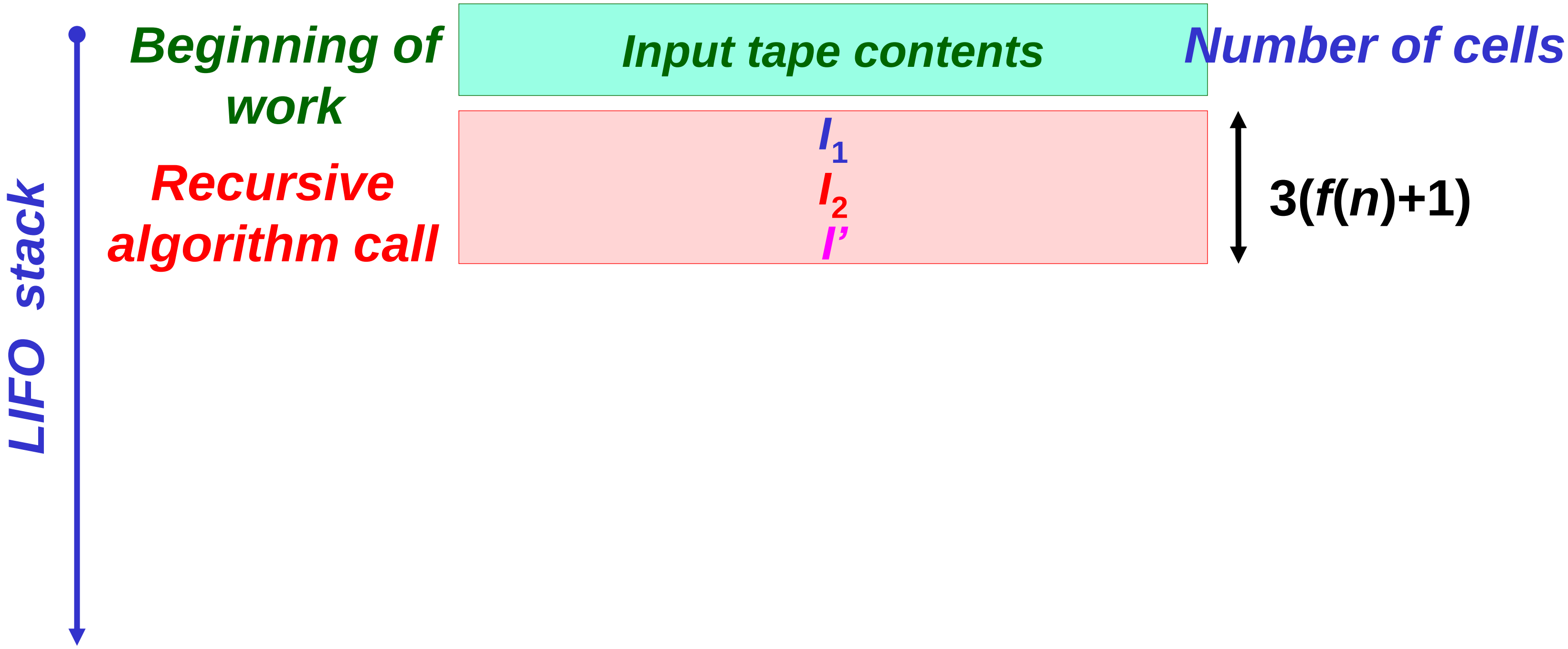
LIFO stack

Relations between language classes

LIFO stack

Beginning of work	*Input tape contents*	*Number of cells*
Recursive algorithm call	l_1 l_2 l'	

Relations between language classes



Relations between language classes

LIFO stack

Beginning of work — **Input tape contents** — **Number of cells**

Recursive algorithm call

l_1
l_2
l'

$3(f(n)+1)$

binary counter for head position l_1
binary counter for head position l_2
binary counter for head position l'

Relations between language classes

LIFO stack

Beginning of work	*Input tape contents*	*Number of cells*
Recursive algorithm call	I_1 I_2 I'	$3(f(n)+1)$
	binary counter for head position I_1 *binary counter for head position* I_2 *binary counter for head position* I'	$3\ log_2(n+2)$

