Resource bounded derivation

Hiroyuki Seki
Nara Institute of Science and Technology

Workshop on MCFG & Related Formalisms
National Institute of Informatics, Oct 5-6, 2010

Example

G_1: $S(xy)$:- $A(x, y)$.

$A(axb, cyd)$:- $A(x, y)$.

$A(ab, cd)$.

$L(G_1) = \{ a^n b^n c^n d^n \mid n \geqq 1 \}$

$S(a^n b^n c^n d^n)$

|

$A(a^n b^n, c^n d^n)$

|

:

|

$A(aabb, ccdd)$

|

$A(ab, cd)$

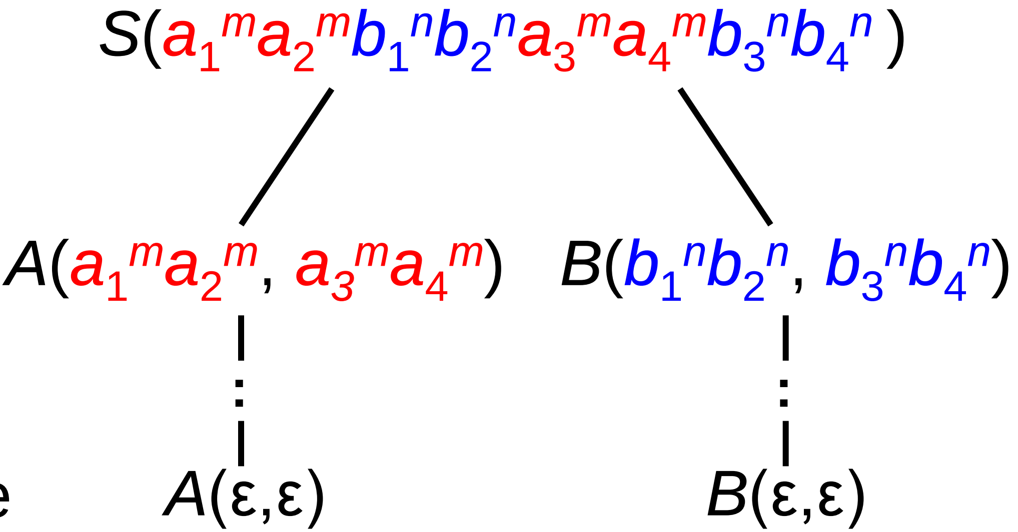
Proof tree

Example

G_2: $S(x_1 y_1 x_2 y_2)$:- $A(x_1, x_2), B(y_1, y_2)$.

$A(a_1 x a_2, a_3 y a_4)$:- $A(x, y)$. $A(\varepsilon, \varepsilon)$.

$B(b_1 x b_2, b_3 y b_4)$:- $B(x, y)$. $B(\varepsilon, \varepsilon)$.



Proof tree

$L(G_2) = \{ a_1^m a_2^m b_1^n b_2^n a_3^m a_4^m b_3^n b_4^n \mid m, n \geqq 0 \}$

Fixed vs. universal recognition

Recognition problem for ***fixed*** language L

Input: string w

Problem: $w \in L$?

Universal recognition problem (URP)

Input: grammar G, string w

Problem: $w \in L(G)$?

Example

Recognition problem for *fixed* language L:

For a string w, decide $w \in L$.

$O(|w|^3)$ time when L is CFL.

Universal recognition problem (URP):

For a grammar G & string w, decide $w \in L(G)$.

$O(|G|\,|w|^3)$ time & P-complete when G is CFG.

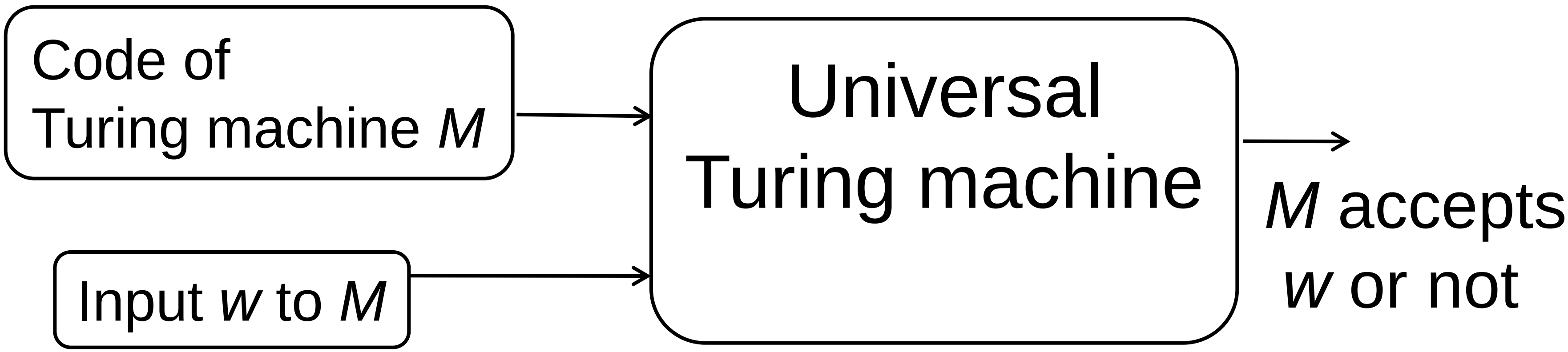
DEXP time-hard when G is GPSG†.

† GPSG (Generalized Phrase Structure Grammar[GKPS85])

