Exact Exploration

Nachum Dershowitz

June 10, 2009

In the Beginning

Heaven and Earth

Algorithm

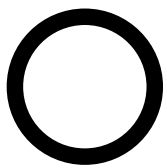
Program

Law and Order

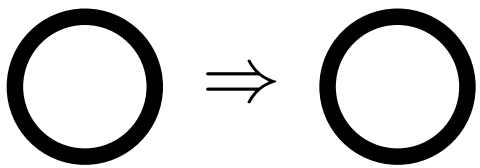
Postulates

ASM

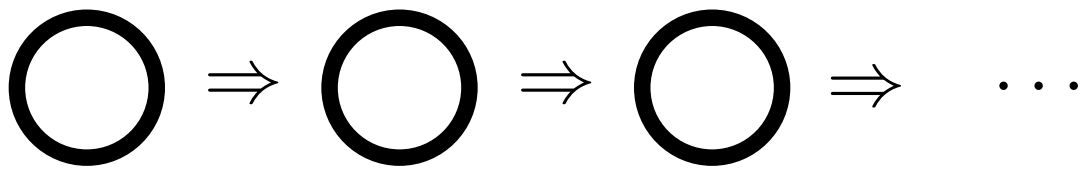
State



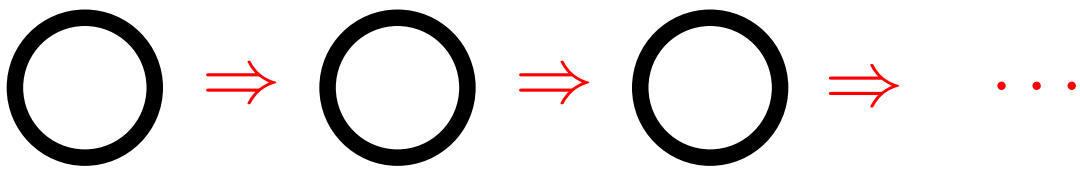
State Transition



Run



Emulation

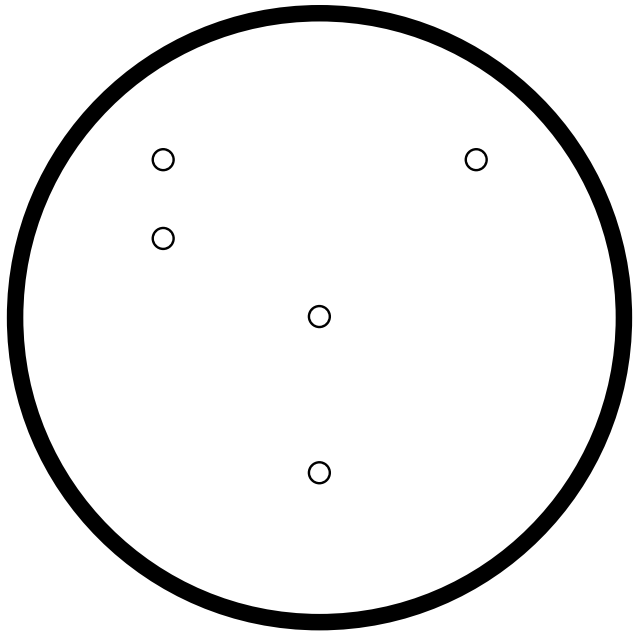


Sequential Time Postulate

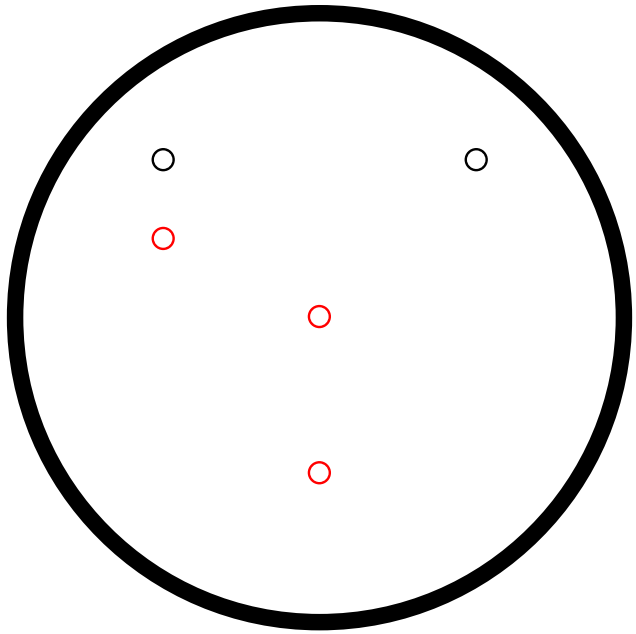
An algorithm determines:

1. A nonempty set $\mathcal{S}$ of *states*.
2. A nonempty subset $\mathcal{I} \subseteq \mathcal{S}$ of *initial states*.
