1 A review of automatic theorem proving

1.1 Introduction

– Proving B follows from A: conditions of **first-order predicate calculus**
– Single non-empty collection D: **universe of discourse**
– Relation symbols on D^n (n is degree of relation)
– Relation symbols with $n = 0$, truth value already determined
– Variables, arbitrary values of D
– Terms, from functions composed with their arguments
– Atomic formulae or atoms: relation symbol followed by a parenthesized list of terms; either T or F
– B is said to follow from A iff no way for D to be chosen, so that $(A \wedge \neg B)$ is T.
– Statement can't be T is **unsatisfiable**. If choice that is T, satisfiable.

1.2 A quick summary of the background theory

– Show statment, S, containing $n \geq 0$ vars, is unsatisfiable S is treated as a combination of atoms.
– $S(x_1, ..., X_n)$, show no interpretation of *vocabularly* $R_1..., R_k$, $f_1, ...f_m$ of S which makes S T.

2 Introduction

This is a test

3 MATHEMATICAL THEORY OF AUTOMATA

I FINITE AUTOMATA

word on Σ for a finite sequence $x = \sigma_0\sigma_1...\sigma_{n-1}, \sigma_i \in \Sigma$
The length $l(x)$ is the no. of elements in x
Λ is the empty word.
"Σ^* is a free semigroup under" concatenation
$\sigma \in \Sigma$ are "free generators" of this semi-group
For events (or subsets) $A \subseteq \Sigma^*, B \subseteq \Sigma^*$
$A^* = A^0 \cup A...$

DEFINITION 1: "A *finite automata (f.a.)* over Σ is a system $\mathfrak{U} =< S, M, s_0, F >$ where S is a finite set (the set of *states*), $M : S \times \Sigma \to S$ (the *table of transitions* of $\mathfrak{U}$) $s_0 \in S$ (s_0 is the *initial state*), $F \subseteq S$ (F is the set of *designated* final states)."

M "can be uniquely extended to" $M^* : S \times \Sigma^* \to S$; $M^*(s, \Lambda) = s, s \in S$
$M^*(s, x\sigma) = M(M^*(s, x), \sigma, s \in S, \sigma \in \Sigma, x \in \Sigma^*$

M is the state-transition function with respect to $\mathfrak{U}$.
In a similar manner, M^* describes state-transitions under input *words* $x \in \Sigma^*$.

DEFINITION 2: "The set $T(\mathfrak{U}) = \{x | x \in \Sigma^*, M^*(s_0, x) \in F\}$
is the *set* (event) *defined* by $\mathfrak{U}$. A set $A \subseteq \Sigma^*$ is called a *regular event* if for some finite automaton $\mathfrak{U}, A = T(\mathfrak{U})$."

I.1 F.a. mappings

Little distinction between automata in DEFINITION 1 and those that "yield a mapping from input sequences to output sequences".
Associate "state $s \in F$ the output 1 and with each $s \in S - F$ the output 0".
Associate $\mathfrak{U}$ the mapping $T : \Sigma^* \to \{0, 1\}^*$ s.t. $T(\sigma_0\sigma_1...\sigma_{n-1})$,
Where for $i \in [0, n-1], \tau_i = 1$ "if $M^*(s_0, \sigma_0...\sigma_i) \in F$ and and $r_i = 0$ otherwise."

Generalizations possible:
1. Partitioning S into disjoint "union of k sets", where $S = F_0 \cup ... \cup F_{k-1}$; mapping $T : \Sigma^* \to \{0, 1, ..., k-1\}^*$ into words from a k-letter alphabet.
2. Output of $F_i, i \in, [0, k-1]$ as a word, w_i on alphabet Ω. Mapping is $T : \Sigma^* \to \Omega^*$ ("not even length preserving")
3. Output as function $f(s, \sigma)$ with respect to the current state and current input of $\mathfrak{U}$

I.2 Nonderministic automata

$\mathfrak{U}$ in state s with input σ; can go into any one of a number of states $s' \in S'$, where $S \subset S$ is a set depending on s and σ.
As a result of DEFINITION 1: $M : S \times \Sigma \to P(S)$; "$P(S)$ is the power set of S and s_0 is replaced by a set $S_0 \subseteq S$ of initial states."
If $x = \sigma_0\sigma_1...\sigma_{n-1}$, the sequence of states $(s_0, s_1, ..., s_n)$ is **compatible** w/ x "if $s_0 \in S_0$" and for $i \in [0, n-1]$.
$\mathfrak{U}$ accepts x if for some sequence of states through s_n is compatible w/ x, $s_n \in F$.
The set $T(\mathfrak{U})$ consists of all words accepted by $\mathfrak{U}$.