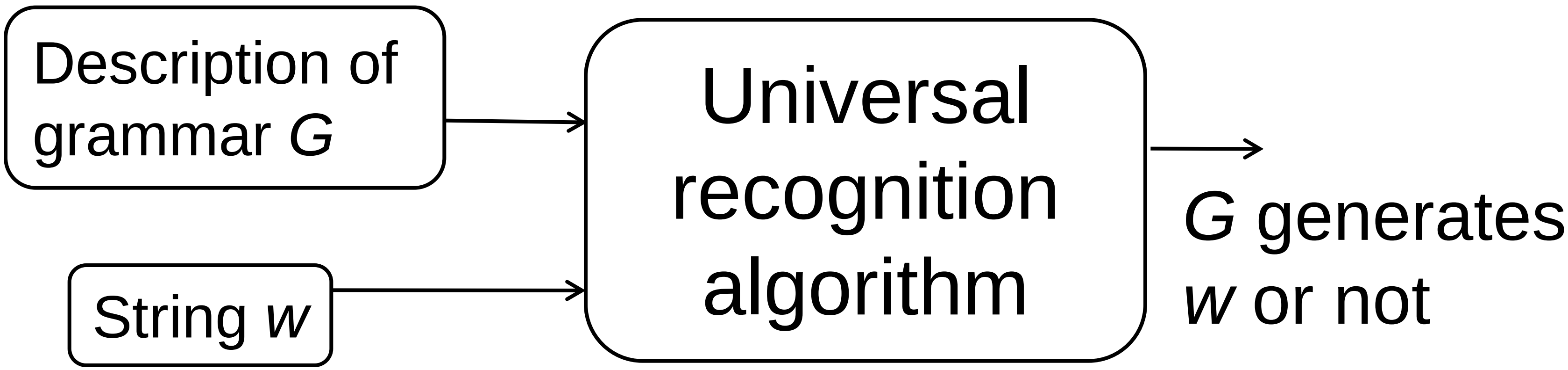
GPSG < CFG in generative power

⇒ Generative power ≠ Complexity of universal recognition

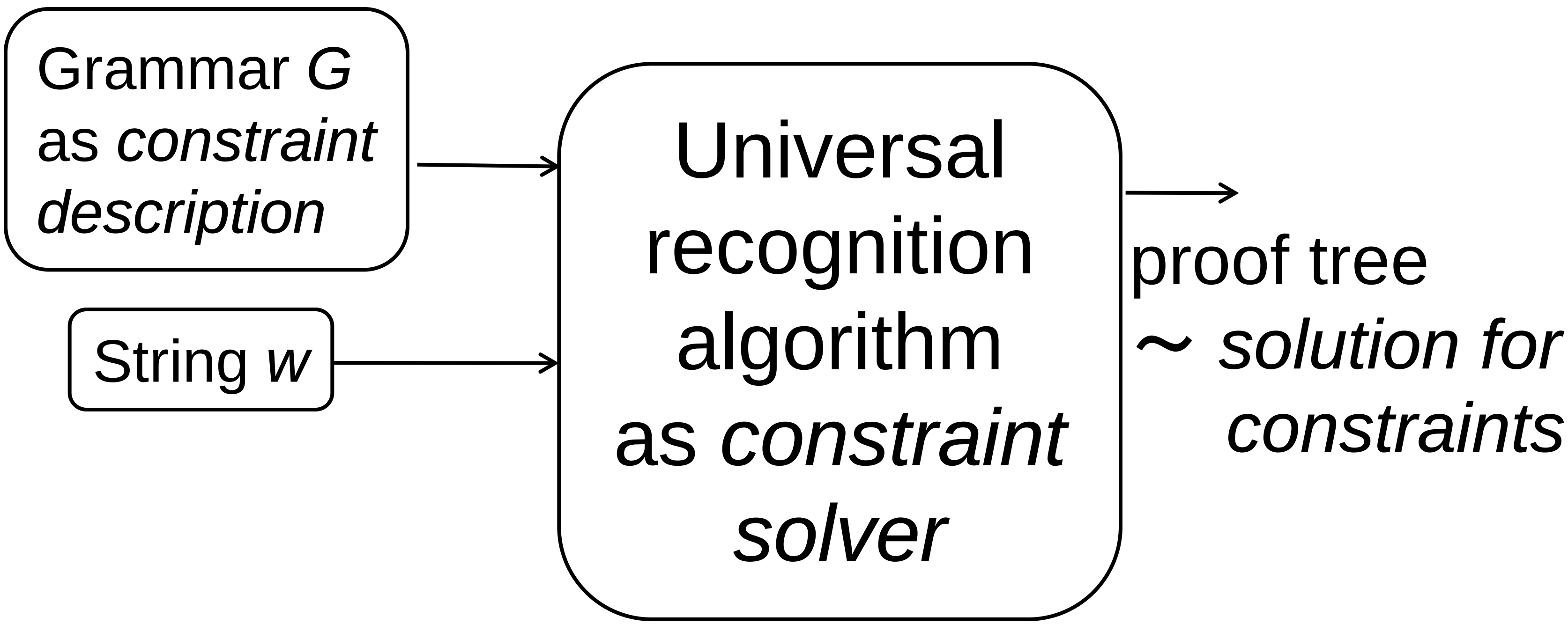
Why URP interesting?



Why URP interesting?

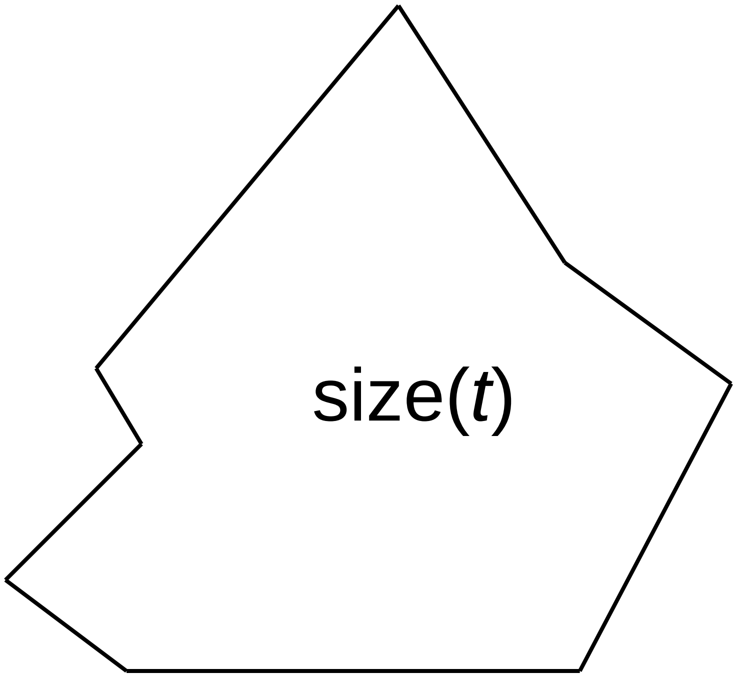


Application



E.g., profile-based structure prediction of biological sequence.   (*G:* structure profile,
w: biological sequence,
algorithm: structure prediction tool)

URP of MCFG and tree size



proof tree *t* of *w*

For a (proof) tree *t*, let
size(*t*) := *num. nodes* in *t*.

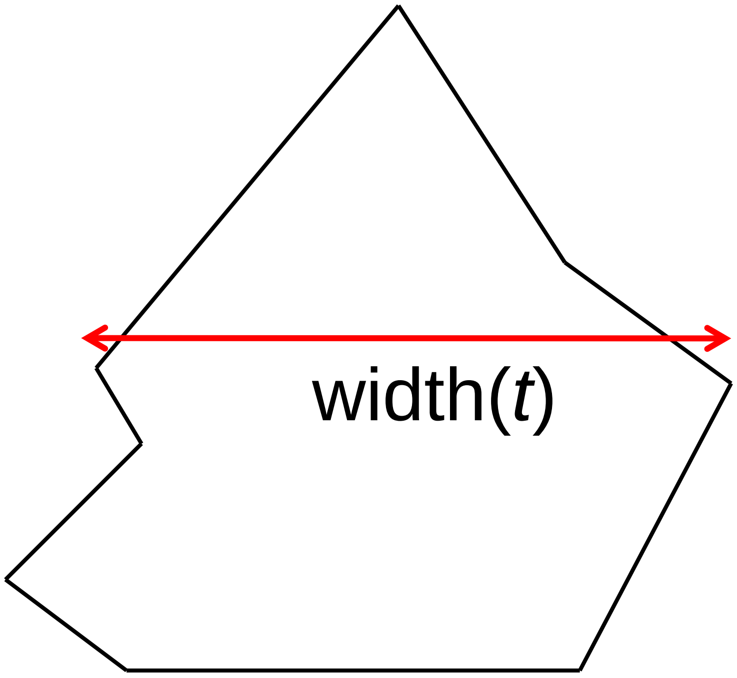
~ nondeterministic time complexity for URP.

$O(c_G|w|)$ in general

$O(|G|^2|w|)$ if dimension of G is bounded ⇒ in NP

where c_G: exponential to $|G|$.