3. A subset $\mathcal{O} \subseteq \mathcal{S}$ of *terminal states*.
4. A *next state* transformation $\tau : \mathcal{S} \setminus \mathcal{O} \to \mathcal{S}$.

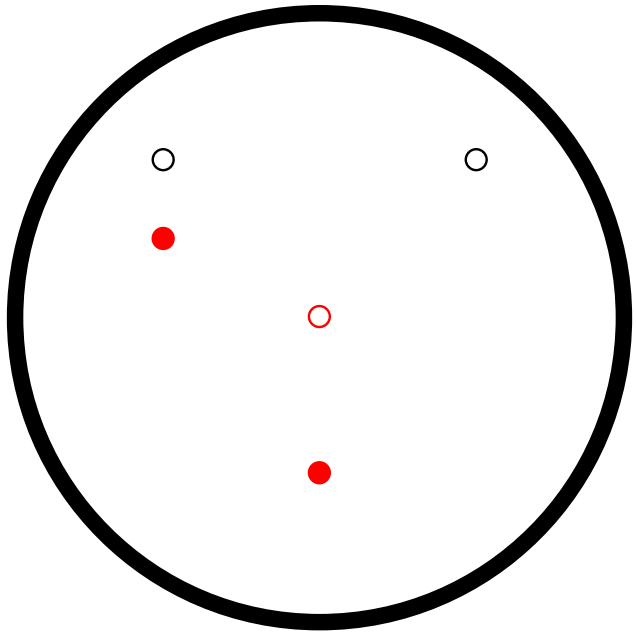
Locations



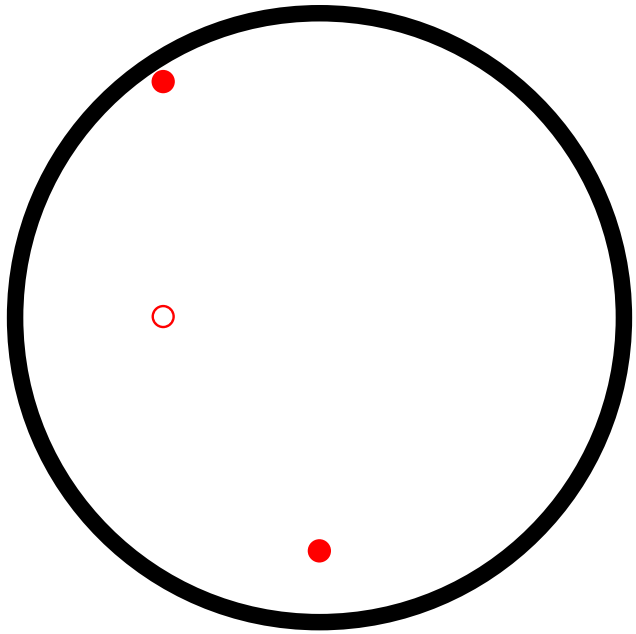
Exploration



Update



State Dependent



ASM Programs

One of the following:

- An *assignment* statement $f(s_1, \ldots, s_n) := t$.
- A *parallel* statement $[P_1 \parallel \cdots \parallel P_n]$, $n \geq 0$.
- A *conditional* statement **if** $C_1 \wedge \cdots \wedge C_n$ **then** P.