Relations between language classes

LIFO stack		Number of cells
Beginning of work	*Input tape contents*	
Recursive algorithm call	I_1 I_2 I'	$3(f(n)+1)$
	binary counter for head position I_1 *binary counter for head position* I_2 *binary counter for head position* I'	$3\ log_2(n+2)$
	binary counter for parameter i	

Relations between language classes

LIFO stack

Beginning of work	***Input tape contents***	***Number of cells***
Recursive algorithm call	I_1 I_2 I'	$3(f(n)+1)$
	binary counter for head position I_1 ***binary counter for head position*** I_2 ***binary counter for head position*** I'	$3\ log_2(n+2)$
	binary counter for parameter i	$f(n)\ log_2 c$

Relations between language classes

	Input tape contents	Number of cells
Beginning of work	I_1 I_2 I'	$3(f(n)+1)$
Recursive algorithm call	binary counter for head position I_1 binary counter for head position I_2 binary counter for head position I'	$3\ log_2(n+2)$
	binary counter for parameter i	$f(n)\ log_2 c$

LIFO stack

$$f(n)+1 \approx f(n) \wedge$$
$$(log_2(n+2) \leq f(n)) \wedge$$
$$(log_2 c \text{ is constant})$$

$$\Rightarrow 3(f(n)+1) + 3\ log_2(n+2) + f(n)\ log_2 c \approx$$
$$3f(n) \quad + \quad 3f(n) \quad + log_2 c\ f(n) \approx f(n)$$