URP for MCFG and tree width



proof tree *t* of *w*

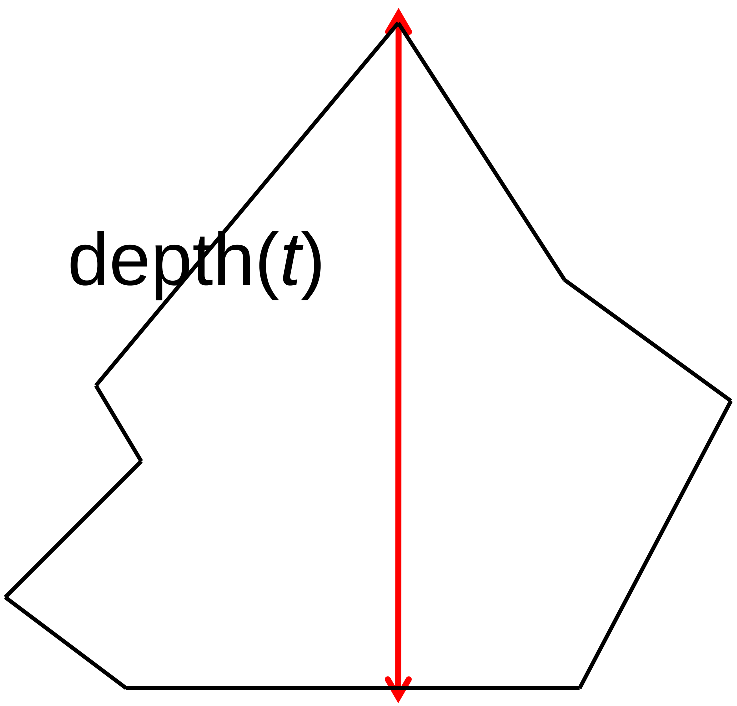
For a proof tree *t*, width(*t*) := *max length* of 'sentential form.'

∼ space complexity needed for URP

$O(|G|\cdot|w|)$ for non-deleting MCFG

⇒ in PSPACE.

URP for MCFG and tree depth



proof tree *t* of *w*

For a (proof) tree *t*,
depth(*t*) := *max length* of path from root to leaf in *t*.

Generally, depth(*t*) is exponential to |*G*| and |*w*|.

For a proof tree t, what follows if depth(t) is polynomial in |G| and |w|?

MCFG

$G=(N, T, V, P, S)$

N: predicates (nonterminals), T: terminals, V: variables,

P: rules, $S \in N$: start predicate

- A rule $\pi \in P$ is a definite Horn clause s.t.
  - Variables are linear in both head & body,
  - Argument of predicate in body is variable,
  - Argument of predicate in head is string over T and variables in its body.

(Ex) $A(axb, cyd)$:- $A(x, y)$.

$B(x_1y_1, x_2y_2)$:- $C(x_1, x_2), D(y_1, y_2)$.

NG $A(x, ax)$:- $B(x)$.

NG $A(x, y)$:- $B(x, y), C(x)$.

MCFG Language

For an MCFG $G=(N, T, V, P, S)$,

$$L(G) := \{ w \in T^* \mid S(w) \text{ is provable in } G \}$$

is the *multiple context-free language (mcfl)* generated by G.

Non-deleting MCFG

MCFG $G=(N, T, V, P, S)$

Non-deleting: $\forall \pi \in P$, $\mathrm{Var}_{\mathrm{head}}(\pi)=\mathrm{Var}_{\mathrm{body}}(\pi)$.

OK $B(x_1y, x_2)$:- $C(x_1, x_2), D(y)$.

NG $B(y, x_2)$:- $C(x_1, x_2), D(y)$.

Lemma: $\forall$ MCFG G,

$\exists$ non-deleting MCFG G' s.t. $L(G') = L(G)$.

Dimension & rank

MCFG $G=(N, T, V, P, S)$

- rule π: $A_0(\ldots)$:- $A_1(\ldots), A_2(\ldots), \ldots, A_n(\ldots)$.
  dim(π) := max { arity(A_i) | $0 \leqq i \leqq n$}, rank(π) := n.

(Ex)

π_1: $A(axb, cyd)$:- $A(x, y)$.

dim(π_1)=2, rank(π_1)=1.

π_2: $B(x_1y_1, x_2y_2)$:- $C(x_1, x_2)$, D(y_1, y_2).

dim(π_2)=2, rank(π_2)=2.

Dimension & rank

For MCFG $G=(N, T, V, P, S)$,

$\{\text{dim}, \text{rank}\}(G) := \max_{\pi \in P} \{\text{dim}, \text{rank}\}(\pi)$

- q-MCFG(r): MCFG with dim(・)$\leqq q$ and rank(・)$\leqq r$
- q-MCFG: MCFG with dim(・)$\leqq q$
- MCFG(r): MCFG with rank(・)$\leqq r$

URP for MCFG

• General case	DEXP-complete
• Non-deleting MCFG	PSPACE-complete
• *q*-MCFG with fixed *q*	NP-complete
• *q*-MCFG(*r*) with fixed *q*, *r*	P-complete

[KNSK94] Kaji, Nakanishi, Seki & Kasami, Computational Intelligence, 10(4), 440-452, 1994.

Depth-bounded MCFG

- MCFG G is $f(m, n)$ *depth-bounded* if

  $\forall\ w \in L(G)$, $\exists$ proof tree t of w such that depth(t) $\leqq f(m, n)$ where $m=|G|$, $n=|w|$.

- MCFG G is *poly depth-bounded* if

  $\exists$ polynomial p such that G is $p(m, n)$ depth-bounded.

Main result

Theorem: URP for poly depth-bounded MCFG
is PSPACE-complete.

Corollary: URP for poly depth-bounded MCFG(1)
is NP-complete.

PSPACE solvability

Lemma 1: URP for poly depth-bounded MCFG is in PSPACE.

Proof outline: Give a non-deterministic algorithm of poly space complexity that decides $w \in L(G)$ for given poly depth-bounded MCFG G and $w \in T^*$.

A normal form

From given MCFG G =(N, T, V, P, S), we can construct MCFG G' satisfying $L(G')=L(G)$, G' contains no useless symbols/rules and for $\forall A \in N$,

(1) Rule A(…) :- … is at most one, or

(2) Every rule A(…) :- … is rank 1.

Proof.

A(…) :- B(…), C(…).

A(…) :- D(…), E(…).

↓

A(…) :- A'(…). A(…) :- A''(…). Type (2)

A'(…) :- B(…), C(…). Type (1)

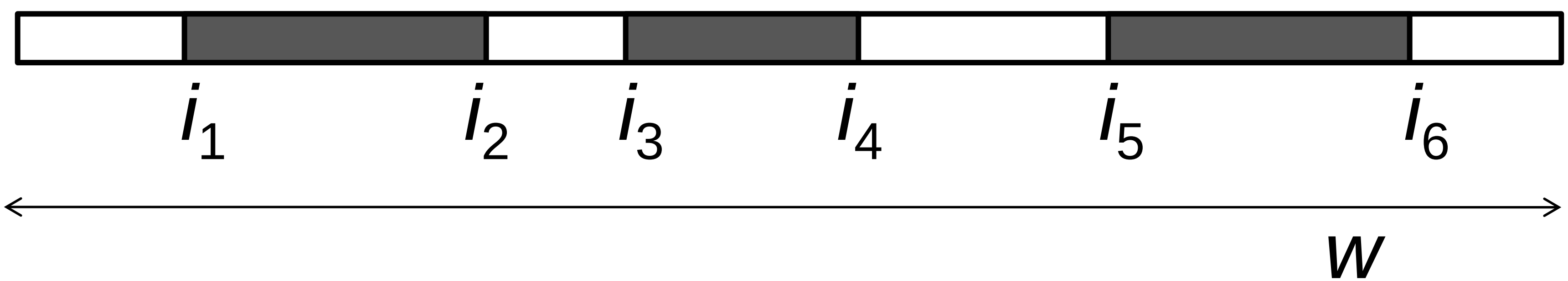
A''(…) :- D(…), E(…). 〃

The construction preserves poly depth-boundedness.