Sorting

$$
\begin{array}{l}
[\ \textbf{if}\ j \neq n\ \textbf{then} \\
\quad [\ \textbf{if}\ F(i) > F(j)\ \textbf{then} \\
\qquad [F(i) := F(j) \parallel F(j) := F(i) \parallel i := 0 \parallel j := 1\ ] \\
\quad \parallel \textbf{if}\ F(j) \geq F(i)\ \textbf{then}\ j := j + 1] \\
\parallel \textbf{if}\ j = n \wedge i + 1 \neq n\ \textbf{then}\ [i := i + 1 \parallel j := i + 2\ ]\ ]
\end{array}
$$

Sorting Reals

What happens when the $F(i)$ are real numbers?

It depends!

Sorting States

	State X such that
$\mathcal{I}$	$[\![n]\!] \geq 0,\ [\![i]\!] = 0,\ [\![j]\!] = 1$
$\mathcal{O}$	$[\![j]\!] = [\![n]\!] = [\![i]\!]$
B	$[\![j]\!] \neq [\![n]\!],\ [\![F(i)]\!] > [\![F(j)]\!]$
S	$[\![j]\!] \neq [\![n]\!],\ [\![F(i)]\!] \leq [\![F(j)]\!]$
R	$[\![j]\!] = [\![n]\!] \neq [\![i]\!] + 1$

Update Set

- $\Delta(X) = \tau(X) \setminus X$.
- $\Delta(X) = \perp$ for $X \in \mathcal{O}$.

Sorting Updates

	$\Delta(X)$
$\mathcal{O}$	$\bot$
B	$F([\![i]\!]) \mapsto [\![F(j)]\!],\ F([\![j]\!]) \mapsto [\![F(i)]\!], i \mapsto 0,\ j \mapsto 1$
S	$j \mapsto [\![j]\!] + 1$
R	$i \mapsto [\![i]\!] + 1,\ j \mapsto [\![i]\!] + 2$

$$
\begin{aligned}
\Delta^{+}_{f(s_1,\ldots,s_n):=t}(X) &= \{f([\![s_1]\!]x,\ldots,[\![s_n]\!]x) \mapsto [\![t]\!]x\} \\
\Delta^{+}_{[P_1 \parallel \cdots \parallel P_n]}(X) &= \Delta^{+}_{P_1}(X) \cup \cdots \cup \Delta^{+}_{P_n}(X) \\
\Delta^{+}_{\textbf{if } C \textbf{ then } P}(X) &= \begin{cases} \Delta^{+}_{P}(X) & \text{if } X \models C \\ \varnothing & \text{otherwise} \end{cases}
\end{aligned}
$$

$$\Delta^0(X) = \{f(\bar{a}) \mapsto [\![f(\bar{a})]\!]_X : \bar{a} \in \mathrm{Dom}\ X\}$$

$$\Delta(X) = \begin{cases} \bot & \text{if } \Delta^+(X) = \varnothing \text{ or is inconsistent} \\ \Delta^+(X) \setminus \Delta^0(X) & \text{otherwise} \end{cases}$$

Abstract State Postulate

The states $\mathcal{S}$ are structures over a finite vocabulary $\mathcal{F}$ such that:

1. If $X \in \mathcal{S}$ and $X \cong Y$, then $Y \in \mathcal{S}$.
2. If $X \in \mathcal{I}$ and $X \cong Y$, then $Y \in \mathcal{I}$.
3. If $X \in \mathcal{O}$ and $X \cong Y$, then $Y \in \mathcal{O}$.
4. $\mathrm{Dom}\ \tau(X) = \mathrm{Dom}\ X$ for all $X \in \mathcal{S} \setminus \mathcal{O}$.
5. If $X \cong_\zeta Y$ for $X, Y \in \mathcal{S} \setminus \mathcal{O}$, then $\tau(X) \cong_\zeta \tau(Y)$.