Relations between language classes

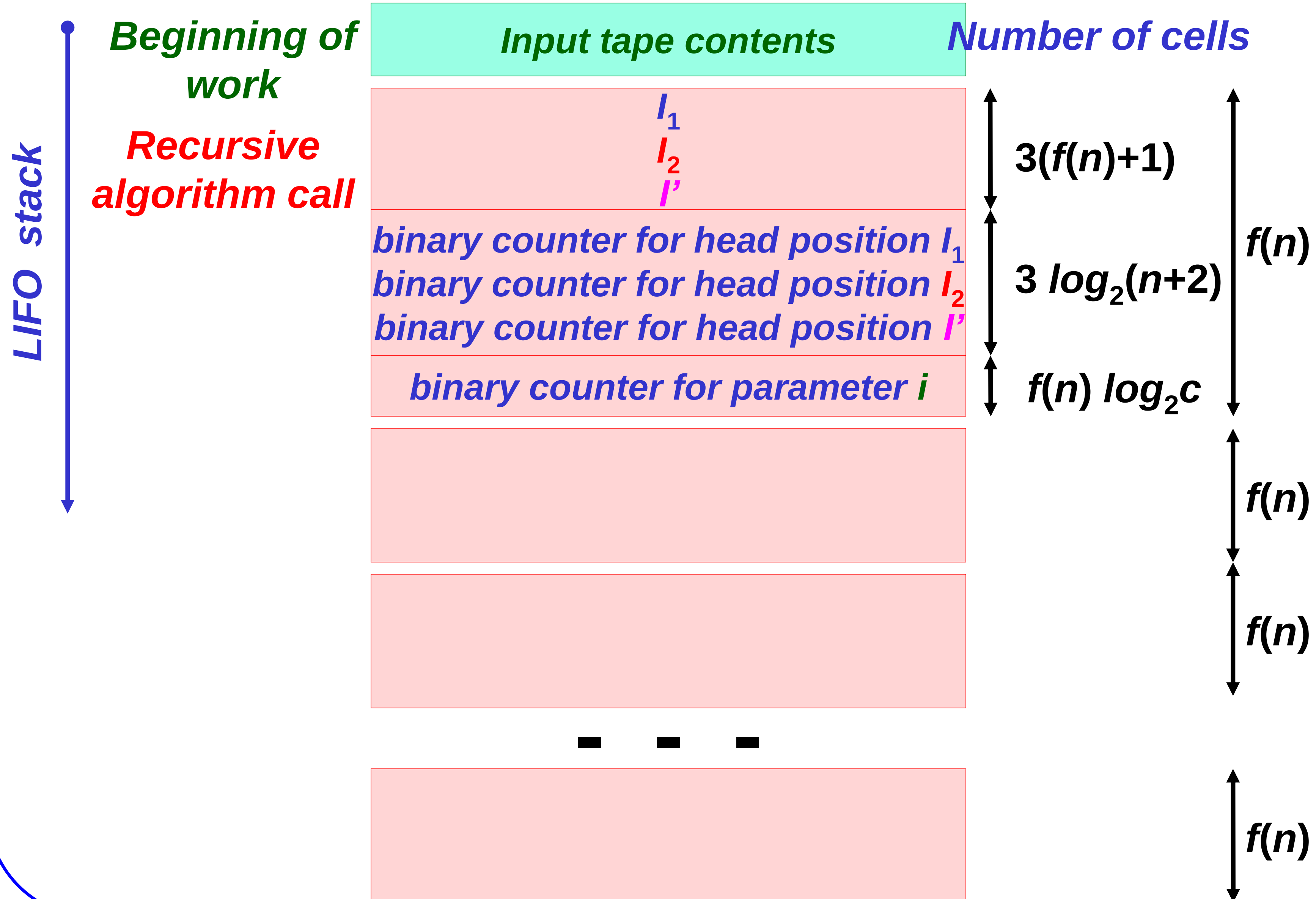
LIFO stack

Beginning of work

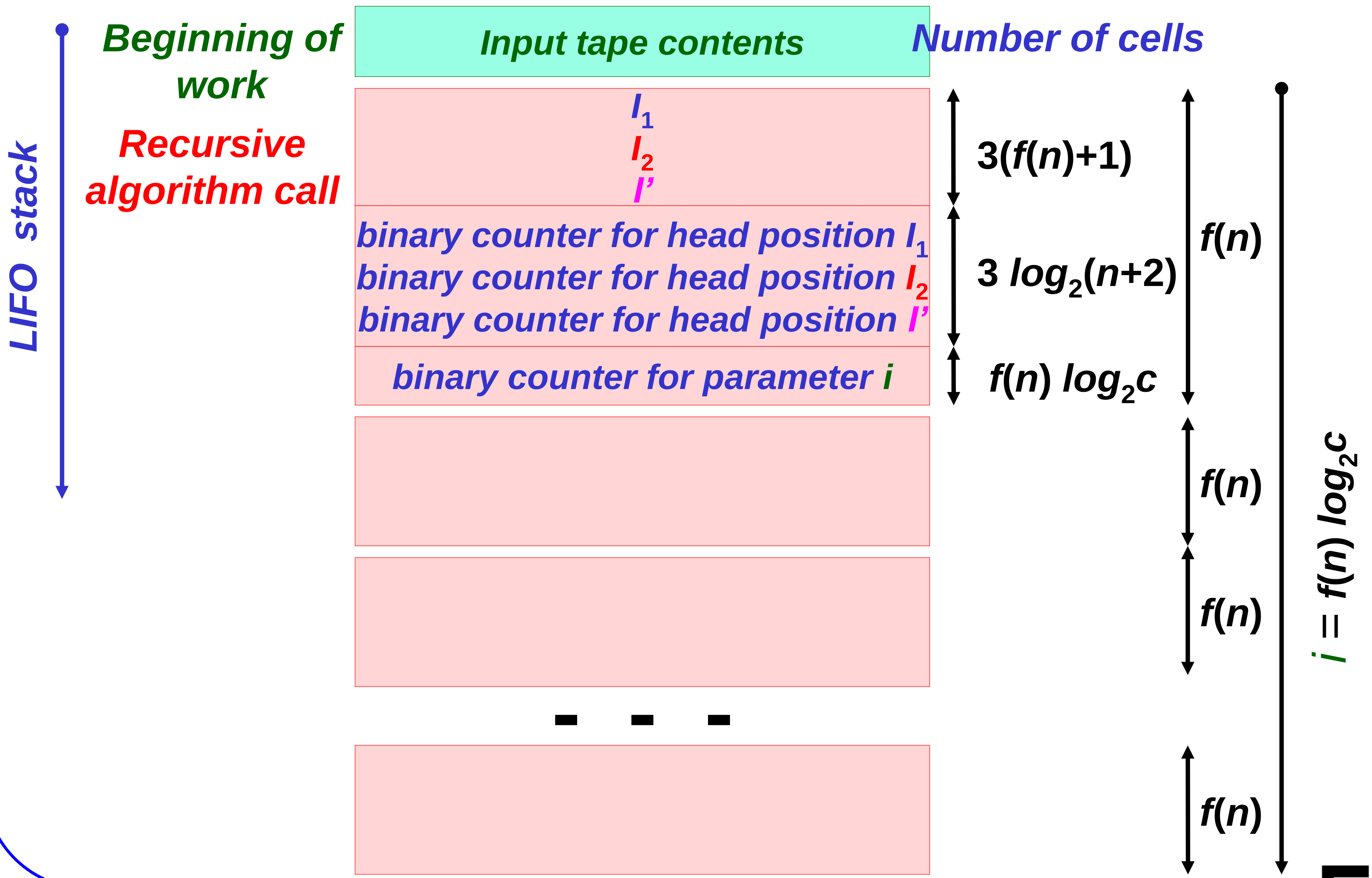
Recursive algorithm call

Input tape contents	*Number of cells*	
I_1 I_2 I'	$3(f(n)+1)$	$f(n)$
binary counter for head position I_1 *binary counter for head position* I_2 *binary counter for head position* I'	$3\ log_2(n+2)$	
binary counter for parameter i	$f(n)\ log_2 c$	

