EQUIVALENCE

$$TM \leq RM \leq FRS \leq REC \leq TM$$

UNARY ALPHABET WITH 0 AS BLANK

REPRESENTING WORDS OVER LARGER ALPHABETS

$\Sigma = \{a, b, c\}$

WORD = $acab$

$0\underbrace{0101110101100}0$

00 SEPARATES WORDS

THUS, WE CAN FOCUS ON TAPE ALPHABET
OR $\{1\}$ WITH BLANK AS 0.

Encoding TM Instantaneous Description

String Approach

$$\ldots 00\underbrace{1010011}\,q_7\,\underbrace{\underline{0}1}\,00\ldots$$

$$1010011\,q_7\,01$$

Record shortest string on right that includes scanned square as rightmost non-blank

Record shortest string on left that includes leftmost non-blank

Place state # to left of scanned square

Integer Approach

$(2, 83, 7)$ for $1010011\,q_7\,01$

Right read R to L

Left read L to R

State inde

Note:

If first number is even, scanned square is 0; if odd, then 1.
Same for rightmost symbol on left

TM ≤ REGISTER MACHINE

CAN STORE TM ID IN JUST THREE REGISTERS

CAN SHIFT LEFT VIA MULTIPLY BY 2

ASSUME $r_2 = 0$; $r_3 = 0$

x. $DEC_{r_1}(x+1, x+4)$
$x+1$. $INC\ r_2\ (x+2)$
$x+2$. $INC\ r_2\ (x+3)$
$x+3$. $INC\ r_3\ (x)$

$r_2 = r_1 * 2$
$r_3 = r_1$
$r_1 = 0$

$x+4$. $DEC_{r_3}(x+5, x+6)$
$x+5$. $INC_{r_1}\ (x+4)$

$r_1 = r_3$
$r_3 = 0$

$x+6$.

CAN SHIFT RIGHT VIA DIVIDE BY 2

DETAILS OF TM ≤ RM ON NOTES 488-494

RM ≤ FRS

ID FOR RM IS

$$P_1^{r_1} P_2^{r_2} \cdots P_n^{r_n} P_{n+j}$$

WHERE r_k IS CONTENTS OF REGISTER k AND WE ARE ABOUT TO EXECUTE INSTR. j.

CAN SIMULATE BY

j. $\text{INCR}_r[i]$

$$P_{n+j}\, x \rightarrow P_{n+i} P_r\, x$$

j. $\text{DEC}_r[s, f]$

$$P_{n+j} P_r\, x \rightarrow P_{n+s}\, x$$

$$P_{n+j}\, x \rightarrow P_{n+f}\, x$$

ALSO

$$P_{n+m+1}\, x \rightarrow x$$

FOR HALTING CONDITION

DETAILS ON PAGES 495–501

RECURSIVE ≤ TURING

SHOW BASE FUNCTIONS ARE TURING COMPUTABLE

$C_a^n(x_1,\ldots,x_n) = a$

$(R\,1)^a R$

$I_i^n(x_1,\ldots,x_n) = x_i$

C_{n-i+1}

$S(x) = x+1$ $\quad C_1\, 1\, R$

NOW SHOW TURING COMPUTABLE CLOSED UNDER COMPOSITION, INDUCTION AND MINIMIZATION

DETAILS ON NOTES PAGES 511–518

UNIVERSAL MACHINE

REALLY AN INTERPRETER FOR PROGRAMS IN SOME MODEL OF COMPUTATION, WRITTEN IN THAT MODEL

$$\mathrm{Univ}(x,y) = \varphi_x(y)$$

WHERE φ_x IS X-TH PROGRAM IN SOME WAY OF ORDERING PROGRAMS, E.G., LEXICALLY.