Algorithm REC solving URP

Input: $p(m,n)$ depth-bounded MCFG $G=(N, T, V, P, S)$ in normal form and $w \in T^*$.

For $A \in N$, $1 \leqq i_j \leqq |w|$, $\mathrm{REC}(A, i_1, \ldots, i_{2*\mathrm{arity}(A)})$ = true iff $A(w[i_1, i_2], \ldots, w[i_{2*\mathrm{arity}(A)-1}, i_{2*\mathrm{arity}(A)}])$ is provable.



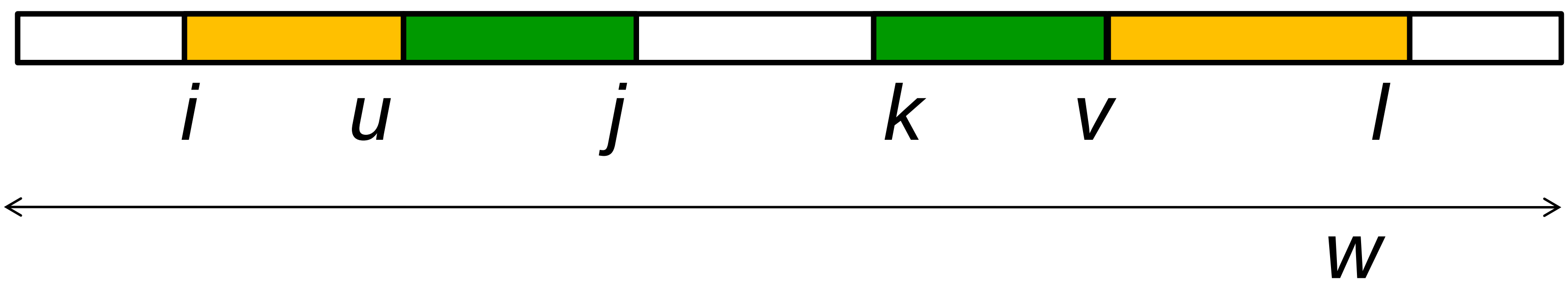
Rank≧2 rule

For $A(y_1z_2, z_1y_2)$:- $B(y_1, y_2)$, $C(z_1, z_2)$,

REC(A, i, j, k, l)

choose u, v ($i \leqq u < j$, $k \leqq v < l$) ;

return REC(B, i, u, v+1, l) ∧ REC(C, u+1, j, k, v) ;



Rank 1 rule

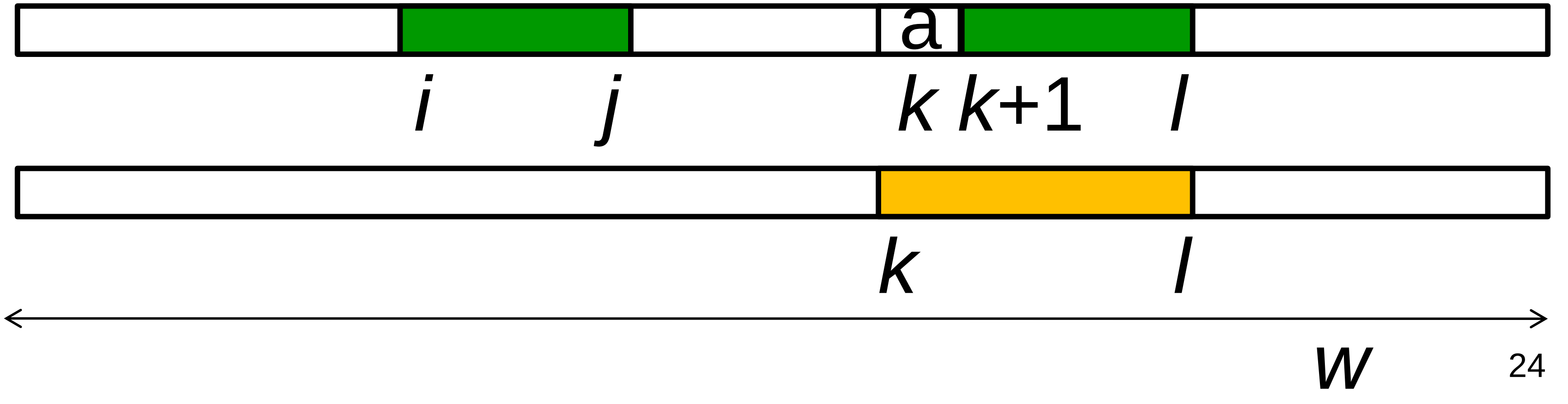
For $A(y_1, ay_2)$:- $B(y_1, y_2)$. $A(\varepsilon, y_2)$:- $C(y_1, y_2)$.,

REC(A, i, j, k, l)

choose

return (REC(B, i, j, k+1, l) $\wedge$ $w[k]=a$) or

return REC(C, *, *, k, l) ;



PSPACE solvability revisited

Lemma 1: URP for poly depth-bounded MCFG is in PSPACE.

Proof outline: By assumption, G is $p(m, n)$ depth-bounded where $m=|G|$ and $n=|w|$.

- Space needed for one instance for REC is $O(m^2 \log n)$.
- Recursion depth of REC can be $p(m, n)$.

⇒ REC is $O(p(m, n)m^2 \log n)$ space bounded.

PSPACE solvability revisited

Lemma 1: URP for poly depth-bounded MCFG is in PSPACE.

The algorithm REC can be expressed by poly time bounded alternating Turing machine.

Alternating Turing machine

ATM is $M = (Q, \Sigma, \Gamma, B, \delta, q_s, Q_F, Q_U, Q_E)$

$Q = Q_U \cup Q_E \cup Q_F \cup \{q_s\}$: state set

Σ ($\subseteq \Gamma$ - {B}): input symbols

Γ: tape symbols   B: blank symbol

$\delta \subseteq (Q \times \Gamma) \times (Q \times \Gamma \times \{\rightarrow, \leftarrow\})$: transition relation

q_s: initial state Q_F: final (accepting) states

Q_U: set of ***universal*** states

Q_E: set of ***existential*** states

Alternating Turing machine

ATM $M = (Q, \Sigma, \Gamma, B, \delta, q_s, Q_F, Q_U, Q_E)$

ID (instantaneous description) is $\alpha q \beta$ where $q \in Q$ & $\alpha, \beta \in \Gamma^*$.