Bounded Exploration Postulate

There is a finite set of terms T, such that, for $X, Y \in \mathcal{S}$

$$\text{if } X =_T Y \text{ then } \Delta(X) = \Delta(Y)$$

Critical Terms

$$j = n$$

$$j \neq n$$

$$i + 1 \neq n$$

$$F(i) > F(j)$$

$$F(j) \geq F(i)$$

$$0,\ 1$$

$$j + 1,\ i + 1,\ i + 2$$

plus subterms

ASM Theorem

$$\textbf{if } (F(i) > F(j)) = \text{true} \wedge j = n \wedge i + 1 \neq n \textbf{ then } j := i + 2$$

ASM Theorem

Every algorithm satisfying these postulates is emulated by an ASM.

The Problem

The ASM may do too much!

Exact Terms

$j = n$

$j \neq n$

$i + 1 \neq n$

$F(i) > F(j)$

$F(j) \geq F(i)$

$0,\ 1$

$j + 1,\ i + 1,\ i + 2$

plus subterms

Gaussian Elimination

- Tests pivot $p \neq 0$ before dividing $a[i, j]/p$.
- Program includes expression $a[i, j]/p$.
- Emulating ASM evaluates $a[i, j]/p$, regardless of p.
- Want ASM that only explores $a[i, j]/p$ when $p \neq 0$.

Matrix Inversion

- Reals; computable reals; complex numbers.
- First compute adjugate.
- Then divide through by determinant.
- No need for equality or disequality.
- Only if $\det A = 0$ is $A^{-1} = \bot$.

Parsimony

- Remove duplicates from a file system.
- First check that sizes are the same.
- ASM always compares *both* size and content!
- Want ASM to only check what algorithm does.

Exact Exploration

There are finite sets of terms $\Gamma(X)$, such that, for $Y \in \mathcal{S}$

$$\text{if } X =_{\Gamma(X)} Y \text{ then } \Delta(X) = \Delta(Y)$$

Exact Exploration

There are finite sets of terms $\Gamma(X)$, such that, for $Y \in \mathcal{S}$

$$\text{if } X =_{\Gamma(X)} Y \text{ then } \Gamma(X) = \Gamma(Y)$$

Explore Sets

	$\Gamma(X)$ (omitting subterms)
$\mathcal{O}$	$j = n,\ j \neq n,\ i+1 \neq n$
B	$j = n,\ j \neq n,\ i+1 \neq n,\ F(i) > F(j),\ F(j) \geq F(i),\ 0,\ 1$
S	$j = n,\ j \neq n,\ i+1 \neq n,\ F(j) > F(i),\ F(j) \geq F(i),\ j+1$
R	$j = n,\ j \neq n,\ i+1 \neq n, i+1,\ i+2$

Exact Terms

$$
\begin{aligned}
\Gamma_{[P_1 \parallel \cdots \parallel P_n]}(X) &= \bigcup_j \Gamma_{P_j}(X) \\
\Gamma_{f(s_1,\ldots,s_n):=t}(X) &= \bigcup_i \Gamma_{s_i}(X) \cup \Gamma_t(X) \\
\Gamma_{f(s_1,\ldots,s_n)}(X) &= \{f(s_1,\ldots,s_n)\} \cup \bigcup_i \Gamma_{s_i}(X) \\
\Gamma_{\textbf{if } C \textbf{ then } P}(X) &= \Gamma_C(X) \cup \begin{cases} \Gamma_P(X) & \text{if } X \models C \\ \varnothing & \text{otherwise} \end{cases} \\
\Gamma_{c_1 \wedge \cdots \wedge c_k}(X) &= \bigcup_j \Gamma_{c_j}(X) \\
\Gamma_{\neg c}(X) &= \Gamma_c(X)
\end{aligned}
$$

Exact Exploration

An algorithm determines a partially-ordered finite explore set $(\Gamma(X), \prec_X)$ of ground terms such that:

- **Determination.** For every other state $Y \in \mathcal{S}$, if $X =_{\Gamma(X)} Y$, then $\Delta(X) = \Delta(Y)$.
- **Discrimination.** For every other state $Y \in \mathcal{S}$, and for every $t \in \Gamma(X) \setminus \Gamma(Y)$, there is a Boolean term $s \in \Gamma(X)$ such that $s \prec_X t$ and $[\![s]\!]_X \neq [\![s]\!]_Y$.
- **Limitation.** The set $\cup_{X \in \mathcal{S}} \Gamma(X)$ of all explore terms is finite.