$$\varphi(x,y) = \mathrm{Univ}(x,y)$$

HALTING PROBLEM

<u>ASSUME</u> THERE EXISTS AN ALGORITHM HALT

SUCH THAT $HALT(x, y) = 1$ IF $\varphi_x(y)\downarrow$

$= 0$ OTHERWISE

DEFINE DISAGREE BY

$$DISAGREE(x) = \mu y[HALT(x,x) == 0]$$

NOTE: IF $HALT(x,x) = 0$ THEN

$DISAGREE(x) = 0$

IF $HALT(x,x) = 1$ THEN

$DISAGREE(x)\uparrow$ (DIVERGES)

AS DISAGREE IS CLEARLY A μ-RECURSIVE FUNCTION, IT IS A COMPUTABLE FUNCTION AND IT HAS AN INDEX, CALL IT d. $\varphi_d = D$

BUT $D(d)\downarrow$ IFF $HALT(d,d) = 0$

IFF $\varphi_d(d)\uparrow$

IFF $D(d)\uparrow$

CONTRADICTION. THUS, HALT CANNOT EXIST

ENUMERATION THEOREM

DEFINE

$$W_n = \{x \in \mathbb{N} \mid \Phi(n,x)\downarrow\}$$

φ_n SEMI-DECIDES W_n SO

THEOREM: A SET B IS RE IFF $\exists n$ SUCH THAT $B = W_n$

THIS ALLOWS US TO ENUMERATE THE RE (SEMI-DECIDABLE) SETS

UNIFORM HALTING PROBLEM

AKA TOTAL φ_f f

$$\text{TOTAL} = \{ f \mid \varphi_f \text{ IS AN ALGORITHM} \}$$
$$= \{ f \mid \forall x\, \varphi_f(x) \downarrow \} \qquad \forall x\, \varphi_f(x)\downarrow$$
$$\forall x\; f(x)\downarrow$$

ASSUME TOTAL IS RE AND A IS AN ENUMATING ALGORITHM

$A(x) = A_x$

WHERE $A_0, A_1, A_2, \ldots$ IS LIST OF INDICES OF ALL AND ONLY ~~THE~~ ALGORITHMS

DEFINE $G(x) = \text{UNIV}(A(x), x) + 1$
$$= \varphi_{A(x)}(x) + 1$$

BUT THEN G IS AN ALGORITHM, SAY THE g-TH ONE

THAT IS, $A(g) = A_g = G$

THUS, $G(g) = \varphi_{A(g)}(g) + 1 = G(g) + 1$

BUT THAT IS A CONTRADICTION SINCE G IS AN ALGORITHM

NOTE: IF G WERE A PROCEDURE, THIS IS NOT NEC. A CONTRADICTION — WHY?

MORE ON TOTAL

$$\text{TOTAL} = \{ f \in \mathbb{N} \mid \forall x\ \varphi_f(x)\downarrow \}$$
$$= \{ f \in \mathbb{N} \mid \omega_f = \mathbb{N} \}$$

OUR RESULT THAT TOTAL IS NOT RE MEANS THERE IS NO COMPLETE MODEL OF COMPUTATION THAT DOES NOT INCLUDE PROCEDURES THAT ARE NOT ALGORITMS.

THAT IS, NO GENERATIVE SYSTEM (E.G., GRAMMAR) CAN PRODUCE DESCRIPTIONS OF ALL AND ONLY ALGORITHMS

<u>AND</u>

NO PARSING SYSTEM CAN ACCEPT ALL AND ONLY ALGORITHMS

THAT IS, REAL COMPUTER LANGUAGES MUST SUPPORT INFINITE COMPUTATION (LOOPS)