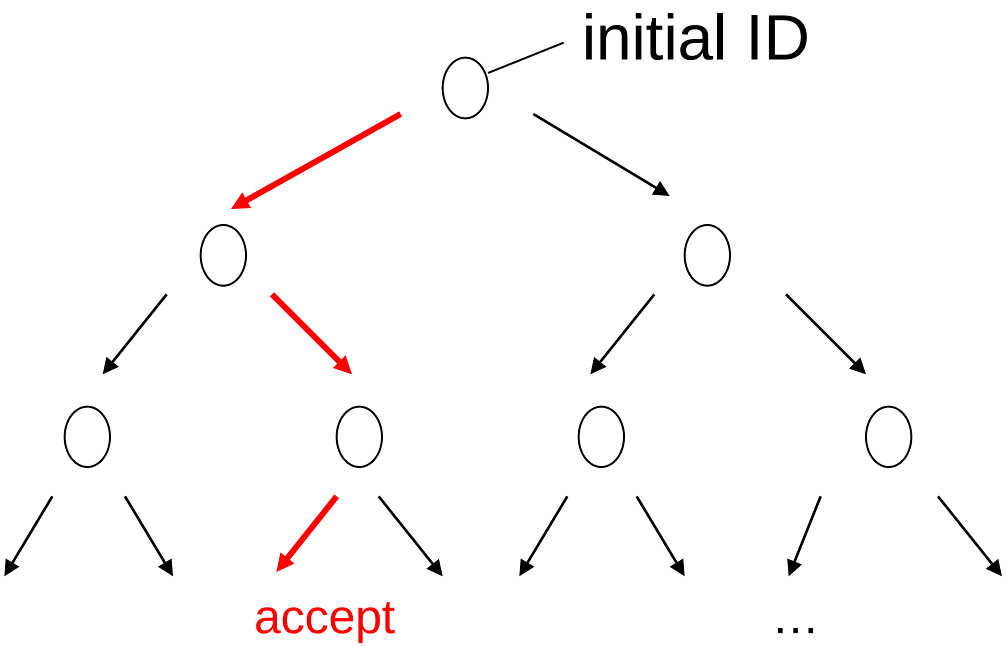
Move relation $\alpha q \beta \rightarrow_M \alpha' q' \beta'$ over IDs is defined in a usual way.

For an input w, ID $q_S w$ is the <u>initial</u> ID for w.

ID $\alpha q \beta$ is <u>accepting</u> if $q \in Q_F$.

Nondeterministic Turing machine

Input *w* is accepted if there is a run that reaches an accepting ID from the initial ID for *w*.

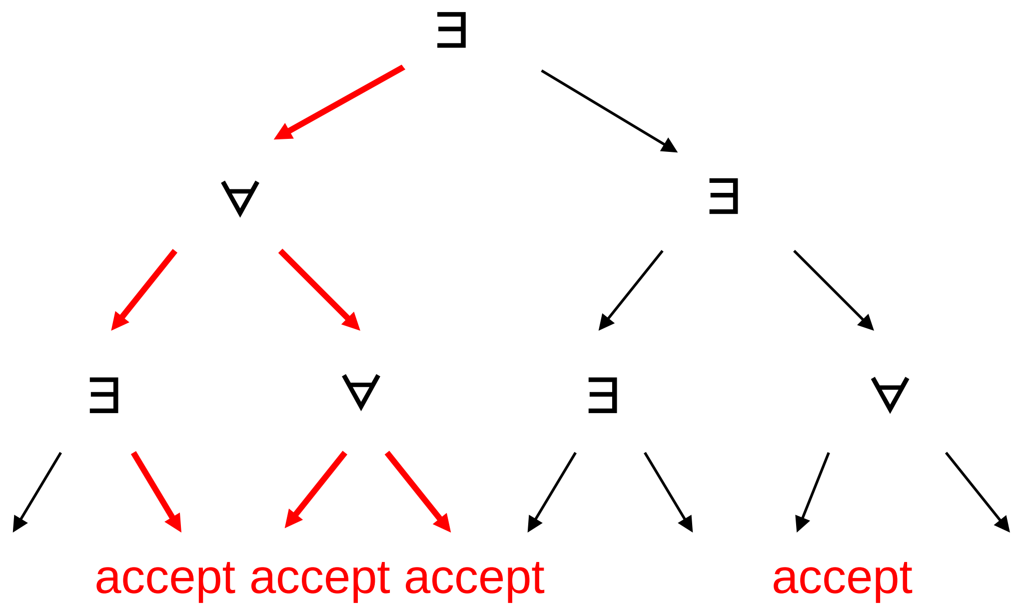


Alternating Turing machine

For $q \in Q_F$, ID $\alpha q \beta$ (trivially) is accepting.

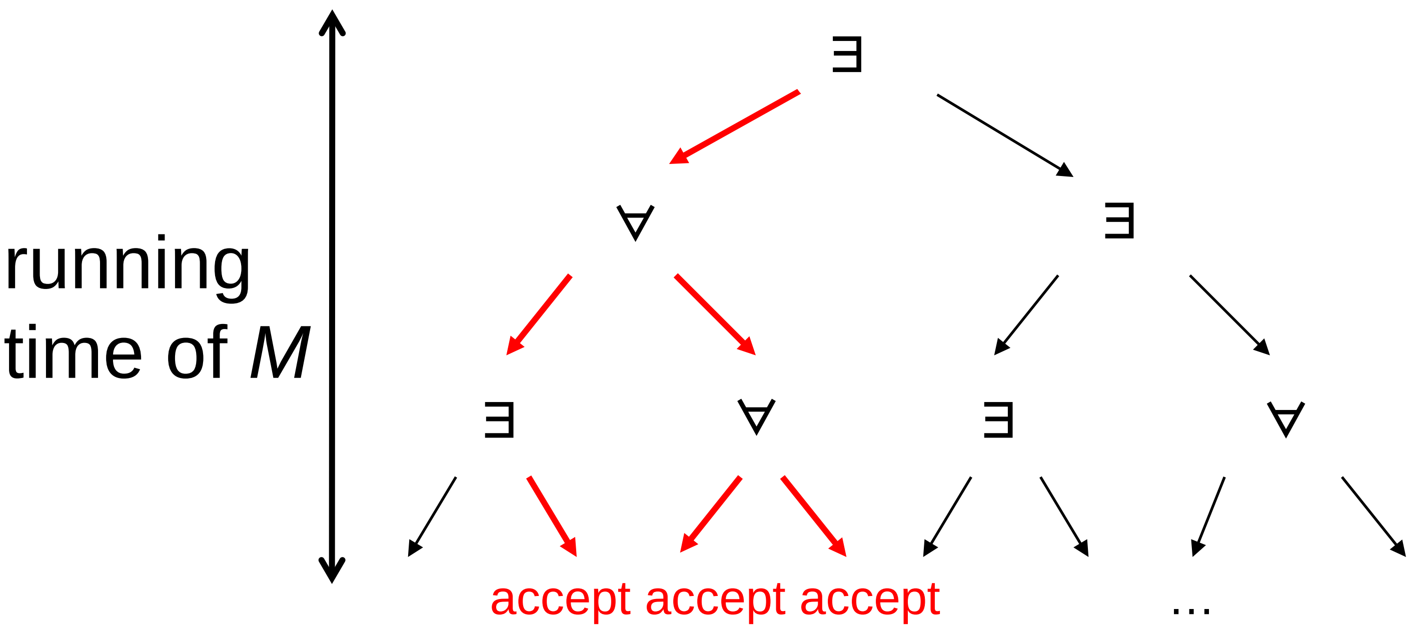
For $q \in Q_E$, ID $\alpha q \beta$ is accepting if *there is* ID' such that ID $\rightarrow$ ID' & ID' is accepting.