Exploration Order

	Exploration order $\prec_X$ (omitting subterms)
$\mathcal{O}$	
B	$j \neq n \prec_X F(i) > F(j) \prec_X 0,\ 1;\quad j \neq n \prec_X F(j) \geq F(i)$
S	$j \neq n \prec_X F(j) \geq F(i) \prec_X j+1;\quad j \neq n \prec_X F(j) > F(i)$
R	$i+1 \neq n,\ j = n \prec_X i+1,\ i+2$

$$
\begin{array}{l}
[\ \textbf{if } d \textbf{ then if } b \textbf{ then if } c \textbf{ then } x := 0 \ \| \\
\quad \textbf{if } d \textbf{ then if } \neg b \textbf{ then } x := 1 \ \| \\
\quad \textbf{if } d \textbf{ then if } \neg c \textbf{ then } y := 2 \ ]
\end{array}
$$

Magic Program

$$[\ \textbf{if}\ b\ \textbf{then}\ [b := b \parallel c := c] \quad \parallel \quad \textbf{if}\ \neg b\ \textbf{then}\ a := a\ ]$$

whenever a is true

$$[\ \textbf{if}\ c\ \textbf{then}\ [b := b \parallel c := c] \quad \parallel \quad \textbf{if}\ \neg c\ \textbf{then}\ a := a\ ]$$

whenever a is false

Exact Exploration implies Bounded Exploration

- Let $T = \cup_{X \in \mathcal{S}} \Gamma(X)$ be all the explore terms of the algorithm in question.
- By Limitation, T is finite.
- By Determination, T determines behavior.

Bounded Exploration implies Exact Exploration

- Let $\Gamma(X) = T$.
- Let $\prec_X$ be empty.

Equivalence

Two algorithms $\mathcal{P}$ and $\mathcal{Q}$ are *equivalent* if

- they operate over the same states $\mathcal{S}$,
- have the same initial states $\mathcal{I}$ and terminal states $\mathcal{O}$,
- for all states $X \in \mathcal{S}$, $\Delta_{\mathcal{P}}(X) = \Delta_{\mathcal{Q}}(X)$, and
- for all states $X \in \mathcal{S}$, $\Gamma_{\mathcal{P}}(X) = \Gamma_{\mathcal{Q}}(X)$.

Sets of States

- $\Gamma(V) = \cap_{X \in V} \Gamma(X)$.
- C is *agreeable* if $X =_{\Gamma(V)} Y$ for all $X, Y \in V$.
- Agreeable states should have uniform behavior!

Uniformity

- A set V of states is *uniform* if $\Gamma(X) = \Gamma(Y)$ for all $X, Y \in V$.
- When V is agreeable, then it should also be uniform.

For any discriminating algorithm, agreeability of a set of states implies its uniformity

- Suppose not all states in V have the same explore set.
- Let $t \in \Gamma(X)$ be a minimal explore term of some $X \in V$ that is not also an explore term of all other states.
- Let $Y \in V$ be such that $t \notin \Gamma(Y)$.
- By Discrimination, there is an $s \in \Gamma(X)$ such that $s \prec_X t$ and $[\![s]\!]_X \neq [\![s]\!]_Y$.
- By agreeability, $s \notin \Gamma(V)$.
- Then s must be a smaller choice of an explore term of X than is t, since $s \notin \Gamma(Z)$ for some $Z \in V$.