WAYS TO DIVERGE

WHILE LOOPS

GOTO'S AS IN TMs AND RMs

MINIMIZATION AS IN REC FUNCTIONS

FIXED POINT AS IN FRS

NON-DETERMINISM IN MODELS

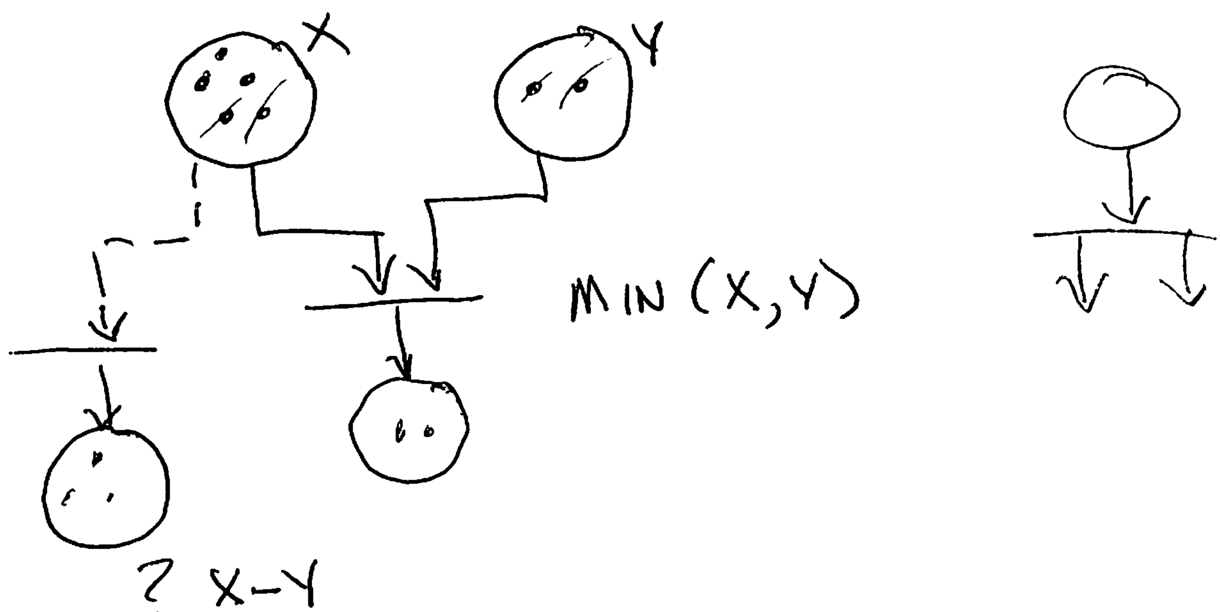
HELPS FOR PDAs

DOESN'T HELP OR HURT FOR

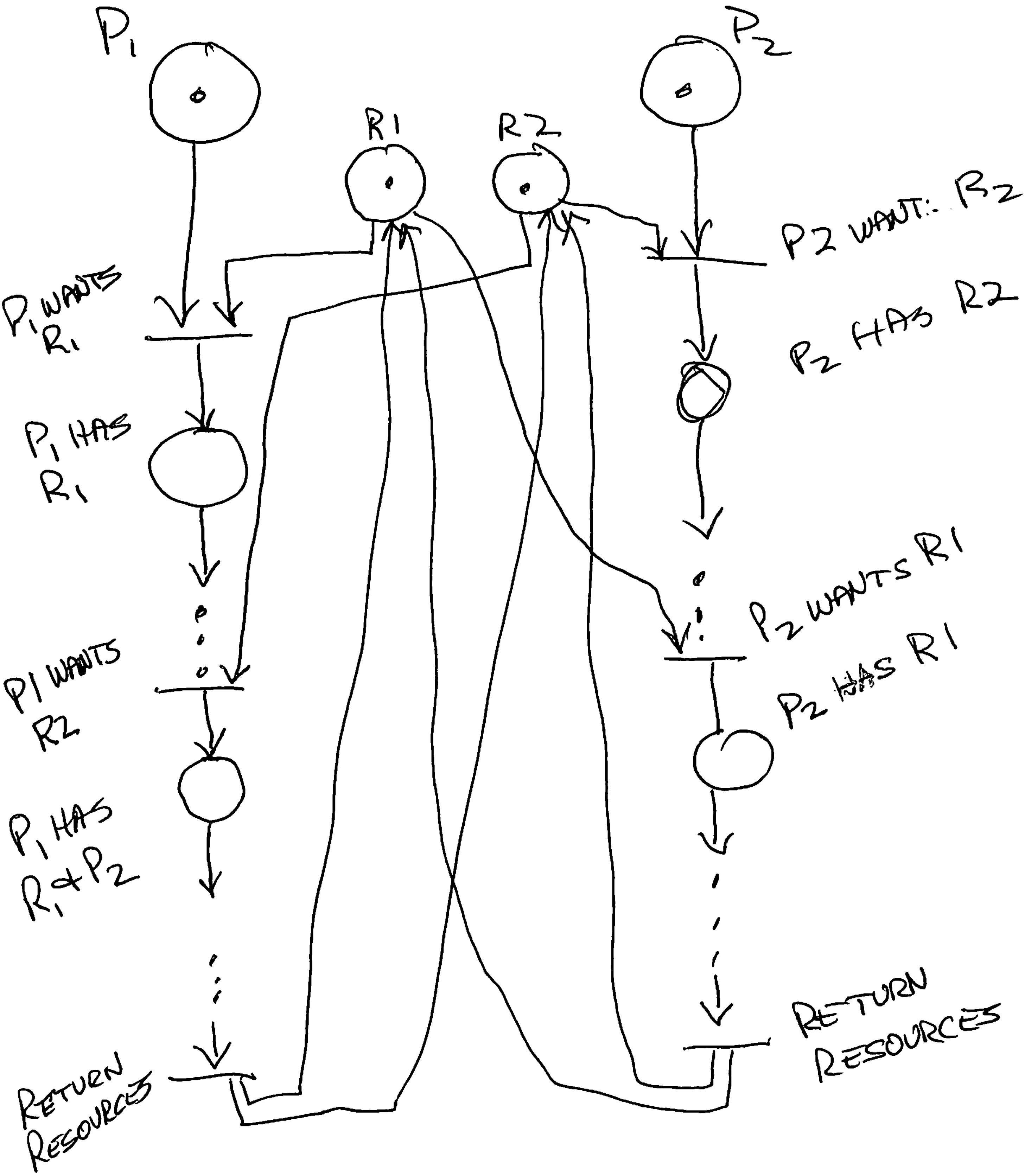
- FINITE STATE AUTOMATA
- LBAs
- TURING MACHINES

WEAKENS FOR FRS AND PETRI NETS

PETRI NET



Petri Net (NonDet) And Concurrency



HOW HARD IS IT TO ANALYZE PETRI NETS?

TO DETERMINE IF SOME MARKING CAN EVENTUALLY ARISE IS IN

EXPSPACE (N)

SOLVABLE, BUT TAKES EXPONENTIAL SPACE

TIME IS ACTUALLY 2^{2^N}

IF PRIORITY ADDED TO TRANSITIONS, PETRI NETS ARE COMPLETE MODELS OF COMPUTATION.

INTRO TO REDUCTION

$A \leq_m B$ IF THERE EXISTS SOME COMPUTABLE ALGORITHM $f \ni$

$x \in A \Leftrightarrow f(x) \in B$

IF B IS EASY TO SOLVE THEN SO IS A IF f DOES NOT ADD TO COMPUTATIONAL COMPLEXITY

HOWEVER, IF A IS KNOWN TO BE HARD (OR EVEN UNSOLVABLE) AND f DOES NOT CHANGE THE COMPLEXITY LANDSCAPE, THEN B MUST BE HARD AT LEAST WITHIN THE ORDER OR fs AND A'S COMPLEXITY, IF A IS UNSOLVABLE THEN SO IS B.

SOME REDUCTIONS

HALT $\leq_m$ TOTAL

LET f BE ARBITRARY FUNCTION

LET x BE ARBITRARY INPUT TO f

WANT TO KNOW IF $f(x)\downarrow$

THAT IS, $\langle f,x\rangle \in$ HALT

DEFINE

$$\forall y \; f_x(y) = f(x)$$

THEN $f_x \in$ TOTAL IFF $\langle f,x\rangle \in$ HALT

IF WE CAN SOLVE TOTAL, WE CAN SOLVE HALT