For $q \in Q_U$, ID $\alpha q \beta$ is accepting if *every* ID' satisfying ID $\rightarrow$ ID' is accepting.



Accepting language

$L(M) = \{ w \mid \text{Initial ID } q_S w \text{ is accepting} \}$



TM and ATM

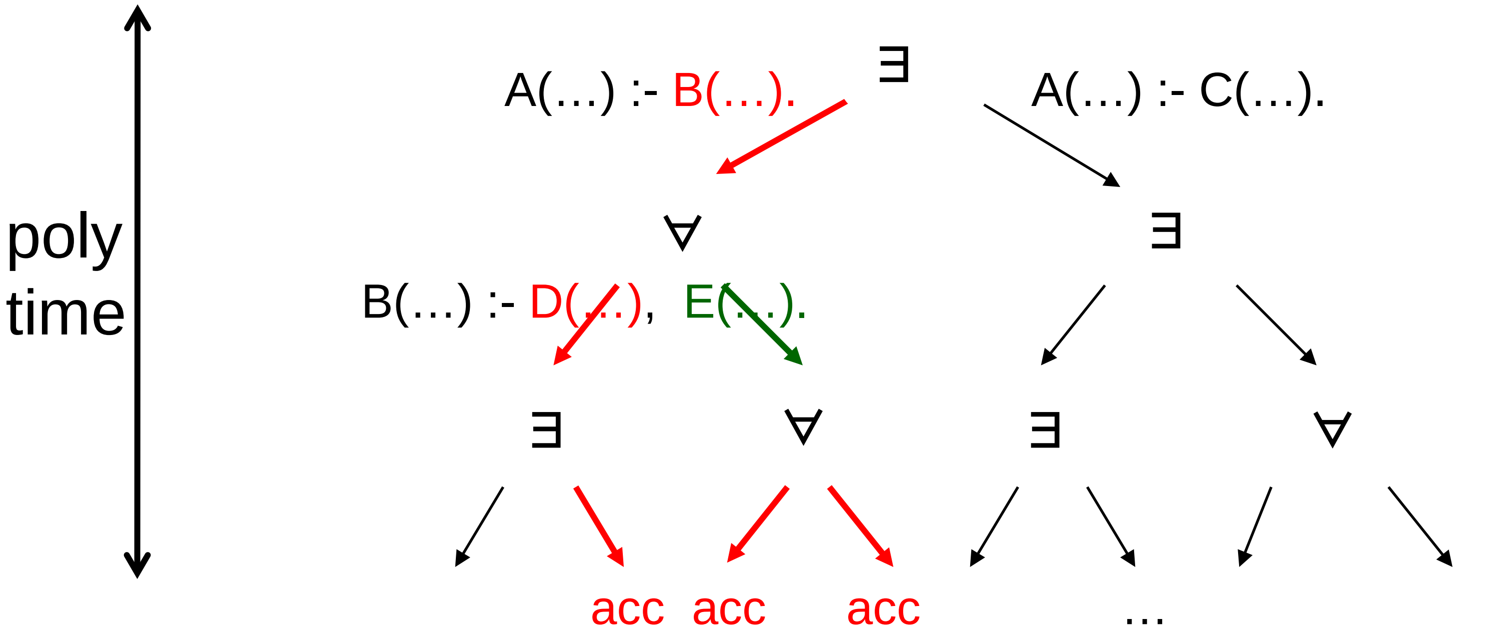
Propositon[CS80]:

PSPACE=APTIME, DEXPTIME=APSPACE

APTIME := class of problems solvable by poly time-bounded ATM

APSPACE := class of problems solvable by poly space-bounded ATM

Recognition by APTIME



PSPACE-hardness

Lemma 2: URP for poly depth-bounded MCFG is PSPACE-hard.

Proposition: Q3SAT (Quantified 3CNF satisfiability) is PSPACE-complete.

Proof outline. Give a poly time reduction from Q3SAT to URP.

Quantified Boolean formula (QBF)

- $\exists \boldsymbol{x}[E] \quad = E_0 \vee E_1$
- $\forall \boldsymbol{x}[E] \quad = E_0 \wedge E_1$

where E_0/E_1 are formulae obtained by replacing (free) x with 0/1.

- $\forall \boldsymbol{x} \exists \boldsymbol{y} [\boldsymbol{x} \vee \boldsymbol{y}]$

  $= \exists \boldsymbol{y} [0 \vee \boldsymbol{y}] \wedge \exists \boldsymbol{y} [1 \vee \boldsymbol{y}]$

  $=((0 \vee 0) \vee (0 \vee 1)) \wedge ((1 \vee 0) \vee (1 \vee 1))$

  $=1 \wedge 1$

  $=1$

Q3CNF: $Q_1 \boldsymbol{x_1} Q_2 \boldsymbol{x_2} \ldots Q_n \boldsymbol{x_n} F$

where $Q_k = \exists$ or $\forall$ $(1 \leqq k \leqq i)$, F is 3CNF (3-conjunctive normal form, $\wedge$'s of $\vee$'s of 3 literals).

Q3SAT

Problem Q3SAT (Q3CNF satisfiability)

Decide whether a given quantified 3CNF

$Q_1 \boldsymbol{x}_1 Q_2 \boldsymbol{x}_2 \ldots Q_n \boldsymbol{x}_n \, F$

is true or not.

Proposition: Q3SAT is PSPACE-compete.

Q3SAT $\leqq_P$ URP

$F \equiv Q_{\mathbf{1}}\boldsymbol{x}_{\mathbf{1}}Q_{\mathbf{2}}\boldsymbol{x}_{\mathbf{2}}\ldots Q_{\boldsymbol{n}}\boldsymbol{x}_{\boldsymbol{n}}\ F_1 \wedge F_2 \wedge \ldots \wedge F_q$: Q3CNF.

Construct poly depth-bounded MCFG $G=(N, T, V, P, S)$ such that

$F = 1$ if and only if $\varepsilon \in L(G)$.

ε encodes 1(true) and # encodes 0(false).

Example

$\forall \boldsymbol{x}_1 \exists \boldsymbol{x}_2(\boldsymbol{x}_1 \vee \neg \boldsymbol{x}_2)$

$= \exists \boldsymbol{x}_2(0 \vee \neg \boldsymbol{x}_2) \wedge \exists \boldsymbol{x}_2(1 \vee \neg \boldsymbol{x}_2)$

$= \exists \boldsymbol{x}_2(\neg \boldsymbol{x}_2) \wedge 1$

$= 1 \vee 0$

$= 1$

Example

$\forall \boldsymbol{x}_1 \exists \boldsymbol{x}_2(\boldsymbol{x}_1 \vee \neg \boldsymbol{x}_2)$

$= \exists \boldsymbol{x}_2(0 \vee \neg \boldsymbol{x}_2) \wedge \exists$

$= \exists \boldsymbol{x}_2(\neg \boldsymbol{x}_2) \wedge 1$

We can make the formula true when $\boldsymbol{x}_1$:=1 regardless the values of the other variables.