THEOREM 1: "For every nonderministic automata $\mathfrak{U}$ there exists a f.a. $\mathfrak{B}$" s.t. "$T(\mathfrak{U}) = T(\mathfrak{B})$. If $\mathfrak{U}$ has n states, then $\mathfrak{B}$" has less than "2^n states."

I.3 Regular expressions and events

Let $Y_1, Y_2, ...$ be vars "ranging over subsets of Σ^* The set of regular terms in $Y_1, Y_2, ...$, $\mathbf{R}$ "is the smallest set satisfying the conditions":
1. $Y_n \in \mathbf{R}$, where $n \in [1, \infty)$
2. if $R_1 \in \mathbf{R}, R_2 \in \mathbf{R}$ then $R_1 \cup R_2 \in \mathbf{R}, (R_1R_2) \in \mathbf{R}, R_1^* \in \mathbf{R}$

Every element of $\mathbf{R}$ is called a regular expression $R(\sigma_1, ..., \sigma_n)$, which is a singleton set.
A regular expression a way of expressing subsets of Σ^*
THEOREM 2. "A set $T \subseteq \Sigma^*$ is f.a. definable (regular)" iff $\exists R(\sigma_1, ..., \sigma_n)$ where $\sigma_i \in \Sigma, i \in [1, k]$ s.t. $T = R(\sigma_1, ..., \sigma_k)$
Every regular event is representable by a regular expression.
Two regular terms $R(Y_1, ..., Y_n)$ and $Q(Y_1, ..., Y_n)$ are called equivalent if $\forall A_1 \subseteq \Sigma^*, ..., \forall A_n \subseteq \Sigma^*$,
$R(A_1, ..., A_n) = Q(A_1, ..., A_n)$

The "equivlance of regular terms is effectively solvable."

If fixed letters of Σ are part of "term formulation", $Y_n \in \mathbf{R}', \sigma \in \mathbf{R}', n = 1, 2, ... \sigma \in \Sigma$. Regular expressions yield f.a. definable sets.

I.4 Algebraization of f.a.

E is an equiv. relation on Σ^* is a right-invariant if xEy implies $xzEyz \forall x, y, z \in \Sigma^*$
left-invariant(?)
congruence on Σ^*: given a set T that is a a subset of Σ^*, one can define two relations...

THEOREM 3.
$T \subseteq \Sigma^*$ is regular iff it's "the union of equiv. classes of a right-invariant relation E (on Σ^*) with finite index." T is regular iff "E_T has a finite index."
$\exists \mathfrak{U}$ "with index(E_T) states" s.t. $T = T(\mathfrak{U})$. "No automaton with fewer than index(E_T) states defines T."

THEOREM 3 "is very useful in show that certain sets $T \subseteq \Sigma^*$ are or are not regular."

THEOREM 4."A set $T \subseteq \Sigma^*$ is regular" iff "it is a union of equivalence classes of a congruence $\equiv$ (on Σ^* with finite index." "...T is regular" iff "index($\equiv_T$) is finite."
"Given a congruence relation" on Σ^*, the partitioned set $\Sigma^*/\equiv$ of equivalence classes of Σ^* with respect to $\equiv$, into semi-group so that mapping $\phi_0 : x \to [x]_\equiv$ of ea. $x \in \Sigma^*$ into equiv. class is a homomorphism of Σ^* onto $\Sigma^*/\equiv$.
Conversely, to ea. homomorphism $\phi_1 : \Sigma^* \to M$ of Σ^* onto semi-group, M - corresponding congruence $\equiv$ def by $x \equiv y$ iff $\phi_1(x) = \phi_1(y)$.