For any classical algorithm, if every agreeable set of states is uniform, then it's discriminating

$\prec_X := \varnothing$
$V := \mathcal{S}$
while $\Gamma(V) \subsetneq \Gamma(X)$ **do**
  $\prec_X := \prec_X \cup\ \Gamma(V) \times (\Gamma(X) \setminus \Gamma(V))$
  $V := \{Y \in V : Y =_{\Gamma(V)} X\}$

- Consider any $t \in \Gamma(X) \setminus \Gamma(Y)$ for a $Y \in \mathcal{S}$.
- Initially, $t \notin \Gamma(V) = \Gamma(\mathcal{S})$.
- At the end, $t \in \Gamma(V) = \Gamma(X)$.
- So Y is not in the final V.
- When Y is removed from V, there must be an $s \in \Gamma(V)$ that discriminates between X and Y.
- By construction, $s \prec_X t$.

Every ASM program is a discriminating algorithm

- First two postulates hold for ASMs.
- Limitation holds, since programs are finite and all terms in $\Gamma(X)$ appear there.
- If $X =_{\Gamma(X)} Y$, then all tests performed by the algorithm have the same outcome in Y as in X.
- So $\Gamma^+(X) = \Gamma^+(Y)$, and, hence, $\Gamma(X) = \Gamma(Y)$.
- The algorithm for explore sets returns $\Gamma(V)$.
- If V is agreeable, then whenever some state in V does not satisfy C, none do.
- So the computation of $\Gamma(V)$, for agreeable V, proceeds precisely as does the computation of $\Gamma(X)$ for any $X \in V$.
- So agreeability of V implies its uniformity.

Assignments

For every discriminating algorithm, if $f(\bar{a}) \mapsto b$ is an update in $\Delta(X)$ for some state X, then there are terms t and $\bar{s}$ in $\Gamma(X)$ such that $[\![t]\!]_X = b$ and $[\![s_i]\!]_X = a_i$ for each a_i of $\bar{a}$.

Each discriminating algorithm has an equivalent ASM

ASM is $P(\mathcal{S})$, defined as follows:

$$P(V) = \begin{cases} [\,] & \text{if } V = \varnothing \\ [\textbf{if } C_1 \textbf{ then } P(V \restriction C_1) \parallel \cdots \parallel \textbf{if } C_k \textbf{ then } P(V \restriction C_k)] & \\ & \text{if } V \text{ is not agreeable} \\ \textbf{if } C \textbf{ then } R & \text{if } V \text{ is agreeable} \end{cases}$$

C is a conjunction of all the Boolean terms $c \in \Gamma(V)$, or their negations $\neg c$, depending on whether $V \models c$ or $V \models \neg c$.
R is a parallel collection of assignments for all the updates in $\Delta(X)$, for any one state $X \in V$.

Case Statements

$$\textbf{case } q_1, \ldots, q_n \textbf{ of } \begin{cases} \textbf{when } a_{11}, \ldots, a_{1n} \textbf{ then} & S\ ; \\ \textbf{when } a_{21}, \ldots, a_{2n} \textbf{ then} & R\ ; \\ \quad \vdots & \\ \textbf{when } a_{m1}, \ldots, a_{mn} \textbf{ then} & S_m \end{cases}$$

More than Boolean

- **Discrimination.** *For every other state* $Y \in \mathcal{S}$*, and for every* $t \in \Gamma(X) \setminus \Gamma(Y)$*, if any, there is a discrimination term* $s \in \Gamma(X)$ *such that* $s \prec_X t$ *and* $[\![s]\!]_Y \neq [\![s]\!]_X \in \mathbf{K}$.