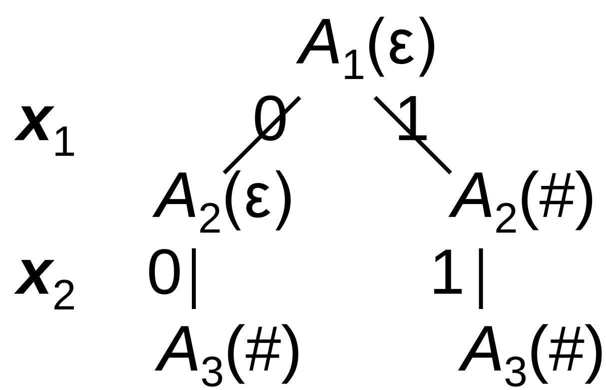
$\boldsymbol{x}_1$:=0 $\boldsymbol{x}_1$:=1

$A_1(y \cdot \varepsilon)$:- $A_2(y)$, $A_2(z)$.

$A_2(\varepsilon)$ $A_3(y)$. $A_2(z)$:- $A_3(z)$.

$A_3(\#)$

Whether the formula becomes true depends on the values of the other variables when $\boldsymbol{x}_1$:=0.



Q3SAT$\leqq_P$ URP

$F \equiv Q_{\mathbf{1}}\boldsymbol{x}_{\mathbf{1}}Q_{\mathbf{2}}\boldsymbol{x}_{\mathbf{2}}\ldots Q_{\boldsymbol{n}}\boldsymbol{x}_{\boldsymbol{n}}\ F_1 \wedge F_2 \wedge \ldots \wedge F_q$: Q3CNF

Construct poly depth-bounded MCFG $G=(N, \{\#\}, T, V, P, S)$ such that $F = 1$ if and only if $\varepsilon \in L(G)$ as follows. Invariant:

$A_k(\ldots, \varepsilon, \ldots)$ iff F_j can be made 1 for some $\boldsymbol{x}_{\boldsymbol{k}}, \ldots, \boldsymbol{x}_{\boldsymbol{n}}$.

j

$N = \{ S, A_1, A_2, \ldots, A_{n+1} \}$ with $\dim(A_j)=q$

P consists of:

- $S(y_1 \cdot \ldots \cdot y_q)$:- $A_1(y_1, \ldots, y_q)$.

Q3SAT $\leqq_P$ URP

- For $1 \leqq k \leqq n$,

  if $Q_k = \exists$, $\boldsymbol{x_k}$:=0

  $A_k(\alpha_1, \ldots, \alpha_q)$:- $A_{k+1}(y_1, \ldots, y_q)$.

  $A_k(\beta_1, \ldots, \beta_q)$:- $A_{k+1}(z_1, \ldots, z_q)$.

  if $Q_k = \forall$, $\boldsymbol{x_k}$:=1

  $A_k(\alpha_1\, \beta_1, \ldots, \alpha_q\, \beta_q)$:- $A_{k+1}(y_1, \ldots, y_q), A_{k+1}(z_1, \ldots, z_q)$.

  $\boldsymbol{x_k}$:=0 where $\alpha_j = \varepsilon$ if F_j contains $\neg \boldsymbol{x}_k$ and $\alpha_j = y_j$ otherwise

  $\boldsymbol{x_k}$:=1 and $\beta_j = \varepsilon$ if F_j contains $\boldsymbol{x_k}$ and $\beta_j = z_j$ otherwise.

- $A_{n+1}(\#, \ldots, \#)$.

Example

$\forall \boldsymbol{x}_1 \exists \boldsymbol{x}_2 \forall \boldsymbol{x}_3 \exists \boldsymbol{x}_4 (\boldsymbol{x}_1 \vee \neg \boldsymbol{x}_2 \vee \boldsymbol{x}_4)(\boldsymbol{x}_1 \vee \boldsymbol{x}_3 \vee \neg \boldsymbol{x}_4)$

$= \exists \boldsymbol{x}_2 \forall \boldsymbol{x}_3 \exists \boldsymbol{x}_4 (\neg \boldsymbol{x}_2 \vee \boldsymbol{x}_4)(\boldsymbol{x}_3 \vee \neg \boldsymbol{x}_4) \wedge 1$

$= \forall \boldsymbol{x}_3 \exists \boldsymbol{x}_4 (\boldsymbol{x}_3 \vee \neg \boldsymbol{x}_4) \vee \forall \boldsymbol{x}_3 \exists \boldsymbol{x}_4 (\boldsymbol{x}_4)(\boldsymbol{x}_3 \vee \neg \boldsymbol{x}_4)$

$= (\exists \boldsymbol{x}_4 (\neg \boldsymbol{x}_4) \wedge 1) \vee (\exists \boldsymbol{x}_4 (\boldsymbol{x}_4)(\neg \boldsymbol{x}_4) \wedge \exists \boldsymbol{x}_4 (\boldsymbol{x}_4))$

$= 1 \vee 0$

$= 1$

Reduction to Q3CNF

Goal:

$A_k(\ldots, \varepsilon, \ldots)$ iff F_j can be made 1 for some $\boldsymbol{x_k}, \ldots, \boldsymbol{x_n}$.

j

$\forall \boldsymbol{x}_1 \exists \boldsymbol{x}_2 \forall \boldsymbol{x}_3 \exists \boldsymbol{x}_4 (\boldsymbol{x}_1 \vee \neg \boldsymbol{x}_2 \vee \boldsymbol{x}_4)(\boldsymbol{x}_1 \vee \boldsymbol{x}_3 \vee \neg \boldsymbol{x}_4)$

$S(y_1 y_2)$:- $A_1(y_1, y_2)$.

$A_1(y_1\, \varepsilon, y_2\, \varepsilon)$:- $A_2(y_1, y_2), A_2(z_1, z_2)$.

$A_2(\varepsilon, y_2)$:- $A_3(y_1, y_2)$. $A_2(z_1, z_2)$:- $A_3(z_1, z_2)$.