THEOREM 4 (restated).
"A set $T \subseteq \Sigma^*$ is regular" iff $\exists M, H \subseteq M$ and a homomorphism $\phi : \Sigma^* \to M$ s.t. $T = \phi^{-1}(H)$, where M is a finite semigroup.

Associate with ea. $\sigma \in \Sigma$ a function f_σ. The system $< A, a_0, f_\sigma >_{\sigma \in \Sigma}$ is an *algebra of type* Σ if $a_0 \in A$

I.5 Decision problems and algorithms

Skipped.

I.6 Finite automata and infinite sequences

A f.a. $\mathfrak{U}$ accepting a word, $x = \sigma_0\sigma_1...\sigma_n...$, x is a denumerable seq. of letters $\sigma_n \in \Sigma$. Let $\mathfrak{U}$ be a NFA. A denumerable seq. of states $(s_0, s_1...)$ "is called *compatible* with x if $s_0 \in S_0$
and $S_{n+1} \in M(s_n, \sigma_n), 0 \leq n < \infty$."

DEFINITION 3. "$\mathfrak{U}$ *accepts* x if for s" (a sequence), "$\{n|S_n \in F\}$ is infinite." "Set of all denumerable" seq. "accepted by $\mathfrak{U}$ will be denoted by $T_\infty(\mathfrak{U})$"

THEOREM 6 (Büchi)."If $H_1 \subseteq \Sigma_\infty$ and $H_2\Sigma_\infty$" are f.a. definable, so are the union, intersection of H_1 and H_2 as well as $\Sigma_\infty - H_1$

II PROBALISTIC AUTOMATA

Not relevant to problem.

III TREE-AUTOMATA

Tree automata have a tree for input.
A version of tree automata can be applied to context-free languages [Mezei and Wright, "Generalized Algol-like Languages"].
$\mathfrak{U} =< S, M, S_0, F >$ f.a over Σ; $x = \sigma_0\sigma_1\sigma_2 \in \Sigma$. Terminal node has no value. Except for the initial state s_0, elements of F not indexed.

$M : S \times S \times \Sigma \rightarrow S$ (note: M earlier defined as transition table). Inputs for $\mathfrak{U}$ finite binary trees, nodes val. by Σ.

III.1 Binary trees

Paper defines trees "as sets of words on alphabet $\Omega = \{0, 1\}$." Let $x, y \in \Omega^*, x$ is "...the prefix of y if some $z \in \Omega^*, y = xz$." "...Subset $H \subseteq \Omega^*$ is *prefix closed* if $y \in H$ and x prefix of y imply $x \in H$. "The *prefix closure* of" $G \subseteq \Omega^*$ "is the smallest prefix closed set H..." s.t. $G \subseteq H$. All "prefix closed set[s]" have the empty word as a member.

"A (binary) tree is a prefix-closed subset $H \subseteq \Omega^*$." ."..Elements $x \in H$" are nodes. "The node $\Lambda \in H$ is the" tree's root. (why use the empty word symbol? (ZO: If you think of the root node as the start of the binary string, then Λ is the

empty word $\{\}$. Following an edge from the root Λ to either 0 or 1 would add a character to the string, producing the string $\{0\}$ or $\{1\}$. Following another edge from that 0-node would give either $\{0, 0\}$ or $\{0, 1\}$, and so on.))
If $x \in H$ and $x_0 \notin H$ and $x_1 \notin H$, x is a terminal node.
H is *frontiered* if $\forall\ x \in H$ (x is non-terminal) we have $x_0 \in H$ and $x_1 \in H$. $Ft(H)$ is "the set of all terminal nodes of H." (ZO: $Fr(H)$ seems to mean the same thing as $Ft(H)$. I think $Ft(H)$ is a typo in the original.)

III.2 Tree-automata

A Σ-valued tree is a pair (v, H), where H is a finite frontiered tree; $v : (H - Fr(H)) \to \Sigma$. Each non-terminal node $x \in H$ is assigned a symbol $v(x) \in \Sigma$.
"The set of all valued trees" is denoted by $V(\Sigma, \Omega)$.