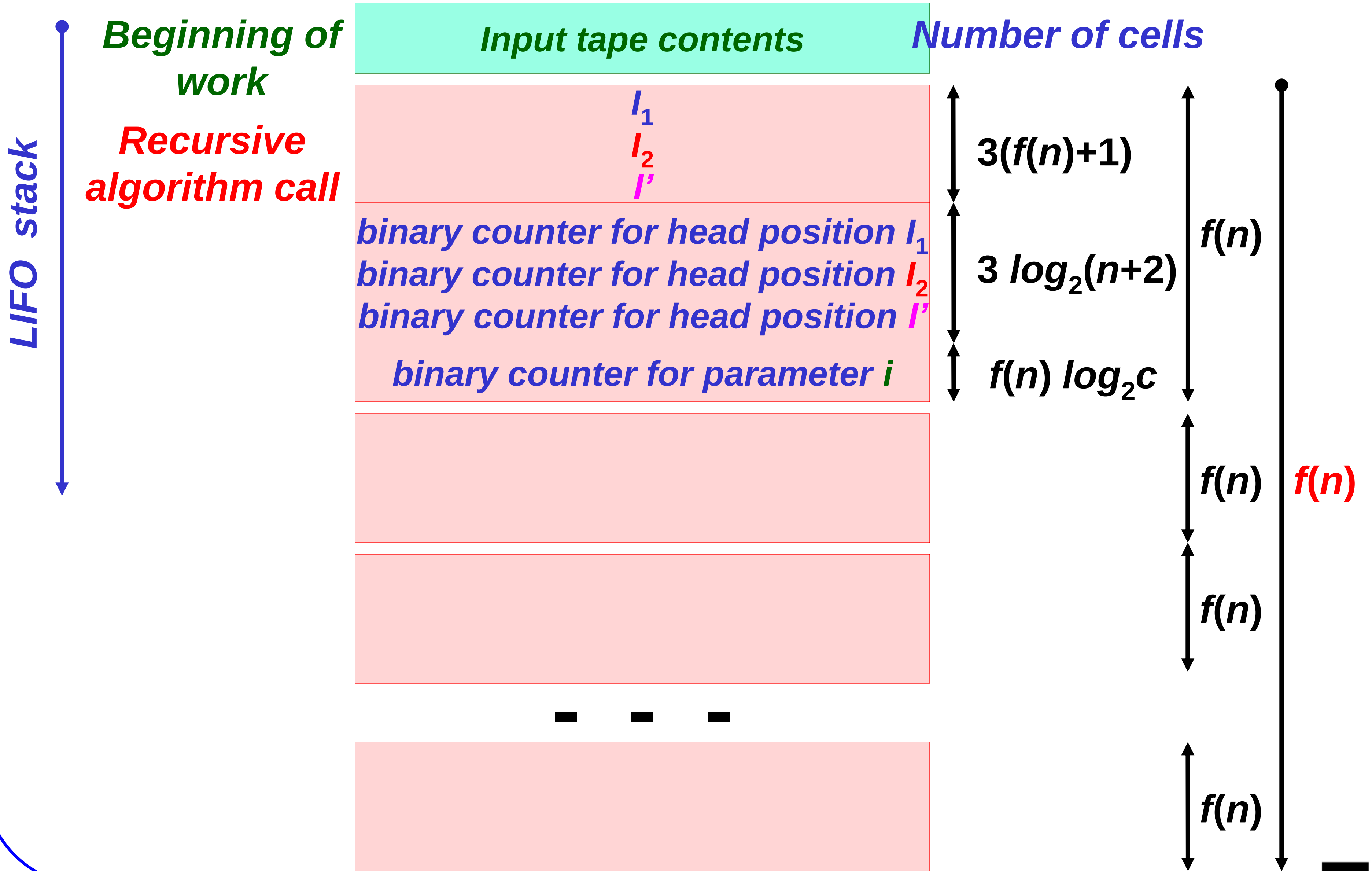
Relations between language classes



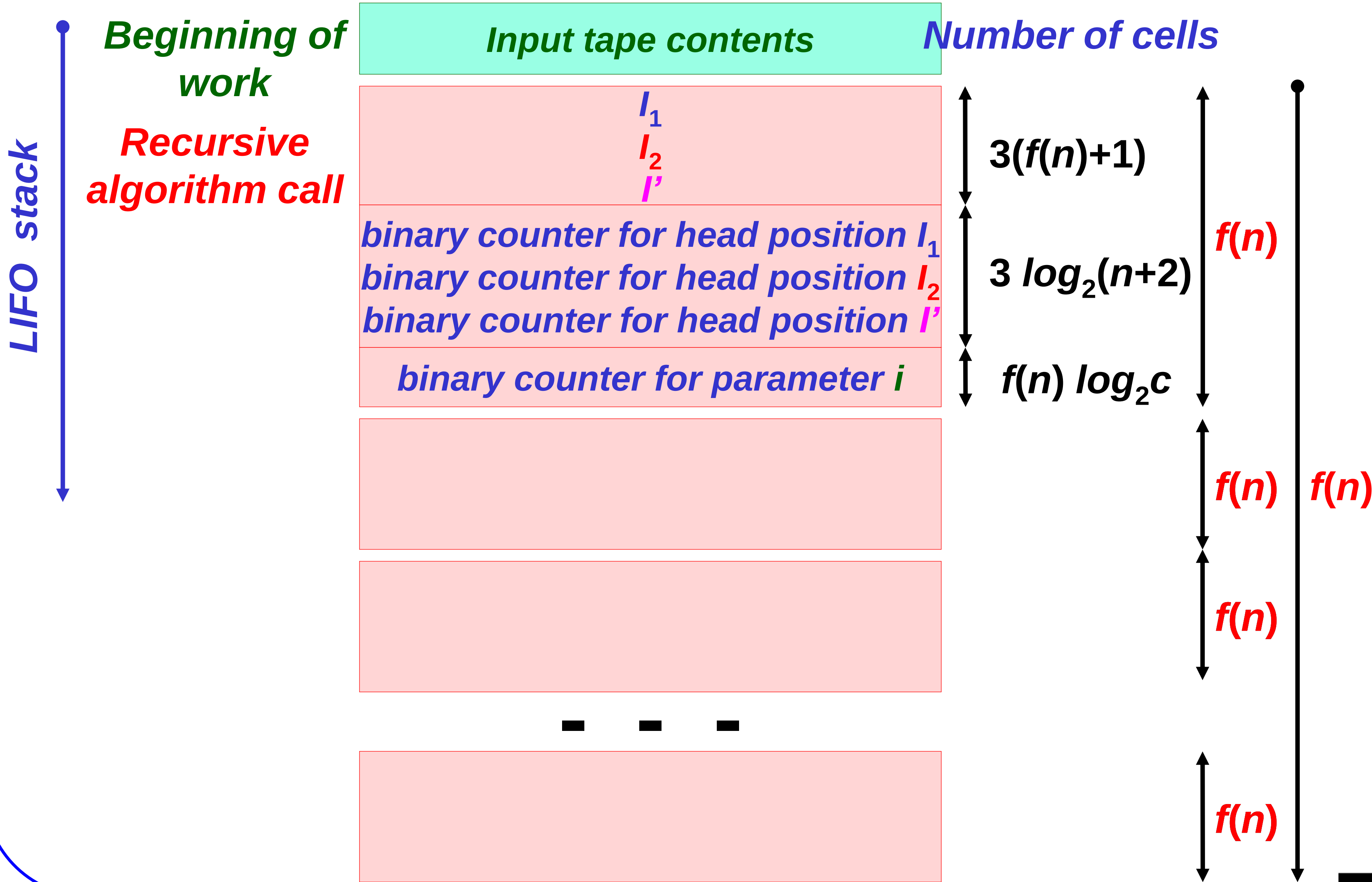
Relations between language classes