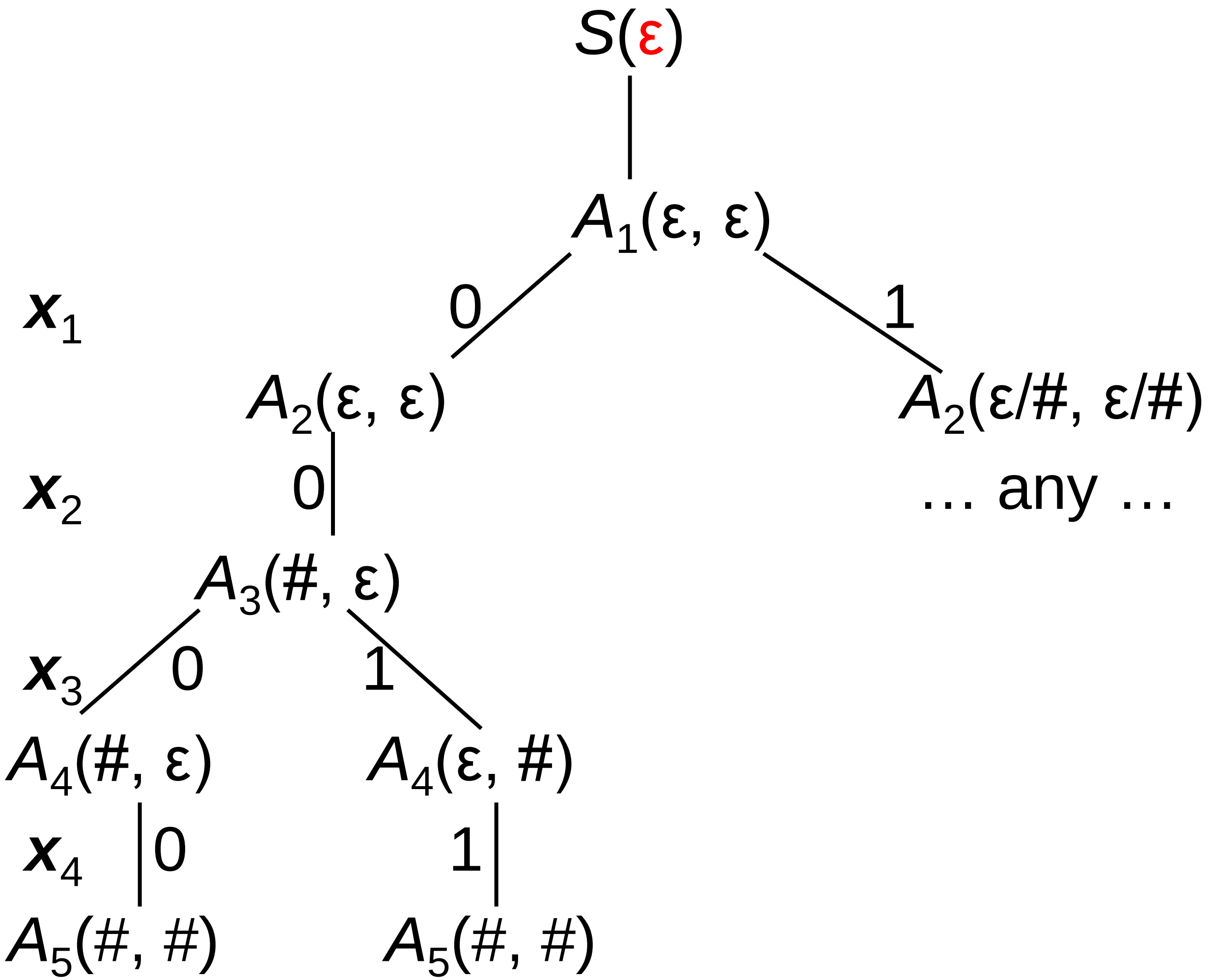
$A_3(y_1 z_1, y_2\, \varepsilon)$:- $A_4(y_1, y_2), A_4(z_1, z_2)$.

$A_4(y_1, \varepsilon)$:- $A_5(y_1, y_2)$. $A_4(\varepsilon, z_2)$:- $A_5(z_1, z_2)$.

$A_5(\#, \#)$. (ε encodes 1 and # encodes 0.)

Reduction to Q3CNF

$$\forall \boldsymbol{x}_1 \exists \boldsymbol{x}_2 \forall \boldsymbol{x}_3 \exists \boldsymbol{x}_4 (\boldsymbol{x}_1 \lor \neg \boldsymbol{x}_2 \lor \boldsymbol{x}_4)(\boldsymbol{x}_1 \lor \boldsymbol{x}_3 \lor \neg \boldsymbol{x}_4)$$



Summary

non-deletion	dimension	depth	Universal recognition
-	-	-	DEXP-complete
required	-	-	PSPACE-complete*
-	bounded	-	NP-complete
-	-	poly	PSPACE-complete
-	-	poly, (*i*-1)-bounded alternation	Σ_i/π_i-complete
-	-	poly, rank 1	NP-complete

[KNSK94] (first three rows)

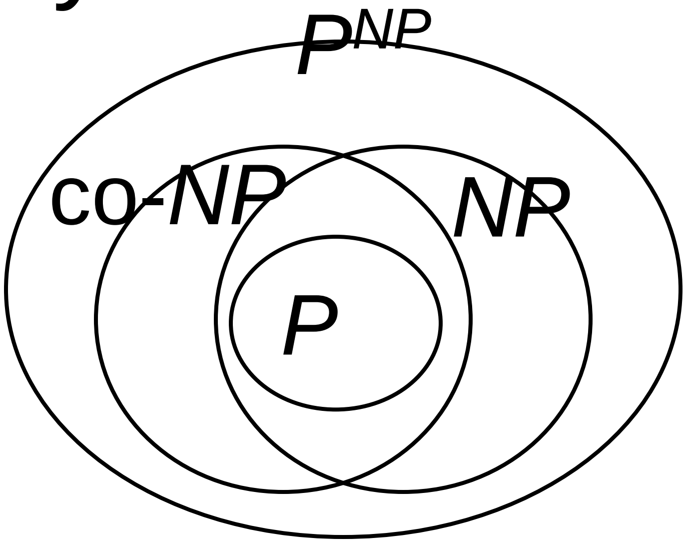
This talk (last three rows)

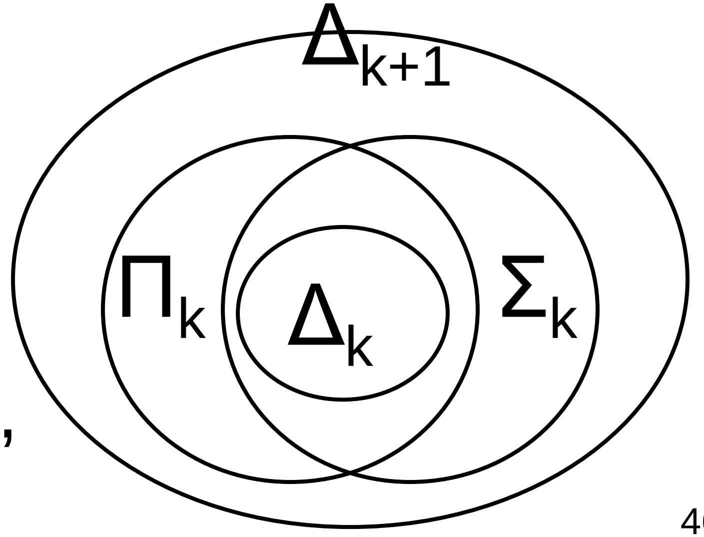
*Proof is more complicated since non-deletion is assumed.

Polynomial hierarchy

Let C^X denote complexity class C with oracle X.

- $\Sigma_0 = \Pi_0 = \Delta_0 = P$
- For $k \geqq 0$,
  - $\Delta_{k+1} = P^{\Sigma k}$
  - $\Sigma_{k+1} = NP^{\Sigma k}$
  - $\Pi_{k+1} = \text{co-}\Sigma_{k+1}$





By definition, $\Delta_1 = P$, $\Sigma_1 = NP$, $\Pi_1 = \text{co-}NP$.

Complete problem for Σ_k

Problem Σ_k3SAT

Decide whether a given Q3CNF with *k*-1 alternations

$$\exists \boldsymbol{x_1} \forall \boldsymbol{x_2} \ldots Q \boldsymbol{x_k} \, F$$

is true or not.

(Q=∃ if k is odd and Q=∀ if *k* is even, and *F* is 3CNF.)

Property: Σ_k3SAT is Σ_k-complete.

Conclusion

URP for poly depth-bounded MCFG was shown to be PSPACE-complete.

Future work:

- Find a syntactical (or decidable) characterization of being poly depth-bounded.
- Complexity of URP for well-nested MCFG.