DEFINITION 9: "A *finite tree automaton* (f.t.a) is a system $\mathfrak{U} =< S, M_t, s_0, F >$."
Symbols S, s_0, F in DEFINITION 1. $M_t : S \times S \times \Sigma \to S$
DEFINITION 10: "The f.t.a. $\mathfrak{U}_t$ is said to accept $(v, H) \in V(\Sigma, \Omega)$, if $\exists\ \phi : H \to S$ satisfying:"
1. "$\phi(x) = s_0$ for $x \in Fr(H)$"
2. "if $x \in Fr(H)$ (hence, $x_0, x_1 \in H$) then $\phi(x) = M_t(\phi(x_0)\phi(x_1), v(x))$"
3. "$\phi(\Lambda) \in F$"

Set of all inputs accepted denoted $T(\mathfrak{U})$. T is the set defined by $\mathfrak{U}_t$. $T \subseteq V(\Sigma, \Omega)$ is f.t.a. definable if for some $\mathfrak{U}_t, T = T(\mathfrak{U}_t)$.
Note: The function ϕ is unique.
non-deterministic f.t.a. – $\phi : H \to S$ compatible with (v, H), $T(\mathfrak{U}_t)$ defined by a non-deterministic f.t.a.
Results that follow from f.a. to finite tree automata:
"...Class of all f.t.a. definable subsets $T \subseteq V(\Sigma, \Omega)$ is a boolean algebra." "For every non-deterministic f.t.a" $T(\mathfrak{U}_t)\exists$ a deterministic f.t.a. $\mathfrak{B}_t$ s.t. $T(\mathfrak{U}_t) = T(\mathfrak{B}_t)$.
"The emptiness and equivalence problems for f.t.a are solvable." (? what is the emptiness problem (ZO: "The emptiness problem is the question of determining whether a language is empty given some representation of it, such as a finite-state automaton." [Sipser, *Introduction to the Theory of Computation*, 2012]. In other words, is there some input that the automaton will accept?); does the equivalence problem have to do with the eq. relation material earlier? (ZO: Maybe?))

III.3 Valued trees and terms

There is "a one-to-one correspondence between terms and valued trees..."
$T_0 = (v_0, H_0)$ and $T_1 = (v_1, H_1)$ are Σ-valued trees and $\sigma \in \Sigma$. H_0 and H_1 are trees, which can be combined: $H = \{\Lambda\} \cup 0H_0 \cup 1H_1$.
combined valued tree is denoted $C(\sigma, T_0, T_1)$ is defined to be (v_0, H) where $v(\Lambda) = \sigma, v(0x) = v_0(x)$ for $x \in H_0, v(1x) = v_1(x)$ for $x \in H_1$.
"Associate with each $\sigma \in \Sigma$" a binary function (operation) symbol F_σ and let u_0 be a fixed symbol. V_2, set of all terms, is "...the smallest set" s.t. "$u_0 \in V_2$ if $t_0 \in V_2$ and $t_1 \in V_2$ then $F_\sigma(t_0, t_1) \in V$ for $\sigma \in \Sigma$."

ϕ is a one-to-one correspondence between $V(\Sigma, \Omega)$ and V_2, defined recursively by all of the following statements.

$$
\begin{aligned}
\phi(\Lambda) &= u_0 && (1)\\
\phi(C(\sigma, T_0, T_1)) &= F_\sigma(\phi(T_0), \phi(T_1)) && (2)\\
\sigma &\in \Sigma && (3)\\
T_0, T_1 &\in V(\Sigma, \Omega) && (4)
\end{aligned}
$$

"A set $K \subseteq V_2$ of terms is...*f.t.a definable* if for some definable $T \subseteq V(\Sigma, \Omega)$ we have $K = \phi(T)$.
If terms $t \in V_2$ are words on the alphabet $\{F_\sigma | \sigma \in \Sigma\}$, definable sets that are not always regular events per DEFINITION 2.
"t is often stored in memory in tree (list) form and the evaluation of" $\phi(x)$ for each node done successively, "starting from the terminals for which $\phi(x) = s_0$, ending with $\phi(\Lambda)$."