$$\Delta^{+}_{\textbf{case } \bar{q} \textbf{ of } \overline{W}}(X) = \bigcup_j D^{\bar{q}}_{W_j}(X)$$

$$D^{\bar{q}}_{\textbf{when } \bar{a} \textbf{ then } P}(X) = \begin{cases} \Delta^{+}_{P}(X) & \text{if } X \models \bar{q} = \bar{a} \\ \{\bullet\} & \text{if } \bar{a} \notin K^n \\ \varnothing & \text{otherwise} \end{cases}$$

$$\Gamma_{\textbf{case } \bar{q} \textbf{ of } \overline{W}}(X) = \{q_1, \ldots, q_n\} \cup \bigcup_j G^{\bar{q}}_{W_j}(X)$$

$$G^{\bar{q}}_{\textbf{when } \bar{a} \textbf{ then } P}(X) = \begin{cases} \Gamma_{P}(X) & \text{if } X \models \bar{q} = \bar{a} \\ \varnothing & \text{otherwise} \end{cases}$$

$$\Delta(X) = \begin{cases} \bullet & \text{if } \bullet \in \Delta^{+}(X) \\ \bot & \text{if } \Delta^{+}(X) = \varnothing \text{ or is inconsistent} \\ \Delta^{+}(X) \setminus \Delta^{0}(X) & \text{otherwise} \end{cases}$$

Black Holes

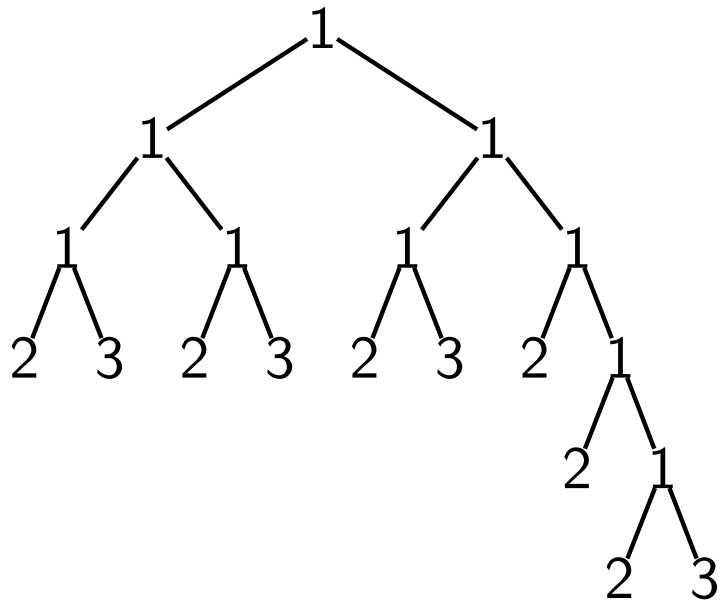
- $\Delta(X) = \tau(X) \setminus X$.
- $\Delta(X) = \perp$ for $X \in \mathcal{O}$.
- $\Delta(X) = \bullet$ for $X \in \mathcal{O}$, but don't know!

Refinement

1. Critical terms are used in the conditional tests of **if** statements and in the queries of the **case** statement.
2. The contents of location indicated by right-hand side of assignement is copied into another location.
3. Critical terms are used indirectly to determine locations needed for tests or updates.

Partition $\Gamma(X)$ into three parts:

1. $\Gamma^D(X)$ for the discriminating terms used in conditional and case statements;
2. $\Gamma^C(X)$ for obtaining the contents of locations indicated by right-hand sides of assignments;
3. $\Gamma^A(X)$ for addressing locations.


1
1
1
1
1
1
1
1
2
3
2
3
2
3
2
1
2
1
2
3

Flat ASMs

$$
\left[\begin{array}{l}
\textbf{if } C_{11} \textbf{ then if } C_{12} \textbf{ then } \cdots \textbf{ then } f_1(s_{11}, \ldots, s_{1n_1}) := t_1 \ \| \\
\textbf{if } C_{21} \textbf{ then if } C_{22} \textbf{ then } \cdots \textbf{ then } f_2(s_{21}, \ldots, s_{2n_2}) := t_2 \ \| \\
\textbf{if } C_{31} \textbf{ then if } C_{32} \textbf{ then } \cdots \textbf{ then } [] \ \| \\
\vdots \\
\textbf{if } C_{k1} \textbf{ then if } C_{k2} \textbf{ then } \cdots \textbf{ then } f_k(s_{k1}, \ldots, s_{kn_k}) := t_k
\end{array}\right]
$$