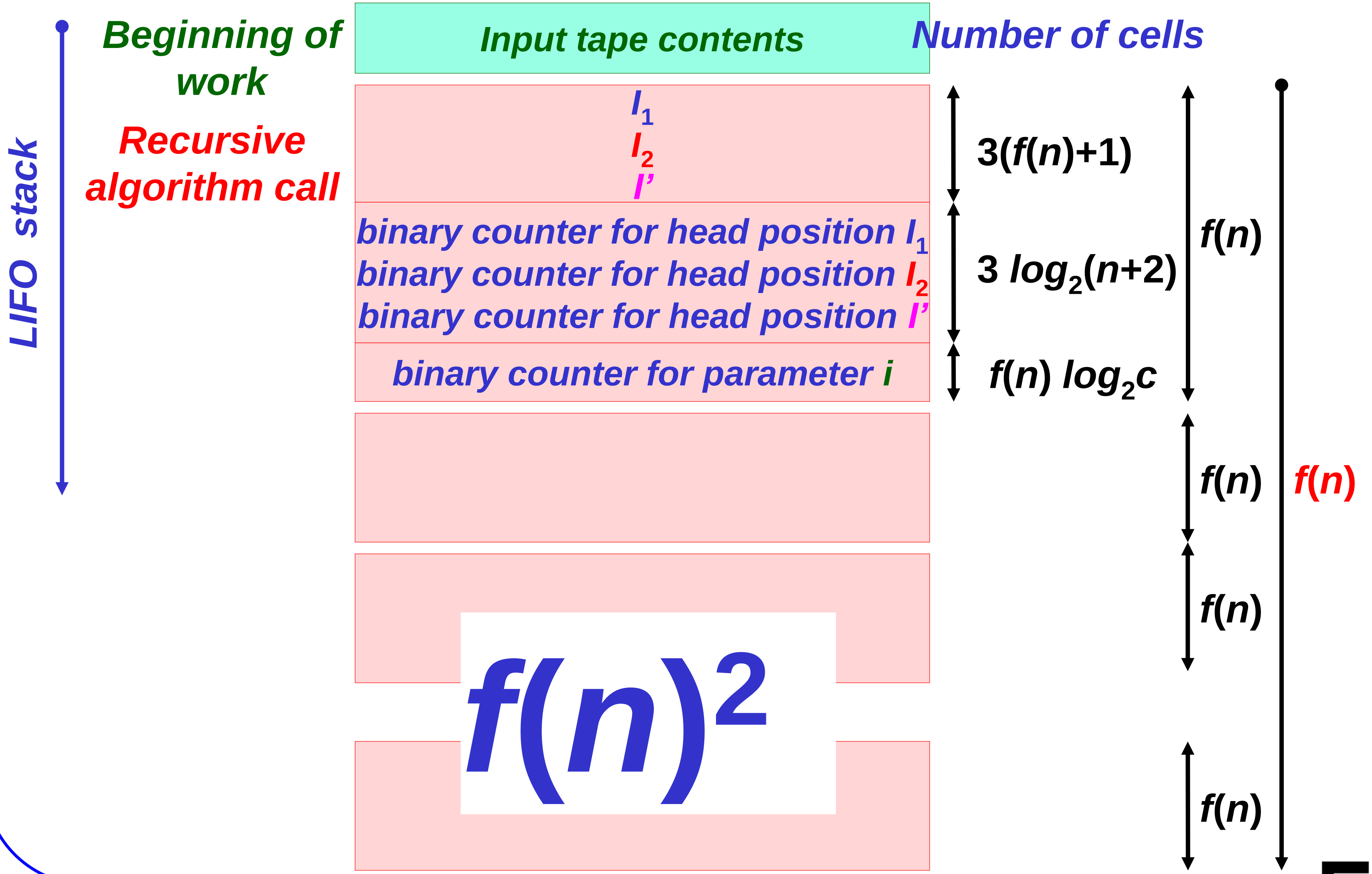
Relations between language classes



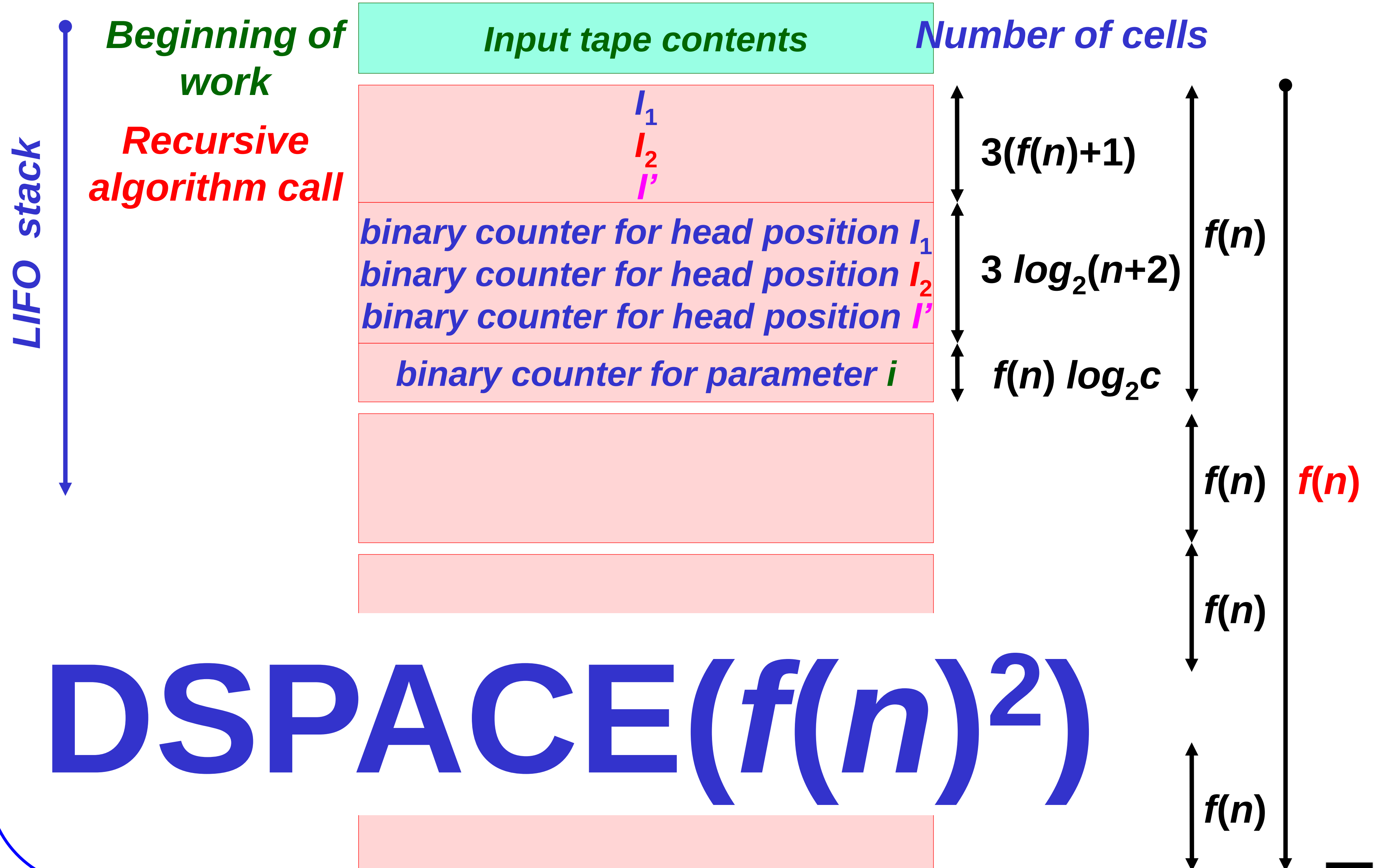
Relations between language classes



Relations between language classes



Relations between language classes



DSPACE($f(n)^2$)

Relations between language classes